Girls Code Club

Emma Montgomery
Lindsay Klatzin
Sarah Karpel
&& our wonderful volunteers!

DEBUGGING

When you hear the word “debugging”, what do you think that means?

If you break down the word, you get DE-BUG-GING.

DE - “DE” means to undo something.

ING - The suffix “ING” is added to words to show an action that is currently in progress.

BUG - The word “BUG” here really does stand for bug!

Back in the day, computers were really big machines that were programmed using really big rolls of paper that had holes punched into specific spots. The machine knew how to read it, kind of like bumps in Braille! A bug could happen, much like it does today, due to a programmer just punching in the wrong holes. Or, a bug could happen because there was literally a bug (insect) in the big machine messing with the paper and pieces of machinery.

In Girls Code Club, you’re asked to write code. Sometimes, the code won’t work the first time you write it - so myself and the other coaches read through your code and try to find the mistakes. What we are doing is called DEBUGGING!

RUBBER DUCKY DEBUGGING

What's the Rubber Duck method?

It's when you talk to a rubber duck about the problem that you're having with your code. Talking about a problem out loud helps you think about a problem in a different way. Sometimes, a bug in the code will jump right out and be more obvious. It's also helpful because the rubber ducky cannot judge you, or make you feel like you're asking the wrong question - because NO question is wrong, but also, it's a rubber duck!

When you're talking to your rubber duck, do the following steps:

1. Mention the code that isn't working, what it's doing, and what is SHOULD be doing.
2. Explain your code - line by line. What is it doing?
3. Spell words out loud, your duck wants to learn too!

What's really going on here?

You are audibly talking about your problem, which helps your brain process it in a different way.

ACTIVITY: UNPLUGGED DEBUGGING

Use your rubber ducky!

- Put your duck on the starting spot.
- Follow the suggested steps.
- Did you make it to the pond? No? Figure out why!
    - Hint: There will only be **one step** that needs to be fixed.
- Write down the correct steps to get to the pond.

Suggested Path to the Pond

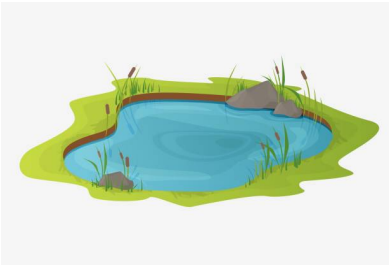
- Move down 1 space
- Move right 3 spaces

Which step is wrong? _____

What are the correct steps?

#1 _____

#2 _____

 **START**		
		

Suggested Path to the Pond

- Move right 1 space
- Move down 1 space
- Move left 1 space

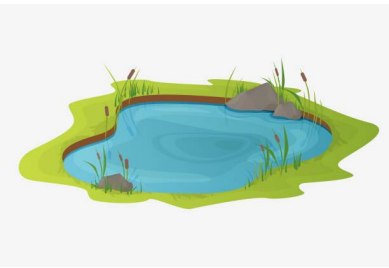
Which step is wrong? _____

What are the correct steps?

#1 _____

#2 _____

#3 _____

	 **START**	
		

Suggested Path to the Pond

- Move down 1 space
- Move right 1 space
- Move up 1 space
- Move left 2 spaces

Which step is wrong? _____

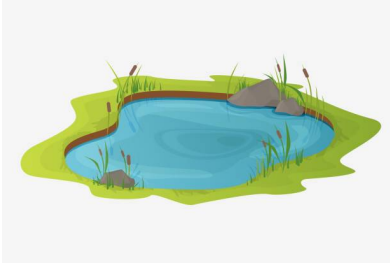
What are the correct steps?

#1 _____

#2 _____

#3 _____

#4 _____

		
	 **START**	

SHERO #1: GRACE HOPPER

Video - https://www.youtube.com/watch?v=Fg82iV-L8ZY

What did you learn about Grace Hopper? Was there anything you really liked about this SHEro of the month?

CODEPEN NOTICE: PARENTS

In Girls Code Club, one of the applications we'll be using is called CodePen. This is a free platform that allows coders to practice and experiment with HTML/CSS/JavaScript. We will be using this for all Cohorts because there's even a Python editor pinned to our account!

