Name: ____

Expressions & Equations

Fall 2025 Student Workbook - WeScheme Edition

Workbook v0.9-beta

Brought to you by the Bootstrap team:

- Emmanuel Schanzer
- Kathi Fisler
- Shriram Krishnamurthi
- Dorai Sitaram
- Joe Politz
- Ben Lerner
- Nancy Pfenning
- Flannery Denny
- Rachel Tabak

Table of Contents

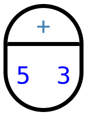
Translation and Equivalence

Translation

Circles of Evaluation help us visualize the structure of mathematical expressions.

- Every Circle of Evaluation must have one - and only one! - operator (or function!) written at the top.
- The inputs of the operator are written left to right, in the middle of the Circle.
- Circles of Evaluation can contain other Circles of Evaluation.

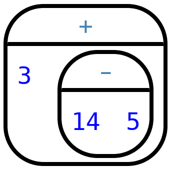
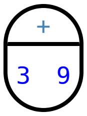
We can translate any arithmetic expression into a Circle of Evaluation or a verbal expression. Below, we've translated the arithmetic expression $5 + 3$ into a Circle of Evaluation and then a Verbal Expression.

$5 + 3$ →  → the sum of 5 and 3

Math is precise, but that precision is difficult to preserve when we switch to words. Often, sentences can be ambiguous, meaning that there is more than one way to interpret them! One reason that Circles of Evaluation are so powerful is that they eliminate the ambiguity we encounter when representing expressions with words.

Equivalence and Computation

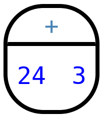
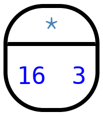
Arithmetic expressions are **equivalent** when they simplify to the same value. Here is an illustration (with Circles of Evaluation) that can help us visualize simplifying a more complex expression into a single numeric value.

 →  → 12

Computation is one tool (of many!) that can allow us to determine if two expressions are equivalent.

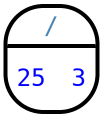
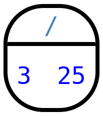
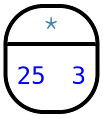
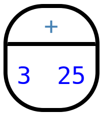
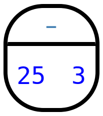
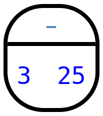
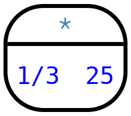
Translating

Each row represents a single arithmetic expression, written in three different forms. Fill in the empty spaces so that all three forms represent the same expression. *For each expression, there must be two equivalent expressions in words.*

	Expression in Words	Circle of Evaluation	Math Expression
1)	24 increased by 3 ______		
2)	the product of 9 and 11 ______		9×11
3)	______ ______		$\frac{24}{8}$
4)	$^1/_3$ less than 4 ______		
5)	______ ______		
6)	half of 100 ______		
7)	the difference between 20 and 8 ______		
8)	______ ______		$^1/_3 \times 4$

Matching Words to Circles of Evaluation

Draw a line from the words on the left to the Circle on the right. Some Circles have more than one correct translation.

Words			Circle of Evaluation
25 tripled	**1**		
3 less than 25	**2**		
25 less than 3	**3**	**A**	
one-third of 25	**4**	**B**	
add 3 and 25	**5**	**C**	
divide 25 into 3 groups	**6**	**D**	
the quotient of 3 and 25	**7**	**E**	
the quotient of 25 and 3	**8**	**F**	
25 decreased by 3	**9**		
the product of 25 and 3	**10**		

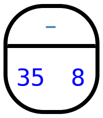
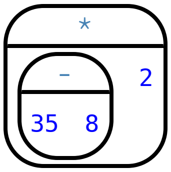
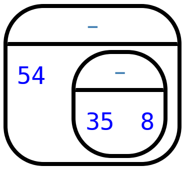
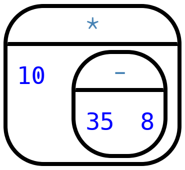
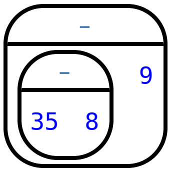
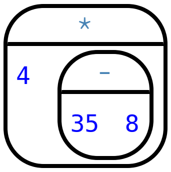
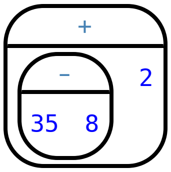
Translating from Words to Circles

For each expression in words on the left, draw a Circle of Evaluation on the right.

	Math Expression	Circle of Evaluation
1)	the sum of 12 and 4	
2)	double the sum of 12 and 4	
3)	the difference between 100 and the sum of 12 and 4	
4)	Find the sum of 12 and 4. Take half.	
5)	10 more than the sum of 12 and 4	
6)	3 less than the sum of 12 and 4	
7)	the product of 6 and the sum of 12 and 4	

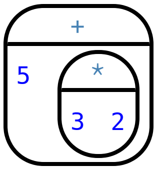
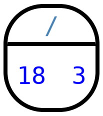
Translating from Circles to Words

Translate each Circle of Evaluation into words. The first one is done for you.

	Circle of Evaluation	Expression in Words
1)		the difference between 35 and 8
2)		
3)		
4)		
5)		
6)		
7)		

Translation Table (1)

Fill in any missing spaces on the table below so that the mathematical expression, the Circle of Evaluation, and Verbal Expression are all equivalent.

	Math	Circle of Evaluation	Verbal Expression
1)			Find the sum of 5 and 3
2)	$(5 + 3) \times 2$		Find the sum of 5 and 3, then double it.
3)			
4)			
5)	$(18 \div 3) + 6$		
6)			Find the sum of 3 and 6. Divide 18 by that sum.

Translation Table (2)

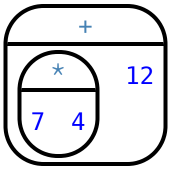
For each provided arithmetic expression, draw an equivalent Circle of Evaluation and write a translation in words.

	Math	Circle of Evaluation	Verbal Expression
1)	24 - 6		
2)	24 - (6 ÷ 3)		
3)	(24 - 6) ÷ 3		
4)	$^1/_2 \times 100$		
5)	$(^1/_2 \times 100) + 10$		
6)	$^1/_2 \times (100 + 10)$		

The Ambiguity of Words

How many different ways can each sentence be interpreted? For each way, draw the Circle and write the arithmetic expression. We've started the first one for you.

1) The product of seven and four increased by twelve

	
$(7 \times 4) + 12$	

2) The quotient of ten and two decreased by one

3) Three more than nine multiplied by four

4) Half of ten tripled

5) The sum of six and three increased by five

Rewriting Ambiguous Expressions

All of the verbal expressions below are ambiguous. Rewrite each expression two times:

- The first time, write the expression to indicate that either multiplication or division happens first.
- The second time, write the expression to indicate that either addition or subtraction happens first.

Use parentheses to indicate which operation comes first. Give both the arithmetic and verbal expression. We've done the first one for you.

	Ambiguous Expression	**Multiplication/division first.**	**Addition/subtraction first.**
1)	The product of 10 and 8 decreased by 5	5 less than the product of 10 and 8 by 5 (10 × 8) - 5	Multiply the difference between 8 and 5 by 10 10 × (8 - 5)
2)	The product of 1/3 and 30 increased by 4		
3)	The difference between 100 and 6 multiplied by 9		
4)	The sum of 6 and 12 divided by 3		
5)	The quotient of 60 and 10 increased by 5		

Ambiguous or Clear?

Decide if the expression in words is ambiguous or clear.

- If it is ambiguous, rewrite it in words two times - once with multiplication / division first, and once with addition / subtraction first.
- If it is clear, draw the Circle of Evaluation.

	Verbal Expression	Rewrite if ambiguous. Draw a Circle of Evaluation if clear.
1)	The product of 12 and 8 decreased by 5	
2)	The quotient of 36 and the sum of 10 and 8.	
3)	Half of 20 decreased by 6.	
4)	Increase the product of 10 and 2 by 7.	
5)	The difference between 20 and 8 multiplied by 2.	
6)	Seven more than one-third of 90.	

Introduction to Programming in a Nutshell

The **Editor** is a software program we use to write Code. Our Editor allows us to experiment with Code on the right-hand side, in the **Interactions Area**. For Code that we want to *keep*, we can put it on the left-hand side in the **Definitions Area**. Clicking the "Run" button causes the computer to re-read everything in the Definitions Area and erase anything that was typed into the Interactions Area.

Data Types

Programming languages involve different ***data types***, such as Numbers, Strings, Booleans, and even Images.

- Numbers are values like `1`, `0.4`, `1/3`, and `-8261.003`.
    - Numbers are *usually* used for quantitative data and other values are *usually* used as categorical data.

- Strings are values like `"Emma"`, `"Rosanna"`, `"Jen and Ed"`, or even `"08/28/1980"`.
    - All strings *must* be surrounded by quotation marks.

- Booleans are either `true` or `false`.

All values evaluate to themselves. The program `42` will evaluate to `42`, the String `"Hello"` will evaluate to `"Hello"`, and the Boolean `false` will evaluate to `false`.

Operators

Operators (like `+`, `-`, `*`, `<`, etc.) are treated the same way as functions: after all, they have inputs and outputs and obey the same rules!

Applying Functions

Functions (and operators!) work much the way they do in math. Every function has a name, takes some inputs, and produces some output. The function name is written first, followed by a list of ***arguments***.

- In math this could look like $f(5)$ or $g(10, 4)$.

- In WeScheme, these examples would be written as `(f 5)` and `(g 10 4)`.

- Applying the operator $+$ to the inputs 1 and 2 would look like `(+ 1 2)`.

- Applying a function to make images would look like `(star 50 "solid" "red")`.

- There are many other functions in WeScheme, for example `sqr`, `sqrt`, `triangle`, `square`, `string-repeat`, etc.

Functions have ***contracts***, which help explain how a function should be used. Every Contract has three parts:

- The *Name* of the function - literally, what it's called.

- The *Domain* of the function - what *type(s) of value(s)* the function consumes, and in what order.

- The *Range* of the function - what *type of value* the function produces.

Strings and Numbers

Make sure you've loaded WeScheme, clicked "Run", and are working in the ***Interactions Area*** *on the right. Hit Enter/return to evaluate expressions you test out.*

Strings

String values are always in quotes.

- Try typing your name *(in quotes!)* .
- Try typing a sentence like "I'm excited to learn to code!" *(in quotes!)* .
- Try typing your name with the opening quote, but *without the closing quote.* Read the error message!
- Now try typing your name *without any quotes.* Read the error message!

1) Explain what you understand about how strings work in this programming language. ______________________________

____

Numbers

2) Try typing `42` into the Interactions Area and hitting "Enter". Is `42` the same as `"42"` ? Why or why not?

____

3) What is the largest number the editor can handle?

____

4) Try typing `0.5` . Then try typing `.5` . Then try clicking on the answer. Experiment with other decimals.

Explain what you understand about how decimals work in this programming language. ______________________________

____

5) What happens if you try a fraction like `1/3` ? ____

____

6) Try writing **negative** integers, fractions and decimals. What do you learn? ______________________________

____

____

Booleans

Boolean-producing expressions are yes-or-no questions, and will always evaluate to either `true` ("yes") or `false` ("no").
What will the expressions below evaluate to? *Write down your prediction, then type the code into the Interactions Area to see what it returns.*

	Prediction	Result		Prediction	Result
1) `(<= 3 4)`			2) `(string>? "a" "b")`		
3) `(= 3 2)`			4) `(string<? "a" "b")`		
5) `(< 2 4)`			6) `(string=? "a" "b")`		
7) `(>= 5 5)`			8) `(string<>? "a" "a")`		
9) `(>= 4 6)`			10) `(string>=? "a" "a")`		
11) `(<> 3 3)`			12) `(string<>? "a" "b")`		
13) `(<> 4 3)`			14) `(string>=? "a" "b")`		

15) In your own words, describe what `<` does. ____

____

16) In your own words, describe what `>=` does. ____

____

17) In your own words, describe what `<>` does. ____

____

	Prediction:	Result:
18) `(string=? "a tree" "trees")`		
19) `(string=? "tree" "tree")`		
20) `(string-contains? "catnap" "cat")`		
21) `(string-contains? "cat" "catnap")`		

22) In your own words, describe what `string-contains` does. Can you generate another expression using `string-contains` that returns true?

____

★ There are infinite string values ("a", "aa", "aaa"...) and infinite number values out there (...-2,-1,0,-1,2...). But how many different *Boolean* values are there? ____

Applying Functions

Open WeScheme and click "Run". We will be working in the Interactions Area on the right.

Test out these two expressions and record what you learn below:

- `(regular-polygon 40 6 "solid" "green")`
- `(regular-polygon 80 5 "outline" "dark-green")`

1) You've seen data types like Numbers, Strings, and Booleans. What data type did the `regular-polygon` function produce? __________

2) How would you describe what a regular polygon is? ______________________________

____

3) The `regular-polygon` function takes in four pieces of information (called arguments). Record what you know about them below.

	Data Type	Information it Contains
Argument 1		
Argument 2		
Argument 3		
Argument 4		

There are many other functions available to us in Pyret. We can describe them using ***contracts***. The Contract for `regular-polygon` is:

```
; regular-polygon :: Number, Number, String, String -> Image
```

- Each Contract begins with the function name: *in this case* `regular-polygon`
- Lists the data types required to satisfy its Domain: *in this case* Number, Number, String, String
- And then declares the data type of the Range it will return: *in this case* Image

Contracts can also be written with more detail, by annotating the Domain with *variable names*:

`; regular-polygon :: ( Number , Number , String , String ) -> Image`

size number-of-sides fill-style color

4) We know that a square is a regular polygon because ______________________________

____

5) What code would you write to make a big, blue square using the `regular-polygon` function?

____________ (________ , ________ , ________ , ________)

function-name size :: Number number-of-sides :: Number fill-style :: String color :: String

6) Pyret also has a `square` function whose contract is: `; square :: ( Number , String , String ) -> Image`

size fill-style color

What code would you write to make a big blue square using the `square` function?

____________ (________ , ________ , ________)

function-name size :: Number fill-style :: String color :: String

7) Why does `square` need fewer arguments to make a square than `regular-polygon`? ______________________________

____

★ Where else have you heard the word ***contract*** used before?

Practicing Contracts: Domain & Range

Note: The contracts on this page are not defined in WeScheme and cannot be tested in the editor.

is-beach-weather

Consider the following Contract:

```
; is-beach-weather :: Number, String -> Boolean
```

1) What is the **Name** of this function? ______________________________

2) How many arguments are in this function's **Domain**? ______________________________

3) What is the **Type** of this function's **first argument**? ______________________________

4) What is the **Type** of this function's **second argument**? ______________________________

5) What is the **Range** of this function? ______________________________

6) Circle the expression below that shows the correct application of this function, based on its Contract.

A. `(is-beach-weather 70 90)`
B. `(is-beach-weather 80 100 "cloudy")`
C. `(is-beach-weather "sunny" 90)`
D. `(is-beach-weather 90 "stormy weather")`

cylinder

Consider the following Contract:

```
; cylinder :: Number, Number, String -> Image
```

7) What is the **Name** of this function? ______________________________

8) How many arguments are in this function's **Domain**? ______________________________

9) What is the **Type** of this function's **first argument**? ______________________________

10) What is the **Type** of this function's **second argument**? ______________________________

11) What is the **Type** of this function's **third argument**? ______________________________

12) What is the **Range** of this function? ______________________________

13) Circle the expression below that shows the correct application of this function, based on its Contract.

A. `(cylinder "red" 10 60)`
B. `(cylinder 30 "green")`
C. `(cylinder 10 25 "blue")`
D. `(cylinder 14 "orange" 25)`

Matching Expressions and Contracts

Match the Contract (left) with the expression that uses it correctly (right).
Note: The contracts on this page are not defined in Pyret and cannot be tested in the editor.

Contract			Expression
`; make-id :: String, Number -> Image`	1	A	`(make-id "Savannah" "Lopez" 32)`
`; make-id :: String, Number, String -> Image`	2	B	`(make-id "Pilar" 17)`
`; make-id :: String -> Image`	3	C	`(make-id "Akemi" 39 "red")`
`; make-id :: String, String -> Image`	4	D	`(make-id "Raïssa" "McCracken")`
`; make-id :: String, String, Number -> Image`	5	E	`(make-id "von Einsiedel")`

Contract			Expression
`; is-capital :: String, String -> Boolean`	6	A	`(show-pop "Juneau" "AK" 31848)`
`; is-capital :: String, String, String -> Boolean`	7	B	`(show-pop "San Juan" 395426)`
`; show-pop :: String, Number -> Image`	8	C	`(is-capital "Accra" "Ghana")`
`; show-pop :: String, String, Number -> Image`	9	D	`(show-pop 3751351 "Oklahoma")`
`; show-pop :: Number, String -> Number`	10	E	`(is-capital "Albany" "NY" "USA")`

Contracts for Image-Producing Functions

Log into WeScheme and click "Run". Experiment with each of the functions listed below, trying to find an expression that will build. Record the contract and example code for each function you are able to successfully build!

Name		Domain		Range
; triangle	::	Number, String, Sting	->	Image
(triangle 80 "solid" "green")				
; star	::		->	
; circle	::		->	
; rectangle	::		->	
; text	::		->	
; square	::		->	
; ellipse	::		->	
; regular-polygon	::		->	
; rhombus	::		->	
; right-triangle	::		->	
; isosceles-triangle	::		->	
; radial-star	::		->	
; star-polygon	::		->	
; triangle/sas	::		->	
; triangle/asa	::		->	

Catching Bugs when Making Triangles

Learning about a Function through Error Messages

1) Type `triangle` into the Interactions Area of WeScheme and hit "Enter". What do you learn? ______________________________

2) We know that all functions will need an open parenthesis and at least one input! Type `(triangle 80)` in the Interactions Area and hit Enter/return. Read the error message. What hint does it give us about how to use this function?

____

3) Using the hint from the error message, experiment until you can make a triangle. What is the contract for `triangle`?

____

What Kind of Error is it?

- **syntax errors** - the computer cannot make sense of the code because of unclosed strings, missing commas or parentheses, etc.
- **contract errors** - the function isn't given what it needs (the wrong type or number of arguments are used)

4) In your own words, the difference between **syntax errors** and **contract errors** is: ______________________________

____

Finding Mistakes with Error Messages

The following lines of code are all BUGGY! Read the code and the error messages below. See if you can find the mistake WITHOUT typing it into WeScheme.

5) `(triangle 20 "solid")`

```
triangle: expects 3 arguments, but given 2: 20 solid at: line 1, column 0, in <interactions>
```

This is a ______________ error. The problem is that ______________________________
contract / syntax

6) `(triangle "solid" "red" 20)`

```
triangle: expects a non-negative number as 1st argument, but given: solid; other arguments were:
red 20 at: line 1, column 0, in <interactions>
```

This is a ______________ error. The problem is that ______________________________
contract / syntax

7) `(triangle 20 40 "solid" "red")`

```
triangle: expects 3 arguments, but given 4: 20 40 solid red at: line 1, column 0, in
<interactions>
```

This is a ______________ error. The problem is that ______________________________
contract / syntax

8) `(triangle 20 solid "red")`

```
solid: this variable is not defined at: line 1, column 0, in <interactions>
```

This is a ______________ error. The problem is that ______________________________
contract / syntax

★ `(triangle 20 "striped" "red")`

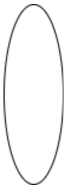
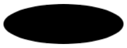
```
triangle: expects a style ("solid" / "outline") or an opacity value [0-255]) as 2nd argument,
but given: "striped"; other arguments were: 20 "red" at: line 1, column 0, in <interactions>
```

This is a ______________ error. The problem is that ______________________________
contract / syntax

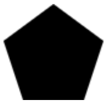
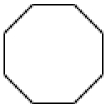
Using Contracts

For questions 1,2,4,5,8 & 9, use the contracts provided to find expressions that will generate images similar to the ones pictured. Test your code in WeScheme before recording it.

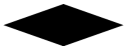
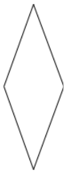
```
; ellipse :: (___Number___, ___Number___, ___String___, ___String___) -> Image
                 width         height       fill-style      color
```

1)		
2)		
3)	Write an expression using `ellipse` to produce a circle.	

```
; regular-polygon :: (___Number___, ___Number___, ___String___, ___String___) -> Image
                      side-length  number-of-sides  fill-style      color
```

4)		
5)		
6)	Use `regular-polygon` to write an expression for a square!	
7)	How would you describe a **regular polygon** to a friend?	

```
; rhombus :: (___Number___, ___Number___, ___String___, ___String___) -> Image
                  size        top-angle     fill-style      color
```

8)		
9)		
10)	Write an expression to generate a `rhombus` that is a square!	

Triangle Contracts

Respond to the questions. Go to WeScheme to test your code.

1) What kind of triangle does the `triangle` function produce? ______________________________

There are lots of other kinds of triangles! And WeScheme has lots of other functions that make triangles!

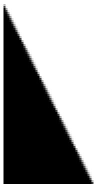
```
; triangle :: ( Number , String , String ) -> Image
                 size   fill-style  color
; right-triangle :: ( Number , Number , String , String ) -> Image
                       base    height fill-style  color
; isosceles-triangle :: ( Number , Number , String , String ) -> Image
                           leg     angle  fill-style  color
```

2) Why do you think `triangle` only needs one number, while `right-triangle` and `isosceles-triangle` need two numbers?

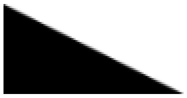
____

____

3) Write `right-triangle` expressions for the images below using `100` as one argument for each.

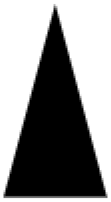


____

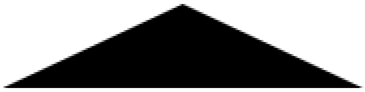


____

4) Write `isosceles-triangle` expressions for the images below using `100` as one argument for each.

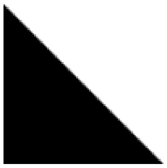


____



____

5) Write 2 expressions that would build **right-isosceles** triangles. Use `right-triangle` for one expression and `isosceles-triangle` for the other expression.



____

____

6) Which do you like better? Why? ______________________________

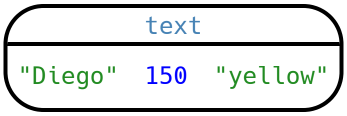
Composing with Circles of Evaluation

Notice and Wonder

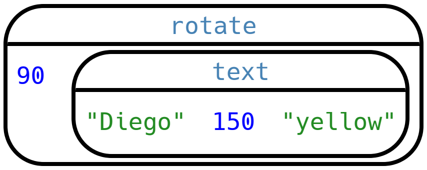
Suppose we want to see the `text` *"Diego" written vertically in yellow letters of size 150. Let's use Circles of Evaluation to look at the structure:*

We can start by generating the Diego image.

