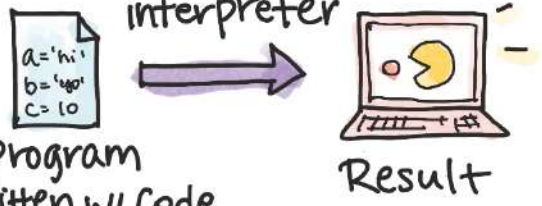
Introduction to Programming & Tools



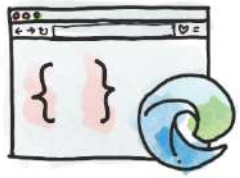
Tools

- Editors...
  - where you write code + debug. you may run the code too!

e.g. VS Code, Atom

- Browsers...
  - run your code on web
  - view visual elements
  - use DevTools to inspect & debug

e.g. Edge, Chrome, Firefox



- Command Line Tools...

Send commands (lines of text) to execute tasks

Less graphical option

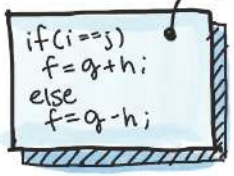
e.g. PowerShell, Terminal, Bash

- Docs...
  - where you learn

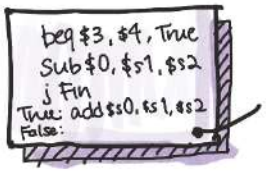
e.g.
- Mozilla Developer Network
- Front end masters

Languages

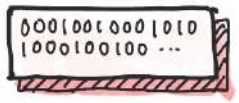
High-level programming language
↓
Low-level programming language
↓
machine language



JavaScript, Python, Ruby



ARM Assembly



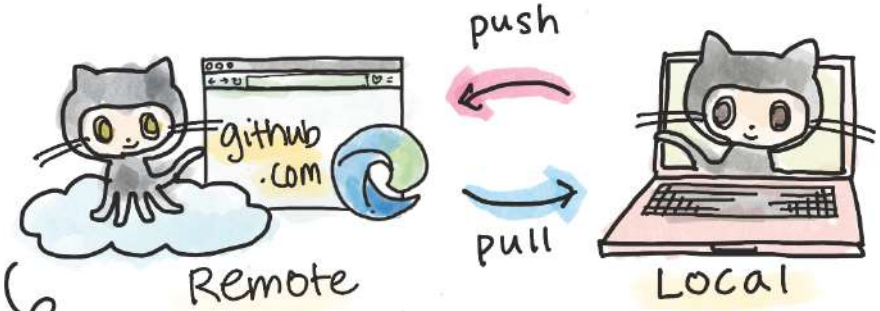
Introduction to GitHub

Git = a distributed version control system for tracking changes in your code history

GitHub = a cloud-based hosting where you can manage your Git repositories

Your first project with Git + GitHub

git-scm.com



0. Download + install Git

Then, set up your local Git profile

config

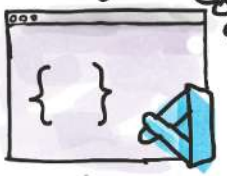
```
git config --global ...
```

1. On github.com, Sign up / Sign in + Create a repository.

Terminal

2. cd my project
3. git init
4. git status
5. git add .

your code

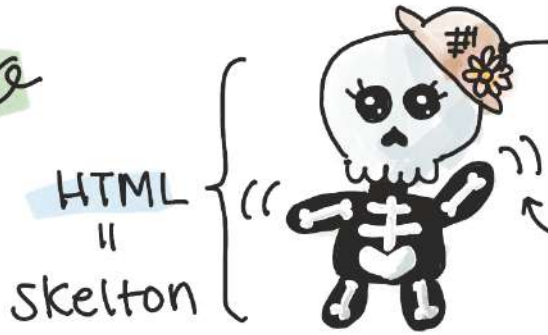


6. git commit -m "first commit"
7. git remote add origin http://github.com/—.git
8. git push -u origin main