This platform is completely safe, and your coder will only ever be on it while logged into the Girls Code Club of Lancaster (private) account. This means any content they make WILL NOT be public for the world to see. As an added precaution, each girl will name their projects after the username they had created during Session 1. Coders are welcome to log into our CodePen account in their free time to continue working on projects or practicing.

Codepen => https://codepen.io/

Username: **girlscodeclublancaster**

Password: **Girl$Rock!**

BEGINNER: INTRO TO JAVASCRIPT

JavaScript is a programming language! We started to use it last week when we learned about variables, and will be using it for the rest of our Girls Code Club sessions.

Fun Fact: JavaScript is the most used programming language in the world. It's popular!

This language is used for creating dynamic websites, and sometimes phone apps.
What does dynamic mean?

dy-nam-ic
adjective

1. (of a process or system) characterized by constant change, activity, or progress.
2. (of a person) positive in attitude and full of energy or new ideas.

Can you think of something in your life that's dynamic?

JavaScript is **dynamic** because it provides movement or change to a website. A static website has the same information and colors all the time, no matter what. When a website is dynamic, you can:

- Click a button to submit a form
- Check that what the user typed into the form is correct
- Flip through pictures, so that a new one shows every couple of seconds
- Show data
    - Example: Amazon - you're shopping for a new cat bowtie. Amazon is able to dynamically show you:
        - What colors the store has available
        - How many bowties they still have
        - Whether or not a bowtie is sold out

ADVANCED: INTRO TO PYTHON

Python, like JavaScript, can be used to build websites and software. It's more frequently used for automating tasks, math, or analyzing data.

What does it mean to automate something?

au-to-mate
verb
convert (a process or facility) largely automatic operation

au-to-mat-ic
adjective

1. (of a device or process) working by itself with little or no direct human control
2. Done or occurring spontaneously, without conscious thought or intention

Can you think of any examples in your life of something that's automatic?

Python is considered **general-purpose**, which means it can be used for a variety of different types of programs - it's not specialized for anything.

What else makes Python cool?

- It's simple - you don't need to type as much as you need to with some other languages.
    - Fewer keys clicked, fewer letters, and fewer lines! Less typing!
- It works on different operating systems (i.e. Windows, Mac, Linux, etc.)
- Python looks a lot like English, so it's easier to understand.
- The code can be run as soon as it's written - it doesn't need to take time for the computer to understand and process code, like with JavaScript.

Types of Variables (all Cohorts)

Some of this will be review from last session!

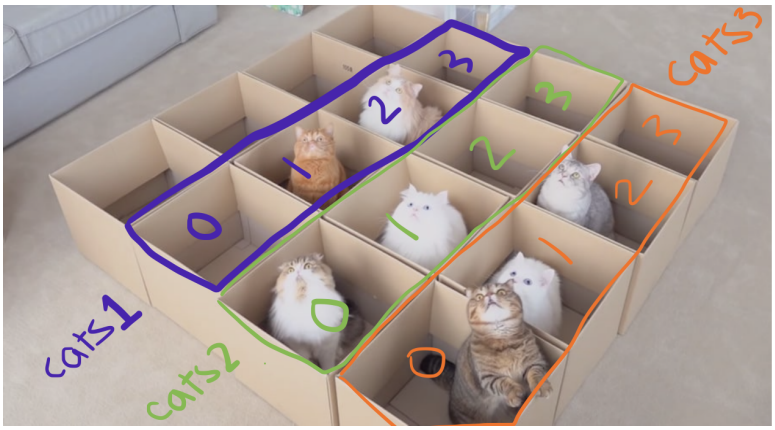
Data Types. Represent different types of data - strings, numbers, boolean, etc. A string is wrapped in quote marks (“ “) so that the computer can understand it’s reading text or regular English words. Numbers look like regular numbers, and boolean data types are true or false, which is the same as saying YES or NO. Examples:

- What data type is **“coders”**? string
- What data type is **101**? number
- What data type is **“true”**? string
- What data type is **false**? boolean
- What data type is **“208”**? String

Variables always have a data type. If you remember from last session, a variable can be any data type, but once it has a value, it cannot be changed! Variables are like putting something in a box, and labeling/naming the box with what’s inside.

