

## Unit 1 - Course A

The first 10 hour course provides students with the foundational skills and knowledge to begin using computer programming as a tool to learn about and develop algebraic functions. Students will be introduced to a graphical programming language designed for Algebra instruction, through which they will gain a deeper understanding of the order of operations, create images with algebraic expressions, and learn a technique for creating functions called the Design Recipe.

By the end of Course A students will have the tools necessary to turn word problems from their own Algebra class into functions that can be used as mini apps.

# Lesson 1: Evaluation Blocks and Arithmetic Expressions

## Overview

Students will begin using Evaluation Blocks to explore the concept of math as a language, and more specifically, a programming language. By composing arithmetic expressions with Evaluation Blocks, students will be able to visualize how expressions follow the order of operations.

## Purpose

In this lesson students get their first taste of the programming language and environment that they'll use throughout this course. While the environment and block-based language we're using here look a lot like things students may have seen in Scratch or the Hour of Code, it's important to understand that the kind of programming we're doing here is a bit different. In order to better align with the rules of Algebra we are following a programming paradigm known as **Functional Programming**. This difference may not matter to your students, but for those who have some existing programming experience you may ask them to research this different paradigm and report back on the similarities and differences.

## Agenda

**Activity: Evaluation Blocks**

**Online Puzzles**

**Getting Started**

**Learning a Language**

## Anchor Standard

### Common Core Math Standards

- **A.SSE.1** - Interpret expressions that represent a quantity in terms of its context.

## Objectives

### Students will be able to:

- Convert arithmetic expressions to and from code.
- Use Evaluation Blocks to reflect the proper order of operations for an expression.

## Links

### For the Students

- **Evaluation Blocks Worksheet** ([https://code.org/curriculum/docs/algebra/worksheets/evaluation\\_blocks.pdf](https://code.org/curriculum/docs/algebra/worksheets/evaluation_blocks.pdf)) - Worksheet
- **CS in Algebra Lesson 2 Slide Deck** ([https://docs.google.com/a/code.org/presentation/d/1\\_OOPjfAQUfpONIOHOnHqIegnW96trR-GUT1qg-rpcjw/](https://docs.google.com/a/code.org/presentation/d/1_OOPjfAQUfpONIOHOnHqIegnW96trR-GUT1qg-rpcjw/)) - Slide Deck

## Vocabulary

- **Evaluate** - Perform the computation in an expression, producing an answer.
- **Evaluation Block** - A block of code that represents the structure of an expression
- **Expression** - Any valid unit of code that resolves to a value.
- **Function** - A mathematical object that takes in some inputs and produces an output.
- **Value** - A specific piece of data, like 5 or "hello".

# Teaching Guide

## Activity: Evaluation Blocks

### Online Puzzles

The programming language you are going to learn uses Evaluation Blocks to visually represent mathematical functions. Each block of code is either a Function, or a Value - head to **CS in Algebra, Course A Stage 1** in Code Studio to get started programming.

### Getting Started

Welcome to Code.org CS in Algebra! In this course you'll be learning a new programming language - a way to tell computers exactly what you want them to do. Just like English, Spanish or French, a programming language has its own vocabulary and grammar that you'll have to learn. Fortunately, the language you'll be using here has a lot in common with the simple math that you already know!

### Learning a Language

Discuss as a class the nature of language, for example: - What makes a language? - Does anyone speak a second (or third) language? Do you speak a different language than your parents/grandparents? - Are there languages that share features, such as a common root (Romance, Germanic) or a similar alphabet (Latin, Cyrillic, Arabic, Kanji)? - Are there languages that are designed for specific purposes or within certain constraints (sign language, Esperanto)? - Math is a language, just like English, Spanish, or any other language! - We use nouns, like "bread", "tomato", "mustard" and "cheese" to describe physical objects. Math has values, like the numbers 1, 2 or 3, to describe quantities. - We also use verbs like "toast", "slice", "spread" and "melt" to describe operations on these nouns. Mathematics has functions like addition and subtraction, which are operations performed on numbers. - Just as you can "slice piece of bread", a person can also "add four and five".

A mathematical expression is like a sentence: it's an instruction for doing something. The expression  $4+5$  tells us to add 4 and 5. To evaluate an expression, we follow the instructions in the expression. The expression  $4+5$  evaluates to 9.

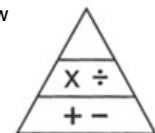
Sometimes, we need multiple expressions to accomplish a task. If you were to write instructions for making a sandwich, it could matter very much which came first: melting the cheese, slicing the bread, spreading the mustard, etc. The order of functions matters in mathematics, too. If someone says "four minus two plus one," they could mean several things:

- Subtract two from four, then add one:  $(4 - 2) + 1$
- Add two and one, and subtract the result from four:  $4 - (2 + 1)$

Depending on which way you read the expression, you might have very different results! This is a problem, because we often use math to share calculations between people. For example, you and your cell phone company should agree upfront on how much you will pay for sending text messages and making calls. Different results might mean that your bill looks wrong. We avoid problems by agreeing on the order in which to use the different operations in an expression. There are two ways to do this:

1. We can all agree on an order to use
2. We can add detail to expressions that indicate the order

Mathematicians didn't always agree on the order of operations, but now we have a common set of rules for how to evaluate expressions. When evaluating an expression, we begin by applying the operations written at the top of the pyramid (multiplication and division). Only after we have completed all of those operations can we move down to the lower level. If both operations are present (as in  $4 - 2 + 1$ ), we read the expression from left to right, applying the operations in the order in which they appear.