And then use the `rotate` function to rotate it 90 degrees.



→



```
(text "Diego" 150 "yellow")
```

```
(rotate 90 (text "Diego" 150 "yellow"))
```

1) What do you Notice? ____

____

2) What do you Wonder? ____

____

Let's Rotate an Image of Your Name!

Suppose you wanted the computer to show your name in your favorite color and rotate it so that it's diagonal...

Write your name (any size), in your favorite color

3) Draw the circle of evaluation:

5) Convert the Circle of Evaluation to code:

`rotate` **the image so that it's diagonal**

4) Draw the circle of evaluation:

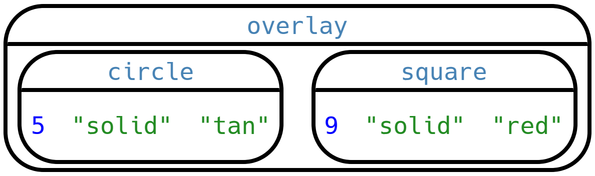
6) Convert the Circle of Evaluation to code:

Circle of Evaluation to Code (Scaffolded)

Complete the Code by Filling in the Blanks!

Finish the Code by filling in the blanks.

1)

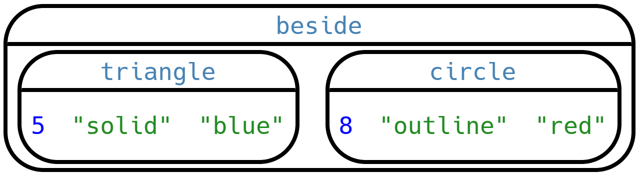


```
(overlay (circle ____ "solid" ____________________) ( ____________________ 9 ____________________ "red"))
```

Complete the Code by adding Parentheses

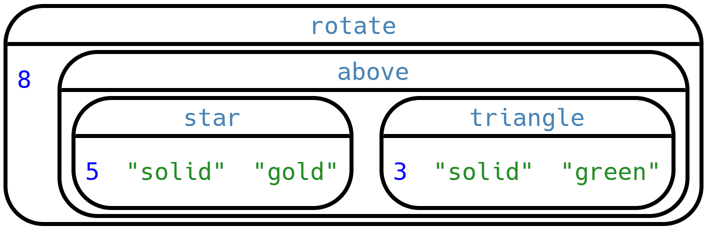
For each Circle of Evaluation, finish the Code by adding parentheses.

2)



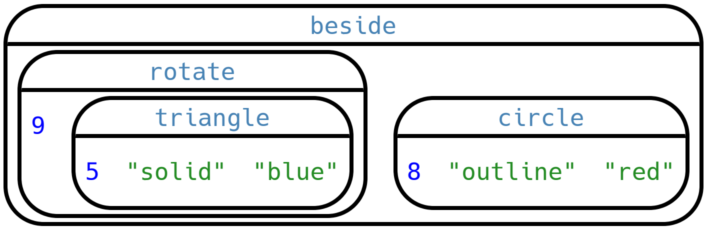
```
beside    triangle    5    "solid"    "blue"    circle    8    "outline" "red"
```

3)



```
rotate    8    above    star    5    "solid"    "gold"    triangle    3    "solid"    "green"
```

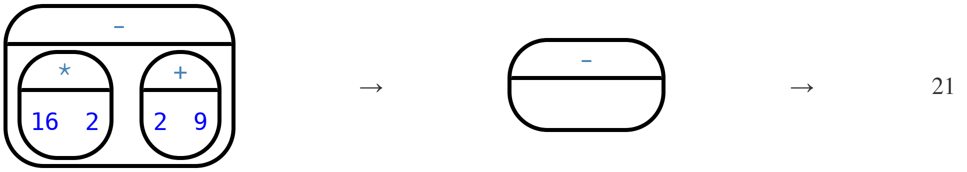
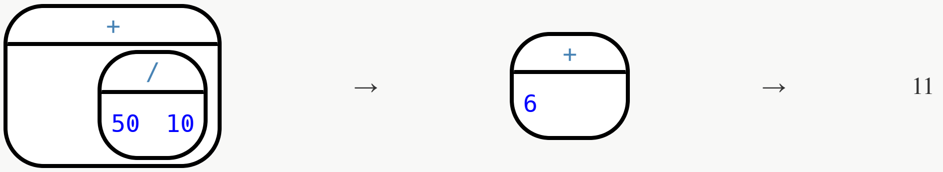
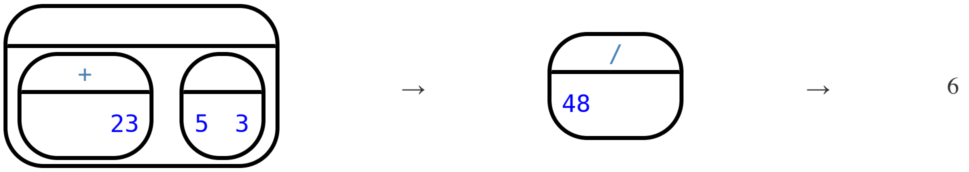
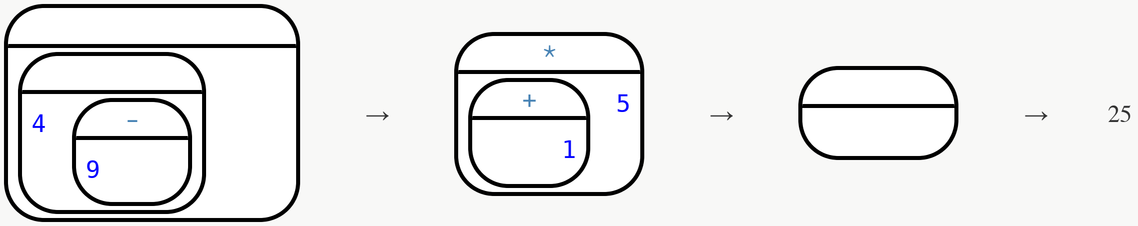
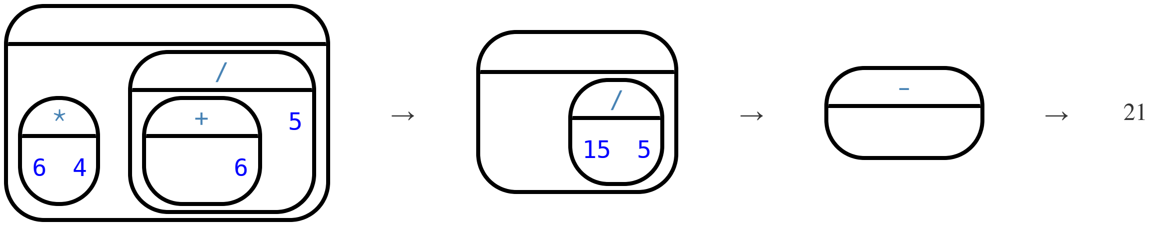
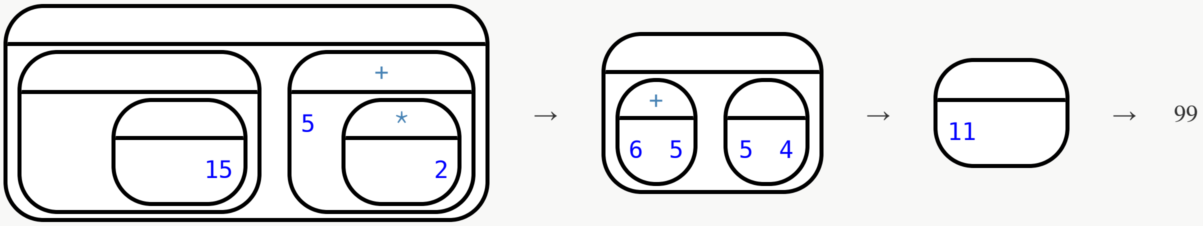
4)



```
beside    rotate    9    triangle    5    "solid"    "blue"    circle    8    "outline"    "red"
```

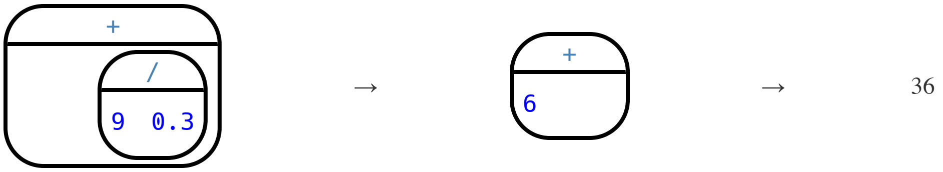
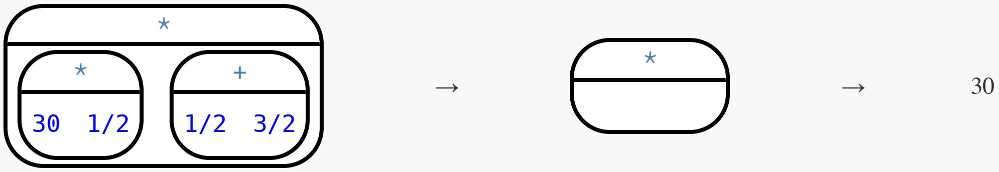
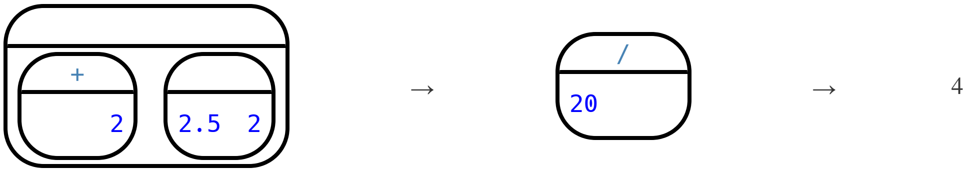
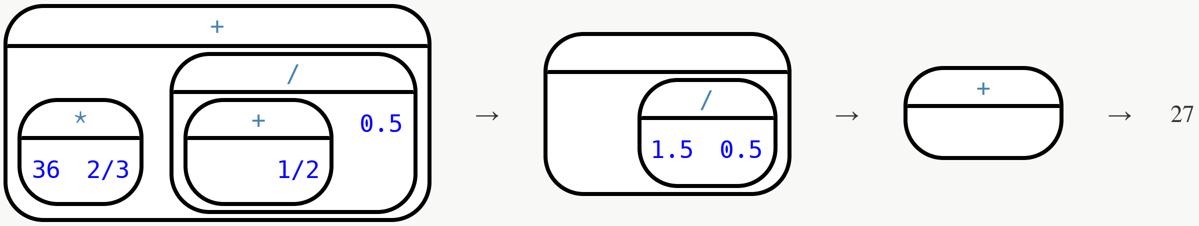
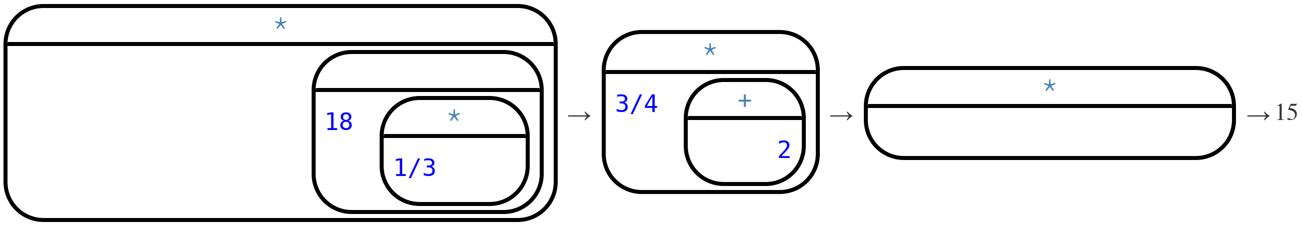
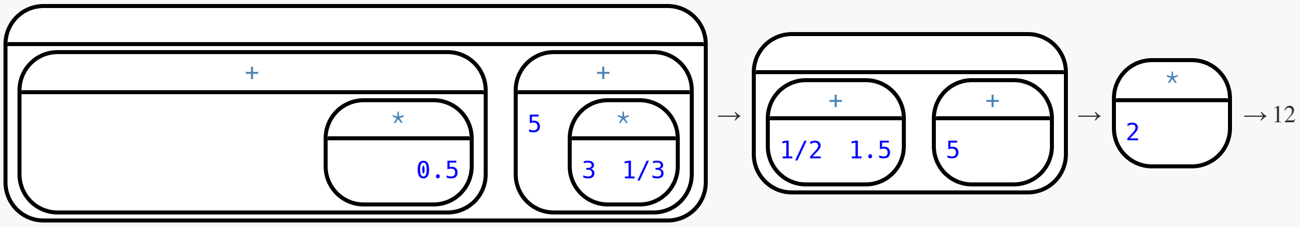
Computation (Whole Numbers)

The Circles of Evaluation below represent a step-by-step computation, which results in an answer. Some of the steps are missing numbers and operators! Fill in those numbers and operators so that each sequence of circles will end with the answer shown on the right.

	Circle Solving Diagram
1)	
2)	
3)	
4)	
5)	
6)	 15 + 5 * 2 → + 6 5 5 4 → 11 → 99

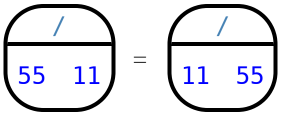
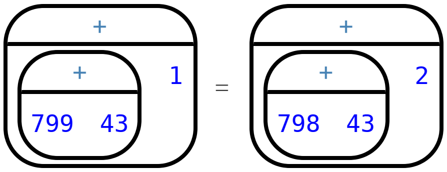
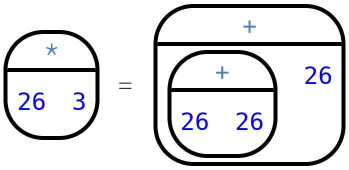
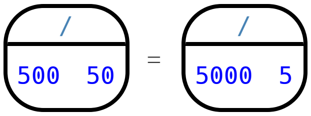
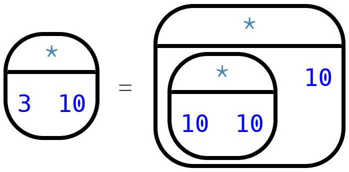
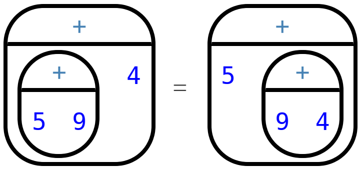
Computation (Fractions and Decimals)

The Circles of Evaluation below represent a step-by-step computation, which results in an answer. Some of the steps are missing numbers! Fill those numbers in so that each sequence of circles will end with the answer shown on the right.

	Circle Solving Diagram
1)	
2)	
3)	
4)	 + * 36 2/3 / + 1/2 0.5 → / 1.5 0.5 → + → 27
5)	
6)	 + * 0.5 + 5 * 3 1/3 → + 1/2 1.5 + 5 → * 2 → 12

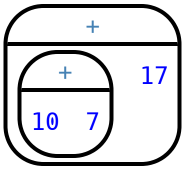
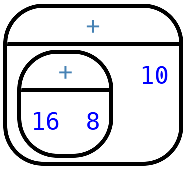
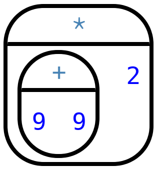
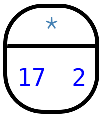
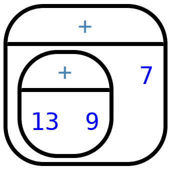
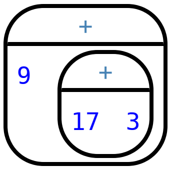
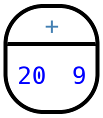
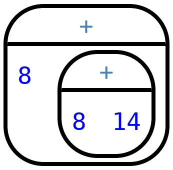
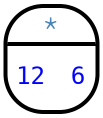
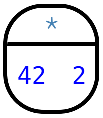
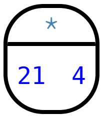
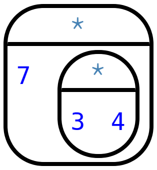
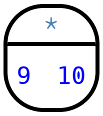
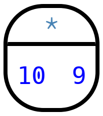
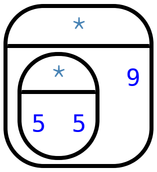
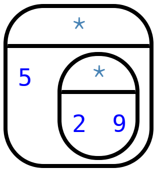
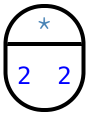
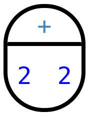
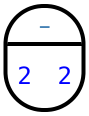
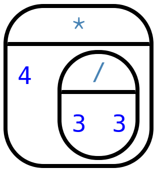
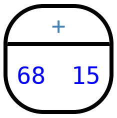
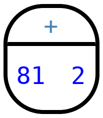
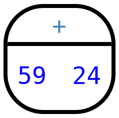
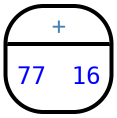
True or False? Computation

Is the equation represented by the two Circles true or false? Explain your response.

	Circles	True or false? Explain.
1)		
2)		
3)		
4)		
5)		
6)		

Which One Doesn’t Belong? Computation

Cross out the Circle that does NOT belong with the others, and then explain your choice.

	Which one doesn’t belong?	Explain
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	
6)	   	

Are They Identical?

Are the images produced by the two lines of code identical - or will they look different? With your partner, make a prediction, referring to your contracts as needed. Test the code in WeScheme. Explain your response. We've partially completed the first one for you.

	Code	Prediction	Result	Explain

1)	`(square 100 "solid" "red")`
	`(square (* 50 50) "solid" "red")`
			No	The shapes are not equivalent because 50 * 50 is not equal to 100.

2)	`(square 40 "solid" "deeppink")`
	`(rectangle 40 40 "solid" "deeppink")`			

3)	`(radial-star 6 20 50 "solid" "red")`
	`(radial-star 20 6 50 "solid" "red")`			

4)	`(radial-star 6 20 50 "solid" "red")`
	`(radial-star (+ 3 3) (* 4 5) 50 "solid" "red")`			

5)	`(circle 60 "outline" "tomato")`
	`(ellipse 60 60 "outline" "tomato")`			

6)	`(star 75 "outline" "pink")`
	`(star 75 "pink" "outline")`			

7)	`(circle 60 "solid" "lime")`
	`(ellipse 120 120 "solid" "lime")`			

Writing Equivalent Code

After testing the provided line of code in WeScheme, write a *different*, equivalent line of code (one that produces an identical image). Refer to your contracts as needed. You may find it useful to draw Circles of Evaluation as you develop your code.

	Provided Code	Your Code
1)	`(ellipse 80 80 "solid" "violet")`	`(circle ______ ______________ ______________)`
2)	`(square 95 "outline" "olive")`	`(rectangle ______ ______ ______________ ______________)`
3)	`(rotate 270 (rectangle 20 50 "solid" "blue"))`	`(rectangle ______ ______ ______________ ______________)`
4)	`(rotate 135 (isosceles-triangle 100 90 "solid" "black"))`	`(right-triangle ______________ ______________ ______________ ______________)`
5)	`(square 60 "solid" "red")`	`(scale ______ ____________________________)`
★	`(flip-horizontal (flip-vertical (text "Azara" 150 "yellow")))`	`(rotate ______________ ____________________________)`

Laws of Arithmetic

Laws of arithmetic can be applied to expressions with numbers and/or variables. By applying laws of arithmetic, we can determine if algebraic expressions are equivalent without assigning values to the variables.

The Commutative Property. For expressions involving only addition or only multiplication, changing the order of the numbers will not change the result. 4×3 is equivalent to 3×4 on account of the Commutative Property of Multiplication. The Commutative Property does not hold for subtraction or division.

The Associative Property. When adding three numbers or multiplying three numbers, it does not matter whether you start with the first pair or the last. The same is true when either adding or multiplying four numbers, five numbers, etc.

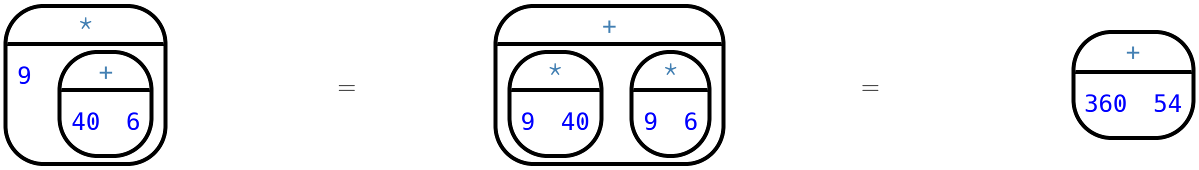
The Associative Property often results in simpler mental computations. For instance, $(14 + 6) + 7 + 2$ is simpler to evaluate than $14 + (6 + (7 + 2))$, although they both produce the same result of 29 .

The Additive Inverse Property. Adding a number and its opposite always produces zero. For instance, $8 + \text{-}\,8 = 0$.

The Multiplicative Inverse Property. Multiplying a number and its reciprocal always produces 1. For instance, $8 \times \frac{1}{8} = 0.$

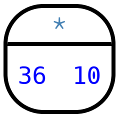
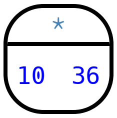
The Identity Property. Multiplying or dividing an expression by 1 does not change its value; similarly, adding or subtracting 0 results in the original value. Due to the Identity Property, $5 + 9$ produces the same result as $(5 + 9) \times 0$ and $(5 + 9) + 0$.

The Distributive Property. Multiplying the sum of two addends by a number produces the same result as multiplying *each* addend by that number before finding the sum. In other words: $a \times (b + c) = ab + ac$. For instance:



Applying the Distributive Property often results in simpler computations that can be completed using mental math.

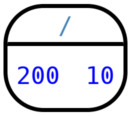
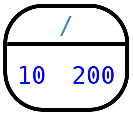
Discover the Commutative Property (1)

	
36 × 10 = ?	10 × 36 = ?

1) What do you notice about the Circles of Evaluation above?

These Circles of Evaluation demonstrate the Commutative Property of Multiplication! **The Commutative Property is true for any expression where all *orders* of the numbers produce the same result.** Draw another example of the Commutative Property of Multiplication with any two numbers, below. Evaluate each Circle to confirm that they are equivalent.

2) Examine and evaluate the Circles of Evaluation below to decide if the Commutative Property holds for problems involving **division**.

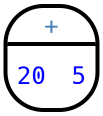
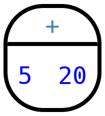
	
200 ÷ 10 = ?	10 ÷ 200 = ?