Remember Example: `var cats = 3` or (`cats = 3` in Python)

But, what if I want to represent a row of boxes, that are also full of cats? I would use an Array!



Array. A single variable that stores a series of elements. Each element in your array has to be the same data type.

To access things in an array, you use what is called the array’s **index**. The index is an item’s order or position in the array, and it will always start at 0 (instead of 1).

As an example, let's think of a grocery list. My list looks like this:

- Apple
- Mango
- Banana
- Grape
- Blueberry
- Kiwi
- Papaya

Because "apple" is first on my grocery list, it's position would be **0**.

Accessing an item at an array's index looks a lot like this: **fruits[2]**

`fruits[2]` - the word "fruits" is the name of our array variable
`fruits[2]` - the opening and closing brackets tell our code that we want the **index**
`fruits[2]` - 2 is the actual index

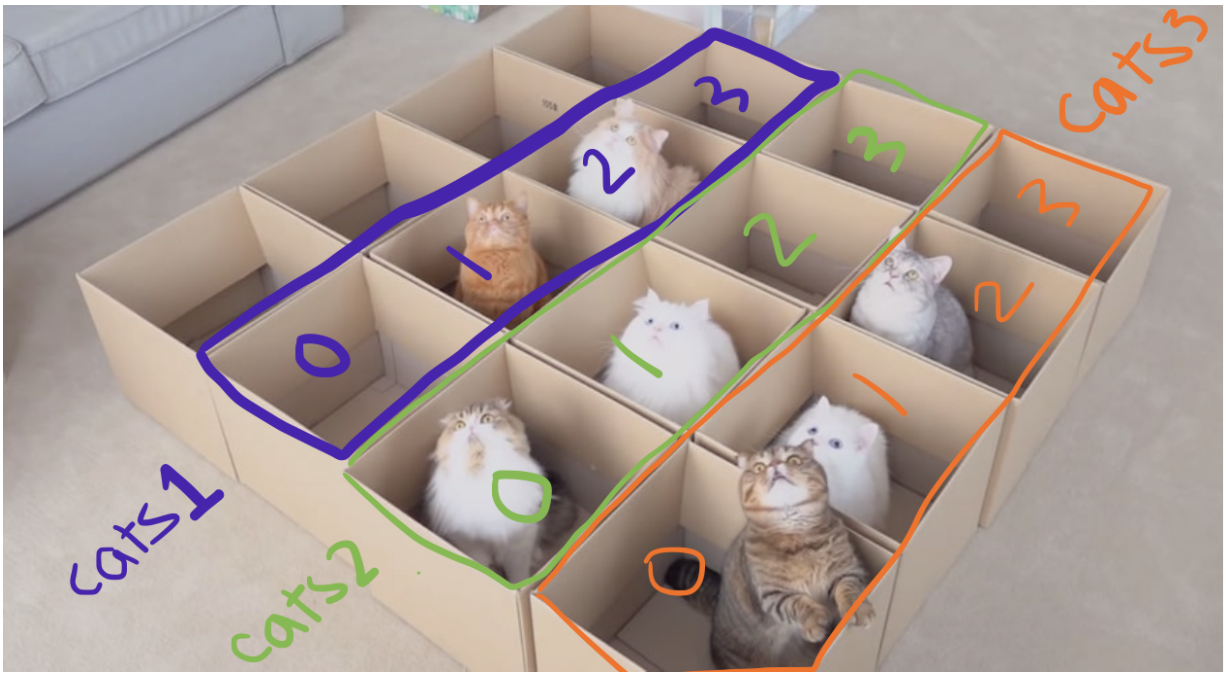
Given this example, what is **`fruits[2]`?**
What index holds the blueberries?

Exercise:

I need 3 volunteers!

Come up to the dry erase board, and show me how I would get the apple. Talk amongst yourselves to make sure you all understand what's going on, and then explain your answer to the class.

Another example!



This is a group of arrays. I've color coded them to make it easier to understand what's going on here!

The purple array is called **`cats1`**.
The green array is called **`cats2`**.
The orange array is called **`cats3`**.

In order to pick up a cat and snuggle it, I have to select the correct array and index.

What do I pick up, if I select **`cats2[2]`**?
What do I pick up, if I select **`cats3[0]`**?