Evaluation Blocks provide a visual way to indicate the order of operations in an expression.

All Evaluation Blocks follow three rules:

- **Rule 1:** Each block must have one function, which is displayed at the top of the block.
- **Rule 2:** The values for that function are placed below, in order from left to right.
- **Rule 3:** If a block contains another block as a value, that inner block must be evaluated before the outer block.

Before students get started on the computers, you can have them work through the **Evaluation Blocks Worksheet** in the student workbook.

### Standards Alignment

#### Common Core Math Standards

- ▶ **EE** - Expressions And Equations
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking

- ▶ **Q** - Quantities
- ▶ **REI** - Reasoning With Equations And Inequalities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 2: Strings and Images

### Overview

To compute more than just numbers, students will need to learn about two new data types, Strings (any string of alphanumeric characters) and Images. Using these new data types, we'll compose programs that produce and manipulate images.

### Purpose

The programming language we are using in this course is a **strongly typed** language, meaning that we must explicitly identify the types of data we're using.

### Agenda

**Activity: Strings and Images**

**Online Puzzles**

**Getting Started**

**Introduction**

### Anchor Standard

#### Common Core Math Standards

- **A.SSE.1** - Interpret expressions that represent a quantity in terms of its context.

### Objectives

#### Students will be able to:

- Write and evaluate expressions for generating Strings and Images.

### Vocabulary

- **Data Type** - A general kind of data, such as a Number, String, or Image
- **Image** - A type of data used for graphics or pictures.
- **String** - Any sequence of characters between quotation marks (ex: "hello", "42", "this is a string!").

# Teaching Guide

## Activity: Strings and Images

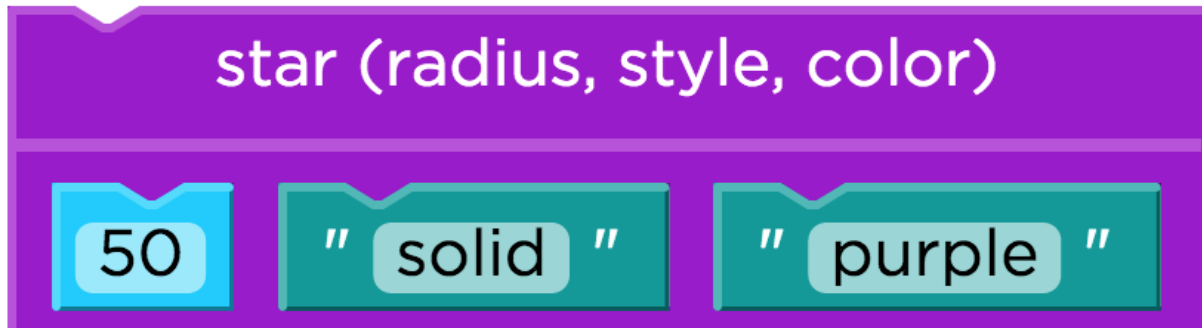
### Online Puzzles

In this activity you'll use the new data types String and Image to compose art with Blocks of Evaluation - head to **Course A Stage 2** in Code Studio to get started programming.

### Getting Started

### Introduction

In the previous stage, students only worked with a single type of value - Numbers. In this next stage they will get a chance to write programs with new data types to output text (Strings) and pictures (Images).



Show students the 'star' function, and ask them to discuss the following questions:

- What is the name of this function?
- How many arguments are being given to this function?
- What do you think this function will do?

Students are not expected to know all the answers here - the goal is for them to apply what they know about Evaluation Blocks to a novel expression, and discuss for themselves what they think it might mean. Ask them to justify their answers, and to explain why they think they are correct. Encourage students to look for patterns among these new blocks (such as colors, or quotation marks around the words "solid" and "purple" - what might those patterns mean?

### Standards Alignment

#### Common Core Math Standards

- ▶ **EE** - Expressions And Equations
- ▶ **F** - Functions
- ▶ **IF** - Interpreting Functions
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities
- ▶ **REI** - Reasoning With Equations And Inequalities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 3: Contracts, Domain, and Range

### Overview

Contracts provide a way for students to better understand and discuss functions. Through this lesson, students will look at known functions and come up with the contracts that describe those functions.

### Agenda

#### Getting Started

- What's in a Function?
- Contracts have three distinct parts

#### Activities

- Reading Contracts
- Writing Contracts

#### Wrap-up

- Keep up your Contracts

### Anchor Standard

#### Common Core Math Standards

- F.IF.1** - Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If  $f$  is a function and  $x$  is an element of its domain, then  $f(x)$  denotes the output of  $f$  corresponding to the input  $x$ . The graph of  $f$  is the graph of the equation  $y = f(x)$ .

### Objectives

#### Students will be able to:

- Describe a function in terms of its name, domain, and range.
- Create contracts for arithmetic and image-producing functions.

### Links

#### For the Teacher

- CS in Algebra Lesson 3 Slide Deck** (<https://docs.google.com/presentation/d/1kmA6p6xCsoN1h4HDdi0la7tlwhLg55Yvc5AS6k7DzOg>) - Slide Deck

#### For the Students

- Contract Log** ([https://code.org/curriculum/docs/algebra/worksheets/contract\\_log.pdf](https://code.org/curriculum/docs/algebra/worksheets/contract_log.pdf)) - Worksheet

### Vocabulary

- Contract** - A statement of the name, domain, and range of a function.
- Domain** - The type of data that a function expects.
- Range** - The type of data that a function produces.