Explain your response.

Draw another example like the one above to confirm what you observed about the Commutative Property and division.

Discover the Commutative Property (2)

1) Now look at two more Circles of Evaluation to decide if the Commutative Property holds for problems involving **addition**.

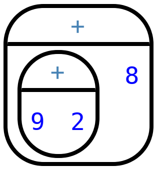
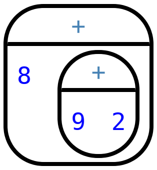
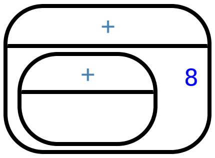
	
20 + 5 = ?	5 + 20 = ?

What do you notice? ____

____

____

2) Now look at two more Circles of Evaluation to see how the Commutative Property holds for problems involving **addition of three values**. Can you fill in a third Circle so that the *order* changes, but not the *groupings*?

		
(9 + 2) + 8 = ?	8 + (2 + 9) = ?	

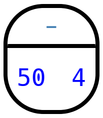
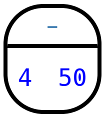
What do you notice? ____

____

____

These Circles of Evaluation *all* represent the Commutative Property of Addition! Notice how, when we used three values, there were multiple ways of reconfiguring the numbers. (*Do you think that is true, also, for the Commutative Property of Multiplication?*)

3) Evaluate the Circles below to decide if the Commutative Property holds for problems involving **subtraction**.

	
50 - 4 = ?	4 - 50 = ?

Explain your response. ____

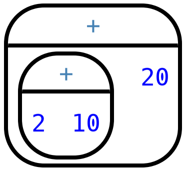
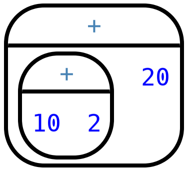
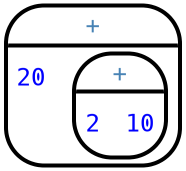
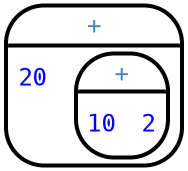
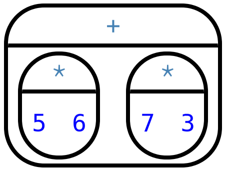
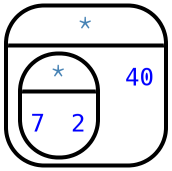
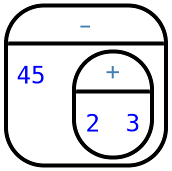
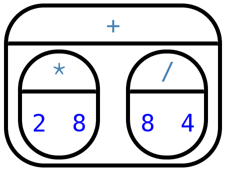
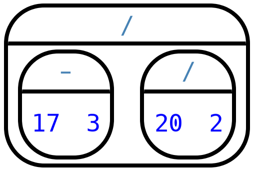
____

____

4) On a separate page, draw two additional examples - one pair of Circles that confirms what you observed about the Commutative Property and *addition*, and another pair of Circles that confirms what you observed about the Commutative Property and *subtraction*. Evaluate each Circle to verify your response.

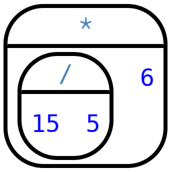
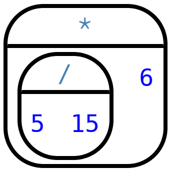
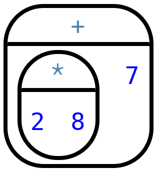
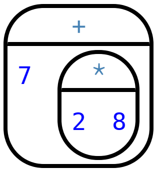
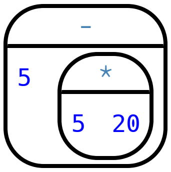
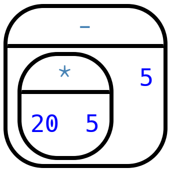
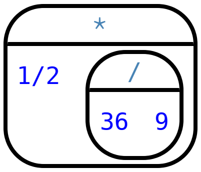
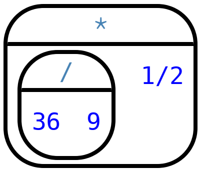
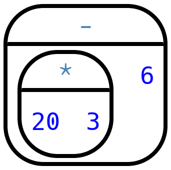
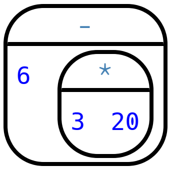
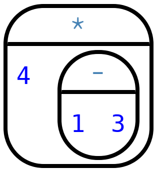
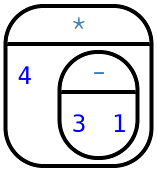
Commutative Property Table

For each Circle of Evaluation, apply the Commutative Property as many times as you can in order to produce equivalent expressions. You may fill as many or as few of the boxes in a row as is appropriate.

	Circle of Evaluation	Equivalent Circle 1	Equivalent Circle 2	Equivalent Circle 3
1)				
2)				
3)				
4)				
5)				
6)				

Which Circle of Evaluation is Correct?

For each of the expressions in words, look at the Circles of Evaluation that Claire and Walker drew. Then, decide who is correct: Claire, Walker, or both? Justify your response.

	Expression in words:	Claire's Circle:	Walker's Circle:	Who is correct? Justify.
1)	Find the quotient of 15 and 5. Multiply it by 6.			
2)	Double 8. Now add 7.			
3)	5 less than the product of 5 and 20.			
4)	One half of the quotient of 36 and 9.			
5)	Subtract 6 from 20 tripled.			
6)	The product of 4 and the difference of 3 and 1.			

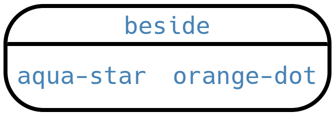
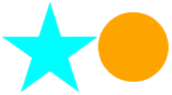
Commutativity and Code (Images)

Open the Commutativity and Associativity Starter File, which you will use to investigate three functions:

```
; beside :: Image, Image -> Image
; above :: Image, Image -> Image
; overlay :: Image, Image -> Image
```

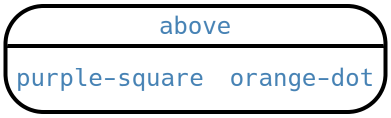
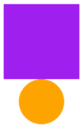
For each function, draw a second Circle of Evaluation that changes the order of the arguments. Translate the Circles of Evaluation to code, then sketch the image that you think your Circle will return. Finally, test your code in Pyret.

Is **beside** Commutative?

Circle of Evaluation		
Code	`(beside aqua-star orange-dot)`	
Sketch		

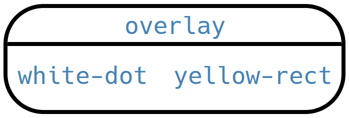
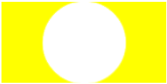
1) Did both expressions produce *identical* images? ______________ Is the `beside` function commutative? ______________

Is **above** Commutative?

Circle of Evaluation		
Code	`(above purple-square orange-dot)`	
Sketch		

2) Did both expressions produce *identical* images? ______________ Is the `above` function commutative? ______________

Is **overlay** Commutative?

Circle of Evaluation		
Code	`(overlay white-dot yellow-rect)`	
Sketch		

3) Did both expressions produce *identical* images? ______________ Is the `overlay` function commutative? ______________

Commutativity and Code (String, Number, Color Blending)

Open the Commutativity and Associativity Starter File, which you will use to investigate four functions:

```
; string-contains :: String, String -> Boolean        ; max :: Number, Number -> Number
; min :: Number, Number -> Number                     ; blend-images :: Image, Image -> Image
```

For each function, draw a second Circle of Evaluation that changes the order of the arguments. Translate the Circles of Evaluation to code, then sketch the image that you think your Circle will return. Finally, test your code in Pyret.

Is `string-contains` Commutative?

Circle of Evaluation	string-contains "rainbow" "bow"	
Code	(string-contains "rainbow" "bow")	
Result	true	

1) Did both expressions produce *identical* images? ______________ Is the `string-contains` function commutative? ______________

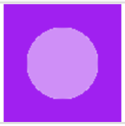
Is `min` Commutative?

Circle of Evaluation	min 200 23	
Code	(min 200 23)	
Result	23	23

2) Did both expressions produce the same result? ______________ Is the `min` function commutative? ______________

3) Make a prediction. Do you think `max` is commutative? ________________ Test your prediction. Were you right? __________________

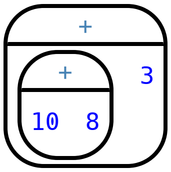
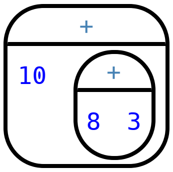
Is `blend-images` Commutative?

Circle of Evaluation	blend-images purple-square white-dot	blend-images white-dot purple-square
Code	(blend-images purple-square white-dot)	
Sketch		

4) Did both expressions produce *identical* images? ______________ Is `blend-images` commutative? ______________

Discover the Associative Property (1)

1) Evaluate the Circles of Evaluation below to help you decide if the Associative Property holds for problems involving **addition**.

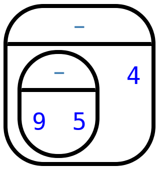
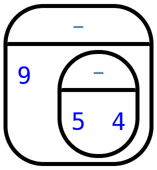
	
(10 + 8) + 3 = ?	10 + (8 + 3) = ?

What do you notice? ____

____

These Circles of Evaluation represent the Associative Property of Addition, which tells us that **when you add three numbers, it does not matter whether you start by adding the first pair of numbers or the last pair of numbers.** Draw another example of the Associative Property of Addition with any **three** numbers, below. Evaluate each expression to confirm that they are equivalent.

2) Evaluate the Circles of Evaluation below to help you decide if the Associative Property holds for problems involving **subtraction**.

	
(9 - 5) - 4 = ?	9 - (5 - 4) = ?

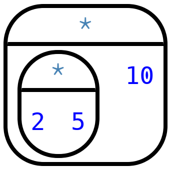
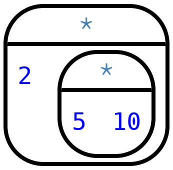
Explain your response. ____

____

Draw another example like the one above to confirm what you observed about the Associative Property and subtraction. Evaluate each Circle to confirm your response.

Discover the Associative Property (2)

1) Evaluate the three Circles of Evaluation below to help you decided if the Associative Property holds for problems involving **multiplication**:

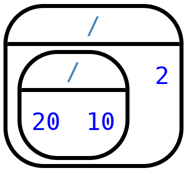
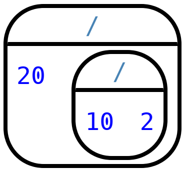
	
(2 × 5) × 10 = ?	2 × (5 × 10) = ?

What do you notice? ____

____

These Circles of Evaluation illustrate the Associative Property of Multiplication, which tells us that **when you multiply three numbers, it does not matter whether you start by multiplying the first pair of numbers or the last pair of numbers**. Draw another example of the Associative Property of Multiplication with any three numbers, below. Make sure that each expression includes a *different* pair of numbers grouped together. Evaluate your expressions to confirm that they are equivalent.

2) Evaluate the Circles of Evaluation below to help you decide whether or not the Associative Property holds for problems involving **division**.

	
(20 ÷ 10) ÷ 2 = ?	20 ÷ (10 ÷ 2) = ?

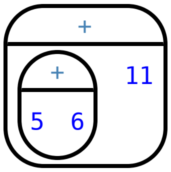
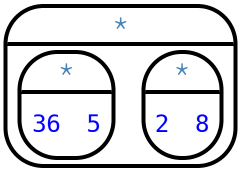
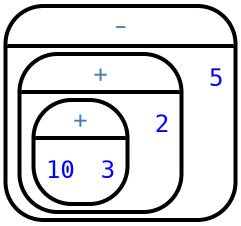
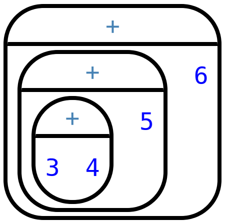
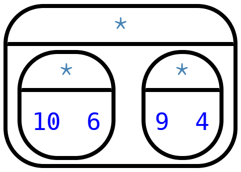
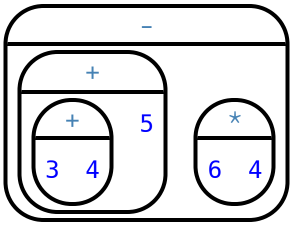
Explain your response. ____

____

Draw another example like the one above to confirm what you observed about the Associative Property and division.

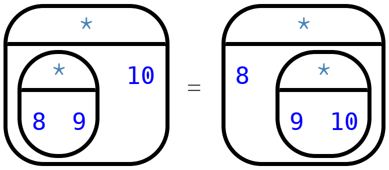
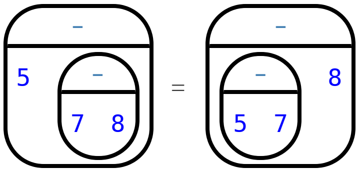
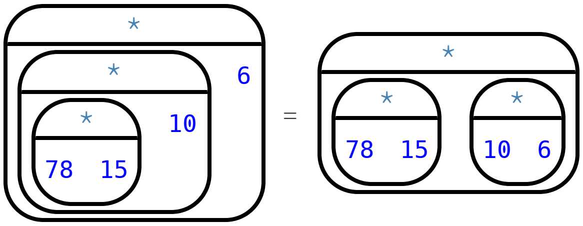
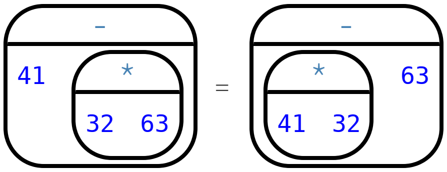
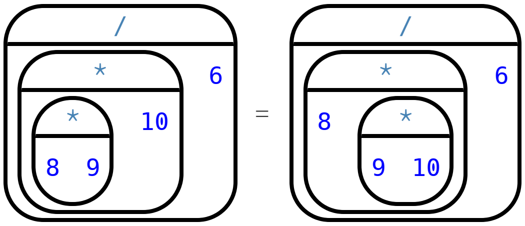
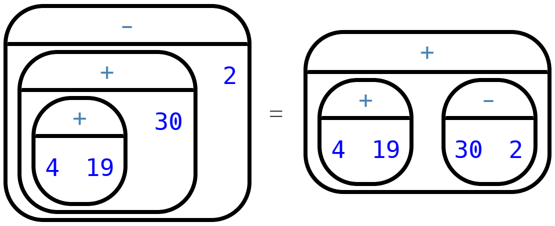
Associative Property Table

For each Circle of Evaluation, apply the Associative Property to create two equivalent Circles. Make sure you use different *groupings* in each Circle. Note: There are multiple possible responses here!

	Original Circle of Evaluation	Equivalent Circle of Evaluation 1	Equivalent Circle of Evaluation 2
1)			
2)			
3)			
4)			
5)			
6)			

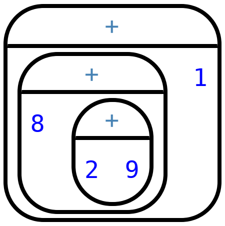
True or False? Associative Property

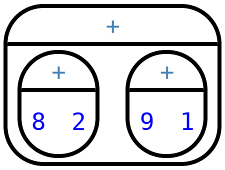
Is the equation represented by the two Circles of Evaluation true or false? Explain your response.

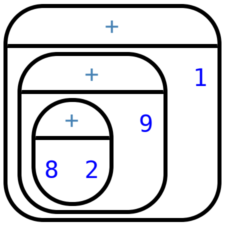
<table>
<tr><th></th><th>Circles of Evaluation</th><th>True or False? Explain</th></tr>
<tr><td>1)</td><td>
</td><td></td></tr>
<tr><td>2)</td><td>
</td><td></td></tr>
<tr><td>3)</td><td>
</td><td></td></tr>
<tr><td>4)</td><td>
</td><td></td></tr>
<tr><td>5)</td><td>
</td><td></td></tr>
<tr><td>6)</td><td>
</td><td></td></tr>
</table>

So Many Groupings!

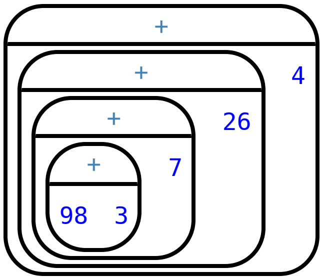
1) *Example.* How many different ways can you group the following expression: $8 + 2 + 9 + 1$? Below are three possibilities. For each example, order stays the same, but groupings change. Can you think of any more?

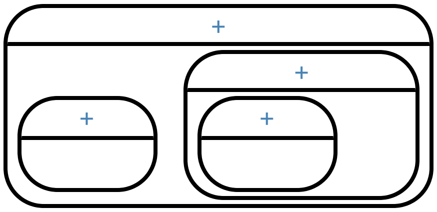






2) Your turn! Draw as many equivalent Circles of Evaluation as you can in the boxes below for the expression $98 + 3 + 7 + 26 + 4$. Then, answer the questions at the bottom of the page. For each equivalent expression, **change the groupings but not the order**. To get you going, we've completed one sample Circle of Evaluation and started a second one.





3) Which Circle (above) seems like it would be the *most difficult* to solve in your head? **Put a star by that one.** (You may award more than one star if it feels right.) Then, in the space below, explain what makes that Circle challenging to evaluate. ______________________________

____

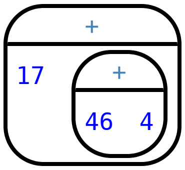
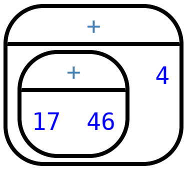
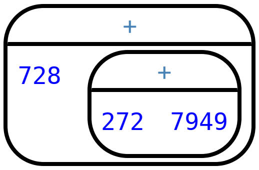
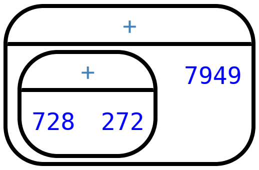
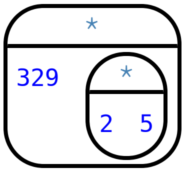
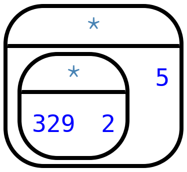
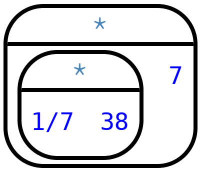
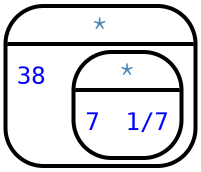
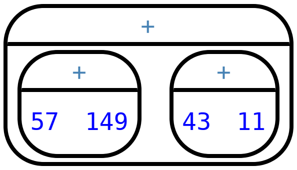
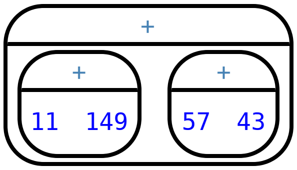
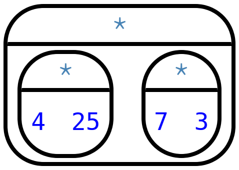
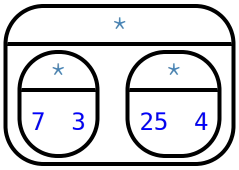
____

4) Which Circle of Evaluation seems like it would be the *easiest* to solve in your head? **Put a check mark by that one.** (You may award more than one star if it feels right.) Then, in the space below, explain what makes that one easier to evaluate. ______________________________

____

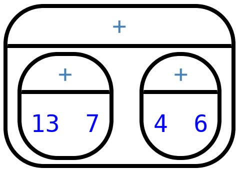
Which Grouping Makes the Computation Easier?

Put a check mark by the Circle of Evaluation which applies the Associative Property to make computation simpler. Then, evaluate the expression.

	Arithmetic Expression	Option 1	Option 2	Evaluate
1)	17 + 46 + 4			
2)	728 + 272 + 7949			
3)	329 × 2 × 5			
4)	$^{1}/_{7} \times 38 \times 7$			
5)	57 + 149 + 43 + 11			
6)	4 × 3 × 25 × 7			

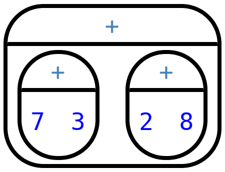
Associativity Makes Computation Easier (1)

Apply the Associative Property to draw the Circle of Evaluation that will make the computation the simplest. Evaluate the expression. The first one is done for you.

$13 + 7 + 4 + 6$  *20 + 10* *30*	$23 + 17 + 31 + 14$
$13 \times 125 \times 8$	$60 + (74 \times 5 \times 2)$
$(15 \times 25 \times 4) + 13 \times 20 \times 5$	$2 + (33 \times 5 \times 2)$
$468 \times 0.5 \times 20$	$^{7}/_{9} + ^{2}/_{9} + 223 + 7$

Restructuring Addition Expressions

For each addition expression, re-order and re-group so that solving is easier. Represent your simpler expression as a Circle of Evaluation, then evaluate. We've done the first one for you.

	Original Expression	Equivalent Circle of Evaluation	Solution
1)	7 + 8 + 2 + 3		20
2)	21 + 75 + 79		
3)	25 + 49 + 11 + 75		
4)	24 + 65 + 6		
5)	125 + 38 + 75 + 2		
6)	450 + 770 + 550 + 230		

Restructuring Multiplication Expressions

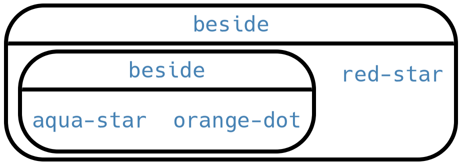
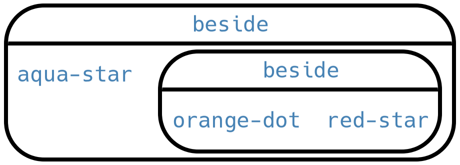
For each multiplication expression, re-order and re-group so that solving is easier. Represent your simpler expression in a Circle of Evaluation, then evaluate.

	Original Expression	Equivalent Circle of Evaluation	Solution
1)	25 × 27 × 4		
2)	5 × 133 × 2		
3)	200 × 38 × 5		
4)	2 × 87 × 50 × 10		
5)	5 × 9 × 4 × 7 × 5		
6)	25 × 5 × 20 × 3		

Associativity and Code

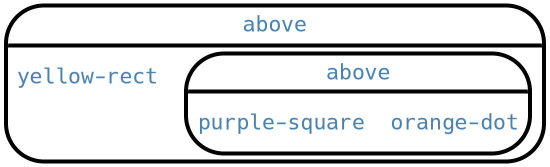
Open the Commutativity and Associativity Starter File. Complete the exploration to determine if `beside`, `above`, and `overlay` are associative.

1) **Is `beside` associative?** Make a prediction, then translate the two Circles of Evaluation into code. Test your code in WeScheme.