9. To add more changes:
   - git add .
   - git commit -m "typo fix"
   - git push

@AzureAdvocates / @girlie_mac

Introduction to HTML



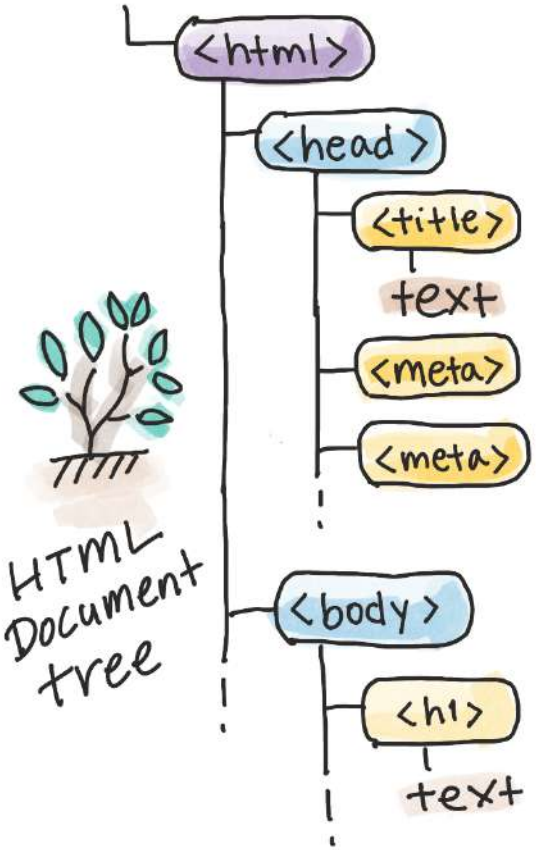
Semantic Markup

HTML tags = represent data

not defining how it look!

- screen reader
- SEO

Document



HTML Document tree

```
<!DOCTYPE html>
<html>
 <head>
  <title>Welcome!</title>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible"...>
  <meta name="viewport"....>
 </head>
 <body>
  <h1>my Terrarium</h1>
  <div id="page">
   <img alt="plant"
        src="img/p1.png">
  </div>
 </body>
</html>
```

@AzureAdvocates
@girlie_mac

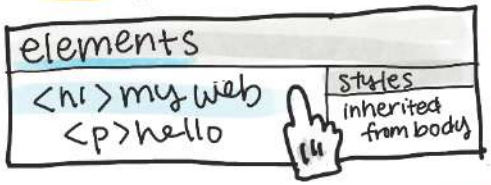
Introduction to CSS

makes web render nicely on a variety of screen sizes!

Inheritance

```
body {
  font-family: helvetica, arial, sans-serif; }
```

Inspect in browser dev tools



H1's font is inherited from the body!

Cascade

index.html

```
<link rel="stylesheet"
      href="style.css">
<h1 style="color: red">
```