# Teaching Guide

## Getting Started

### What's in a Function?

You've already seen several functions that take in two Numbers, such as `+`, and `-`. Other functions like `star`, take in a Number and two Strings. Different functions take in different inputs, and we need a way to keep track of the requirements for each function.

- What does the `'+'` function do?
  - What does it take as input?
  - What does it return as output?
- How about the `'triangle'` function?
- What do these different functions have in common?

Let's look at a simple way to describe any function, it's called a "contract"

- What is a Contract?
  - A formal agreement
  - A description of expected behavior
- What do Contracts tell us?
  - What a function should do
  - What inputs a function needs
  - What a function returns

Encourage students to think about contracts in the real world. What purpose do they serve? If a contract is signed, do we expect it to be followed?

### Contracts have three distinct parts

1. Name
2. Domain
3. Range

**The Name of a function briefly describes what the function does.**

**The Domain of a function is the data that the function expects.**

**The Range of a function is the data that the function produces.**

By keeping a list of all the functions in a language, and their Domains, programmers can easily look up how each function is used. However, it's also important to keep track of what each function produces! For example, a program wouldn't use `star` if they were trying to produce a Number, because `star` only produces Images.

Domains and Ranges help programmers write better code, by preventing silly mistakes and giving themselves hints about what to do next. A programmer who wants to use `star` can look up the Domain and immediately know that the first input has to be a Number (like 100), without having to remember it each time. Instead of writing a single value there, a programmer could write a whole expression, like `(25 * 4)`. We know this code will return an appropriate value (Number) by looking at the Range for `*`; therefore, the result of `*` can be used in place of any Number value.

When programmers write down the Domains and Ranges of each function, they write what are called **contracts**, to keep track of what each function needs.

### Activities

#### Reading Contracts

Let's look at a few example contracts - for each contract we'll identify the Name, Domain, and Range

- `+`: Number Number -> Number



- `triangle`: Number String String -> Image





- rotate: Number Image -> Image



## Writing Contracts

Let's see if we can come up with contracts for some of the functions you've already seen. You'll want to make sure that you've got your **Contract Log - Worksheet**, as this is where you'll keep a running document of all contracts you write - both for existing functions and ones of your own creation.

- We'll start with contracts for simple arithmetic functions
- +, -, \*, /

Those were pretty easy as arithmetic functions only deal in Numbers. When it comes to writing functions that deal with multiple data types, looking at the Evaluation Block can give us some helpful clues.

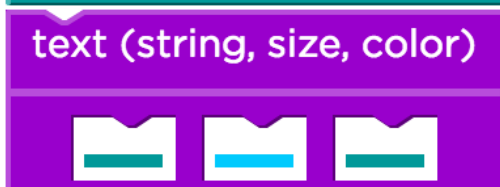
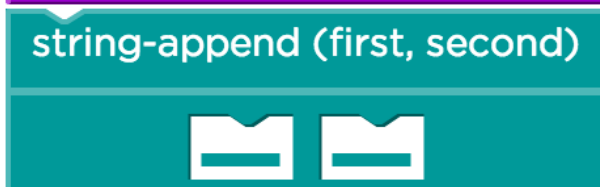
- The Name of each function is at the top
- There will be a slot for each Domain element
- The color of each slot tells you Domain type
- The color of the whole block tells you Range
- Color codes: Number String Image

Display each of the following Evaluation Blocks and ask students:

- What is the Name of this function?
- What is the Domain of this function?
- What is the Range of this function?
- Add this function's contract to your reference

### Lesson Tip

Common mistakes when students first write down contracts include: writing values (such as "red") instead of types (such as "String") and forgetting arguments. Read your students' contracts carefully, as they often indicate misconceptions that will persist and affect them later on.



## Wrap-up

## Keep up your Contracts

As you continue programming, make sure that you document a contract for every new function you encounter or write. In the next unit, you'll learn how to create your own functions to save work in writing expressions (this will turn out to be an essential part of writing a game). You'll also start customizing your game with images for the elements in your game design.

## Standards Alignment

### Common Core Math Standards

- ▶ **EE** - Expressions And Equations
- ▶ **F** - Functions
- ▶ **IF** - Interpreting Functions
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 4: Writing Contracts

### Overview

Students will work their way through a number of new functions, first using each to solve a problem, and then writing a contract which describes it.

### Agenda

Writing Contracts

Online Puzzles

Getting Started

Introduction

### Anchor Standard

#### Common Core Math Standards

- **8.G.1** - Verify experimentally the properties of rotations, reflections, and translations:

### Objectives

#### Students will be able to:

- Decompose existing functions.
- Write contracts that describe functions.
- Experiment with basic geometric transformations.

### Vocabulary

- **Rotate** - To turn a shape around a center point.
- **Scale** - To increase the dimensions of a shape by the same factor in all directions.
- **Translate** - To move a shape from one location to another.

# Teaching Guide

## Writing Contracts

### Online Puzzles

In this stage you'll be looking at some functions, some of which you've seen before and some which are brand new. For each function you'll first get a chance to use the function, and then you'll write a Contract for it. Make sure to document any new Contracts on your Contract Log. Head to **Course A stage 4** in Code Studio to get started programming.

