



# Unit 2

## Contracts, Strings and Images

### Unit Overview

Students are introduced to a set-mapping representation for functions, in which the function object exists as a means of translating points from a Domain into a Range. Coupled with their understanding of Circles of Evaluation, students generalize their understanding of functions to include other datatypes, including Strings and Images.

### Learning Objectives

Students will:

- Learn Racket syntax and grammar, using the Circle of Evaluation.
- Learn how to define values to be used later, as variables.
- Students learn syntax for simple string and image-composition expressions

### Product Outcomes

- Students can make and manipulate the basic elements of their games -- numbers, strings, and images.

### State Standards

See [Bootstrap Standards Matrix](#) provided as part of the Bootstrap curriculum.

**Length: 90min**

### Materials and Equipment

- Student [workbook](#) folders - in pairs! - with names on covers.
- Pens/pencils for the students, fresh whiteboard markers for teachers
- Class posters (List of rules, language table, course calendar)
- Language Table (see below)

### Preparation

- Write agenda on board
- Display Class posters, Language Table
- Seating arrangements: ideally clusters of desks/tables
- Optional: demo machine with projector to show the interactions and definitions windows

### Language Table

Types	Functions
Number	+ - * / sqrt expt

### Agenda

- 25min [Introduction](#)
- 15min [Contracts](#)
- 10min [Strings](#)
- 35min [Creating Images](#)
- 5min [Closing](#)

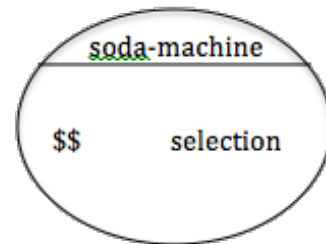
**Introduction****Time: 25 minutes**

- Welcome back! Who can remember what we did in the last class?
- At the end of class, you learned something extremely important: the Circles of Evaluation, and how to write simple programs in Racket.
- Suppose I have the math expression "2+3." How would I draw a circle of evaluation for this? How would I convert this into code?
- How about "2+3 divided by 1-2"?
- Let's do a review challenge:
  - Turn to **Page 6** in your workbooks, where it says "Circles Competition." On each row there is a mathematical expression in the first column, with room to draw the Circle of Evaluation and the Racket Code on the second and third columns.
  - Each row will be a round for this activity, so for Round 1 we'll just be looking at the first row. For this first row, we've even helped you out a little.
  - Each team will have one minute to draw the Circle of Evaluation, and then to convert that Circle to Racket code. As always, everyone must have the answer written down for your group to win the point. GO!
  - During the minute, walk around and see how groups are doing. Comment on good teamwork when you see it. Don't tutor much; let kids fail: they'll get it in review before the next round. When time is up (really 1 minute: the idea is to go quickly), give them a countdown: "30... 10... 5... 4... 3... 2... 1... PENCILS DOWN, EYES UP HERE!" Wait for total silence and complete attention.
  - Review and discuss. Assign points.
  - Repeat for each additional round.
- *Have kids complete the competition - give as little help as possible at this point. After the time is up, have students volunteer their answers. Review as necessary.*

**Contracts****Time: 15 minutes**

- Suppose you have to describe a soda machine. You know that the input to this machine will be money, and you have to get a soda in return. When you use a soda machine, does it only give you one type of soda? Wouldn't it be lame if you had to have a different machine for Coke than you did for Sprite?
- A soda machine, for example, takes in money and outputs soda.
 

```
; Soda-machine : Money Selection -> Soda
; Takes money and gives soda
```
- This description talks about outputs without having to say what those outputs are.
- This explanation of a soda machine just talks about some soda, but not any specific type of soda. It just says what kind of thing comes out, not which particular thing.
- How would you describe a coffeemaker? What kind of stuff goes into a coffeemaker? What kind of stuff comes out?
- How would you describe a lightbulb? *Have the students brainstorm a few other machines, and their inputs and outputs*
- The things that go into a machine are called the *inputs*, and what comes out is called the *output*. But the **kinds** of things that go in is the *Domain*, and the kind of thing that comes out is the *Range*. A soda machine's domain is money. Can you think of an example input? (any type of money: quarters, dimes, dollars) It's range is soda. What is the domain and range of a coffeemaker?
- Functions in Racket are the same: the addition function needs two numbers as its domain, but those numbers don't have to be 4 and 5. They can be any numbers.
- Okay, so you already know about a bunch of functions in Racket from math class: plus, minus, times, divide, etc.
- In Racket, a description of a function is called the **contract**. The contract tells you just what you need to know to use the function.
- A contract is also a promise: if you give the machine the kinds of things in its Domain, then you'll get something in the Range. If you give a pizza shop your order and some money, it promises to give you some hot, delicious pizza.
- Same thing in Racket: if you give plus two numbers, it will give you another number back.
- We'd like to create a list of contracts for ourselves, so that we can keep track of these functions and exactly how they work. Once we've practiced entering these contracts into our book, I'll show you new functions that let you work with words and pictures!
- Are you ready? Turn to the front of your workbooks, to the MOST IMPORTANT PAGE, where it says **Contracts**.
- A contract specifies
  1. the name of the function,
  2. the domain, and
  3. the range.