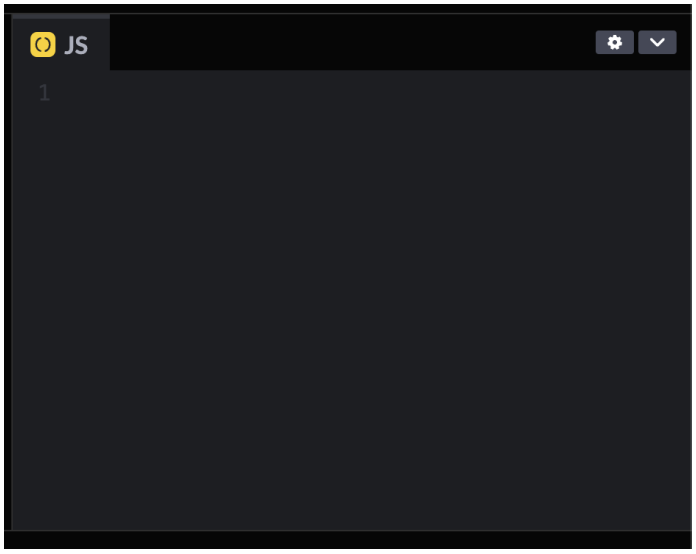
Exercise: Again, I will need 3 volunteers.
Go up to the dry erase board and tell me how to pick up the **orange cat**. Make sure you each understand the answer you've chosen, and then explain it to the class.

Exercise #2: (different group) tell me how to pick up the **gray and white** cat.
Exercise #3: (different group) tell me how to pick up a **cat with long hair**.

Let's Practice! CodePen - JavaScript.

To practice **JavaScript**:

- Log into the Girls Code Club account
- Under "Create" on the left, click "Pen"
- Name the pen after the username created in Session 1
- Code in the JavaScript window
    - On the rigjt, "JS", yellow icon



Remember that **`console.log()`** let's you print things out in the console at the bottom of the page. If you need help accessing this console, please ask one of us for help!

Exercise #1

Write the following code:

Use **`console.log()`** to print out the colors, in the order they would appear in a rainbow (ROYGBPP). For example, if you typed **`console.log(colors[0])`**, what would you get?

Exercise #2

Think about everything that goes into your favorite sandwich. What kind of bread do you like? How many slices are there? Are there tomatoes? Your sandwich must have at least 5 ingredients.

Create an array that has all the ingredients for your favorite sandwich. Example:

```
JS
1  var sandwich = ["bread", "peanut butter", "jelly", "more bread"];
```

Use the console to print instructions telling me how to make this sandwhich. Here's another example!

In this example, I'm logging my sentence as a string. When I want to include my variable in the sentence, I end the string with a quote mark ". I then use + signs to add my variable to the sentence. Kind of like math, but with letters!

