Unit 1
Everything you learned in Bootstrap 1...

Unit Overview

Agenda

5 min	Introduction
30 min	Expressions
20 min	Contracts
30 min	The Design Recipe
5 min	Closing

Students review almost all of the material from Bootstrap 1, including Circles of Evaluation, Contracts, Expressions dealing with strings and images, Variable definitions, Function definitions, and the Design Recipe.

Learning Objectives:

- Students will become comfortable writing complex expressions, defining variables and functions, and using the Design Recipe

Product Outcomes:

- Students define functions to meet a goal, or problem statement.

State Standards See our Common Core Standards Table provided as part of the Bootstrap curriculum.
Length: 90 minutes

Materials and Equipment:

- *Computers w/DrRacket or WeScheme*
- *Student workbooks*
- *Design Recipe Sign*
- *Language Table*
- *Bootstrap Review [Review.rkt from source-files.zip | WeScheme], already open on students' computers.*

Preparation:

- *Language Table Posted*
- *Seating arrangements: ideally clusters of desks/tables*

Introduction (Time 5 minutes)

- Welcome back to Bootstrap!
- *Introduce teaching staff. Have students introduce themselves as well. Review the rules:*
  - *Unless we say otherwise, raise your hand to speak*
  - *If you're not programming, the computer monitors need to be OFF. When we say "monitors off", everyone should shut off the monitors, stop talking, and give their attention to the teachers. Let's try this one out - everyone make sure the monitors are on...now start talking.....louder!....MONITORS OFF! Repeat as necessary.*
  - *No one in this class is allowed to put each other down. I don't ever want to hear anyone call each other "stupid", "smart one", "idiot", or anything else. Nobody tells each other to shut up, and everyone is respectful. Is that clear?*

  Divide the class into teams of 3-4 students. Give each team a name, and draw a scoreboard at the front of the room.
- Today, we're going to be covering everything you did in Bootstrap 1. Let's see how much you remember! We're going to start with a bit of a competition today.

Expressions (Time 30 minutes)

- *Monitors on! The "review" file should already be open on students' machines and your projector. Review UI of the DrScheme or WeScheme IDE.*
- At the top of the definitions window, there are a number of variables defined. What is the name of one of the variables?
- *Raise your hand if you can tell me:*
  What value is that variable is defined to have?
- When I click "Run"
  (click run)
  , and type the name of that variable into the interactions window, what should I get back? Try it out for yourselves on your own computers.
- *1, 2, 3...Monitors off! Let's focus on the code on the projector for now.*
- *Raise your hand if you can tell me:*
  What is the name and value of another variable that is defined here?
  (Repeat as necessary)
- Look at the variable OUTLINE. What shape will this draw? How big do you think it will be? Will it be solid or outline? What color will it be?
- *Monitors on.*
  Try it out! Was the fill what you expected it to be? It says outline in the code... why did it turn out solid? Raise your hand if you can explain this.
- The problem is that we used a very confusing variable name! The name was outline, but the value was "solid"! Let's remember this: always choose your variable names carefully!
- *Have teams define two additional Numbers, and practice using them in the interactions window. Have teams define two additional Strings. Have teams define one more Boolean. Have teams define one more Image.*
- Okay, we've got variables, and we know how to define them.
- However, we've also got some pre-built functions, which let us play around with these values. Do you know a function that works on numbers?
- *Review circles of evaluation with numbers, as well as how to convert them to Scheme code*
- Let's have a competition! Turn to Page 3 in your workbooks.
- Each row has a math expression. You'll have to convert that math expression into Circles of Evaluation, and then convert the Circle into Racket code.
- *Each team starts with three points. For every row you get correct, you will get two points. If you get stuck, your team can ask any teacher for help...but it will cost you one point. Questions? On your mark, get set...go!*

Contracts (Time 20 minutes)

- It's important for us to keep track of how our functions work - in Bootstrap 1 you also learned about Contracts. Every contract has three parts...raise your hand if you know what they are.
- That's right! Name, Domain and Range!
- *Review contracts for +, -, *, etc, making sure that students write them down in their workbooks. Review contracts for image-producing functions like circle, triangle, rectangle, star, etc, making sure that students write them down in their workbooks.*

The Design Recipe (Time 30 minutes)

- You remember how to define variables, and you know how to use contracts for pre-built functions. Let's see if you remember how to define a function of your own! Turn to Page 4 in your workbook.
- Here we have a function definition:

```
;double : Number -> Number
;takes a number and multiplies it by two
(define (double n)
        (* n 2))
```

- What is the name of this function? `double.` How do you know? It's in the contract!
- How many inputs does it have in its domain? One. How do you know? It's in the contract!
- What kind of data is the domain? A number. How do you know? It's in the contract!
- What is the name given to this input? `n`. How do you know? It's in the beginning of the definition!
- What's the Range of this function? Number. How do you know? The contract!
- Now let's talk about what the function does to its input. Look at the two EXAMPLEs, written below. Raise your hand if you can tell me what happens when I type `(double 5)`. What about the next example?
- What would happen if I changed the name of the input `n` to something else, like `x`. What else would have to change? *Keep asking more questions, to make sure they understand.*
- *Okay, now it's time for a little team competition! I'm going to give your teams some new function descriptions, and you'll need to create the contract, two examples, and the definition.*
- We'll do one to practice: First write only the contract and examples! The contract and each example will be worth one point apiece, for a maximum of three points. Write the contract and examples for a function called `triple`, which takes in a number as its input and multiplies it by 3. You have one minute! GO!
- *Walk around and help, while counting down... 3! 2! 1! Eyes on me!*
- Now look at your two examples. What is the only thing that changes from one to the other? The number being multiplied by three! That is the only thing that is changeable, or variable.
- Now take another minute, and write the function header and body for triple. Don't forget to replace the changing thing with a variable!
- *Walk around and help, while counting down... 3! 2! 1! Eyes on me!*
- *Okay, you all did a great job on the practice round. Now I'm going to give you 2 min. to write the entire function, contract, and examples for the next function description. You MUST write your contract and examples first! When you are done, raise your hand so I can check it! If you move on to writing the definition before I have checked your examples, your team will lose a point! Repeat the competition for each of the following examples:*
-
    - Write a function `plus1`, that takes in a number and adds one to it
    - Write a function `mystery`, that takes in a number and subtracts 4
    - Write a function `redspot`, that takes in a number and draws a solid red circle, using the number as the radius
- *NOTE: the last one is very difficult, and kids probably won't know quite what to do. This is intentional! Give them some time to be frustrated with it, and then cut it short.*

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