### Getting Started

#### Introduction

Review with students the purpose of a Contract:

- Describes three elements of a function
  - Name (what is the function called)
  - Domain (what inputs does it take)
  - Range (what does it output)
- As a class, describe the Contracts for some basic mathematical operators
  - Addition (name +, domain Number Number, range Number)
  - Subtraction (name -, domain Number Number, range Number)
  - Multiplication (name \*, domain Number Number, range Number)
  - Power of two (name sq, domain Number, range Number)

#### Standards Alignment

##### Common Core Math Standards

- ▶ **EE** - Expressions And Equations
- ▶ **F** - Functions
- ▶ **G** - Geometry
- ▶ **IF** - Interpreting Functions
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 5: Defining Variables and Substitution

### Overview

In this activity, students will learn to define variables that can be used to reference values and expressions. Once defined, their variables can be used repeatedly throughout a program as substitutes for the original values or expressions.

### Purpose

As stated earlier, the programming paradigm that students are learning in this course is called **functional programming**, and it varies in several ways from programming paradigms that students may have seen elsewhere. One of the core tenants of functional programming is the concept of **immutability**. Immutable objects may not change state once they have been created, which allows variables to function as we would expect them to in Algebra.

### Agenda

#### Getting Started

#### Introduction

#### Defining Variables and Substitution

#### Online Puzzles

### Anchor Standard

#### Common Core Math Standards

- **6.EE.4** - Identify when two expressions are equivalent (i.e., when the two expressions name the same number regardless of which value is substituted into them). For example, the expressions  $y + y + y$  and  $3y$  are equivalent because they name the same number regardless of which number  $y$  stands for.

### Objectives

#### Students will be able to:

- Define variables by giving them a name and assigning them a value or expression.
- Use variables within Evaluation Blocks.
- Describe a situation where using variables as substitutions for values or expressions is more efficient.

### Vocabulary

- **Define** - Associate a descriptive name with a value.
- **Variable** - A reference to a value or expression that can be used repeatedly throughout a program.

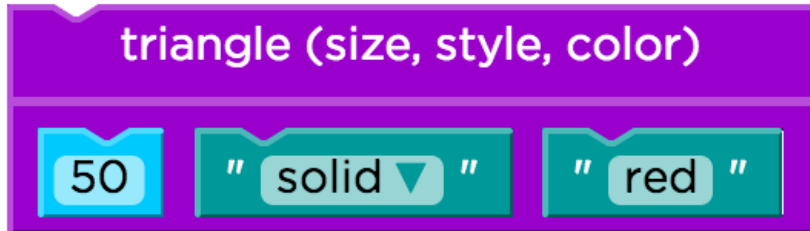
# Teaching Guide

## Getting Started

### Introduction

Suppose we want to make an image with seventy-five identical, solid red triangles. To do so you'd have to create this Evaluation Block seventy-five times!

Even worse, if you decided you wanted seventy-five blue triangles instead, you'd have to go through and change each and every block. There must be a better way!



We can store that red triangle Evaluation Block in a Variable, let's call it "red-triangle." That name "red-triangle" now becomes a shortcut for the blocks inside the variable, and we can use that shortcut over and over in our program. If we decide that we want that red triangle to be 100 pixels instead of 50, we only need to change it in the variable definition.

### Defining Variables and Substitution

#### Online Puzzles

In this stage you'll use variables to reference a variety of values and expressions. Head to **Course A stage 5** in Code Studio to get started programming.

#### Lesson Tip

If students have used variables in other programming languages, it's essential to note that in functional programming, as in math, variables are considered **immutable** - meaning the value can't be changed during the execution of a program. Think about it this way: saying  $x = 50$ , and then  $x = x + 1$  might make sense in Javascript, but it's impossible in Algebra.

### Standards Alignment

#### Common Core Math Standards

- ▶ **CED** - Creating Equations
- ▶ **EE** - Expressions And Equations
- ▶ **IF** - Interpreting Functions
- ▶ **LE** - Linear, Quadratic, And Exponential Models★
- ▶ **MP** - Math Practices
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 6: Fast Functions

### Overview

In this lesson we will build on students' understanding variables by making functions that reduce the number of inputs required. These 'fast functions' allow students to practice using the Design Recipe to develop simple functions without the additional overhead of parsing a word problem.

### Agenda

#### Getting Started

##### Introduction

#### Large Group Activity (15 minutes)

##### Whole Class

#### Small Group Activity (20 minutes)

##### Breakout Groups

### Vocabulary

- **Function** - A mathematical object that takes in some inputs and produces an output.
- **Independent Variable** - An input variable the is controlled by the user.
- **Parameter** - A value or expression belonging to the domain of a function.
- **Variable** - A reference to a value or expression that can be used repeatedly throughout a program.

# Teaching Guide

## Getting Started

### Introduction

Variables in this language are immutable, which means that once they've been defined they can never be changed. This allows us to use variables to use an exact value repeatedly, but what if we wanted to make a slight change to that value each time it's used?

Suppose we wanted to create seventy-five solid-red triangles of varying size? The same frustration exist of creating seventy-five different triangle functions. There must be a better way!

We can store that triangle Evaluation Block in a Function, let's call it `red-triangle`. Now `red-triangle` will ask the user for what size they would like to create keeping the triangle always green and solid, thus reducing the amount of inputs required.

#### 💡 Teaching Tip

This is very similar to a Variable because we are substituting one function in for another. The major difference is that Functions can receive an input from the user. Variables can only be repeated exactly and any changes made inside the code will affect all instances that variable is used. Now we have the ability to ask the user for one changing aspect of what we want and pass that information to as many functions as we want.

### Large Group Activity (15 minutes)

In this stage you will need to create some background stories to engage students into WHY we would want these Fast Functions.

