Web Development

Day 6: Programming

Pair Programming & Intro to JavaScript

Maydm

Welcome Back!

Maydm

Improv!

Maydm

Today's Schedule

Morning:

- Fly Gal
- Pair Programming
- Scratch projects

Afternoon:

- Introduction to JavaScript
- JavaScript Syntax
- The Console
- Variables
- Data Types
- Template literals

Maydm

What Is Pair Programming?

Maydm

Why Pair Programming?

What are some of the benefits of having two programmers working together on the same problem?

- Brainstorming with another programmer leads to better, more efficient code
- Find solutions to problems faster
- Talking about code leads to greater understanding
- Catch and fix bugs in the code sooner

Maydm

The Role of the Driver

The Driver:

- Is the one writing the code
- Is concerned with the small details
- Should explain what they are doing
- Should ask for help if they get lost
- Will switch roles with The Navigator after each level

Maydm

The Role of the Navigator

The Navigator:

- Is concerned with the big picture
- Should offer suggestions on what to do next
- Should watch for bugs in the code
- Should ask if they aren't sure where the Driver is going

Dos and Don'ts

- ✗ **Don't** be rude or insulting when talking to your partner.
- ✓ **Do** be respectful!
- ✗ **Don't** grab the keyboard when you are the Navigator.
- ✓ **Do** switch roles regularly.
- ✗ **Don't** just sit there!
- ✓ **Do** engage with your partner about what you're trying to accomplish.

Maydm

Pair Programming Guidelines

- Ask before taking the keyboard.
- Be respectful when communicating with your partner.
- Talk to each other about the problem you're solving.
- Explain what you're trying to do if you're the Driver.
- Think ahead and offer suggestions if you're the Navigator.
- Switch roles whenever instructed to do so.

Introduction to JavaScript

JavaScript is the programming language for the web. It's used by an estimated 95% of websites.

JavaScript is used to:

- Add interactivity to web pages
- Create web & mobile apps
- Build web servers and backend infrastructure

Maydm

JavaScript Syntax

JavaScript is made up of expressions and statements.

Expressions are bits of code that can be reduced to a value.

Example:

```
var x = 1;
```

Statements are code that will be executed to perform a function.

Example:

```
document.getElementById('test').innerHTML = 'Hello world';
```

Maydm

JavaScript Syntax

In JavaScript, you must **declare** a variable using the keyword **var** before you can do anything with it, like this:

```
var x = 1;

var y = 2;

x + y;
```

JavaScript Syntax

In JavaScript, expressions and statements end with a semicolon (;). What are these expressions doing?

```
var x = 1;

var y = 2;

x + y;
```

Maydm

The Browser Console

Chrome has a tool for looking at JavaScript!

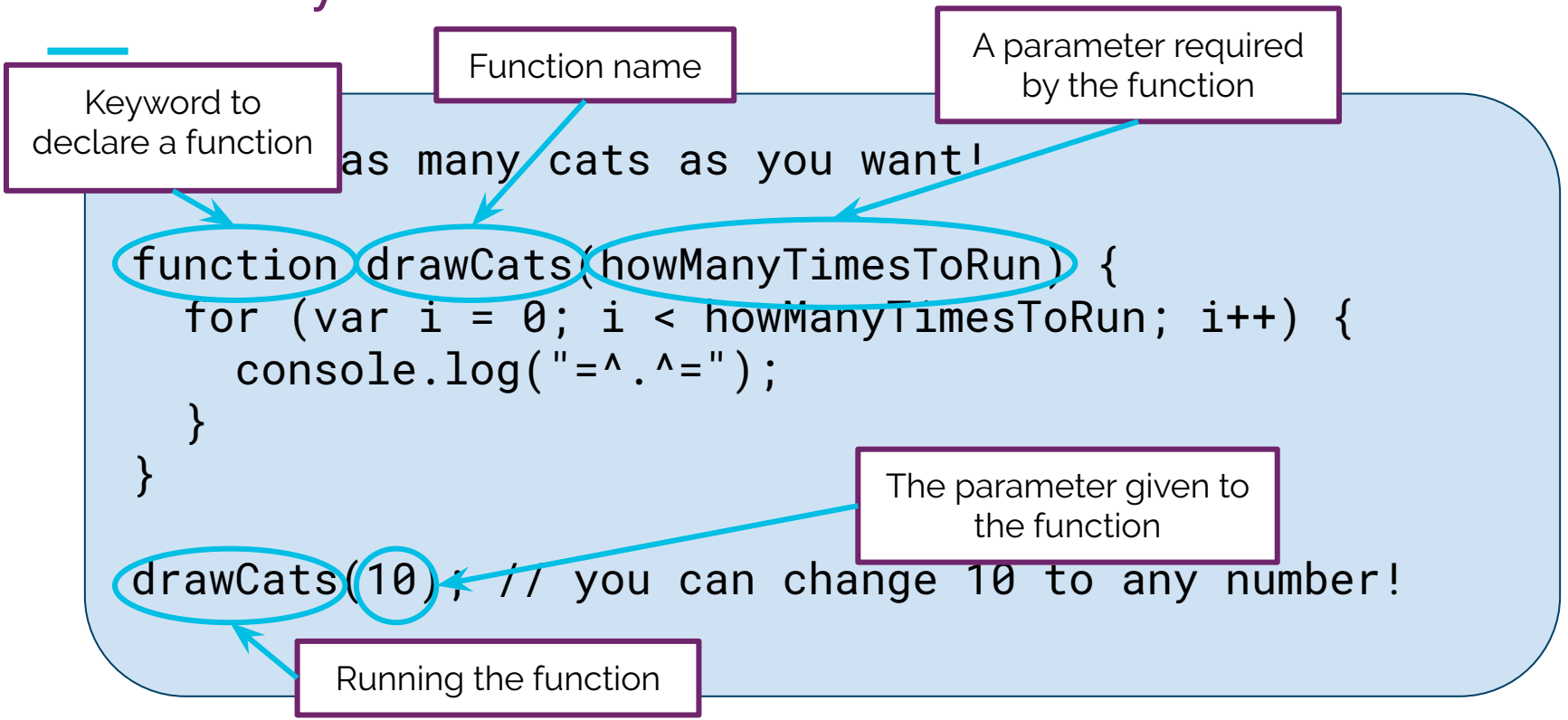
Open a new tab then open the Developer Tools using **CNTL + SHIFT + J** (Windows) or **ALT + CMD + J** (Mac). This will open the DevTools Console.