Circle of Evaluation		
Code		

2) Did both expressions produce *identical* images? ______________ Is the `beside` function associative or not? ____

3) **Is `above` associative?** Make a prediction, then draw a second Circle of Evaluation that changes the *grouping* without changing the *order*. Translate into code and test in WeScheme.

Circle of Evaluation	 above yellow-rect above purple-square orange-dot	
Code		

4) Did both expressions produce *identical* images? ______________ Is the `above` function associative or not? ____

5) **Is `overlay` associative?** Draw two Circles of Evaluation that will help you decide if `overlay` is associative. Translate into code and test in WeScheme.

Circle of Evaluation		
Code		

6) Did both expressions produce *identical* images? ______________ Is the `overlay` function associative or not? ____

Categorizing Functions

Discuss each function with a partner before categorizing as: Associative, Commutative, Both or Neither.

	Function	Associative	Commutative	Both	Neither
1)	`overlay`				
2)	`beside`				
3)	`above`				
4)	`blend-images`				
5)	`string-contains`				
6)	`min`				
7)	`rectangle`				
8)	`triangle`				
9)	`+`				
10)	`-`				
11)	`*`				
12)	`/`				

13) Which functions were *only* commutative? ____

____

14) Which functions were *only* associative? ____

____

15) Which functions were *both* commutative and associative? ______________________________

____

16) Which functions were *neither* commutative nor associative? ____________________________

____

17) Consider the operators listed in rows 9-12 of the table. Do these operators have different categorizations (Associative, Commutative, Both, Neither) in WeScheme versus in math? ____

18) What else did you Notice while completing the above table? What did you Wonder? ______________________

____

____

____

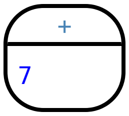
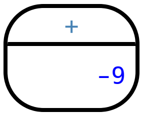
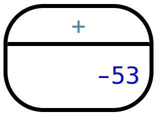
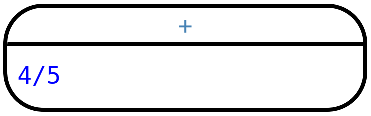
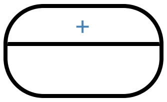
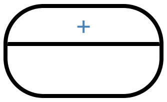
____

The Additive Inverse Property

Fill in the missing numbers to complete each equation. The last row includes some challenge problems!

$12 + \_\_\_\_\_\_\_\_ = 0$	$254 + \_\_\_\_\_\_\_\_ = 0$	$\_\_\_\_\_\_\_\_ + 33 = 0$
$-72 + \_\_\_\_\_\_\_\_ = 0$	$\_\_\_\_\_\_\_\_ + -240 = 0$	$2.5 + \_\_\_\_\_\_\_\_ = 0$
$^{2}/_{3} + \_\_\_\_\_\_\_\_ = 0$	$-27 + \_\_\_\_\_\_\_\_ = 0$	$\_\_\_\_\_\_\_\_ + 32.35 = 0$
$24 + 6 + \_\_\_\_\_\_\_\_\_\_\_\_ = 0$	$-f + \_\_\_\_\_\_\_\_ = 0$	$\_\_\_\_\_\_\_\_\_\_\_\_ + -g + -h = 0$

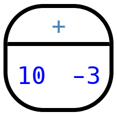
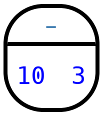
Fill in the missing number in each Circle of Evaluation to complete the equations. For the last two, create your own equations.

 = 0	 = 0
 = 0	 = 0
 = 0	 = 0

Discover Inverse Operations: Addition & Subtraction

From Adding to Subtracting

1) Under each Circle of Evaluation, write the equivalent arithmetic expression.

<table>
<tr><td>
</td><td>=</td><td>
</td></tr>
<tr><td>+</td><td>=</td><td></td></tr>
</table>

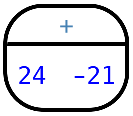
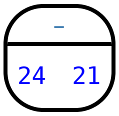
2) What will the first Circle evaluate to? __________ The second Circle? __________

3) What do you Notice? What do you Wonder? ____

____

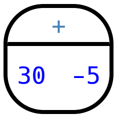
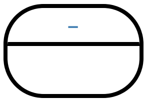
Did you know that adding - 3 **is the same as subtracting** 3**?**

4) Under each Circle of Evaluation, write the equivalent arithmetic expression.

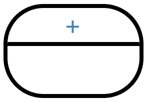
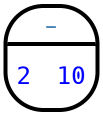
<table>
<tr><td>
</td><td>=</td><td>
</td></tr>
<tr><td>+</td><td>=</td><td></td></tr>
</table>

5) What will the first Circle evaluate to? __________ The second Circle? __________

6) In the table below, fill in the blanks to demonstrate how *adding a negative* produces the same result as *subtracting a positive.* Then write the equivalent arithmetic expression for each of your Circles.

<table>
<tr><td>
</td><td>=</td><td>
</td></tr>
<tr><td>+</td><td>=</td><td>-</td></tr>
</table>

7) This time, we've completed the *subtraction* Circle of Evaluation. You fill in the equivalent addition Circle of Evaluation.

<table>
<tr><td>
</td><td>=</td><td>
</td></tr>
<tr><td>+</td><td>=</td><td>-</td></tr>
</table>

Practice

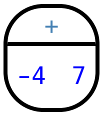
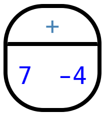
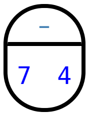
Rewrite addition as subtraction, and subtraction as addition.

30 - 12 = ___30 + - 12___	24 + - 4 = __________	100 - 101 = __________
__________ = 6 - 18	__________ = 0 + - 12	__________ = 6 - 36

Discover Inverse Operations: Addition & Subtraction (2)

What if the expression *starts* with a negative value?

1) In the example below, we've applied the Commutative Property and then the Additive Inverse Property to rewrite an addition expression as subtraction.

Start here:		Apply the Commutative Property:		Apply the Additive Inverse Property:
	→		→	
- 4 + 7		7 + - 4		7 - 4

2) What will the first Circle evaluate to? ___3___ The second Circle? ___3___ The third Circle? ___3___

3) What do you Notice about these three Circles? What do you Wonder? ____

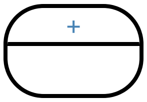
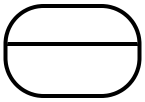
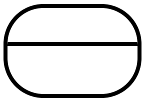
____

____

4) Look at the worksheet you just completed. Why is there an additional step in rewriting the addition expressions above? ______________

____

5) In the table below, draw another example like the one above to show how we can rewrite more complex addition expressions as subtraction.

Start here:		Apply the Commutative Property:		Apply the Additive Inverse Property:
	→		→	
+				

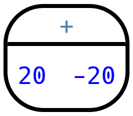
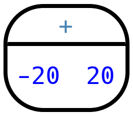
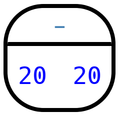
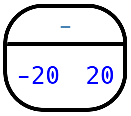
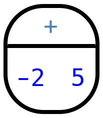
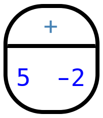
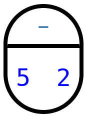
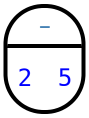
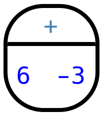
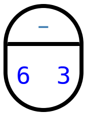
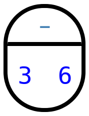
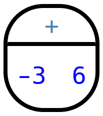
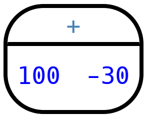
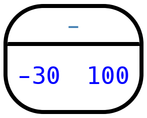
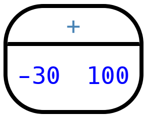
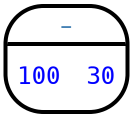
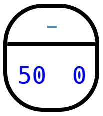
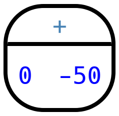
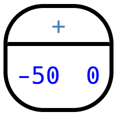
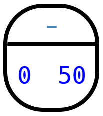
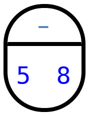
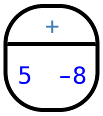
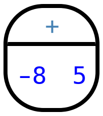
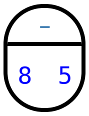
Try it out

Rewrite addition as subtraction, and subtraction as addition.

- 50 + 5 = ________________	- 8 + 4 = ________________	100 - 101 = ________________
________________ = 6 - 18	________________ = 0 + - 12	________________ = 6 - 36

Which One Doesn’t Belong?

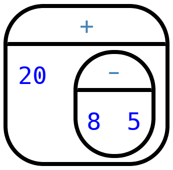
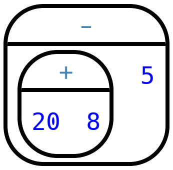
Identify the Circle of Evaluation that does not belong with the others.

	Circles of Evaluation
1)	   
2)	   
3)	   
4)	   
5)	   
6)	   

Subtract First... or Solve Left-to-Right?

For each expression, draw *two* unique Circles of Evaluation.

- In the column on the left, draw a Circle to illustrate evaluating subtraction first, followed by addition. In other words, evaluate from right-to-left.
- In the column on the right, draw a Circle to illustrate evaluating from left to right.
- Evaluate each Circle and make a note of whether or not they produce the same result. We've done the first one for you.

	Expression	Subtract First	Solve Left-to-Right
1)	20 + 8 - 5		
2)	4 + 3 - 2		
3)	12 + 9 - 8		
4)	64 + 92 - 91		
5)	Come up with an example of your own! See if you can come up with an unusual expression.		

6) What did you Notice? What do you Wonder? ______________________________

____

____

7) Can you change the groupings or the order in which you evaluate an expression like this one: 100 - 20 - 5 ? Do you get the same answer when you solve left-to-right and right-to-left? Which way is correct? ______________________________

____

____

Introduction to Examples (Additive Inverse)

Use the Additive Inverse Starter File to complete this page. **Do not click "Run" yet.**

1) In the table below, record your Noticings and Wonderings about what you see in the Definitions Window (left side) of the Additive Inverse Starter File.

Notice	Wonder

2) Click "Run." At the top of the Interactions Area on the right side, a message appears that says, **"7 TESTS PASSED, 4 TESTS FAILED."** Summarize the remaining information that appears by filling in the blanks, below.

- In the first examples block , all 5 tests ____________________.
- In the second examples block the examples at line ________ and ________ failed.
- In the third examples block , the test ____________________.

3) **First, let's explore `examples-block-1`.** In your own words, describe *why* each of the tests in `examples-block-1` passed. __________

____

____

4) Below, place a checkmark next to each of the examples that *passed* from `examples-block-2`.

```
____  (EXAMPLE (- 30 5) (+ 30 5))
____  (EXAMPLE (+ 11 -9) (- 11 9))
____  (EXAMPLE (- 5 30) (- 5 -30))
____  (EXAMPLE (+ 24 -4) (- 24 -4))
____  (EXAMPLE (- 60 55) (+ 60 -55))
```

5) Edit each of the **failing** examples on the left so that *all examples pass* when you click "Run". Be sure to change **only** the part of the example after the `is`! Describe one of the changes you had to make. ______________________________

____

____

6) **Let's explore examples-block-3.** Below line 26 (`3: My Own Example Block`), create a block of 5 passing examples of your own. Hit "Run" to see if your examples pass. If not, revise them until they do. If you encountered an error message along the way, describe it here:

____

____

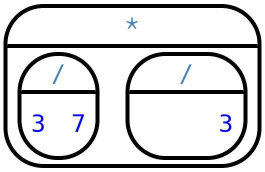
____

The Multiplicative Inverse Property

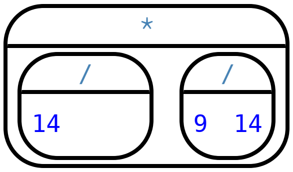
Fill in the missing numbers to complete each equation. The last row includes some challenge problems!

$1 = \frac{3}{\ } \times \frac{5}{3}$	$\frac{8}{9} \times \frac{9}{\ } = 1$	$1 = \frac{\ }{3} \times \frac{3}{10}$
$\frac{\ }{7} \times \frac{7}{4} = 1$	$10 \times \frac{\ }{10} = 1$	$1 = \frac{1}{\ } \times 9$
$1 = \underline{\hspace{4em}} \times {}^{20}/_{19}$	$650 \times \underline{\hspace{4em}} = 1$	$1 = \underline{\hspace{4em}} \times {}^{26}/_{27}$
$2\,{}^{1}/_{3} \times \underline{\hspace{4em}} = 1$	$1 = 4.5 \times \underline{\hspace{4em}}$	$\underline{\hspace{4em}} \times \frac{1}{w} = 1$

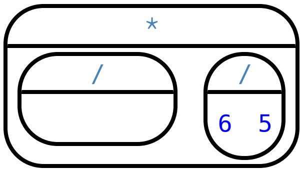
Fill in the missing number in each Circle of Evaluation to complete the equations. For the last two, create your own equations.



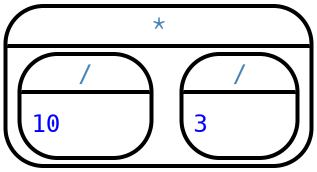
$= 1$



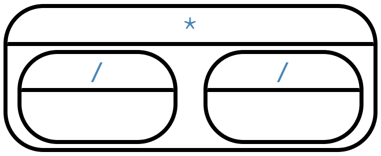
$= 1$



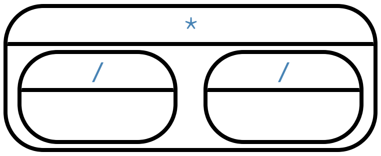
$= 1$



$= 1$



$= 1$

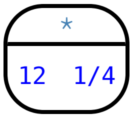
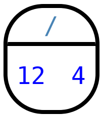


$= 1$

Discover Inverse Operations: Multiplication & Division

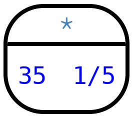
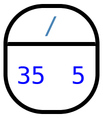
From Multiplying by Fractions to Dividing by Whole Numbers

1) Under each Circle of Evaluation, write the equivalent arithmetic expression.

	=	
×	=	÷

2) What will the first Circle evaluate to? __________ The second Circle? __________

3) Under each Circle of Evaluation, write the equivalent arithmetic expression.

	=	
×	=	÷

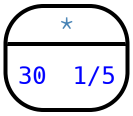
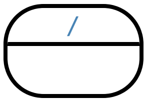
4) What will the first Circle evaluate to? __________ The second Circle? __________

5) What do you Notice? What do you Wonder? ____

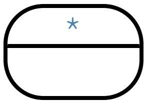
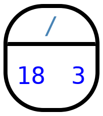
____

Did you know that multiplying by $^{1}/_{4}$ is the same as dividing by 4?

6) In the table below, fill in the blanks to demonstrate how *multiplying by a fraction* produces the same result as *dividing by the fraction's reciprocal.* Then write the equivalent arithmetic expression for each of the Circles.

	=	
×	=	÷

7) This time, we've completed the division Circle of Evaluation. You fill in the multiplication Circle of Evaluation.

	=	
×	=	÷

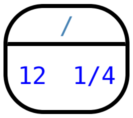
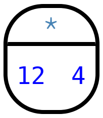
Practice

Rewrite multiplication as division, and division as multiplication.

$20 \times \frac{1}{4} =$ ______________	$100 \div 25 =$ ______________	$27 \times \frac{1}{9} =$ ______________
______________ $= 6 \times \frac{1}{18}$	______________ $= 35 \div 7$	______________ $= 5 \times \frac{1}{6}$

Discover Inverse Operations: Multiplication & Division (2)

From Dividing by Fractions to Multiplying by Whole Numbers

	=	
$12 \div {}^{1}/_{4}$	=	12×4

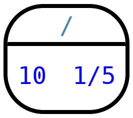
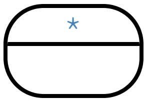
1) What will the first Circle evaluate to? ___80___ The second Circle? ___80___

2) How is this equation similar to the equations on the page you just completed? How is it different? ______________________________

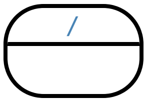
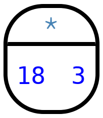
____

Did you know that dividing by ${}^{1}/_{4}$ is the same as multiplying by 4?

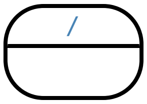
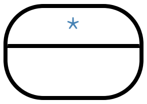
3) In the table below, fill in the blanks to demonstrate how *dividing by a fraction* produces the same result as *multiplying by the fraction's reciprocal.* Then write the equivalent arithmetic expression for each of the Circles.

	=	
$\div$	=	$\times$

4) This time, we've completed the multiplication Circle of Evaluation. You fill in the division Circle of Evaluation.

	=	
$\div$	=	$\times$

5) Provide another example that demonstrates how *dividing by a fraction* produces the same result as *multiplying by the fraction's reciprocal* .

	=	
$\times$	=	$\div$

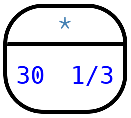
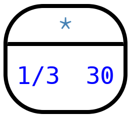
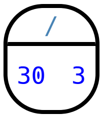
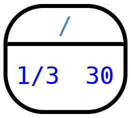
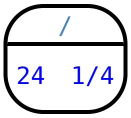
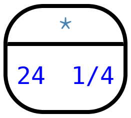
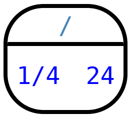
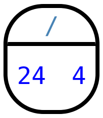
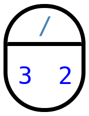
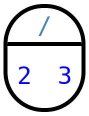
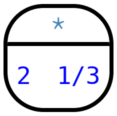
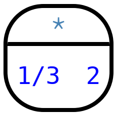
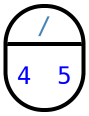
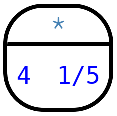
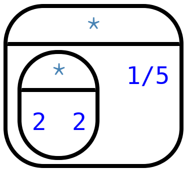
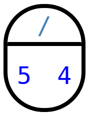
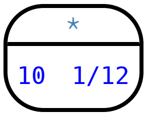
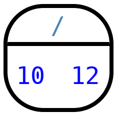
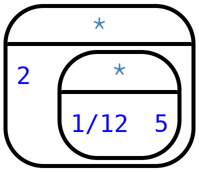
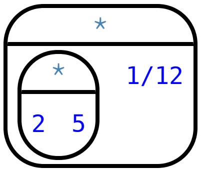
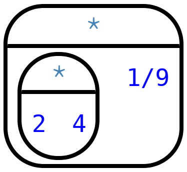
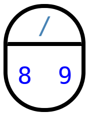
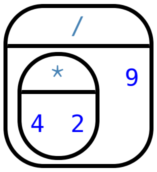
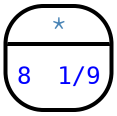
Practice

Rewrite multiplication as division, and division as multiplication.

$20 \times 4 =$ __________________	$21 \div \frac{1}{3} =$ __________________	$12 \times 4 =$ __________________
__________________ $= 6 \times 3$	__________________ $= 14 \div \frac{1}{7}$	__________________ $= 15 \times 2$

Which One Doesn't Belong?

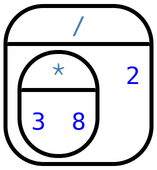
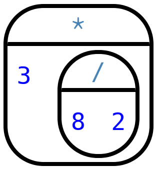
For each row, cross out the Circle(s) of Evaluation that evaluate to do **not** evaluate to the provided quantity. In some cases, you may not cross out any Circles.

	Which Circle(s) evaluate to...	Circles of Evaluation			
1)	10 ?				
2)	6 ?				
3)	$^{2}/_{3}$?				
4)	$^{4}/_{5}$?				
5)	$^{10}/_{12}$?				
6)	$^{8}/_{9}$?				

Divide First… or Solve Left-to-Right?

For each expression, draw *two* unique Circles of Evaluation.

- In the column on the left, draw a Circle to illustrate evaluating from left to right.
- In the column on the right, draw a Circle to illustrate evaluating division first, followed by multiplication.
- Evaluate each Circle and make a note of whether or not they produce the same result. We've done the first one for you.

	Expression	Divide First	Solve Left-to-Right
1)	$3 \times 8 \div 2$		
2)	$4 \times 50 \div 2$		
3)	$5 \times 3 \div 3$		
4)	$5 \times 6 \div 3$		
5)	Come up with an example of your own! See if you can come up with an unusual expression.		

6) What did you Notice? What do you Wonder? ________________________________

____

____

7) Can you change the groupings or the order in which you evaluate an expression like this one: $100 \div 20 \div 5$? On the back of your paper, sketch out a few possible Circles. Which one is correct? Why do we need to use the left-to-right rule here? ____________________

____

____

Programming with the Multiplicative Inverse

Examples and the Multiplicative Inverse

1) Below, place a checkmark next to each of the examples that you **predict** will pass when you click "Run".

```
____ (EXAMPLE (* 30 1/3) 10)
____ (EXAMPLE (/ 25 1/5) 5)
____ (EXAMPLE (* 1/3 2) 2)
____ (EXAMPLE (* (* 2 2) 1/7) 4/7)
```

```
____ (EXAMPLE (* 1/9 (* 2 4)) 8/9)
____ (EXAMPLE (/ 9 10) 10/9)
____ (EXAMPLE (/ 2 5) (/ 2 1/5))
____ (EXAMPLE (* 27/20 20/27) (/ 20 20))
```

2) Open the Multiplicative Inverse Starter File and click "Run." Using the information provided, fill in as many of the blanks as needed below to describe the examples that failed.

Test # ________ failed because ____

Test # ________ failed because ____

Test # ________ failed because ____

Test # ________ failed because ____

3) Edit each of the **failing** examples on the left so that *all examples pass* when you click "Run". Be sure to change **only** the part of the example after the `is`! Describe one of the changes you had to make. ____

____

____

Revisiting "Is the Order of Operations Universal?"

4) Below, place a checkmark next to each of the examples that you **predict** will pass when you click "Run".

```
____ (EXAMPLE (/ (* 40 12) 2) (* 40 (/ 12 2)))
____ (EXAMPLE (* (/ 6 12) 3) (/ 6 (* 12 3)))
____ (EXAMPLE (/ (* 5 9) 15) (* 5 (/ 9 15)))
____ (EXAMPLE (* (/ 8 4) 25) (/ 8 (* 4 25)))
____ (EXAMPLE (/ (* 90 1) 2) (* 90 (/ 1 2)))
```

5) Open the Multiplicative Inverse Starter File 2 and click "Run". Using the information provided, fill in as many blanks as needed below to describe the examples that failed.