### Whole Class

- Define a function `blue-circle`, that takes in a size and produces a solid circle of color blue on your Fast Function Worksheet.

### 🎤 Say (with enthusiasm)

Do you know my favorite Image? I absolutely love blue circles! I like small circles, large circles, and everything in between. I want to make a t-shirt with hundreds of blue circles, can you help me?

Can we use what we already know about variables? But what if I want a variety of sizes? The circle function is just too much work and no body has time for all those inputs. Can we create a function?..."

Students have been working on Contracts and basic Definitions but now we are asking something more complex and abstract. Be sure to slow down the process and focus on "What the Function Should Do". The process of making EXAMPLES is what forces the the programmer to slow down and allows them to make sure the function does what it is supposed to do.

How to Create EXAMPLES:

- Bring down the Function-Name from the contract and ask WHY that should be the function name
- Ask the students what input this function should take. Be sure to require them to use evidence from the Contract.
- Ask students what output should this function DO or CALL. Many students will want to DO the math instead of having the function do the work.
  - For `blue-circle` remind students that if any function calls itself it wouldn't make sense like if you call yourself on your own phone. `blue-circle` needs to call another function to help it create this function (it calls `circle` which is in your Contract Log)
- Build the second Example in the same fashion
- Go piece by piece and CIRCLE what CHANGES and label it as a variable.

How to Create DEFINITION:

- Go piece by piece of the example and bring down pieces.
- Pull down your variable name from the Circled Change of the Example
- Pull down the function the `blue-circle` calls and its inputs from Example. Be sure to replace anything that Changes with a variable.

### Small Group Activity (20 minutes)

#### Breakout Groups

Put students into groups of 3 - each member of the group will represent one step of the Fast Function

1. Contract
2. Examples
3. Function

Their task will be to create the following three Fast Functions:



- Define a function `spot`, that takes in a color and produces a solid circle of radius 50, filled in with that color.
- Define a function `average`, which takes in two numbers and produces their average. (You may need to remind the students that to find the average of two numbers, they should be added together and divided by two.)
- Suppose a company logo is a word drawn in big, red letters, rotated some number of degrees. Define a function `logo`, that takes in a company name and a rotation, and produces a logo for that company.

### Looking Forward

Students are building their understanding to eventually

## Standards Alignment

### Common Core Math Standards

- ▶ **BF** - Building Functions
- ▶ **IF** - Interpreting Functions



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 7: Composite Functions

### Overview

In the past lessons students have defined Variables and written Fast Functions. In this stage, they will continue to explore function writing with ever increasing complexity.

### Agenda

#### Getting Started

#### Introduction

#### Composite Functions

#### Online Puzzles

### Anchor Standard

#### Common Core Math Standards

- **8.F.1** - Understand that a function is a rule that assigns to each input exactly one output. The graph of a function is the set of ordered pairs consisting of an input and the corresponding output.<sup>1</sup>

### Objectives

#### Students will be able to:

- Analyze and use existing functions.
- Modify existing functions.
- Create new functions.
- Create similar shapes by changing size parameters on functions.

### Vocabulary

- **Parameter** - A value or expression belonging to the domain of a function.

# Teaching Guide

## Getting Started

### Introduction

In the previous lessons students began composing functions together to make their lives easier. This Stage should be used as a way to deepen the understanding of students. There are assessments built into the later levels, so be sure to use probing questions to have students think deeply about their work.

Level 1 through 5 are composite image functions and beware of aqua-star as this is where most students struggle transferring their learning. Pause at Level 4 to make sure all students have a firm understanding.

Levels 6 through 9 bring the content back to math. It is crucial that students continue to use their Fast Function process for these problems because that will help make this Design Recipe more applicable to Algebra problems in the future.

Assessment Levels 10 through 15 are a great way to have students check their understanding and are by no means required.

This is a Stage that, as a teacher, you have full flexibility as to which levels you want students to complete.

### Composite Functions

### Online Puzzles

In this stage you'll define simple functions. Head to **Course A stage 7** in Code Studio to get started programming.

Be sure that students have plenty of space on their Fast Functions form so they can build these new functions from the Contract to the Examples to the Definition. Great code starts on paper first!

### Standards Alignment

#### Common Core Math Standards

- ▶ **CED** - Creating Equations
- ▶ **F** - Functions
- ▶ **G** - Geometry
- ▶ **IF** - Interpreting Functions
- ▶ **LE** - Linear, Quadratic, And Exponential Models★
- ▶ **MP** - Math Practices
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities
- ▶ **SSE** - Seeing Structure In Expressions



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 8: The Design Recipe

### Overview

In the last stage, students wrote some very simple functions - but more sophisticated functions demand a more thoughtful approach. The Design Recipe is a structured approach to writing functions that includes writing a purpose statement and test cases to ensure that the function works as expected. Once students have mastered the Design Recipe process, they can apply it to any word problem they encounter.

### Purpose

The Design Recipe process for writing functions introduces students to some best practices in software design that most don't learn until college or later. By writing examples (or test cases, as they are often called in industry) before writing the body of the function students are following a practice called **test-driven development**. In TDD software developers write failing test cases before adding a new features, and then works on their code until the test cases pass. This ensures that developers stay focused on writing code that does what they set out to do while also putting in place automated testing that ensures the program continues working as expected while more features are added.