<head>

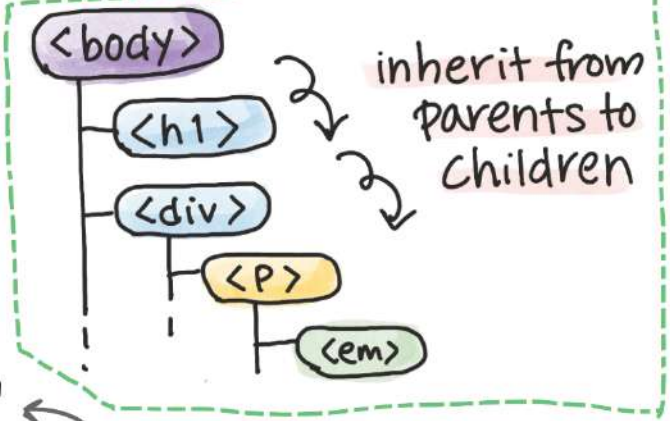
<body>

style.css

```
h1 {
  color: blue;
}
```

Selectors

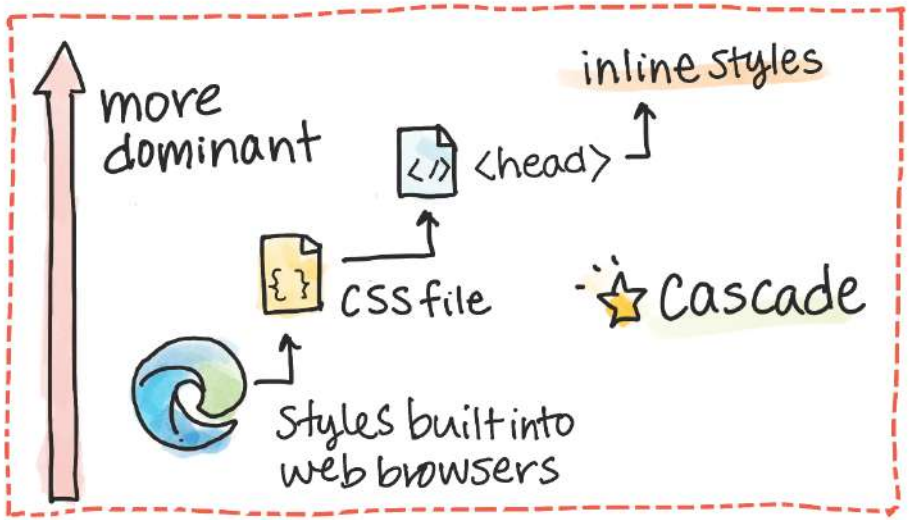
- tag
- id
- class




```
<div id="plant1"
     class="plant">
```

```
#plant1 { .... }
```

```
.plant {
  position: absolute;
  width: 150%;
  z-index: 2;
}
```



Layouts

- positioning
- display rules
- flows etc.