Test # ________ failed because ____

Test # ________ failed because ____

Test # ________ failed because ____

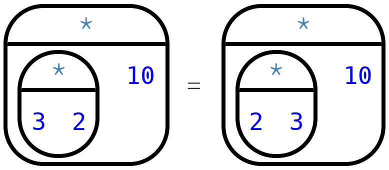
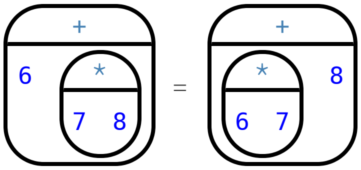
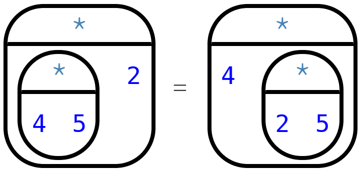
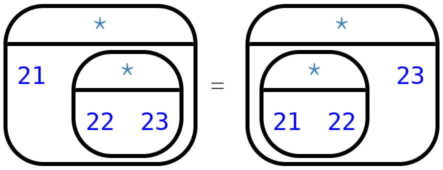
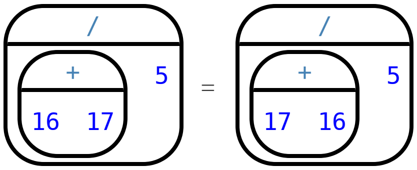
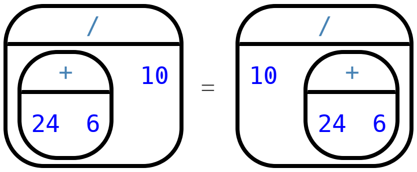
6) Notice that all of the examples appear to follow the same pattern in terms of groupings. Why do you think **some** of the examples passed, but others did not? ____

____

____

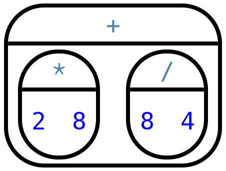
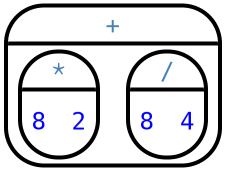
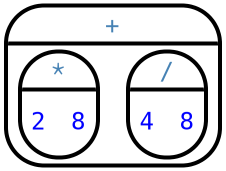
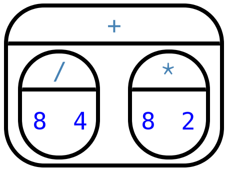
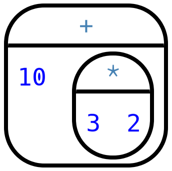
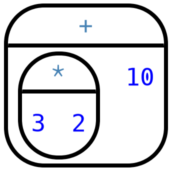
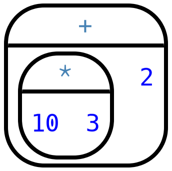
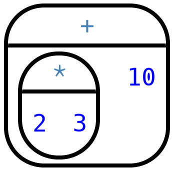
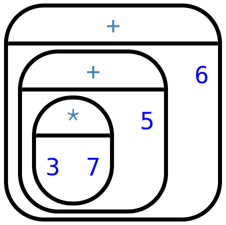
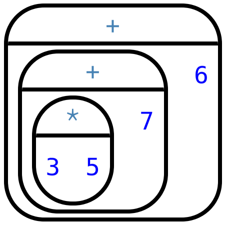
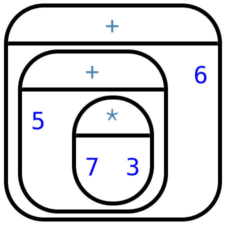
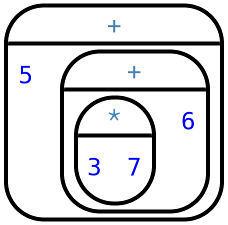
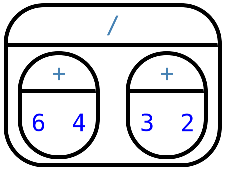
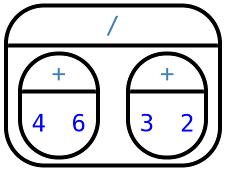
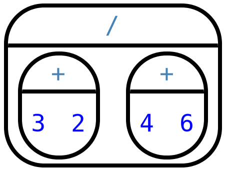
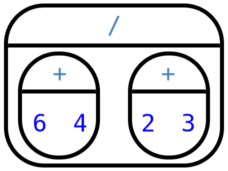
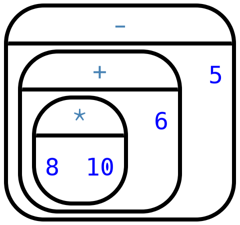
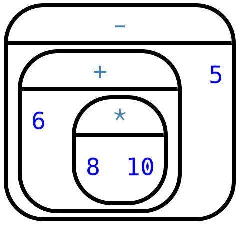
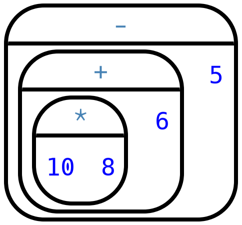
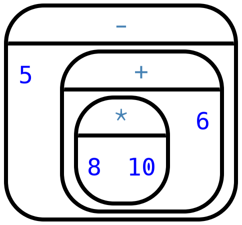
True or False? Commutative and Associative Properties

Is the equation represented by the two Circles of Evaluation true or false? Explain your response.

	Circles of Evaluation	True or False? Explain
1)		
2)		
3)		
4)		
5)		
6)		

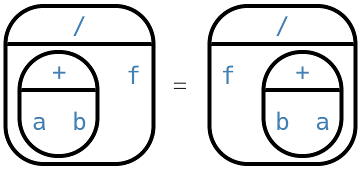
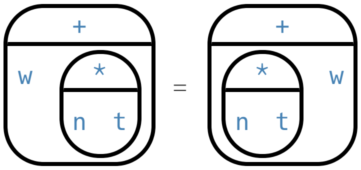
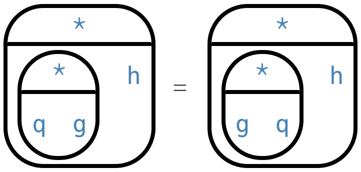
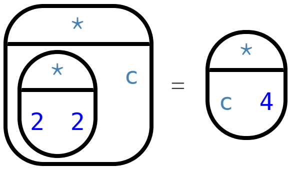
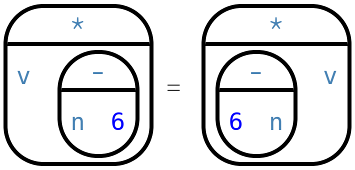
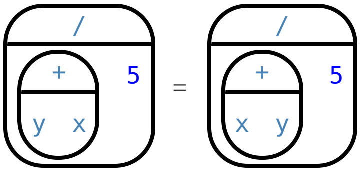
Which One Doesn't Belong?

Cross out the Circle of Evaluation that does NOT belong with the others, and then explain your choice.

	Which one doesn't belong?	Explain
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	

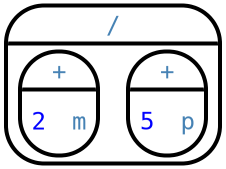
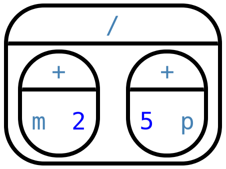
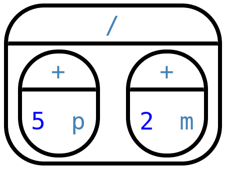
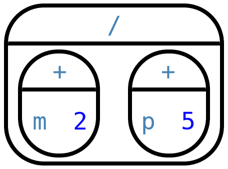
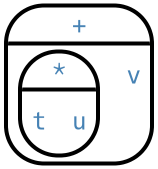
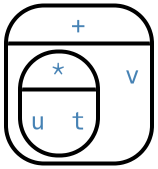
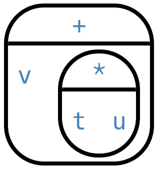
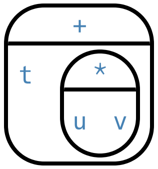
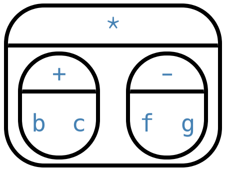
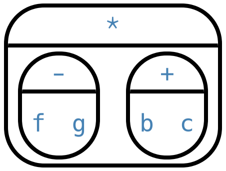
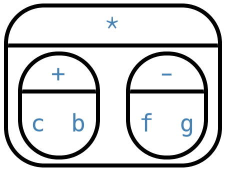
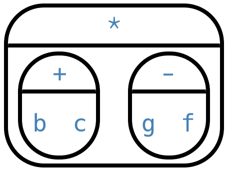
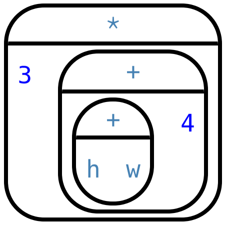
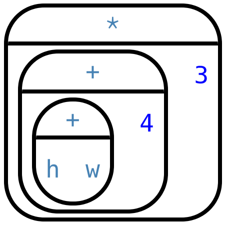
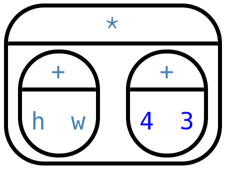
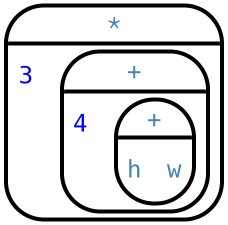
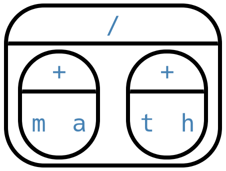
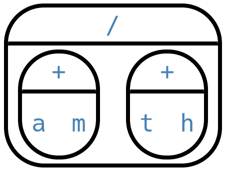
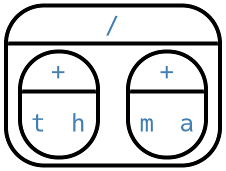
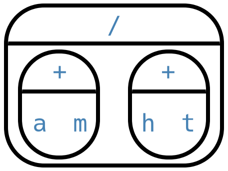
True or False? Variables

Is the equation represented by the two Circles of Evaluation true or false? Explain your response.

	Circles of Evaluation	True or False? Explain
1)		
2)		
3)		
4)		
5)		
6)		

Which One Doesn’t Belong? Variables

Cross out the Circle of Evaluation that does NOT belong with the others, and then explain your choice.

	Which one doesn’t belong?	Explain
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	

Variables and Code (Commutative Property)

1) Open Variables & the Commutative Property Starter File. On the table below, record your Noticings and Wonderings about what you see there.

Notice	Wonder

2) Hit "Run." A message appears that says, "Looks shipshape, all 4 tests passed, mate!"

3) Click "Show Details" on the right side of the green `examples-block-1` rectangle. Describe what you see. ______________________________

____

4) In lines 4-5 of the Definitions Area (left side of the screen), change the variable definitions. This time, use `a = 6` and `b = 10`. What do you predict will happen when you hit "Run"? ______________________________

____

5) Was your prediction correct? __________ In your own words, explain what happened and why. If you need help, click "Show Details".

____

____

____

6) Give three additional pairs of values for `a` and `b` that will cause *both* example 3 and example 4 to fail. Try them out!

`a` = __________ `b` = __________ `a` = __________ `b` = __________ `a` = __________ `b` = __________

7) Are there any pairs of values for `a` and `b` that will cause example 1 or example 2 to fail? If so, list them here: ______________________________

8) Are any of the examples true **every time**, no matter what values we use for `a` and `b`? If so, which ones? ______________________________

9) Are any of the examples true **some of the time**, depending on what values we use for `a` and `b`? If so, which ones? ______________________

10) Maria says, "The Commutative Property applies for every operation. I know because sometimes I can change the order of the numbers being subtracted or divided and the result remains the same." Is she correct? Explain. ______________________________

____

____

____

Variables and Code (Associative Property)

1) Open Variables & the Associative Property Starter File. On the table below, record your Noticings and Wonderings about what you see there. Consider how this starter file is different from Variables & the Commutative Property Starter File. **Don't hit "Run" yet!**

Notice	Wonder

2) Based on what you see in the Definitions Window (left side), predict what will happen when you hit "Run" by circling your choice below.

All examples pass	3 examples pass, 1 fails	2 examples pass, 2 fail	1 example passes, 3 fail

3) Explain how you made your prediction (above). ____

____

____

4) Click "Run". Were you correct? __________ Explain. ____

____

____

5) Give four sets of values for a, b and c that will cause *both* example 3 and example 4 to fail. Try them out!

a = ________ b = ________ c = ________ a = ________ b = ________ c = ________

a = ________ b = ________ c = ________ a = ________ b = ________ c = ________

6) Are there any sets of values for a, b, and c that will cause example 1 or example 2 to fail? If so, list them here: ____________

7) Are any of the examples true **every time**, no matter what values we use for a, b, and c? If so, which ones? ____________

8) Are any of the examples true **some of the time**, depending on what values we use for a, b, and c? If so, which ones? __________

9) Deena wants to edit the starter file's code so that when she hits "Run", all four tests pass. She suggests changing all the values of the variables so that a, b, and c are each equal to zero. Do you agree with her idea? *Before deciding, feel free to test out the idea!* Explain.

____

____

____

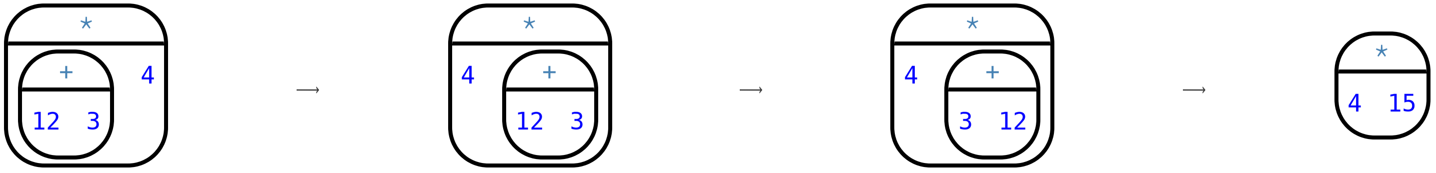
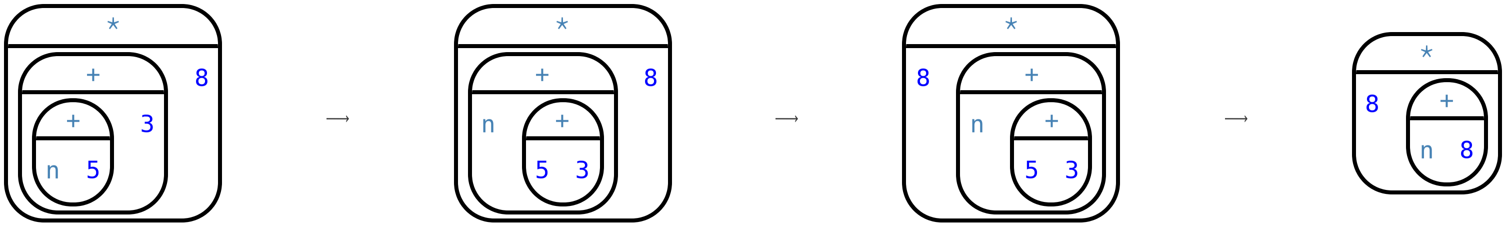
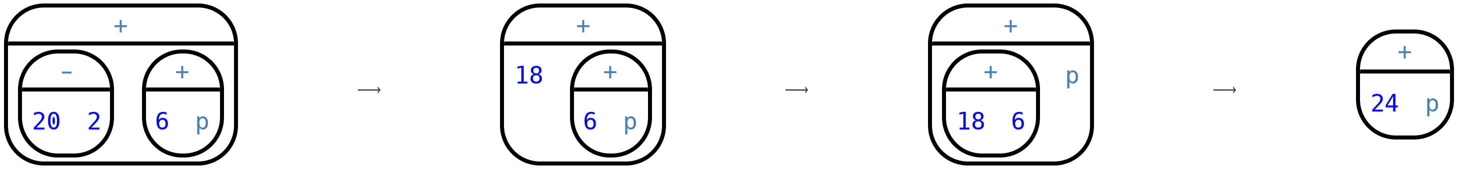
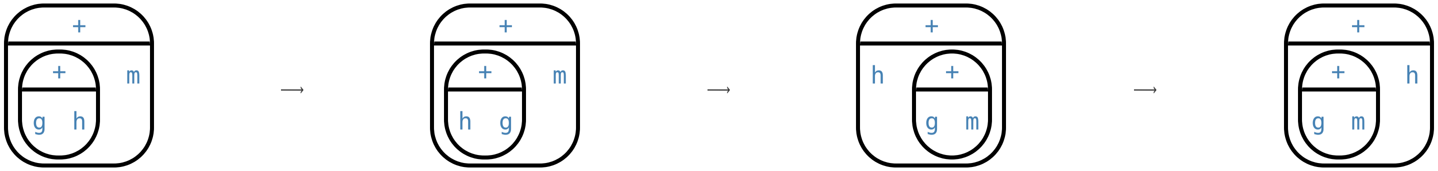
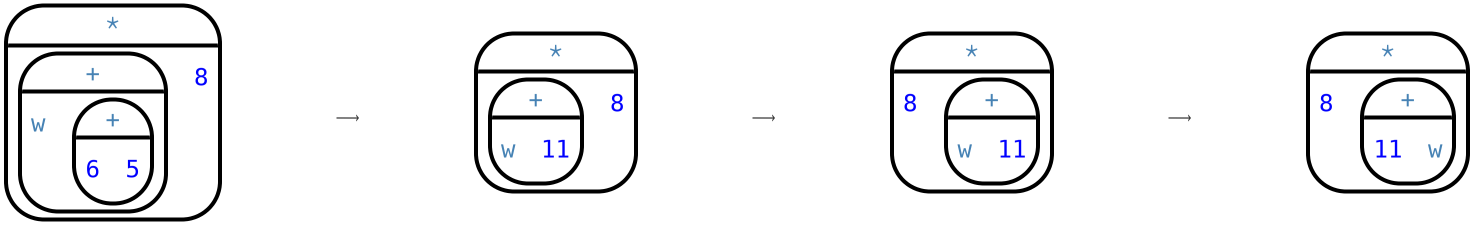
★ Can you think of any values for a, b, and c that will result in all four tests passing? ______________________

____

____

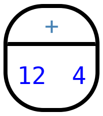
Label the Arrows

Each arrow represents a transformation from an expression to an equivalent expression. Label each arrow with the type of transformation that you observe: Associative Property ("AP"), Commutative Property ("CP"), or Computation ("Comp").

	Circles of Evaluation
1)	
2)	
3)	
4)	
5)	

Discover the Identity Property

1) Read each verbal expression and translate it into a Circle of Evaluation. The first one has been completed for you.

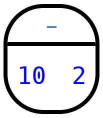
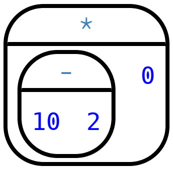
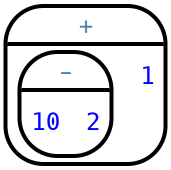
Find the sum of 12 and 4.	Find the sum of 12 and 4. Multiply it by 1.	Find the sum of 12 and 4. Add 0.
		

Are these circles ***equivalent***? Why or why not?

The second and third Circles of Evaluation illustrate the Identity Property!

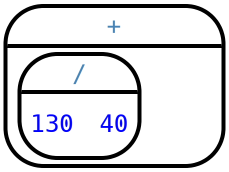
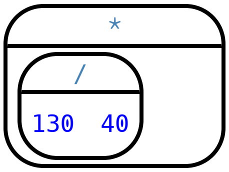
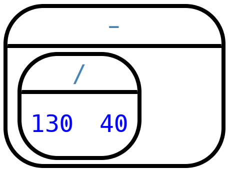
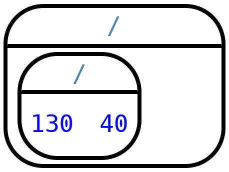
- The Identity Property of Multiplication tells us that a value **does not change** when multiplied by 1.
- The Identity Property of Addition tells us that a value **does not change** when added to 0.

2) Take a look at the *counter-examples* below.

Find the difference of 10 and 2.	Find the difference of 10 and 2. Multiply it by 0.	Find the difference of 10 and 2. Add one.
		

Explain why each Circle of Evaluation above does **not** represent the Identity Property.

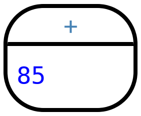
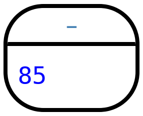
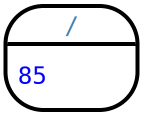
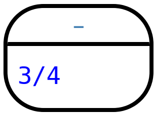
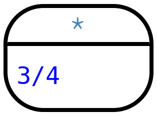
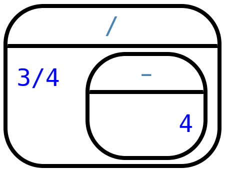
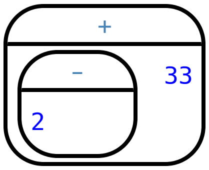
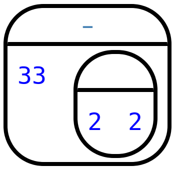
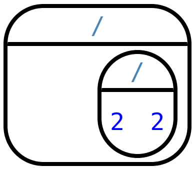
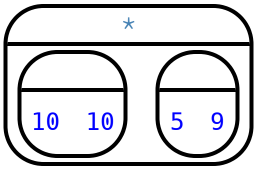
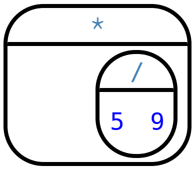
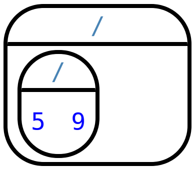
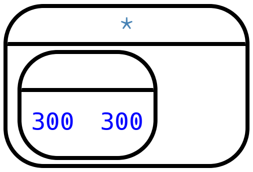
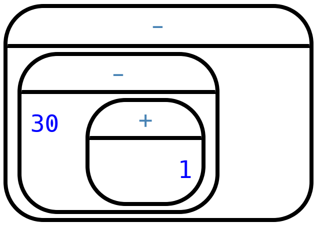
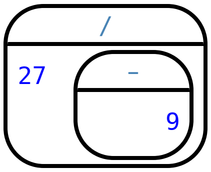
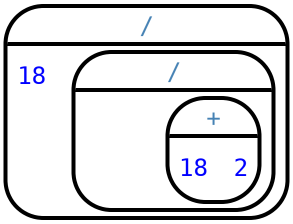
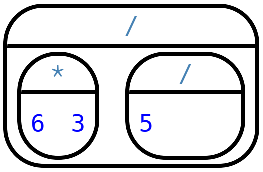
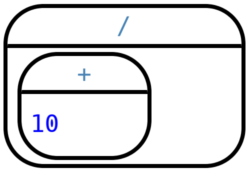
3) The Identity Property of Addition involves adding zero, and the Identity Property of Multiplication involves multiplying by 1. Is there an Identity Property of *Subtraction* and *Division*? Complete the Circles of Evaluation below so that the value doesn't change.