### Agenda

#### Getting Started

- What is the Design Recipe?
- Step 1 - The Contract
- Step 2 - Examples
- Step 3 - Function Definition

#### Activity

- Design Recipe Battle
- Large Group Battle
- Driver - Navigator Battle

### Anchor Standard

#### Common Core Math Standards

- F.BF.1** - Write a function that describes a relationship between two quantities.

### Objectives

#### Students will be able to:

- Use the Design Recipe to identify dependent variables, independent variables, and constants.

### Links

#### For the Teacher

- CS in Algebra Lesson 8 Slide Deck** ([https://docs.google.com/presentation/d/1tG42SkPwG29TvLylzAhg\\_VGHwiaUMNf5\\_Rt45uWMkb8/](https://docs.google.com/presentation/d/1tG42SkPwG29TvLylzAhg_VGHwiaUMNf5_Rt45uWMkb8/)) - Slide Deck

#### For the Students

- Design Recipe Form** ([https://code.org/curriculum/docs/algebra/worksheets/design\\_recipe.pdf](https://code.org/curriculum/docs/algebra/worksheets/design_recipe.pdf)) - Worksheet
- Fast Functions** ([https://code.org/curriculum/docs/algebra/worksheets/fast\\_functions.pdf](https://code.org/curriculum/docs/algebra/worksheets/fast_functions.pdf)) - Worksheet

### Vocabulary

- Constant** - A fixed number in a relationship.
- Dependent Variable** - A value that changes based on an independent variable.
- Design Recipe** - A systematic sequence of steps to document, test, and write functions.
- Independent Variable** - An input variable the is controlled by the user.
- Purpose Statement** - A brief description of what a function does.

# Teaching Guide

## Getting Started

### What is the Design Recipe?

The Design Recipe is a roadmap for defining functions, which programmers use to make sure the code they write does what they want it to do. Each step builds on the last, so any mistakes can be caught early in the process. This roadmap has a series of steps:

1. Write a Contract that describes the word problem. 1a. Write a Purpose Statement that breaks down the problem to isolate the function.
2. Write Examples based on the contract.
3. Define a function that matches the examples.

Let's start out by applying the Design Recipe together to the following problem:

**Define a function** `purple-star` , **that takes in the size of the star and produces an outlined, purple star of the given size.**

### Step 1 - The Contract

**`purple-star: Number -> Image`**

Be sure to include a good Name for each function, and remember that the Domain and Range can only include types like Numbers, Images, Strings, etc.

A Contract is the foundation for a function, which gives programmers just enough information to use it: the name of the function, the type (or types) of data it expects and the type of data it returns.

The Purpose Statement gives students insight into what the function should do in the Example step.

Stop here and analyze student Purpose Statements for depth of understanding. At this step, it is critical for students to have a strong understanding of what the function should do even if they cannot figure which function they need to use.

### Step 2 - Examples

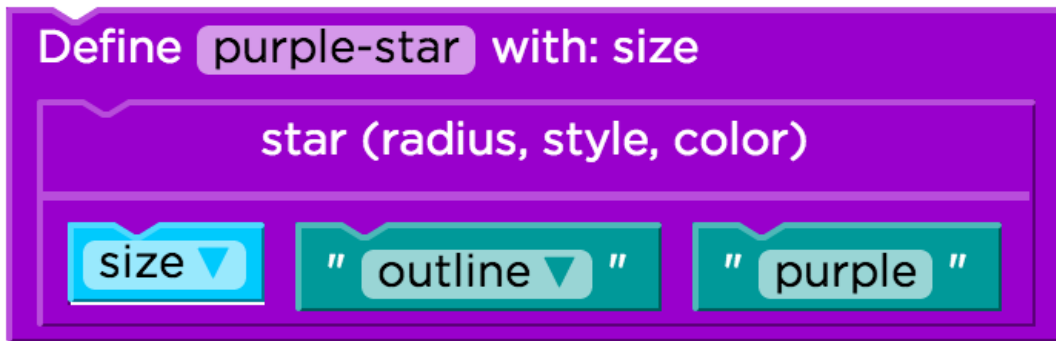
The image shows two examples of function calls and results. Each example is presented in a green box with a white header 'Example (call, result)'. The first example shows a function call 'purple\_star (size)' with an input of '45' and a result of 'star (radius, style, color)' with inputs '45', 'solid', and 'purple'. The second example shows a function call 'purple\_star (size)' with an input of '120' and a result of 'star (radius, style, color)' with inputs '120', 'solid', and 'purple'. The function names and inputs are highlighted in blue boxes, and the results are highlighted in purple boxes.

- Every Example begins with the name of the function. Where could you find the name of the function? (from the Contract)
- Every Example has to include sample inputs. Where could you find out how many inputs this function needs, and what types they are? (from the Contract)
- Every Example has to include an expression for what the function should do when given an input. Where could you look to find out what this function does? (from the Purpose Statement)

Once you have two or more Examples, it should be easy to identify what has changed between them by going word by word and circle what changes. In fact, the number of things that change should match the number of things in the function's Domain: if the Domain has a Number and a String in it, then those two values should be the things that differ between your Examples.

Be sure to label the circled change as a variable that makes sense to this problem before moving on to the Definition.

### Step 3 - Function Definition



By identifying what has changed between these Examples, we can define our actual function.

Remember that the Contract and Purpose Statement can be used to write the Examples, even if a programmer isn't sure how to begin. Then the Examples are used to build the function even if the programmer isn't sure what to do.

## Activity

### Design Recipe Battle

In this activity students in pairs will analyze the Design Recipe to work through the following problems

Students will be expected to defend every action they took on the Design Recipe using evidence only from the previous section.