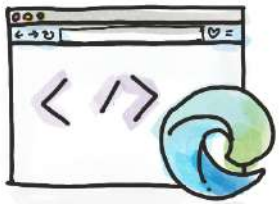
@AzureAdvocates / @girlie_mac

JavaScript DOM & Closure

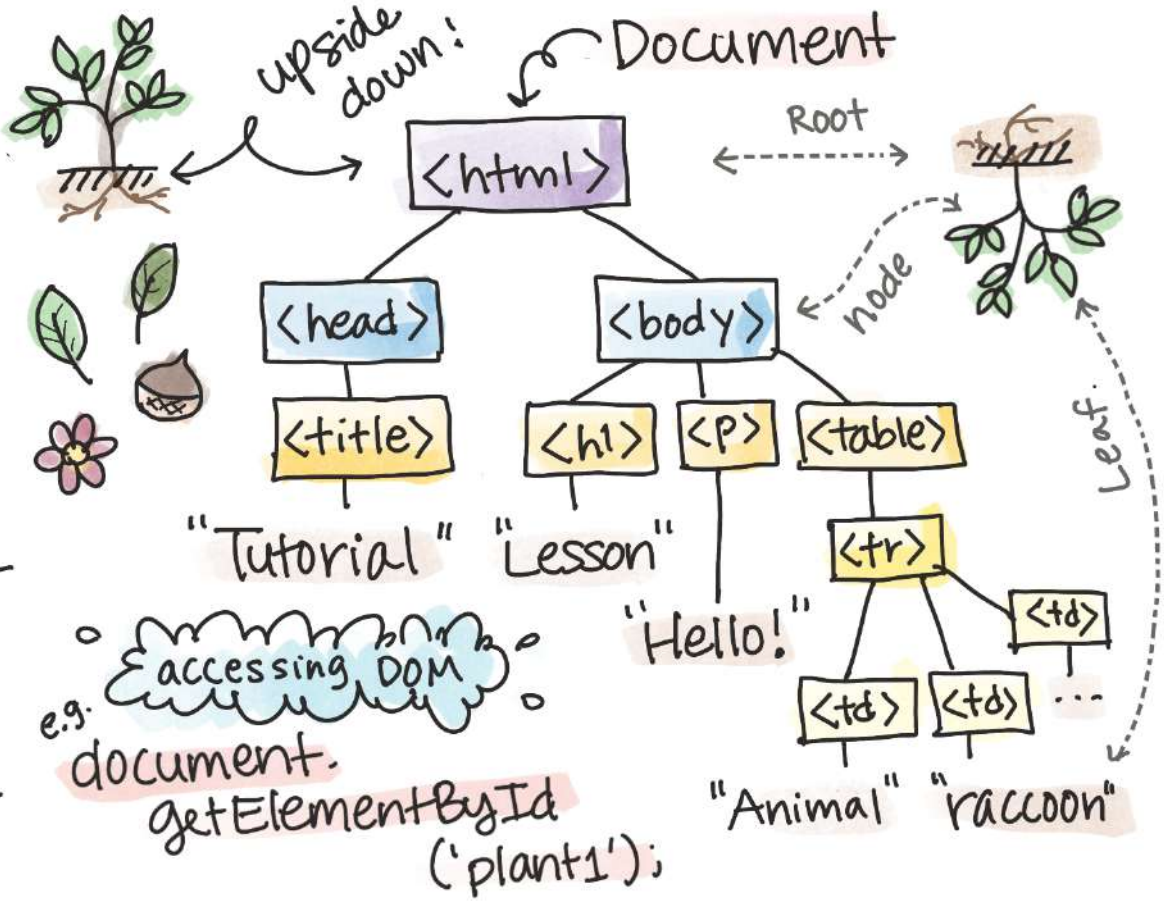
@AzureAdvocates
@girlie_mac

manipulation

Document Object Model



- programming interface for HTML
- data representation of the objects that make up the web structure + content



Closure

↳ an outer function that encloses an inner function

A closure gives you access to an outer function's scope from an inner function!

```
function dragElement(terrariumEl) {
  let pos1=0, pos2=0, pos3= 0;
  terrariumEl.onpointerdown = pointerDrag;
  function pointerDrag(e) {
    e.PreventDefault();
    pos3 = e.clientX;
    ...
  }
}
```

JavaScript Basics
Data Types

Variables

* 3 different keywords

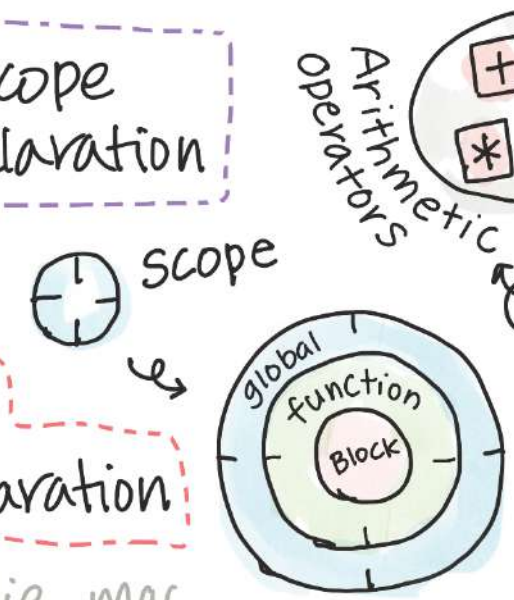
var
- Function scoped
- can be changed in scope
- Avail. before declared!

let
- Block scoped
- can be changed in scope
- Only avail after declaration

Const
- Block scoped
- cannot be changed
- Only avail after declaration

```
const greeting = "Hello";
```

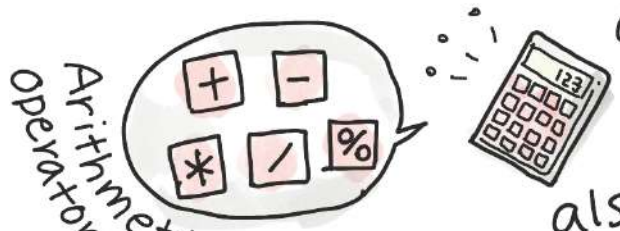
- `const` — Declaring a variable
- `greeting` — variable name
- `=` — assigned value
- `"Hello"` — keyword



Data types

♥ String
a set of characters that reside between single or double quotes.