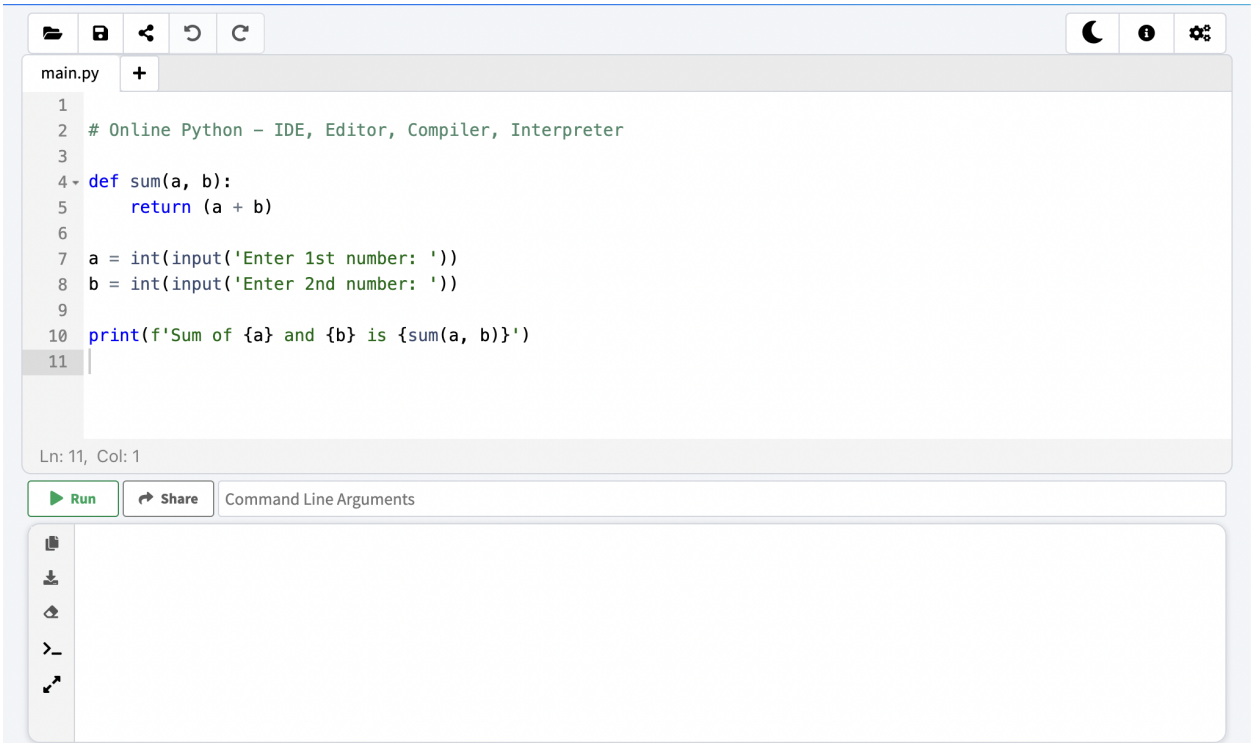
Once most coders are done, we'll have some time go around the room and read our sentences.

Let's Practice! CodePen - Python.

For access to a **Python** code editor:

- Log into the Girls Code Club account.
- Click "Pinned Items" on the left.
- Click "Online Python..."
- Type in the "main.py" editor.
- Click "Run"
- See your changes at the bottom!

This work will not be saved anywhere - it's just for the coders to practice with Python during the introductory course.



Exercise #1

Write the following code:

```
main.py +

1
2  # Online Python - IDE, Editor, Compiler, Interpreter
3
4  colors = ["red", "pink", "green", "yellow", "purple", "orange", "blue"];
5
```

Use **`print()`** to print out the colors, in the order they would appear in a rainbow (ROYGBPP). For example, if you typed **`print(colors[0])`**, what would you get?

Exercise #2

Think about everything that goes into your favorite sandwich. What kind of bread do you like? How many slices are there? Are there tomatoes? Your sandwich must have at least 5 ingredients.

Create an array that has all the ingredients for your favorite sandwich. Example:

```
main.py +

1
2  # Online Python - IDE, Editor, Compiler, Interpreter
3
4  sandwich = ["bread", "peanut butter", "jelly", "more bread"];
5
```

Use the console to print instructions telling me how to make this sandwhich. Here's another example!

In this example, I'm printing my sentence as a string. Python lets you print with strings and variables. We know all about **`print().`**

To print something that includes text and variables, we add an **f** inside the parenthesis, like so: `print(f).`

We then add our text inside our parenthesis, but after the **f**. Our text needs to start and end with a single quote **'**. To get to the single quote, hold your shift key and press the button that has the double quote on it **"**. Our example now looks like this:
`print(f'Before I make a sandwich, I need.')`

Now, let's include our variable!
`print(f'Before I make a sandwich, I need {sandwich[0]}.')`

SHERO #2: JOAN CLARKE

She was born in 1917 in London, England, and was the youngest of 5 children. She attended college in England for math, but was not allowed to actually earn her degree due to being a girl. Her skills did not go unnoticed, because she was soon recruited for the Government Code and Cypher School.

Joan Clarke is best known for her work as a **code breaker** during World War 2.

This school existed for one single purpose - breaking German Enigma code during the war. Germans were using a machine called the Enigma to encode their messages. Encoding a message means to hide a message behind a language or code that most others don't understand.

She was originally placed when an all women's group called "The Girls". Cryptography at the time was not considered to be a job women would be good at, so this group did the clerical work - paperwork, phone calls, etc.