Summarize what you discovered about the Identity Property.

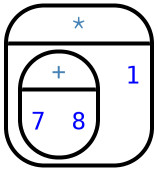
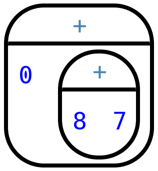
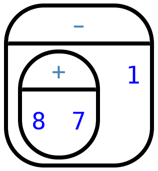
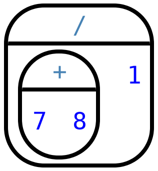
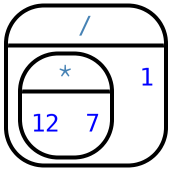
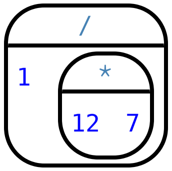
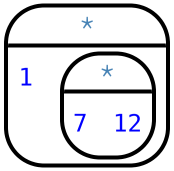
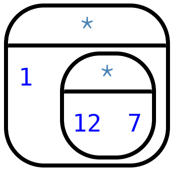
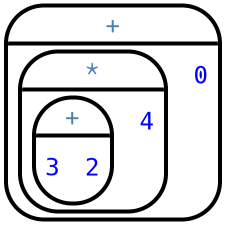
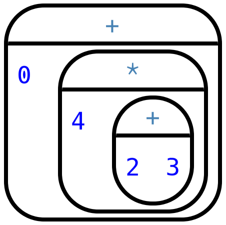
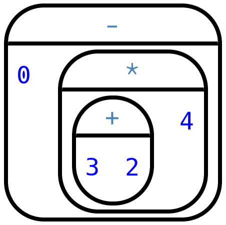
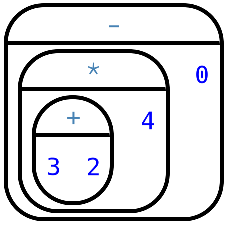
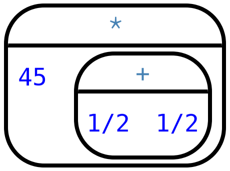
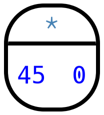
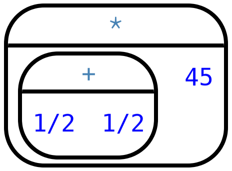
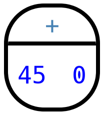
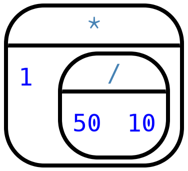
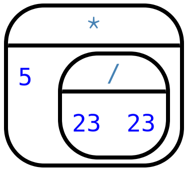
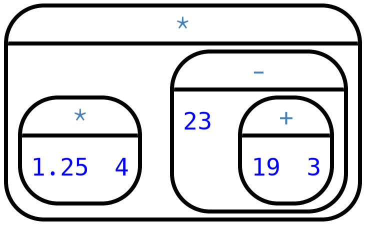
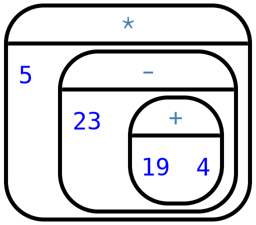
Identity Property Table

Each row in the table below contains a "goal number" in the first column. For each Circle of Evaluation in the row, complete the expression using Identity Property of Addition, Subtraction, Multiplication, or Division so that the expression will always evaluate to the "goal number".

1 *Goal:* *85*			
2 *Goal:* *3/4*			
3 *Goal:* *33*			
4 *Goal:* *5/9*			
5 *Goal:* *27*			
6 *Goal:* *18*			

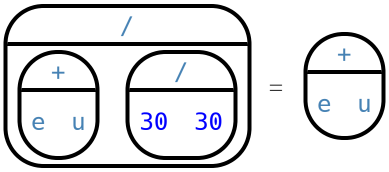
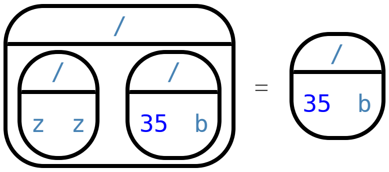
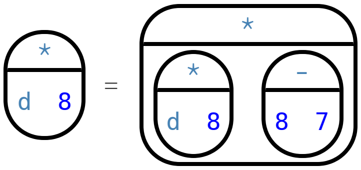
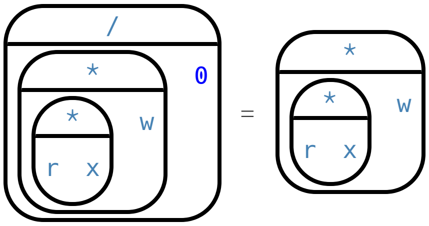
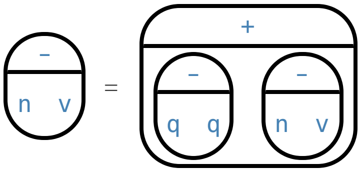
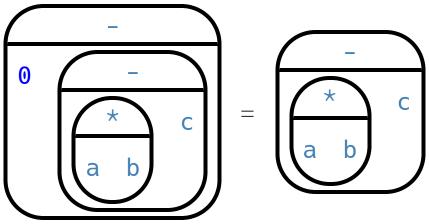
Which One Doesn’t Belong? Identity Property

Cross out the Circle of Evaluation that does NOT belong with the others, and then explain your choice.

	Which one doesn’t belong?	Explain
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	

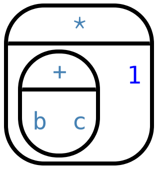
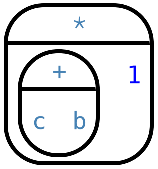
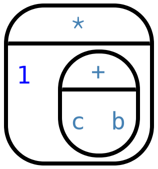
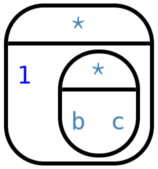
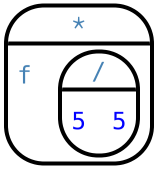
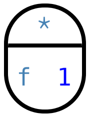
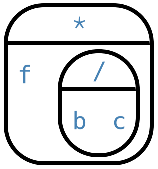
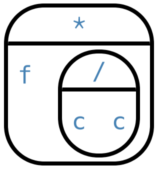
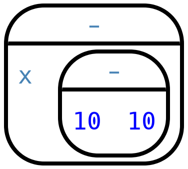
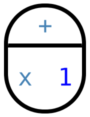
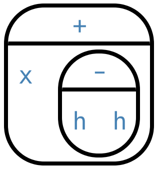
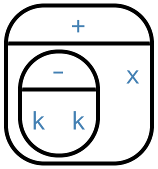
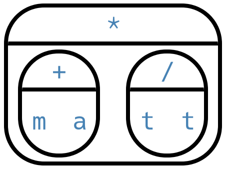
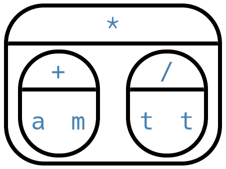
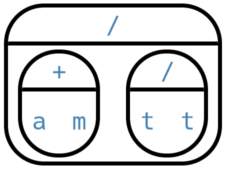
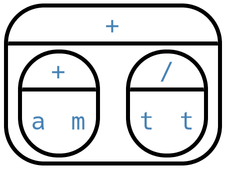
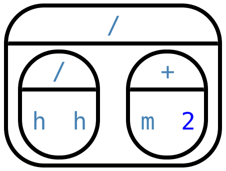
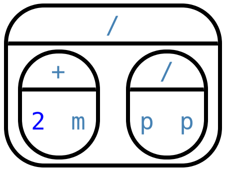
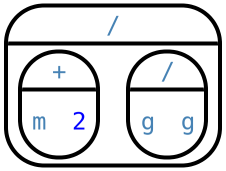
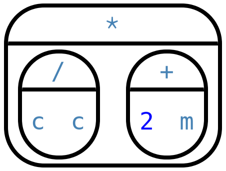
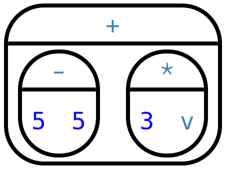
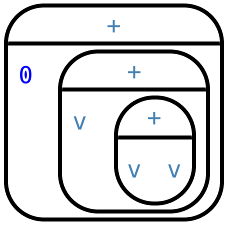
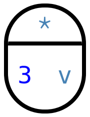
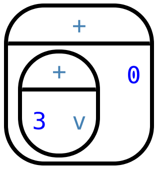
True or False? Identity Property with Variables

Is the equation represented by the two Circles of Evaluation true or false? Explain your response.

	Circles	True or False? Explain
1)		
2)		
3)		
4)		
5)		
6)		

Which One Doesn't Belong? Identity Property with Variables

Cross out the Circle of Evaluation that does NOT belong with the others, and then explain your choice.

	Which one doesn't belong?	Explain.
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	
★	   	

The Identity Property and Images

Use Identity Property Starter File to respond to the questions below.

Scale

1) With your partner, predict what belongs in each blank below. Then, test your prediction using WeScheme to see if you were correct.

Scaling by 1 will produce an image that is ________________________________

Scaling by $^1/_2$ will produce an image that is ________. ________________________________

Scaling by $^1/_{10}$ will produce an image that is ________. ________________________________

Scaling by 0 will produce ________________________________

2) Place a checkmark next to lines of code that you predict will produce an image identical to the original. Then, run the code to test your predictions.

________ `(scale (/ 5 5) dog)`

________ `(scale (* 1/2 2) dog)`

________ `(scale -1 dog)`

________ `(scale (+ -20 20 1) dog)}`

________ `(scale 0 dog)`

________ `(scale (+ 45 -45) dog)`

Rotate

3) In the Interactions Area (right), type `(rotate 90 dog)`. What happened? ________________________________

____

4) Place a checkmark next to the code that you predict will produce an image identical to the original. Run the code to test your predictions.

________ `(rotate 180 dog)`

________ `(rotate -90 dog)`

________ `(rotate 360 dog)`

________ `(rotate 450 dog)`

________ `(rotate 90 dog)`

________ `(rotate -180 dog)`

________ `(rotate -360 dog)`

________ `(rotate (* 360 19) dog)`

5) What did you discover? For what degrees did `rotate` produce an identical image? ________________________________

____

Flip

6) In the Interactions Area (right), try out `(flip-vertical dog)`, then try `(flip-horizontal dog)`. How is the image returned different from the original? ________________________________

____

7) Place a checkmark next to lines of code that you predict will produce an image identical to the original. Then, run the code to test your predictions.

________ `(flip-vertical (flip-horizontal dog))`

________ `(flip-vertical (flip-vertical dog))`

________ `(flip-horizontal (flip-vertical dog))`

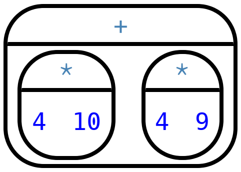
________ `(flip-horizontal (flip-horizontal dog))`

★ Write the longest, most complex line of code you can that applies several transformations to `dog`, but produces an identical output.

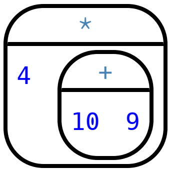
From Sum to Product

Complete the Circles of Evaluation on the right to make them equivalent to the ones on the left. On the lines below each Circle, compute the answer and show your work. For each pair of Circles, put a check by the one you think is easier to answer. We did the first one for you.

1)



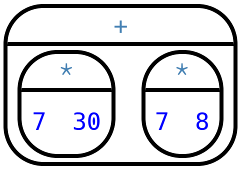
→



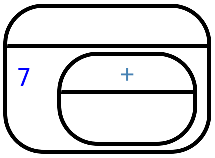
40 + 36 = 76

4 × 19 = 76

2)

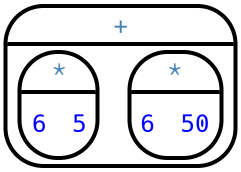


→

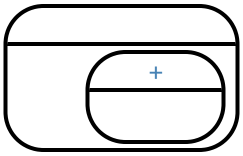


210 + 56 = 266

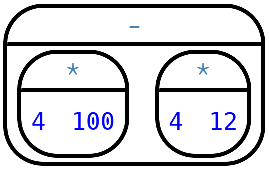
3)



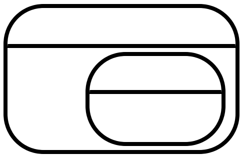
→



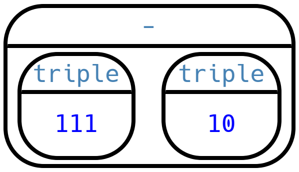
4)



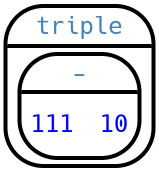
→



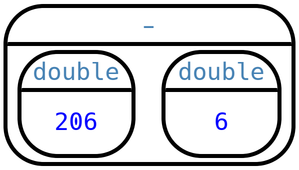
5)



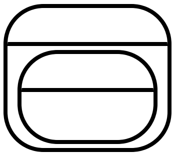
→



6)



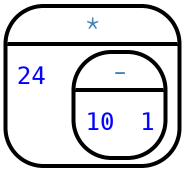
→



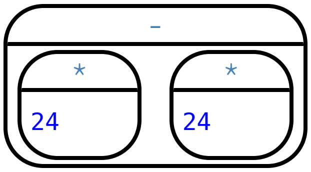
From Product to Sum

Complete the Circles of Evaluation on the right to make them equivalent to the ones on the left. On the lines below each Circle, compute the answer and show your work. The first one has has been done for you.

1)



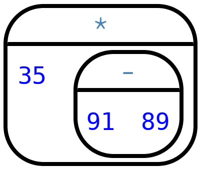
→



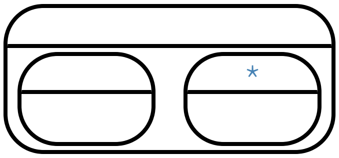
24 × 9 = 216

240 - 24 = 216

2)



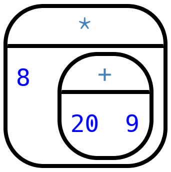
→



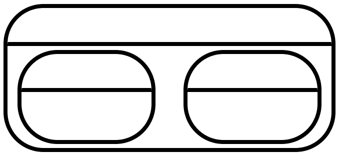
______

______

3)



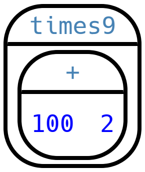
→



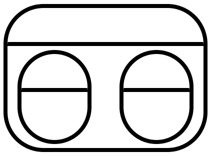
______

______

4)



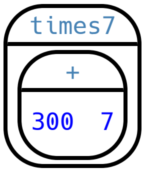
→



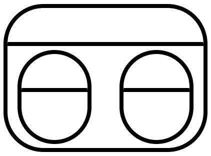
9 × 102 = 918

______

5)



→



______

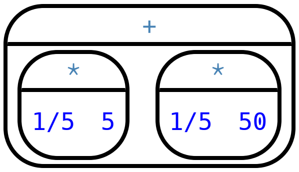
______

★ For each pair of Circles above, put a check next the one you think is easier to answer.

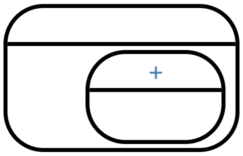
Distribution Challenge

Fill in the blanks for each pair of Circles of Evaluation below to make them equivalent. On the lines below each Circle, compute the answer and show your work.

1)

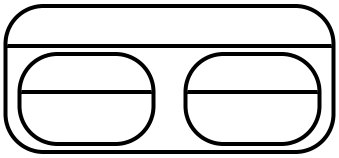


→

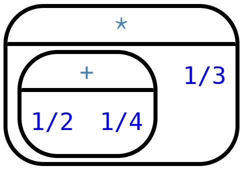


______________________ ______________________

2)

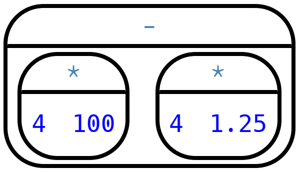


←

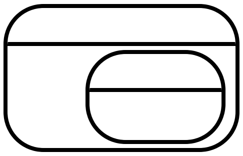


______________________ ______________________

3)

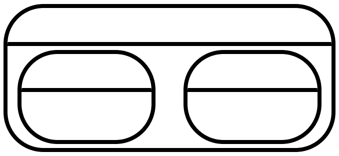


→

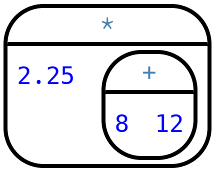


______________________ ______________________

4)

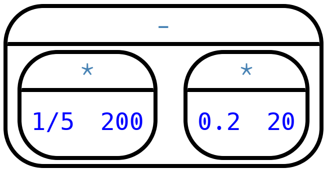


←

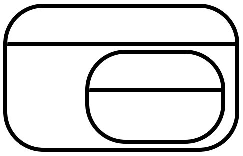


______________________ ______________________

5)



→

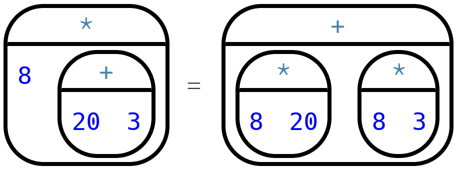
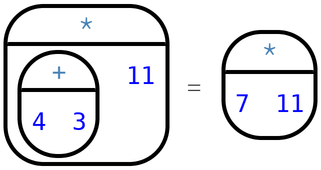
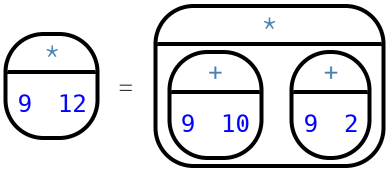
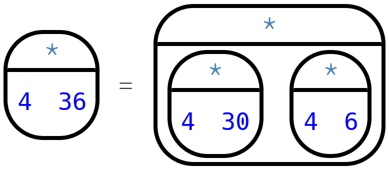
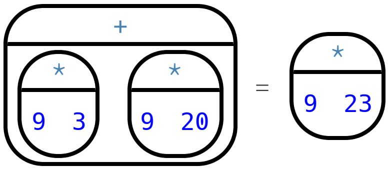
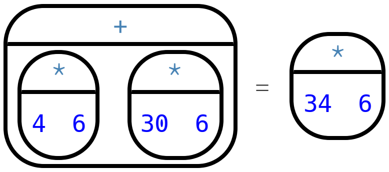


______________________ ______________________

★ For each pair of Circles above, put a check next the one you think is easier to answer.

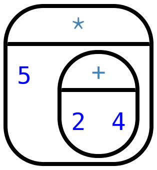
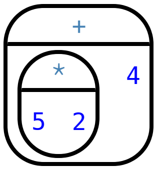
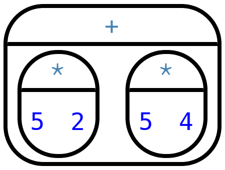
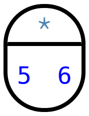
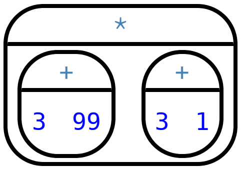
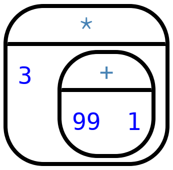
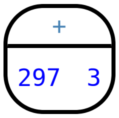
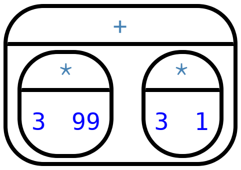
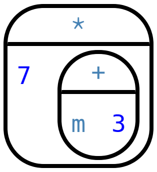
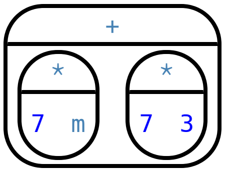
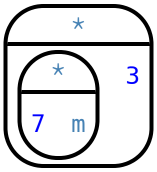
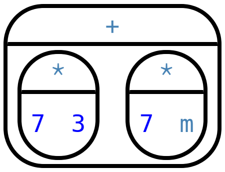
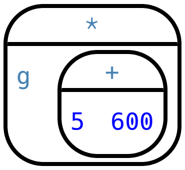
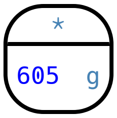
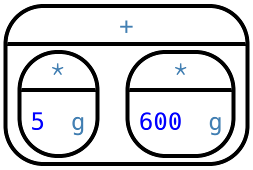
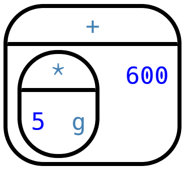
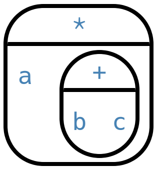
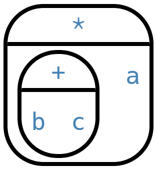
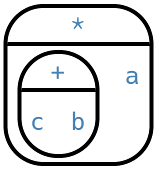
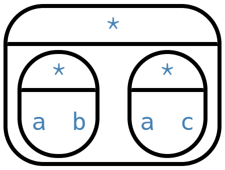
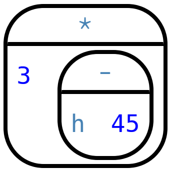
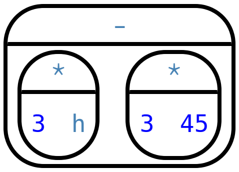
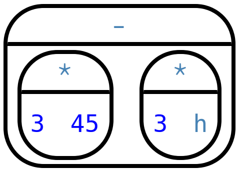
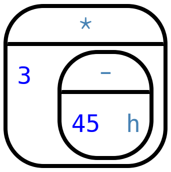
True or False? Distributive Property

Is the equation represented by the two Circles of Evaluation true or false? Explain your response.