### Large Group Battle

As a class you will need to model how battling works. Through your Document Camera, Projector, or just on the board, have a student complete a blank Design Recipe one section at a time.

- Give the student 2 minutes to complete the Contract and Purpose Statement.
- For one minute question every aspect of the Contract and Purpose statement. Be as specific as possible. They have to be able to answer your questions using only the description or word problem of the original problem.
  - Example: Where did you get the function-name from? Where did you get Number for the Domain? How did you know Number was supposed to be the Range? Where did you get the function in your purpose statement?
- Have the student cover up the description or word problem where the original problem came from. Now they must create their examples only using the Contract and Purpose Statement.
- Give 2 minutes for student complete both examples.
- For one minute question every aspect of the examples. Be as specific as possible.
  - Example: Where did you get your function-name? Why did you choose ## as your input? How did you get your output? Did you do the work or did the computer do the work to find the output? Why did you make your examples different? How will the computer know where the variable should go?
- Have the student cover up the Contract and Purpose Statement.
- Give the student 1 minute to Define the Function
- For 1 minute question every aspect of the definition. Be as specific as possible.
  - Example: Where did you get the variable from? How does the computer know what to do with the input? How do you know this function will behave like you want it to?
- You have successfully Battled! Awesome!

To further incentivize students, you can print a list of questions and have students keep score of how many they correctly defend.

### Driver - Navigator Battle

Now that your students have seen the type of questions and intensity of questioning modeled by you, they are ready to battle each other.

It is helpful to give all the Drivers (partner A) one problem and give Navigators (partner B) a different problem.

Your main priority to keep the energy high and time. The students will do the rest:

- 2 minutes to complete the Contract and Purpose Statement
- 1 minute (or 30 seconds depends on your class) for Driver to Battle (ask the questions)
- 1 minute (or 30 seconds depends on your class) for Navigator to Battle (ask the questions)

Have the class collectively take a breath, refocus and cover up the original problem

- 2 minutes to complete both Examples
- 1 minute (or 30 seconds depends on your class) for Driver to Battle (ask the questions)
- 1 minute (or 30 seconds depends on your class) for Navigator to Battle (ask the questions)

Have the class collectively take a breath, refocus and cover up the Contract and Purpose Statement.

- 2 minutes to Define the Function
- 1 minute (or 30 seconds depends on your class) for Driver to Battle (ask the questions)
- 1 minute (or 30 seconds depends on your class) for Navigator to Battle (ask the questions)

Important to note that students will have plenty of mistakes and they need to make revisions as they go through the battle. It is really helpful to battle 2 - 3 times on different problems and rotate partners on this day.

## Standards Alignment

### Common Core Math Standards

- ▶ **BF** - Building Functions
- ▶ **CED** - Creating Equations
- ▶ **EE** - Expressions And Equations
- ▶ **F** - Functions
- ▶ **IF** - Interpreting Functions
- ▶ **LE** - Linear, Quadratic, And Exponential Models★
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 9: Solving Word Problems with the Design Recipe

### Overview

In this stage students practice using the Design Recipe to write functions which solve for word problems. Towards the end of the lesson students should be ready to begin using the Design Recipe on problems from your own math curriculum.

### Agenda

#### Getting Started

##### Introduction

#### Activity (45 minutes)

##### De-Bugging Online Puzzles

#### Textbook Problem Extension

### Anchor Standard

#### Common Core Math Standards

- **F.BF.1** - Write a function that describes a relationship between two quantities.

### Objectives

#### Students will be able to:

- Design functions to solve word problems.
- Continue to practice writing contracts with more complex scenarios.

### Preparation

Print Blank Copies of the **Design Recipe Form - Worksheet**.

Review the **Design Recipe - Word Problem Template** for extensions

### Links

#### For the Teacher

- **Design Recipe - Word Problem Template** ([https://docs.google.com/document/d/1\\_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/](https://docs.google.com/document/d/1_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/)) (PDF) ([https://docs.google.com/document/d/1\\_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/export?format=pdf](https://docs.google.com/document/d/1_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/export?format=pdf)) | **DOCX** ([https://docs.google.com/document/d/1\\_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/export?format=doc](https://docs.google.com/document/d/1_p-UN-iXMMsTx8nRx58hvpX7xSBsXfGmo6aKMq6yFJM/export?format=doc))

#### For the Students

- **Design Recipe Form** ([https://code.org/curriculum/docs/algebra/worksheets/design\\_recipe.pdf](https://code.org/curriculum/docs/algebra/worksheets/design_recipe.pdf)) - Worksheet



# Teaching Guide

## Getting Started

### Introduction

With these incomplete or “bugged” Design Recipes, students will need to become expert analyst to find the errors. It is crucial that students are instructed to deliberately de-bug these levels using the Design Recipe.

It should be noted that the examples must be filled in completely. The error message when the example is incomplete is “You have a block with an unfilled input.”

#### 💡 Teaching Tip

##### **Beware:**

Students will want to just drag and drop to fill in the blanks, often rushing towards a solution without thinking it through.

Challenge students battle their partners and produce a completed version of each broken recipe on the paper form.

### Activity (45 minutes)

#### De-Bugging Online Puzzles

In this stage you'll use the Design Recipe to analyze and de-bug broken functions. Head to **CS in Algebra stage 9** in Code Studio to get started programming.