Despite challenges she was facing due to being a woman, Joan quickly became an actual cryptographer and lead her group to massive victory. Groups called "wolf packs" were sinking 282.000 tons of shipping per month during war, but Joan and her team's efforts saved 220,000 tons a month! She was able to decode Nazi messages and figure out where "wolf packs" were planning their next attack.

What is something you liked or learned about Joan Clarke?

BUILD A GOOGLE LOGO

Editor - https://scratch.mit.edu/projects/177224273/#editor

Google logos are called **doodles**! These doodles are what you see instead of "Google" when you're doing something like a Google Search. Google uses scratch to create these logos. Each letter in the doodle is called a **sprite**.

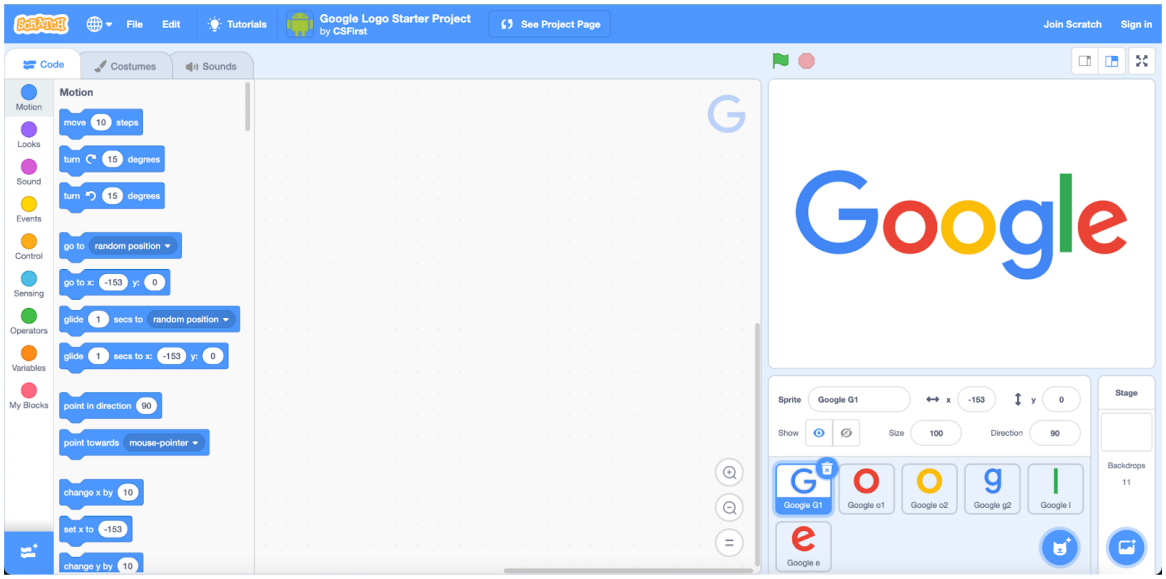
In the editor, you will be dragging and dropping scratch code from the left, to tell the program what you want each letter to do. You can also pick different costumes for each letter!