♥ Number
`let donut = 32;`
can be:
integer, negative decimals, etc.
also: Infinity, BigInt



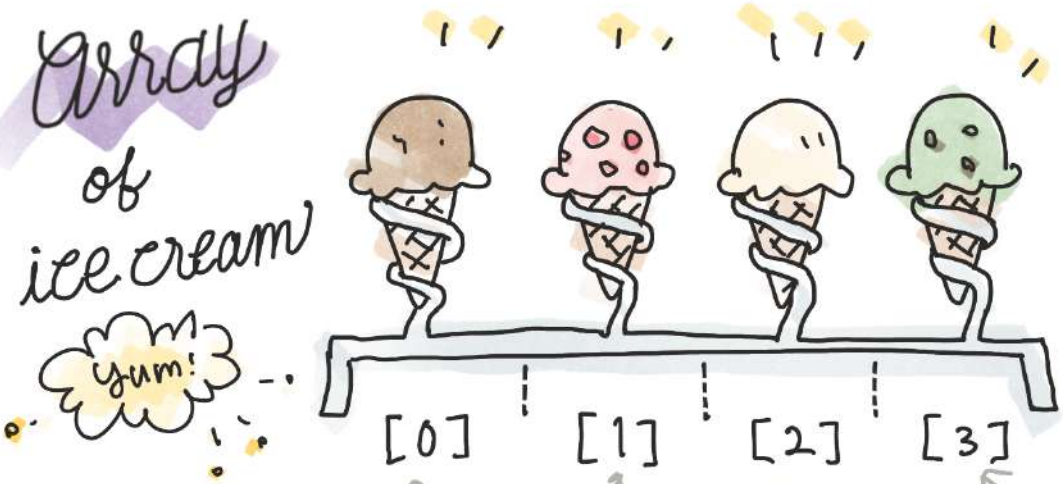
- `+` Addition
- `-` Subtraction
- `*` multiplication
- `/` division
- `%` Remainder

♥ Boolean

@AzureAdvocates / @girlie_mac

JavaScript Basics
Arrays [and Loops]

Array of ice cream




```
let flavors =
['chocolate','strawberry','Vanilla',
 'Pistachio'];

flavors[2]; // 'Vanilla'
flavors[4] = 'Rocky Road';
flavors[4]
    = 'Butter Pecan';
```

@girlie_mac
@AzureAdvocates

Loops

For Loop

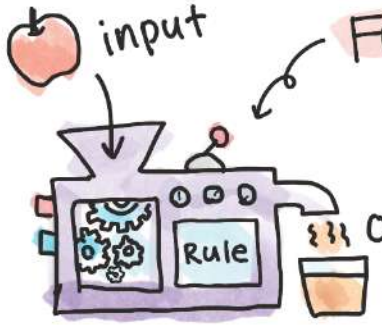
```
for (let i = 0; i < flavors.length; i++){
    console.log (flavors[i]);
}
```

↳ prints out each flavor after each iteration!

While-Loop

The loop will stop when the condition is met!

```
let i = 0;
while (i < flavors.length) {
  console.log (flavors[i]);
  i++;
}
```



Function: a building block of code that we can execute on demand.