- Show the contract for "+" on the board, and have students volunteer the contracts for the other math functions covered so far.
- Add other functions, like "/" and "\*". Leave these contracts written on the board.
- If students already know about square roots or squaring, you can add these functions as well. If not, take a moment to review what they do.

```
sqrt : Number -> Number
sqr  : Number -> Number
```

## Strings

Time: 10 minutes

- One of the following does not belong. Can you find it? 6, "cat", 0, -2, 7.5
- Why doesn't "cat" belong? Why isn't 7.5 the odd one out? Or -2? Circle "cat".
- What you have all stumbled onto is that not all values are the same. Some of them are Numbers and some are words.
- Words like "cat" are called Strings. A String is anything in quotation marks. 5, is not the same as "5"!
- You'll notice that our math functions don't really work on Strings.
- What does it mean to add three and the word "cat"? It doesn't make sense, does it?
- To create a Number in Racket, we just typed it in. To create a String, you type in the word or words that you want, and put quotation marks on either side. **With your partner, try creating a few strings on your own.**
- Racket also gives us functions we can use to work with Strings. For example, there's a function called string-length. What do you think it does? I'll give you one hint: the contract. Write the contract on the board, but without the labels Name, Domain, Range:

```
; string-length: String -> Number
```

- What is the domain of this function? Raise your hand if you can tell me what you have to give to string-length. What about the Range - what will it give you back?
- Any guesses as to what it does? *Take 1-2 guesses.*
- As soon as you've written down the contract and we've checked it, you can turn on your monitors and experiment - see if you can figure out what it does! GO!



## Creating Images

Time: 35 minutes

- Let's talk about graphics.
- On the board, draw the Circle of Evaluation for (`circle 100 "solid" "red"`)
- Can someone tell me how to convert this into Racket code? *Copy their answer on the board.*
- This uses a new function, which you've never seen before! What's its name?
- Every contract has three parts! Raise your hand if you can tell me what they are!
- Can you figure out the contract for `circle`? Based on the example, can you tell me what's in its Domain?  
*Follow along on the board...*

```
; circle: Number String String -> ...
(circle 100 "solid" "red")
```

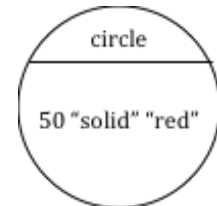
- So what's the Range? What do you think this thing is going to give us back? A Number? A String? Type it in and try it out!
- What it gives back is a new Type: **Image**!
- Now we're going to do the next step as a group.
- I'm going to show you the contract for another function. This time around, you're not allowed to touch the keyboard until everyone in the group has copied the contract into your contract tables, along with the contract for `circle`. Once you've all done that, I want to see which groups can draw a rectangle! GO!

```
; rectangle: Number Number String String -> Image
```

- Now I'm going to write a few more contracts on the board. Once again, you ALL have to have them written down before anyone can touch the keys. You'll have five minutes to figure out how to use each of these functions to make a shape! GO!

```
; ellipse: Number Number String String -> Image
; triangle: Number String String -> Image
; star: Number String String -> Image
; radial-star: Number Number Number String String -> Image
; text: String Number String -> Image
```

- Let kids experiment with these functions and point out interesting results to the class.
- **Going Further** - If time allows, you can go further into [manipulating images](#) or [making flags](#)!



**Closing****Time: 5 minutes**

---

- Who can tell us one thing we learned today?
- Who saw someone else in the class do something great?
- Cleanup, dismissal.

Bootstrap by [Emmanuel Schanzer](#) is licensed under a [Creative Commons 3.0 Unported License](#). Based on a work at [www.BootstrapWorld.org](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available at [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).