Username: **GirlC0dersRule**

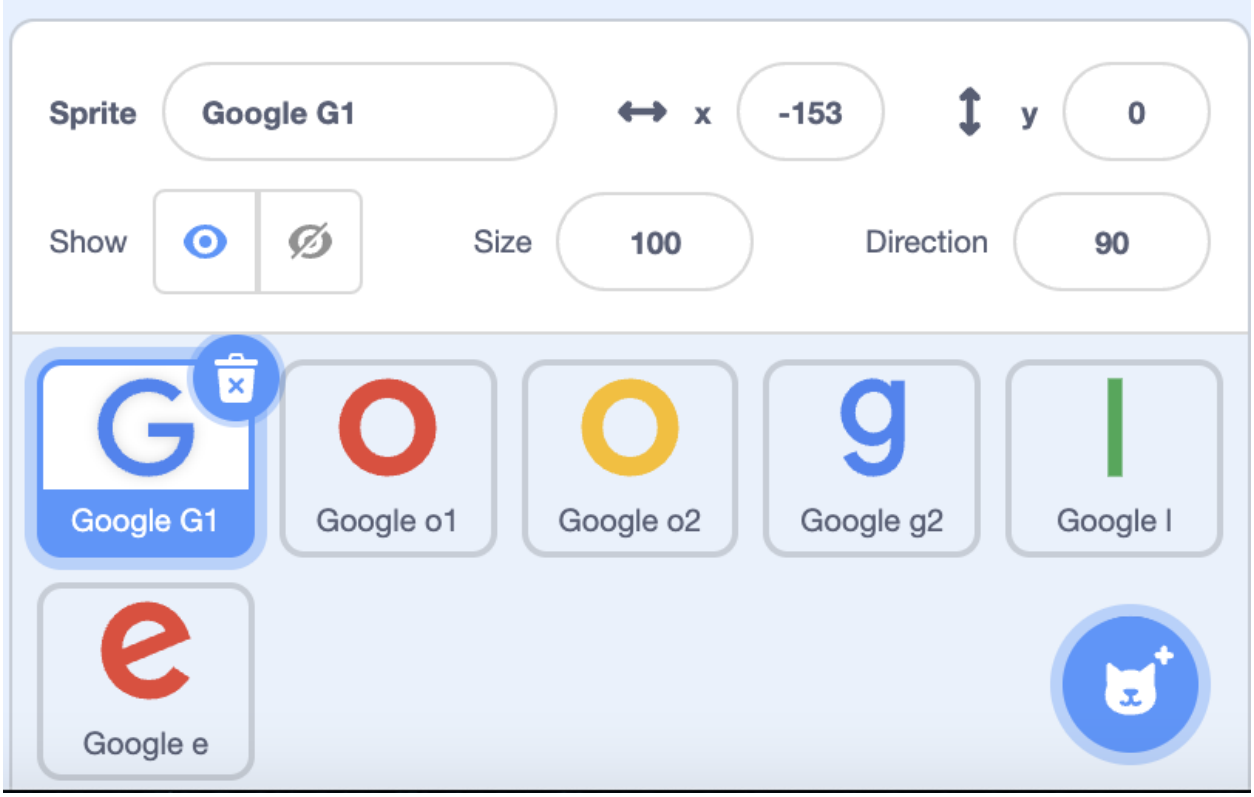
Note: The 0 is a ZERO.

Password: **LSF012208**

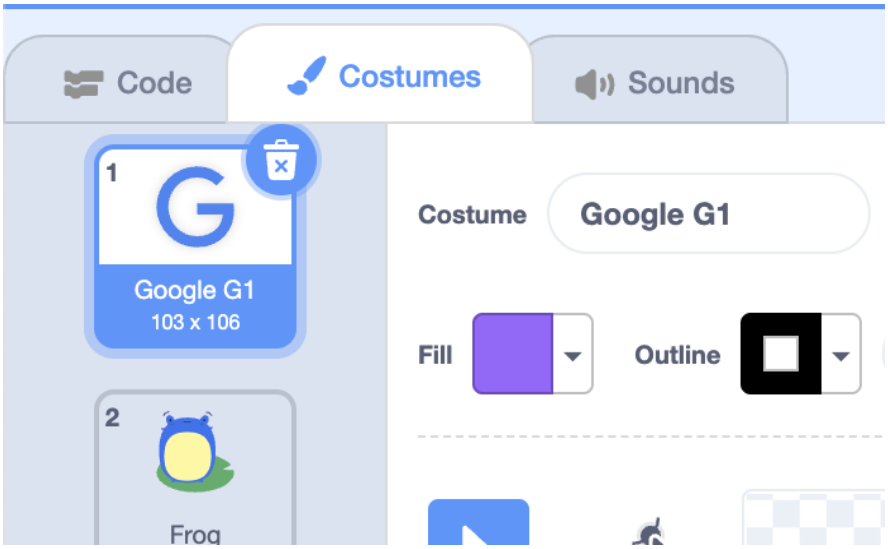
The editor! The code that you'll be working with is in the pane on the left.



In order to switch which letter you're working with, click on a different letter/sprite in the bottom right hand window. Be careful not to delete your sprite!



To get to each sprite's costume options, click the tab here (it's next to your code tab at the top):



Exercise #1: Change colors on click.
Video tutorial =>
https://csfirst.withgoogle.com/c/cs-first/en/create-your-own-google-logo/create-your-own-google-logo/extensions/change-color.html

Exercise #2: Make your own custom logo! Do whatever you want!