	Circles of Evaluation	True or False? Explain
1)		
2)		
3)		
4)		
5)		
6)		

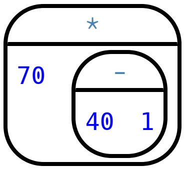
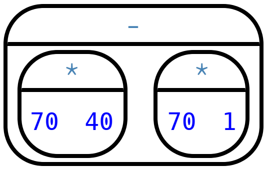
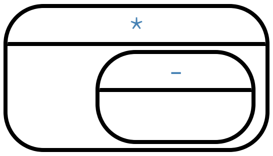
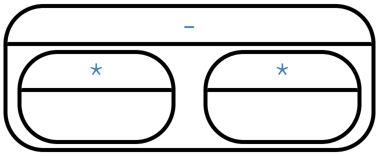
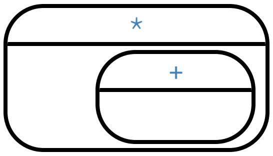
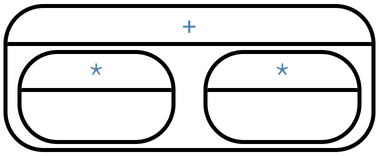
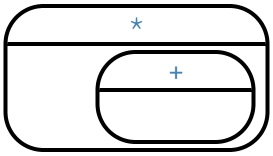
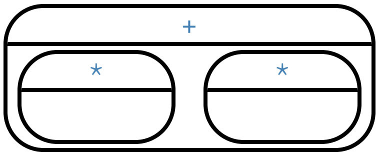
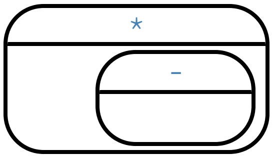
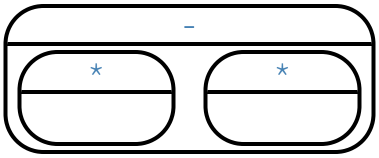
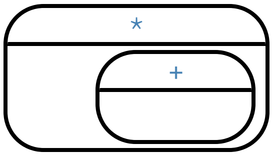
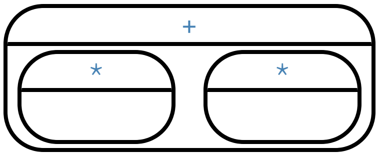
Which One Doesn't Belong? Distributive Property

Cross out the Circle of Evaluation that does NOT belong with the others, and then explain your choice.

	Which one doesn't belong?	Explain
1)	   	
2)	   	
3)	   	
4)	   	
5)	   	
6)	   	

The Distributive Property and Mental Math

On this page, the goal is to **make the math easier** by creating equivalent Circles of Evaluation that we can solve in our heads. In each row, fill in each Circle of Evaluation from left to right. Then, use mental math to compute the answer. The first one is done for you.

	Expression	Product	Sum or Difference	Answer
1)	70 × 39			210
2)	20 × 29			
3)	50 × 51			
4)	25 × 83			
5)	15 × 37			
6)	9 × 54			

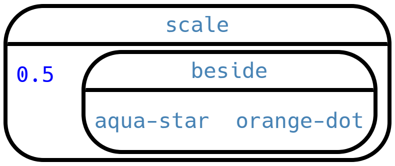
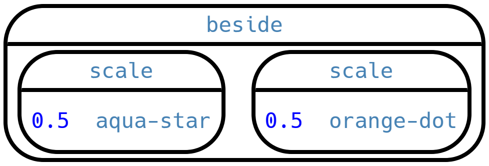
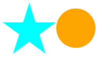
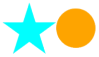
Distribution and Code

Open the Distributive Property Starter File, which you will use to investigate four functions:

```
; beside :: Image, Image -> Image                    ; rotate :: Number, Image -> Image
; above :: Image, Image -> Image                     ; scale :: Number, Image -> Image
```

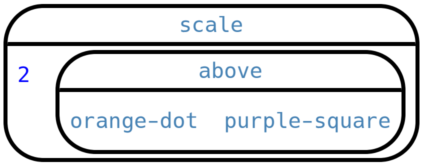
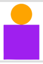
What image operations can be distributed? **Follow the example in the table below.**

Example: Does scale distribute over beside ?

Circle of Evaluation		 beside scale 0.5 aqua-star scale 0.5 orange-dot
Sketch		

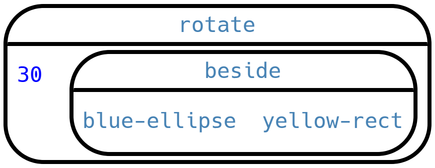
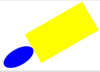
Did both expressions produce *identical* images in WeScheme? Yes.   Is scale distributive over beside? It is.

1) Does scale distribute over above ?

Circle of Evaluation		
Sketch		

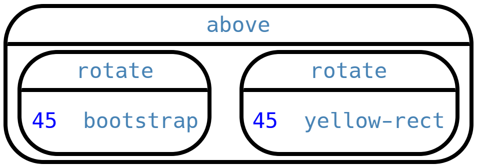
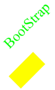
Did both expressions produce *identical* images in WeScheme? ____________   Is scale distributive over above? ____________

2) Does rotate distribute over beside ?

Circle of Evaluation		
Sketch		

Did both expressions produce *identical* images in WeScheme? ____________   Is rotate distributive over beside? ____________

3) Does rotate distribute over above ?

Circle of Evaluation		 above rotate 45 bootstrap rotate 45 yellow-rect
Sketch		

Did both expressions produce *identical* images in WeScheme? ____________   Is rotate distributive over above? ____________

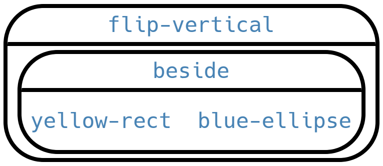
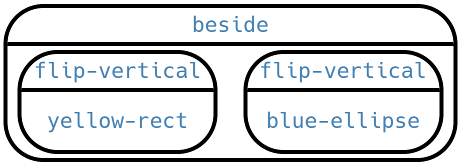
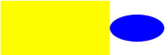
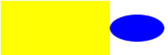
Distribution and Code (2)

Open the Distributive Property Starter File, which you will use to investigate four functions:

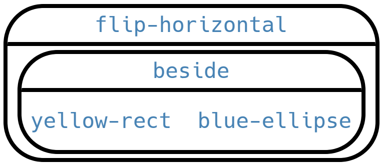
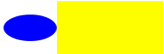
```
; beside :: Image, Image -> Image          ; flip-vertical :: Image -> Image
; above :: Image, Image -> Image           ; flip-horizontal :: Image -> Image
```

What image operations can be distributed? **Follow the example in the table below.**

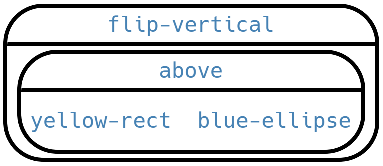
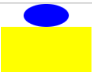
Example: Does `flip-vertical` distribute over `beside` ?

Circle of Evaluation		 beside flip-vertical flip-vertical yellow-rect blue-ellipse
Sketch		
Did both expressions produce *identical* images in WeScheme? ________		Is `flip-vertical` distributive over `beside`? ________

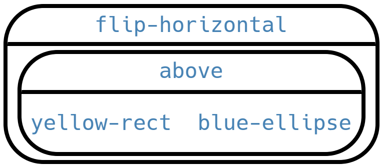
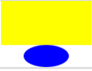
1) Does `flip-horizontal` distribute over `beside` ?

Circle of Evaluation		
Sketch		
Did both expressions produce *identical* images in WeScheme? ________		Is `flip-horizontal` distributive over `beside`? ________

2) Does `flip-vertical` distribute over `above` ?

Circle of Evaluation		
Sketch		
Did both expressions produce *identical* images in WeScheme? ________		Is `flip-vertical` distributive over `above`? ________

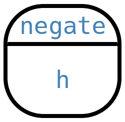
3) Does `flip-horizontal` distribute over `above` ?

Circle of Evaluation		
Sketch		
Did both expressions produce *identical* images in WeScheme? ________		Is `flip-horizontal` distributive over `above`? ________

Absolute Value and Opposite

Opposites are two numbers that are the same distance from zero on the number line, with one negative and one positive. For instance, h is the opposite of $-h$.

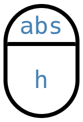
We can represent $-h$ (read: “the opposite of h,” or “negative h”) with a Circle of Evaluation:



Absolute value is the (positive) distance of a number from zero. We annotate absolute value like this: $|h|$, with h being any given number.

When we encounter an expression like $|h|$, we say "the absolute value of h."

We can represent $|h|$ with a Circle of Evaluation:



Because *opposites* are the same distance away from zero, they will always have the same absolute value. So, $|4| = 4$ and $|-4| = 4$.

The algebraic expressions $|h|$ and $-h$ sometimes produce the same outcome, and they sometimes produce different outcomes. $|h|$ is always positive or zero, while $-h$ can be negative, zero, or positive.

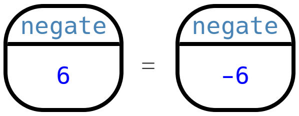
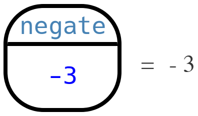
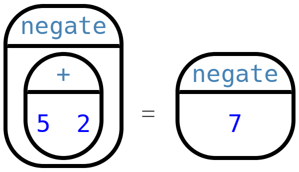
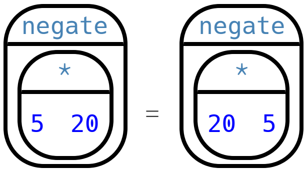
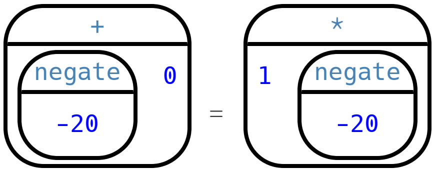
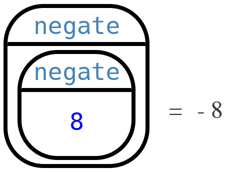
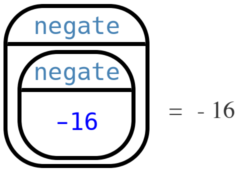
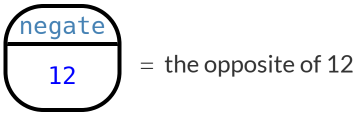
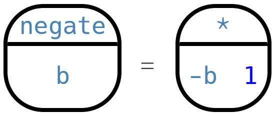
We can also create expressions that utilize both opposite and absolute value. For instance:

- We can find the ***opposite*** of an ***absolute value***: $-|x|$
- We can find the ***absolute value*** of an ***opposite***. $|-x|$

Thinking about the structure of the expression (and studying its Circle of Evaluation) can help us understand if it is positive or negative.

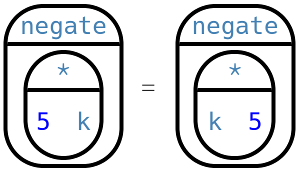
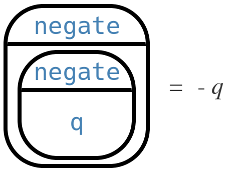
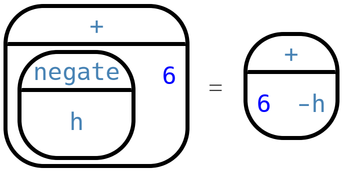
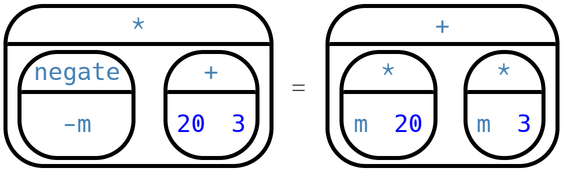
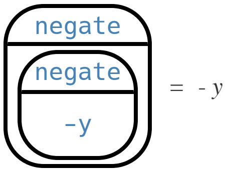
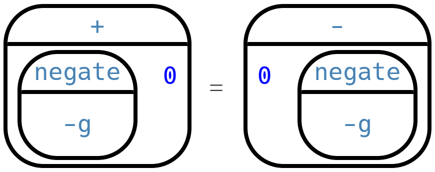
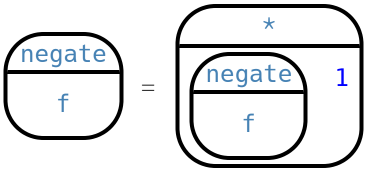
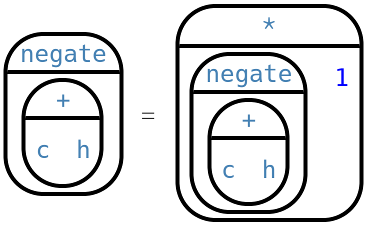
True or False? Negate

Is the equation represented by the two Circles of Evaluation true or false? Evaluate each side of the equation to confirm your response. The first one is done for you.

	Circles of Evaluation	True or False? Justify
1)		False. $-6 \neq 6$
2)		
3)		
4)		
5)		
6)		
7)		
8)		
9)		

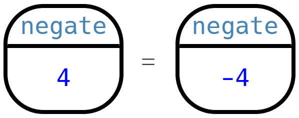
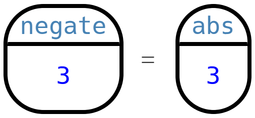
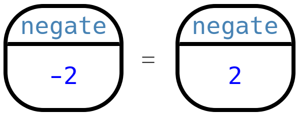
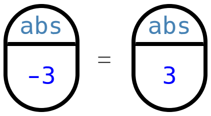
True or False? Negate (2)

Is the equation represented by the two Circles of Evaluation true or false? Evaluate each side of the equation to confirm your response. When applicable, provide the property that confirms the equivalence. The first one is done for you.

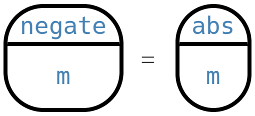
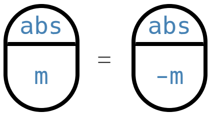
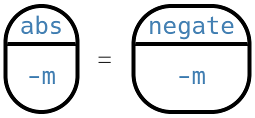
	Circles of Evaluation	True or False? Justify
1)		
2)		
3)		
4)	 * negate -m + 20 3 = + * m 20 * m 3	
5)		
6)		
7)		
8)		

True or False? Absolute Value & Negate

Is the equation represented by the two Circles of Evaluation true or false? Evaluate each side of the equation to confirm your response. The first one is done for you.

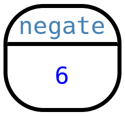
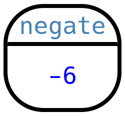
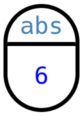
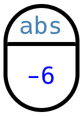
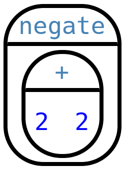
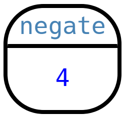
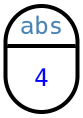
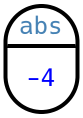
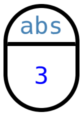
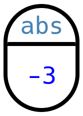
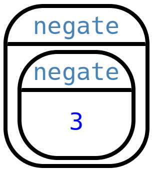
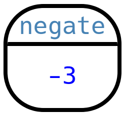
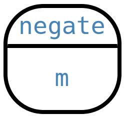
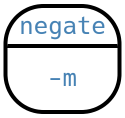
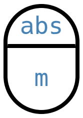
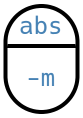
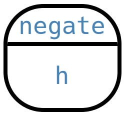
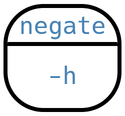
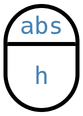
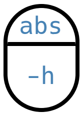
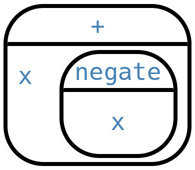
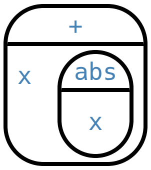
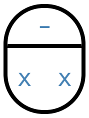
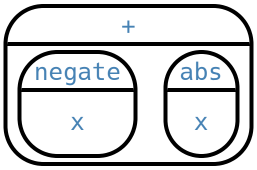
	Circles of Evaluation	True or False? Explain
1)		False. - 4 ≠ 4
2)		
3)		
4)		

On the table below, state whether the equation represented by the Circles is *always true* , *sometimes true* , or *never true* . Explain your response.

	Circles	Always, sometimes, or never true?
5)		
6)		
7)		

Which One Doesn't Belong? Absolute Value & Negate

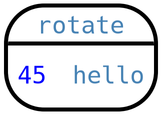
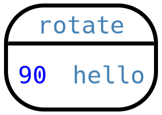
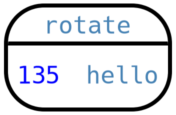
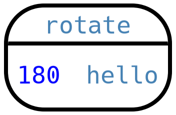
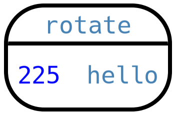
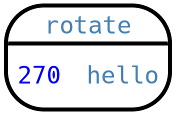
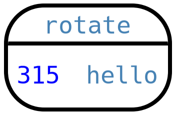
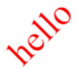
For each row, cross out any Circles of Evaluation that do NOT meet the condition stated on the left. **NOTE:** Some rows might not need anything crossed out!

	Value	Place a check mark by the equivalent Circles of Evaluation
1)	Which Circles evaluate to 6 ?	   
2)	Which Circles evaluate to -4 ?	   
3)	Which Circles evaluate to 3 ?	   
4)	If $m = -3$, which Circles evaluate to 3 ?	   
5)	If $h = 20$, which Circles evaluate to 20 ?	   
6)	If $x = 7$, which Circles evaluate to 0 ?	   

Exploring Rotations

Use the Negation Starter File for this page.

1) Draw the image that each Circle of Evaluation will produce. The first prediction has been done for you.

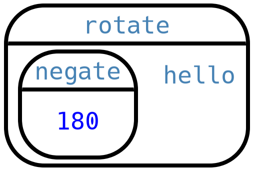
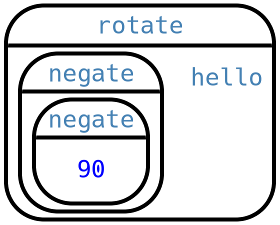
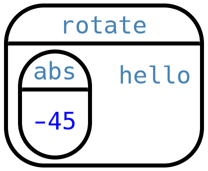
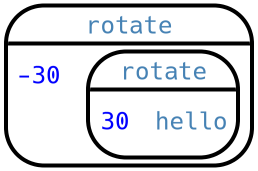
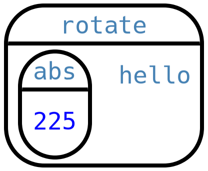
						
						

2) What did you discover? (Some questions to consider: What happens when you rotate an image 90 degrees? 180 degrees? Were rotations clockwise or counter-clockwise?) ___________

____

____

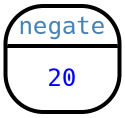
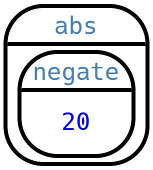
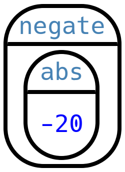
3) The table below includes negation, absolute value, and composed rotations. Draw the image that each Circle of Evaluation will produce.

★ For each Circle in the table above, can you up with a ***simpler, equivalent*** Circle of Evaluation that will produce the same image? Draw them in the empty boxes below.

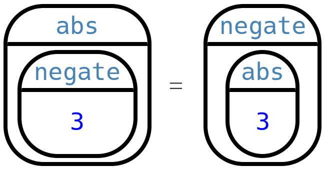
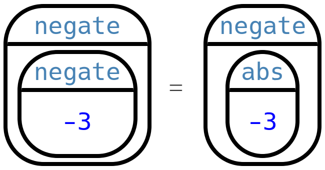
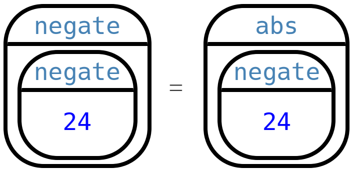
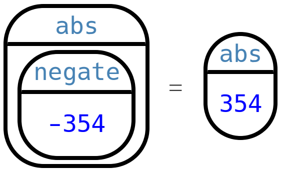
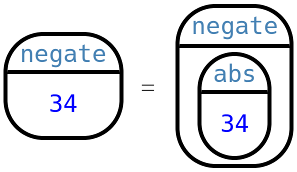
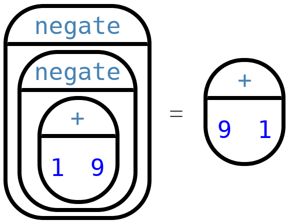
Translating (Absolute Value & Opposite)

Each row represents a single arithmetic expression, written in three different forms. Fill in the empty spaces so that all three forms represent the same expression.

	Circle of Evaluation	Words	Math
1)		the opposite of 20	
2)			$\lvert 20 \rvert$
3)			
4)			$-(-20)$
5)		the opposite of the absolute value of 20	
6)			

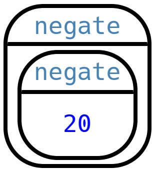
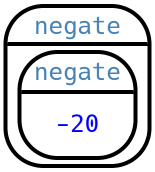
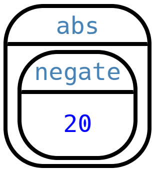
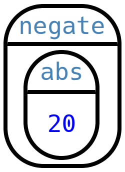
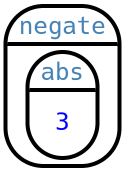
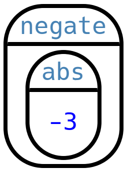
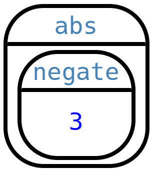
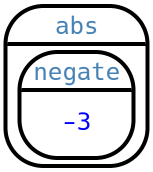
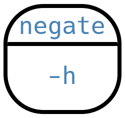
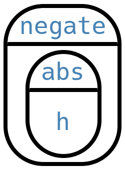
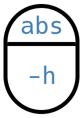
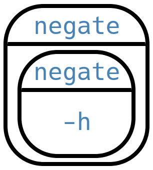
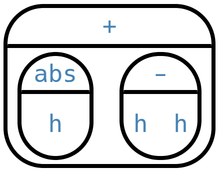
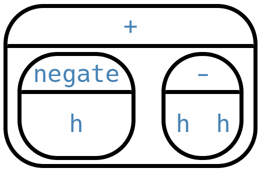
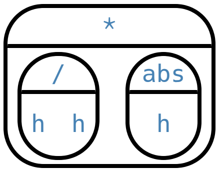
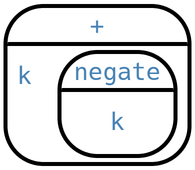
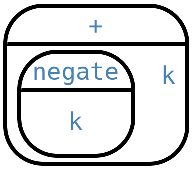
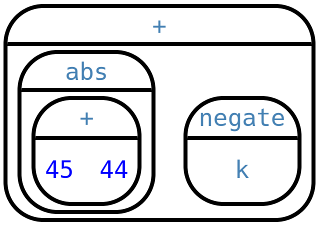
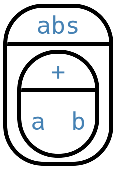
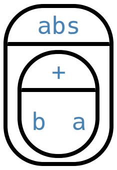
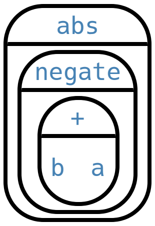
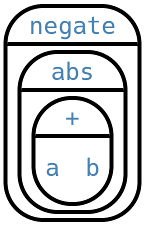
True or False? Absolute Value & Negate

Is the equation represented by the two Circles of Evaluation true or false? Evaluate each side of the equation to confirm your response. The first one is done for you.

