

Functions: Contracts, Examples & Definitions

(Also available in [WeScheme](#))

Students learn to connect function descriptions across three representations: Contracts (a mapping between Domain and Range), Examples (a list of discrete inputs and outputs), and Definitions (symbolic).

Lesson Goals	Students will be able to: • identify patterns where a function would be useful • define their own function • match examples, contracts, and definitions for the same function
Student-Facing Lesson Goals	• Let's define our own functions, by writing contracts and examples to help us identify patterns.
Prerequisites	• Simple Data Types • Contracts
Materials	• PDF of all Handouts and Page • bc Starter File • Lesson Slides • Printable Lesson Plan (a PDF of this web page)
Supplemental Materials	• Additional Printable Pages for Scaffolding and Practice • Matching Examples & Contracts (Desmos) • Matching Examples & Function Definitions (Desmos) • Project: Create Your Own Function

Glossary

contract :: a statement of the name, domain, and range of a function

example :: shows the use of a function on specific inputs and the computation the function should perform on those inputs

function :: a relation from a set of inputs to a set of possible outputs, where each input is related to

exactly one output

function definition :: code that names a function, lists its variables, and states the expression to compute when the function is used

variable :: a name or symbol that stands for some value or expression, often a value or expression that changes

Three Representations of a Function

55 minutes

Overview

Students will practice describing functions using all 3 representations (in programming syntax), and test their code against the examples in the editor.

Launch



- Open the [bc Starter File](#). Look at the Contract, some Examples, and the Function Definition for `gt`.
- What do you Notice? What do you wonder?

We know that...

1 Every **function** has a **Contract**.

```
# gt :: Number -> Image
```

2 We can write **examples** illustrating how a **function** should work to help us identify the pattern.

```
examples:
  gt(10) is triangle(10, "solid", "green")
  gt(15) is triangle(15, "solid", "green")
  gt(20) is triangle(20, "solid", "green")
  gt(50) is triangle(50, "solid", "green")
  gt(100) is triangle(100, "solid", "green")
  gt(200) is triangle(200, "solid", "green")
end
```

3 **Function definitions** replace whatever changes in the **examples** with a **variable** describing what changes.

```
fun
  gt(size):triangle(size, "solid", "green")
end
```



If we use the correct syntax, we can include all three of these function representations in our Pyret files. Let's take a look!

- Click "Run". What message do you get back?
 - *Looks shipshape, all 5 tests passed, mate!*
- What do you think that message means?

- The editor has checked to see whether the 5 examples work with the function definition and they do!
- Change `gt(10) is triangle(10, "solid", "green")` to `gt(15) is triangle(15, "solid", "green")`
- Click "Run". What happens?
 - The editor lets us know that the function doesn't match the examples so that we can fix our mistake!

Examples not only help us to identify the pattern to define a function, they also let us double check that the functions we define do what we intend for them to do!

Investigate



Think about these three representations of functions by completing:

- [Matching Examples and Contracts](#)
- [Matching Examples and Function Definitions](#)
- [Creating Contracts From Examples](#)

For more practice, complete these Desmos card sort activities:

- [Matching Examples & Contracts \(Desmos\)](#)
- [Matching Examples & Function Definitions \(Desmos\)](#)

There are many more materials for students to work with in the Additional Practice section at the end of the lesson!

Synthesize

- What strategies did you use to match the *examples* with the *contracts*?
- What strategies did you use to match the *examples* with the *function definitions*?

Defining bc and Other Functions

Overview

Using `gt` as an example, students will write the *Contract*, *examples*, and *definition* for several other *functions*.

Launch



- Turn to [Contracts, Examples & Definitions - bc](#)
- On the top half of the page, you will see the Contract, examples, and function definition for `gt`.
- Circle what is changing and label it with the word `size`.
- Using `gt` as a model, complete the Contract, examples and function definition for `bc`.

When students have completed the above steps, direct them to type the Contract, Examples and Definition into the Definitions Area. They will then click “Run”, and make sure all of the examples pass!

Check-in with students to gauge their confidence level. (Thumbs up? Thumbs to the side? Thumbs down?) How confident do students feel in writing the *Contract*, *examples* and *function definition* on their own if they were given a word problem about another shape function?

Investigate



- Complete [Contracts, Examples & Definitions - Stars](#).
- Complete [Contracts, Examples & Definitions - Name](#).

As students work, walk around the room and make sure that they are circling what changes in the *examples* and labeling it with a *variable* name that reflects what it represents.

Optional Project: Create Your Own Function

In this lesson, students defined functions given a specific set of parameters. In [Project: Create Your Own Function](#), students get additional practice thinking about functions before developing and defining a function of their own.

Synthesize

- How were each of the representations helpful?
- Why is it important to write examples in our code?

Additional Exercises

- [Do the Examples Have the Same Contracts?](#)
- [Do the Examples Have the Same Contracts? \(2\)](#)
- [Matching Contracts and Examples](#)
- [Matching Contracts and Examples \(2\)](#)