A **console** is a text-only computer interface. In JS, the console is useful for debugging code.

Let's try some code on the console! Go to Day 6 on GitHub and copy the function at the top.

Maydm

Anatomy of a JS Function



Maydm

The Browser Console

Let's try the previous code in the browser console. Type each line in the console and hit return after each.

```
var x = 1;

var y = 2;

x + y;
```

Maydm

Data Types

Variables can be any of the data types in JavaScript. There are many data types in JS but we will be focusing on three to start:

- Numbers
  - Whole numbers
  - Decimal point numbers
- Strings
  - Text wrapped in quotation marks
  - Quotation marks can be single or double, but must match
- Booleans
  - True or False

Maydm

What are strings?

A string is a collection of letters, numbers, or characters that are wrapped in quotation marks. A string can even be a single character!

Maydm

Commenting Code

Comments in code help explain what's going on. They are ignored by the computer. There are two kinds in JS: single line and multi-line.

```
// This is a single line comment.

/*
    This is a multi-line comment. It can go over
    multiple lines.
*/
```

Maydm

Practice with Variables

Work through the Day 6 “Variables” exercises on JS Bin.

Maydm

Booleans

Booleans are a logic-based datatype. They can be either true or false.

```
var pageLoaded = true;

var errorOnPage = false;
```

Operating on Variables

Just like in Algebra, you can operate on variables! Try these on your console.

```
var firstName = ‘Jane’;

var lastName = ‘Doe’;

firstName + lastName;
```

Maydm

Operating on Variables

Just like in Algebra, you can operate on variables! Try these on your console.

```
var x = 10;

var y = x + 20;

(x + y) * 5;
```

Maydm

Practice with Variables

Work through the Day 6 “Operating on Variables” exercises on JS Bin.

Maydm

Operating on Strings

```
var firstName = ‘Jane’;

var lastName = ‘Doe’;

console.log(firstName + lastName);
// returns ‘JaneDoe’
```

Maydm

Operating on Strings

```
var firstName = ‘Jane’;

var lastName = ‘Doe’;

console.log(firstName + ‘ ‘ + lastName);
// returns ‘Jane Doe’
```

Maydm

Operating on Strings

Imagine you want to create a sentence using a bunch of variables, like below. It can get confusing quickly! What are some problems with this?

```
console.log(firstName + ‘ ‘ + lastName + ‘ was born
on ‘ + birthday + ‘ in ‘ + city);

// returns ‘Jane Doe was born on 1/1/1970 in New York
City’
```

Maydm

Template Literals

Instead of adding strings together, template literals use placeholders for variables using **${}**. It makes strings easier to read.

```
firstName + ' ' + lastName + ' was born on ' +
birthday + ' in ' + city

// Same expression as template literal:

`${firstName} ${lastName} was born on ${birthday} in
${city}`
```

Maydm

Template Literals

Anything inside the **${}** placeholder will be processed by JavaScript. That means you can put operate on the variables like this:

```
var balance = 100;
var tax = 0.055;

`The amount owed is ${balance} plus tax of
${balance*tax}`
```

Maydm

Practice with Variables

Work through the Day 6 “Template Literals” exercises on JS Bin.

Maydm

Reflection

Write in your journal about how you feel or what you learned today.

Prompts:

- HTML & CSS not considered programming languages. What do you think about this?
- Do you have any concerns about learning JavaScript? What are they?
- If you have used block coding before, what do you think about the differences between that and using a coding language?
- What did you think about Pair Programming? How is it useful?

Maydm