- Take an input
- Return an output

Declaration — function name — parameter(s)

```
function square(n) {
  return n * n;
}
```

Return an output

JavaScript Basics
Functions

Passing info

```
function juice {
  (apple, orange)
}
```

When calling the function, you'll store the value in a variable

```
Const myNum
  = square(25);
```

Dafault values

```
function displayGreet (name, sal = 'Hello') {
  console.log (`${sal}, ${name}`);
}
```

Anonymous Function

```
setTimeout (3000, function() {
  console.log (···· );
}
```

Rewrite

Function as parameter

```
function displayDone( ) {
  Console.log ('3 sec. elapsed.');
}
SetTimeout (3000, displayDone)
```

Rewrite ES6

Fat Arrow Function

```
setTimeout (3000, ( ) => {
  console.log( ···· );
}
```

@AzureAdvocates / @girlie_mac

Lessons

	Project Name	Concepts Taught	Learning Objectives	Linked Lesson	Author
01	Getting Started	Introduction to Programming and Tools of the Trade	Learn the basic underpinnings behind most programming languages and about software that helps professional developers do their jobs	Intro to Programming Languages and Tools of the Trade	Jasmine
02	Getting Started	Basics of GitHub, includes working with a team	How to use GitHub in your project, how to collaborate with others on a code base	Intro to GitHub	Floor
03	Getting Started	Accessibility	Learn the basics of web accessibility	Accessibility Fundamentals	Christopher
04	JS Basics	JavaScript Data Types	The basics of JavaScript data types	Data Types	Jasmine
05	JS Basics	Functions and Methods	Learn about functions and methods to manage an application's logic flow	Functions and Methods	Jasmine and Christopher
06	JS Basics	Making Decisions with JS	Learn how to create conditions in your code using decision-making methods	Making Decisions	Jasmine
07	JS Basics	Arrays and Loops	Work with data using arrays and loops in JavaScript	Arrays and Loops	Jasmine
08	Terrarium	HTML in Practice	Build the HTML to create an online terrarium, focusing on building a layout	Introduction to HTML	Jen
09	Terrarium	CSS in Practice	Build the CSS to style the online terrarium, focusing on the basics of CSS including making the page responsive	Introduction to CSS	Jen

10	Terrarium	JavaScript Closures, DOM manipulation	Build the JavaScript to make the terrarium function as a drag/drop interface, focusing on closures and DOM manipulation	JavaScript Closures, DOM manipulation	Jen
11	Typing Game	Build a Typing Game	Learn how to use keyboard events to drive the logic of your JavaScript app	Event-Driven Programming	Christopher
12	Green Browser Extension	Working with Browsers	Learn how browsers work, their history, and how to scaffold the first elements of a browser extension	About Browsers	Jen
13	Green Browser Extension	Building a form, calling an API and storing variables in local storage	Build the JavaScript elements of your browser extension to call an API using variables stored in local storage	APIs, Forms, and Local Storage	Jen
14	Green Browser Extension	Background processes in the browser, web performance	Use the browser's background processes to manage the extension's icon; learn about web performance and some optimizations to make	Background Tasks and Performance	Jen
15	Space Game	More Advanced Game Development with JavaScript	Learn about Inheritance using both Classes and Composition and the Pub/Sub pattern, in preparation for building a game	Introduction to Advanced Game Development	Chris
16	Space Game	Drawing to canvas	Learn about the Canvas API, used to draw elements to a screen	Drawing to Canvas	Chris
17	Space Game	Moving elements around the screen	Discover how elements can gain motion using the cartesian coordinates and the Canvas API	Moving Elements Around	Chris
18	Space Game	Collision detection	Make elements collide and react to each other using keypresses and provide a cooldown function to ensure performance of the game	Collision Detection	Chris

19	Space Game	Keeping score	Perform math calculations based on the game's status and performance	Keeping Score	Chris
20	Space Game	Ending and restarting the game	Learn about ending and restarting the game, including cleaning up assets and resetting variable values	The Ending Condition	Chris
21	Banking App	HTML Templates and Routes in a Web App	Learn how to create the scaffold of a multipage website's architecture using routing and HTML templates	HTML Templates and Routes	Yohan
22	Banking App	Build a Login and Registration Form	Learn about building forms and handing validation routines	Forms	Yohan
23	Banking App	Methods of Fetching and Using Data	How data flows in and out of your app, how to fetch it, store it, and dispose of it	Data	Yohan
24	Banking App	Concepts of State Management	Learn how your app retains state and how to manage it programmatically	State Management	Yohan

Refer link for more