# Solving Word Problems

(Also available for Pyret)

Students discover functions as an abstraction over an arithmetic pattern, applying the Design Recipe to traditional word problems.

| | |
|---|---|
| **Prerequisites** | Defining Functions |
| **Relevant Standards**<br>OK<br>CC-Math | *Select one or more standards from the menu on the left (⌘-click on Mac, Ctrl-click elsewhere).* |
| **Lesson Goals** | Students will be able to:<br>• Understand how to use the Design Recipe to break down word problems.<br>• Create a strong purpose statement that details in their own words what the function should do. |
| **Student-Facing Lesson Goals** | • I can use the ***Design Recipe*** to break down word problem when writing a ***function***.<br>• I can identify the ***domain*** and ***range*** and other quantities in a word problem when writing a function.<br>• I can create and revise a strong ***purpose statement*** that explains what the function is doing. |
| **Materials** | • Lesson slides (Google Slides)<br>• Rocket-height starter file (WeScheme) |
| **Preparation** | • Make sure all materials have been gathered<br>• Decide how students will be grouped in pairs |
| **Supplemental Resources** | • Desmos Expression Bundle (Desmos Activities)<br>• Variables and Expressions (Quizizz)<br>• Desmos Functions Bundle (Desmos Activities)<br>• Functions & Relations (Desmos Polygraph Activity)<br>• Functions Quiz (Quizizz)<br>• Function Notation (Quizizz)<br>• Design Recipe Practice (Desmos Activity)<br>• Design Recipe Practice - Blank Template (Desmos Activity) |
| **Key Points for the Facilitator** | • The **purpose statement** is a comment in the code - something the computer doesn't read. It is important for readability of their code - there may be other people looking at their code and using their functions!<br>• Remind students that the domain and range of a function must be one or more of the three ***data types*** (Number, String, Image) they've learned so far.<br>• If students struggle with creating the examples, use the Circle of Evaluation mapping activity or use role-playing to help students build up their understanding around the concept. |

For a textbook-like version of materials similar to these, you may wish to see the prior unit-based version

***Glossary***

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

**datatypes ::** a way of classifying values, such as: Number, String, Image, Boolean, or any user-defined data structure

**design recipe ::** a sequence of steps that helps people document, test, and write functions

**domain ::** the type or set of inputs that a function expects

**function ::** a mathematical object that consumes inputs and produces an output

**purpose statement ::** a brief description of what a function does

**range ::** the type or set of outputs that a function produces

---

# Warmup

Students should have their workbook, pencil, and be logged into WeScheme on their computer.

---

# Writing Linear Functions

25 minutes

## *Overview*

Students are given a non-working program, which uses a linear function to determine the height of a rocket after a given length of time. The "broken" code is provided to lower cognitive load, allowing students to focus on comprehension (reading the code) and making use of structure (identifying where it's broken).

## *Launch*

Ask students to open the rocket-height Starter File and click "Run". By typing `(start rocket-height)`, they will see the simulation start to run on their computer.

> **Notice and Wonder**
>
> What do you notice about this program? What do you wonder?

Survey the class on their "Notices" and "Wonders" and record on the board before moving on to the discussion.

- Is `rocket-height` working?
- Why do you think it's not working?
- What do you think the **purpose** of this function is? How do you know?
- What is the domain of `rocket-height`? *Number*
- What is the range of `rocket-height`? How do you know? *Number, we can tell by looking at the **contract** for the function.*
- As the program is currently written, what happens when I give the function an input of 5? 15? One million? *It always returns 0.*

You've started to master most of the steps of the Design Recipe, but there's one part you haven't seen yet: *writing a purpose statement*. Programmers and Mathematicians alike find it helpful to restate a problem in their own words. After all, if you can't explain a problem to someone else, you probably don't understand it yourself!

## *Investigate*

Let's use the Design Recipe to fix `rocket-height`, and get comfortable with writing **purpose statements**.

- Have students turn to Word Problem: rocket-height (Page 27) and read the problem statement with their partner.
- Now that the students have revised and refined their purpose statement, have them write the ***Contract*** and ***purpose statement*** on Word Problem: rocket-height (Page 27) worksheet.
- Given the contract and purpose statement, write two examples of how `rocket-height` should work after two different lengths of time.
- Circle and label what's changing in the two examples, just as they did with their green triangle function before.
- Choose a good variable name for what's changing.
- Write the function definition using the variable name.
- Once the Design Recipe has been completed in the workbook, students can type the code into the `rocket-height` program, replacing any incorrect code with their own code.

## *Synthesize*

- What was the problem?
- What mistake did the programmer make?
- Where in the Design Recipe did they first go astray?

*The Design Recipe allows us to trace mistakes back to the source!*

---

# More Interesting Functions

flexible

## *Overview*

For teachers who cover quadratic and exponential functions, this activity deepens students' understanding of functions and extends the Design Recipe to include those. This can also be a useful activity for students who finish early, or who need more of a challenge.

## *Launch*

Now that `rocket-height` is working correctly, explore the rest of the file and try the following:

- Remove the comment from before the `(start rocket-height)` and test the program.
- Put the comment back in front of `(start rocket-height)`, remove the comment from `(graph rocket-height)`, and test the program.
- Try out `(space rocket-height)`
- Try out `(everything rocket-height)`

## *Investigate*

- Can you make the rocket fly faster? Slower?
- Can you make the rocket sink down instead of fly up?
- Can you make the rocket *accelerate over time*, so that it moves faster the longer it flies?
- Can you make the rocket blast off *and then land again*?
- Can you make the rocket blast off, *reach a maximum height of exactly 1000 meters*, and then land?
- Can you make the rocket blast off, reach a maximum height of exactly 1000 meters, and then land after exactly 100 seconds?
- Can you make the rocket fly to the edge of the the universe?

### *Synthesize*

Debrief - what did students try? Have students share their experiments with one another!

---

# Additional Exercises:

- Define a function `purple-star`, that takes in the size of the star and produces an outlined, purple star of the given size.
  Word Problem: purple-star
- Define a function `spot`, that takes in a color and produces a solid circle of radius 50, filled in with that color. (Word Problem: spot)
- Define a function `average`, which takes in two numbers and produces their average. (Word Problem: average)
- Do Examples Have the Same Contracts? (1) (original, answers)
- Do Examples Have the Same Contracts? (2) (original, answers)
- Matching Contracts and Examples (1) (original, answers)
- Matching Contracts and Examples (2) (original, answers)