	Circles of Evaluation	True or False? Explain
1)		False. 3 ≠ - 3
2)		
3)		
4)		
5)		
6)		

Which One Doesn’t Belong? (Absolute Value & Negate)

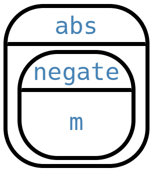
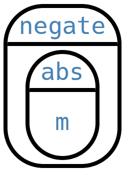
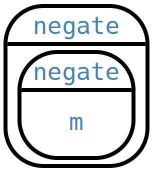
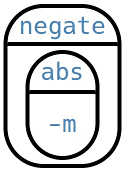
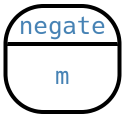
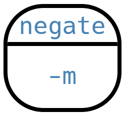
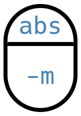
For each row, cross out any Circles of Evaluation that do NOT meet the condition stated on the left. **NOTE:** Some rows might not need anything crossed out!

	Value	Circles of Evaluation
1)	Which Circles evaluate to 20 ?	   
2)	Which Circles evaluate to -3 ?	   
3)	Let $h = 9$. Which Circles evaluate to 9 ?	   
4)	Let $h = 6$. Which Circles evaluate to 6 ?	  
5)	Let $k = 99$. Which Circles evaluate to 0 ?	  
6)	Let $a = 11$ and $b = 20$. Which Circles evaluate to 31 ?	   

Matching Circles and Expressions

Assume m is a non-zero integer. Draw a line from the expression on the left to the Circle of Evaluation on the right.

Note: Some expressions have more than one correct Circle of Evaluation!

Words			Circle of Evaluation
$\| - m\|$	1	A	
$- \|m\|$	2	B	
$- (- m)$	3	C	
$- m$	4	D	
$- \| - m \|$	5	E	
		F	
		G	

6) Look at the expressions in the matching activity above. Are any of them *always* positive, no matter what we substitute in for m ? __________

7) Are any of the above expressions *always* negative? ____

____

8) Are there any expressions that were neither *always positive* nor *always negative*? Why? Explain. ______________________

____

____

Programming with Absolute Value and Opposite

Predict

Which equations in the table below will be *true* when $m = 6$? What about when $m = 12$? $m = 0$? With your partner, put a check ✓ in the boxes when you predict the equations will be true. Note: an equation might be true for some values and false for others!

Example	`m = 6`	`m = -12`	`m = 0`
1) `(EXAMPLE (negate m) (abs m))`			
2) `(EXAMPLE (negate m) (negate (abs m)))`			
3) `(EXAMPLE (negate m) (abs (negate m)))`			
4) `(EXAMPLE (abs (negate m))  (negate (negate m)))`			
5) `(EXAMPLE (abs (negate m)) (negate (abs m)))`			

Test

6) Open the Negation Starter File (2) and click "Run". Using the information provided, fill in as many of the blanks as needed below to describe the examples that failed.

Test # ________ failed because the left side was ________ and the right side was ________.

Test # ________ failed because the left side was ________ and the right side was ________.

Test # ________ failed because the left side was ________ and the right side was ________.

7) Talia says that setting `m` equal to *any* positive value will produce the same results. In other words, she thinks that the **same tests** will fail if `m > 0`? Do you agree? Explain. ____

8) Edit the definition of `m` (Section 1 in the starter file) to try out some other **positive** values. Was your prediction correct? Explain. ________

____

____

9) Change the definition of `m` so that it equals -12. Click "Run". Which tests failed? ______________________________

10) Edit the definition of `m` to test out other **negative** values. What do you observe? ______________________________

____

____

11) Change the definition of `m` so that it equals 0. Click "Run". Which tests failed? ______________________________

Reflect

12) Which equations in the WeScheme file are always **true**? ______________________________

13) Which equations in the WeScheme file are always **false**? ______________________________

Exponent Expressions

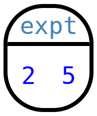
Introduction to Exponents

2^5 is an exponent expression.

- The number on the left is called the ***base***. That number is multiplied by itself when we apply the exponent.
- The smaller, raised number after the base is called the ***exponent***; it indicates how many times to multiply the base.
- "Cubing" is the same as "raising to the third power", and "squaring" is the same as raising to the second power.
- There is no special terminology for any other exponents.

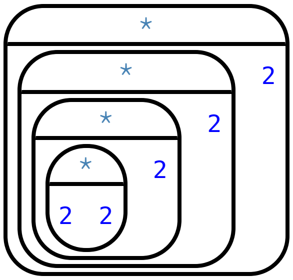
Below, one Circle of Evaluation is written in exponent notation, while the other is written in expanded notation.

2^5



=

$2 \times 2 \times 2 \times 2 \times 2$



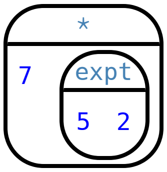
Exponents are valuable because they act as a shorthand.

The Circle of Evaluation with `expt` is a lot shorter, and easier to read!

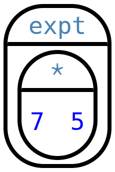
Multi-Step Exponent Expressions

In multi-step exponent expressions with no grouping symbols, we evaluate the exponent before the other operations. The two expressions below are **not** equivalent beccause the parentheses influence the order in which we evaluate.

7×5^2

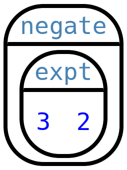


$(7 \times 5)^2$

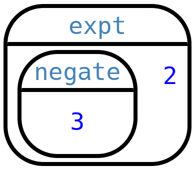


Circles of Evaluation can help us to visualize expressions with exponents and negatives and then determine if their value is positive or negative. Below, the Circles of Evaluation can help us visualize why $-(3^2)$ has a negative value, while $(-3)^2$ has a positive value.

$-(3^2)$



$(-3)^2$



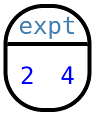
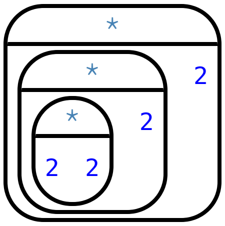
Exponent Basics

On the left, translate the verbal exponent expression into a numeric expression and a Circle of Evaluation.

On the right, write the equivalent expanded numeric expression and the corresponding Circle of Evaluation.

The first one is done for you.

1) Two to the fourth power

2^4	$2 \times 2 \times 2 \times 2$
	

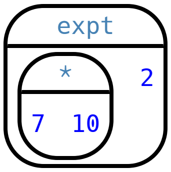
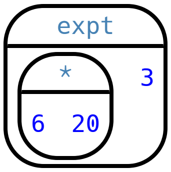
2) Six cubed

3) Ten squared

4) The square of 1/2

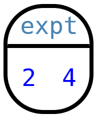
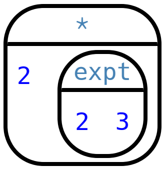
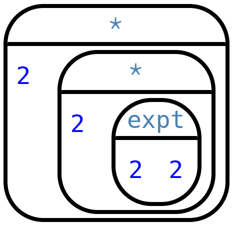
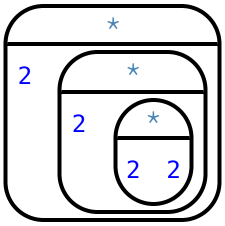
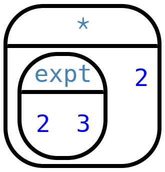
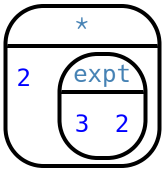
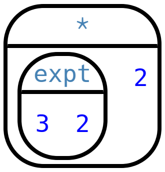
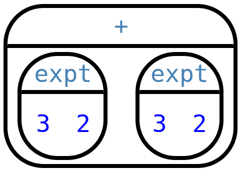
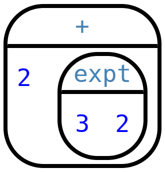
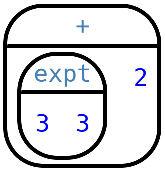
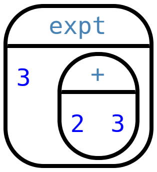
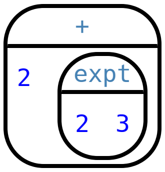
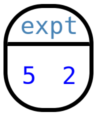
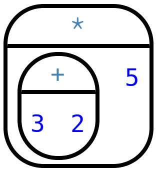
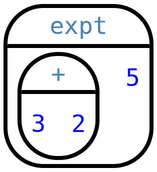
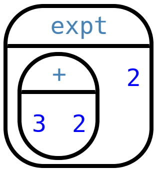
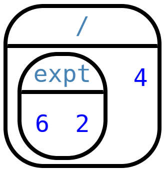
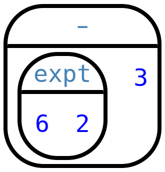
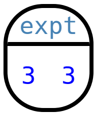
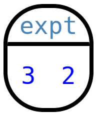
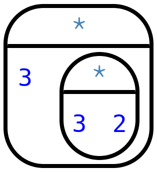
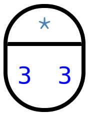
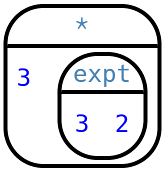
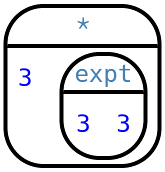
Translating Exponent Expressions

Each row represents a single arithmetic expression, written in three different forms. Fill in the empty spaces so that all three forms represent the same expression.

	Words	Circle of Evaluation	Math
1)	Start with 10. Multiply it by the square of 7.		
2)			
3)			$20^3 \times 6$
4)			
5)	Add 7 to 25 raised to the fifth power.		
6)			$(7 + 25)^5$

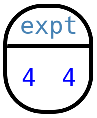
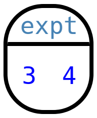
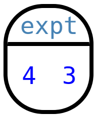
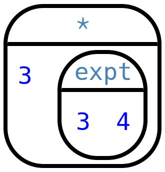
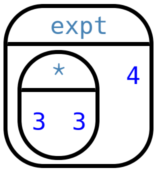
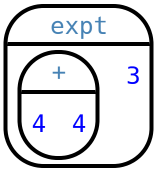
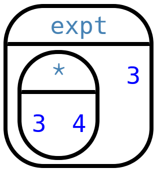
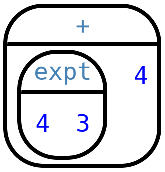
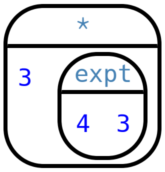
Which One Doesn’t Belong? Exponent Expressions

For each row, cross out any Circles of Evaluation that do NOT evaluate to the provided quantity. **NOTE:** Some rows might not need anything crossed out!

	Which Circle(s) evaluate to...	Circles of Evaluation
1)	16 ?	   
2)	18 ?	   
3)	11 ?	   
4)	25 ?	   
5)	9 ?	   
6)	27 ?	   

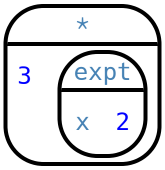
Matching Expressions to Circles of Evaluation

Draw a line from the expression on the left to the equivalent Circle of Evaluation on the right.

Words			Circle of Evaluation
$3 \times 3 \times 3 \times 3$	**1**	**A**	
3×3^4	**2**	**B**	
$(3 \times 3)^4$	**3**	**C**	
3×4^3	**4**	**D**	
$(3 \times 4)^3$	**5**	**E**	
4^3	**6**	**F**	
$4 \times 4 \times 4 \times 4$	**7**	**G**	
$4^3 + 4$	**8**	**H**	
$(4 + 4)^3$	**9**	**I**	

Variable Expressions with Exponents

Create a Circle of Evaluation for the given expression. Once you have drawn a Circle of Evaluation, evaluate the expression by substituting in the value provided in the third column. The first one is done for you.

	Expression	Circle of Evaluation	Evaluate
1)	$3x^2$		Evaluate for $x = 5$. $3x^2 = 75$
2)	$\frac{m^2}{4}$		Evaluate for $m = 10$.
3)	$6 + w^3$		Evaluate for $w = 3$.
4)	$^1/_{25} \times 5^b$		Evaluate for $b = 3$.
5)	$(7 + c)^2$		Evaluate for $c = 13$.
6)	$5w^m$		Evaluate for $w = 6$ and $m = 2$.

Programming with Exponents

Examples and Exponents

1) Below, place a checkmark next to each of the equations that you **predict** will pass when you click "Run".

```
      (EXAMPLE (expt 5 2)
____    (* 2 (* 2 (* 2 (* 2 2)))))
      (EXAMPLE (expt 4 6)
____    (* 4 (* 4 (* 4 (* 4 (* 4 4))))))
____  (EXAMPLE (expt 2 3) (* (* 2 2) 2))
____  (EXAMPLE (expt 8 3) (* 8 (* 8 8)))
      (EXAMPLE (expt 3 5)
____    (+ 3 (+ 3 (+ 3 (+ 3 3)))))
____  (EXAMPLE (expt 1 4) (* 1 (* 1 (* 1 1))))
```

2) Open the Exponents Starter File and click "Run." Using the information provided, fill in as many of the blanks as needed below to describe the examples that failed.

Test # ______ failed because the left side was ______ and the right side was ______.

Test # ______ failed because the left side was ______ and the right side was ______.

Test # ______ failed because the left side was ______ and the right side was ______.

3) Changing **only** the second part of the example , fix the **failing** examples so that *all of them pass* . Describe one of the changes. ______________

____

____

Does it equal 16?

4) A teacher asked her students to make up expressions with exponents that evaluate to 16. She typed their expressions into WeScheme as examples to test if they evaluate to 16. Below, place a checkmark next to each of the examples that you **predict** will pass.

```
____  (EXAMPLE (expt 2 4) 16)
____  (EXAMPLE (+ (expt 2 3) 10) 16)
____  (EXAMPLE (* 4 (expt 1 4)) 16)
____  (EXAMPLE (* 2 (expt 2 3)) 16)
```

```
____  (EXAMPLE (* 2 (* 2 (expt 2 2))) 16)
____  (EXAMPLE (* (expt 4 2) 2) 16)
____  (EXAMPLE (/ (/ (expt 4 3) 2) 2) 16)
____  (EXAMPLE (/ (/ (expt 4 3) 2) 2) 16)
```

5) Open the Is it 16? Starter File and click "Run". Which tests failed? ____

6) The three failing examples are all wrong for the same reason. That's because the students who wrote them doesn't understand something about how exponents work! What do they not understand?? ____

____

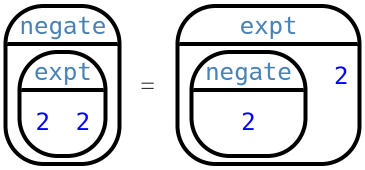
____

7) Come up with a unique exponent expression of your own that evaluates to 25, using any numbers and operators. (We've included one example for you in Section 2 of the starter file.) Write it in mathematical notation (not code) on the line: ____________________

8) Translate your expression to code and add it to the second examples block. Does your example pass? If not, revise it until it does.

True or False? Exponents and Negatives

Draw two Circles of Evaluation to represent the equation. Then, use your Circles of Evaluation to determine if the equation is true or false. The first one is done for you.

	Equation & Circles of Evaluation	True or False?
1)	$-2^2 = (-2)^2$ 	False: $-4 \neq 4$
2)	$-2^3 = (-2)^3$	
3)	$-2^4 = (-2)^4$	
4)	$-2^5 = (-2)^5$	
5)	$-2^6 = (-2)^6$	

6) What do you notice about the Circles of Evaluation on the *left*? ______________________________

____

7) What do you notice about the Circles of Evaluation on the *right*? ______________________________

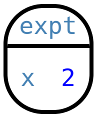
____

8) What do you notice about the *true* equations? ______________________________

____

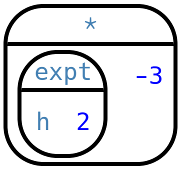
Evaluate and Compare

Create a Circle of Evaluation for the given expression. Once you have drawn a Circle of Evaluation, use it to help you evaluate the expression twice - once for $x = 5$ and once for $x = -5$. The first one is done for you.

	Expression	Circle of Evaluation	$x = 5$	$x = -5$
1)	x^2		25	25
2)	$-x^2$			
3)	x^3			
4)	$-x^3$			
5)	$-2x^3$			
6)	$(-2x)^3$			

Variable Expressions with Exponents and Negatives

Create a Circle of Evaluation for the given expression. Once you have drawn a Circle, evaluate the expression by substituting in the value provided in the third column. The first one is done for you.

	Expression	Circle of Evaluation	Evaluate
1)	$-3h^2$		Evaluate for $h = -5$. $-3h^2 = -75$
2)	$2w^m$		Evaluate for $w = -3$ and $m = 2$.
3)	$-2r^t$		Evaluate for $r = -3$ and $t = 3$.
4)	$-g^2 + -g^3$		Evaluate for $g = 2$.
5)	$(-f)^2 + (-f)^3$		Evaluate for $f = 2$.
6)	$-z^2 + -z^3$		Evaluate for $z = -2$.

Contracts for Expressions And Equations (Wescheme)

Contracts tell us how to use a function, by telling us three important things:

1. The **Name**
2. The **Domain** of the function - what kinds of inputs do we need to give the function, and how many?
3. The **Range** of the function - what kind of output will the function give us back?

For example: The contract `triangle :: (Number, String, String) -> Image` tells us that the name of the function is `triangle`, it needs three inputs (a Number and two Strings), and it produces an Image.

With these three pieces of information, we know that typing `(triangle 20 "solid" "green")` will evaluate to an Image.

Name		Domain		Range
; *	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Number
(1 2)*				
; +	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Number
(+ 1 2)				
; -	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Number
(- 1 2)				
; /	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Number
(/ 1 2)				
; <	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Boolean
(< 3 4) ; produces true				
; <=	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Boolean
(<= 3 3) ; produces true, because 3 is equal to 3				
; =	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Boolean
(= 3 4) ; produces false				
; >	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Boolean
(> "a" "b") ; produces false				
; >=	::	($\underset{a}{\text{Number}}$, $\underset{b}{\text{Number}}$)	->	Boolean
(>= 3 4) ; produces false, because 3 is neither greater-than nor equal-to 4				
; above	::	($\underset{above}{\text{Image}}$, $\underset{below}{\text{Image}}$)	->	Image
(above (circle 10 "solid" "black") (square 50 "solid" "red"))				

Name	Domain	Range
; beside	:: ($\underset{\text{left}}{\texttt{Image}}$, $\underset{\text{right}}{\texttt{Image}}$) ->	Image
(beside (circle 10 "solid" "black") (square 50 "solid" "red"))		
; circle	:: ($\underset{\text{radius}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String,}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(circle 50 "solid" "purple")		
; ellipse	:: ($\underset{\text{width}}{\texttt{Number}}$, $\underset{\text{height}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(ellipse 100 50 "outline" "orange")		
; flip-horizontal	:: (Image) ->	Image
(flip-horizontal (text "Lion" 50 "maroon"))		
; flip-vertical	:: (Image) ->	Image
(flip-vertical (text "Orion" 65 "teal"))		
; isosceles-triangle	:: ($\underset{\text{size}}{\texttt{Number}}$, $\underset{\text{vertex-angle}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(isosceles-triangle 50 20 "solid" "grey")		
; max	:: ($\underset{\text{a}}{\texttt{Number}}$, $\underset{\text{b}}{\texttt{Number}}$) ->	Number
(max 3 4)		
; min	:: ($\underset{\text{a}}{\texttt{Number}}$, $\underset{\text{b}}{\texttt{Number}}$) ->	Number
(min 3 4)		
; overlay	:: ($\underset{\text{top}}{\texttt{Image}}$, $\underset{\text{bottom}}{\texttt{Image}}$) ->	Image
(overlay (circle 10 "solid" "black") (square 50 "solid" "red"))		
; radial-star	:: ($\underset{\text{points}}{\texttt{Num}}$, $\underset{\text{outer}}{\texttt{Num}}$, $\underset{\text{inner}}{\texttt{Num}}$, $\underset{\text{fill-style}}{\texttt{Str}}$, $\underset{\text{color}}{\texttt{Str}}$) ->	Image
(radial-star 6 20 50 "solid" "red")		
; rectangle	:: ($\underset{\text{width}}{\texttt{Number}}$, $\underset{\text{height}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(rectangle 100 50 "outline" "green")		
; regular-polygon	:: ($\underset{\text{size}}{\texttt{Number}}$, $\underset{\text{vertices}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(regular-polygon 25 5 "solid" "purple")		
; rhombus	:: ($\underset{\text{size}}{\texttt{Number}}$, $\underset{\text{top-angle}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(rhombus 100 45 "outline" "pink")		
; right-triangle	:: ($\underset{\text{leg1}}{\texttt{Number}}$, $\underset{\text{leg2}}{\texttt{Number}}$, $\underset{\text{fill-style}}{\texttt{String}}$, $\underset{\text{color}}{\texttt{String}}$) ->	Image
(right-triangle 50 60 "outline" "blue")		
; rotate	:: ($\underset{\text{degrees}}{\texttt{Number}}$, $\underset{\text{img}}{\texttt{Image}}$) ->	Image
(rotate 45 (star 50 "solid" "darkblue"))		
; scale	:: ($\underset{\text{factor}}{\texttt{Number}}$, $\underset{\text{img}}{\texttt{Image}}$) ->	Image
(scale 1/2 (star 50 "solid" "lightblue"))		

Name		Domain		Range
; sqr	::	(Number)	->	Number
(sqr 4)				
; sqrt	::	(Number)	->	Number
(sqrt 4)				
; square	::	(Number , String , String) size fill-style color	->	Image
(square 50 "solid" "red")				
; star	::	(Number , String , String) radius fill-style color	->	Image
(star 50 "solid" "red")				
; star-polygon	::	(Number , Number , Number , String , String) size point-count step-count fill-style color	->	Image
(star-polygon 100 10 3 "outline" "red")				
; string-contains?	::	(String , String) haystack needle	->	Boolean
(string-contains? "hotdog" "dog")				
; string-length	::	(String)	->	Number
(string-length "rainbow")				
; text	::	(String , Number , String) message size color	->	Image
(text "Zari" 85 "orange")				
; triangle	::	(Number , String , String) size fill-style color	->	Image
(triangle 50 "solid" "fuchsia")				
; triangle/asa	::	(Number , Number , Number , String , String) top-left-angle left-side bottom-angle fill-style color	->	Image
(triangle/asa 90 200 10 "solid" "purple")				
; triangle/sas	::	(Number , Number , Number , String , String) bottom-R-side top-R-angle top-side fill-style color	->	Image
(triangle/sas 50 20 70 "outline" "darkgreen")				
	::			
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	

These materials were developed partly through support of the National Science Foundation (awards 1042210, 1535276, 1648684, and 1738598)