Before you let students dive into this lesson, the teacher needs to frame how students are to use their Design Recipe skills to de-bug. It is helpful to do the first one together on the projector, document camera, or board, then let students tackle Level 2 through Level 5 on their own.

With **blank** Design Recipe forms:

- Students will open Stage 9 Level 1 and fill as much of their Design Recipe form is provided. (They can fill out one form per pair)
- As students fill in their forms, post the question “What is Missing?” and “What would we need to make this work?”
- Once students think they have fixed what went wrong on paper, they should be encouraged to **battle** their partner to polish their work.
- After they have **battled** each other have them upload their work into Code Studio and what their functions work!

Make sure you **stop class at Stage 9, Level 5** to regroup. Have students share their process for de-bugging and highlight tricks or tips that students used to help them find all the errors.

Level 6 transitions from broken functions of word-problems to blank, unknown functions. This scaffold is helpful but if glossed over, students will be lost.

At this transition point re-energize students that although the problems look different, their process will work! Have students build their functions using their blank Design Recipe forms, again one per pair is good for this level.

#### Textbook Problem Extension

In this activity you would draft problems from your textbook or classroom materials so the Design Recipe could be used. Building functions is a major standard in Algebra 1 but there are limited tools to help students take in a description, table, points, or graph and create the function. Many times the “Create an Equation” is just assumed to be obvious but is painfully not for many students. By using this template and your own problems you will help students transfer their analytical thinking from CS to just Algebra!

On the free-play, Level 10, have students use the Design Recipe to build functions that would solve word problems from your textbook or course. You can create a worksheet with problems you want to create a function for or you can just have them try one.

The **Design Recipe - Word Problem Template** has 5 example problems that were modified from either teacher’s course materials or a textbook to give students access to building their own functions.

### Standards Alignment

#### Common Core Math Standards

- ▶ **BF** - Building Functions
- ▶ **CED** - Creating Equations
- ▶ **EE** - Expressions And Equations
- ▶ **IF** - Interpreting Functions
- ▶ **LE** - Linear, Quadratic, And Exponential Models★
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

## Lesson 10: Rocket Height

### Overview

Using the Design Recipe, students will work through a series of word problems about calculating the height of a rocket after a given number of seconds from launch. The functions they write will be used to animate the rocket launch.

### Agenda

#### Getting Started

##### Introduction

#### Extension Activities

##### Non-linear Animation

#### Activity

##### Online Puzzles

### Anchor Standard

#### Common Core Math Standards

- **F.LE.1** - Distinguish between situations that can be modeled with linear functions and with exponential functions.

### Objectives

#### Students will be able to:

- Design functions to solve word problems.
- Use the Design Recipe to write contracts, test cases, and function definitions.

### Links

#### For the Students

- **Rocket-Height Design Recipe**  
([https://code.org/curriculum/docs/algebra/worksheets/rocket\\_height.pdf](https://code.org/curriculum/docs/algebra/worksheets/rocket_height.pdf)) -  
Worksheet

# Teaching Guide

## Getting Started

### Introduction

Functions are a key part of animation in computer programs. A function that draws a static picture of a bat, for example, can place the bat at a different location based on the input. When that input changes slightly based on time or user-interaction, the bat will appear to move. This is similar to the way that flip-book animations work, in which each page draws a static image that has changed by a small amount. When the pages are displayed quickly, the images appear to change smoothly.



Putting these images together, we arrive at an animation of the bat turning around.



In the online puzzles, students will see a function box which has no parameter inputs, which represents the function as a Type of data. This allows you to pass your function into the `start` function, where it can be used to control the rocket animation. By passing the `rocket-height` function into the engine we can call it repeatedly with new numbers as each second ticks by instead of running it with a specific value. The periodic execution creates the flip-book effect. As each second passes, the `rocket-height` function is executed again, the new location is calculated, and the rocket is re-drawn in its new location. This drawing and re-drawing in different locations gives the appearance of motion.

### Extension Activities

#### Non-linear Animation

The final puzzle of this stage is a Free Play puzzle that will allow you and your students to experiment with other variations on the rocket-height formula. One activity that students find particularly interesting (and often challenging) is to write functions that produce non-linear acceleration. If your students are familiar with quadratics then you can call this out as such, but even younger students who haven't yet seen quadratics can enjoy this extension challenge.

#### 💡 Lesson Tip

After creating simple linear movement, students will be asked to write functions to animate simple acceleration. Students will be given an input/output table from which to write their new function. You may want to work through these problems as a whole class, so that students can see how you might analyze an input/output table in order to understand the relationship between input and output values.

Place the following input/output tables on the board and see if students can come up with functions that will produce the appropriate animation.

#### Challenge 1

Input	Output
1	10
2	40
3	90
4	160

#### Challenge 2

Input	Output
1	15
2	45
3	95
4	165

Once students have figured out the provided Input Output tables, encourage them to come up with non-linear animation functions of their own.

## Activity

### Online Puzzles

In this stage you'll write functions that manipulate images to create animations. Head to **Course A stage 10** in Code Studio to get started programming.

### Standards Alignment

#### Common Core Math Standards

- ▶ **BF** - Building Functions
- ▶ **CED** - Creating Equations
- ▶ **EE** - Expressions And Equations
- ▶ **IF** - Interpreting Functions
- ▶ **LE** - Linear, Quadratic, And Exponential Models★
- ▶ **MP** - Math Practices
- ▶ **NS** - The Number System
- ▶ **OA** - Operations And Algebraic Thinking
- ▶ **Q** - Quantities



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.