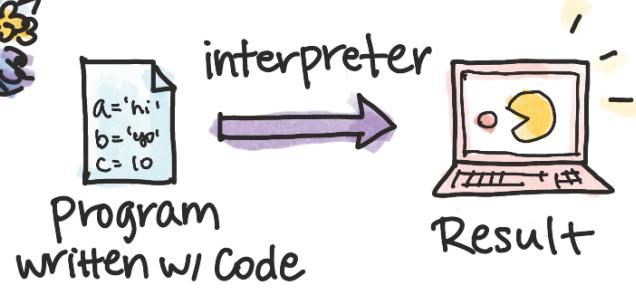


# Introduction to Programming & Tools

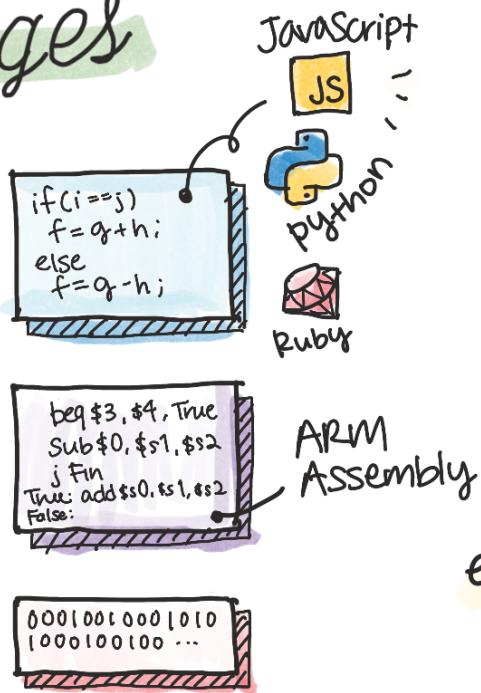


## Languages

High-level programming language

↓  
Low-level programming language

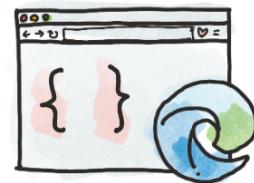
↓  
Machine language



e.g. VSCode Atom

### Editors...

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



### Browsers...

- run your code on web
- view visual elements
- use DevTools to inspect + debug

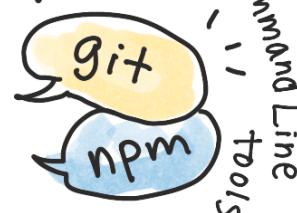
e.g. Edge, Chrome, Firefox

### Command Line Tools...



Send commands (lines of text) to execute tasks

Less graphical option  
e.g. PowerShell  
Terminal  
Bash



### DOCS...

- where you learn

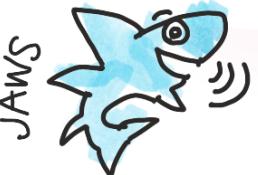
e.g.

- Mozilla Developer Network
- Frontend Masters

# Creating Accessible Webpages

## Tools

- Screen readers
- Contrast checkers
- Lighthouse in Devtools



WCAG color checker



## Designing



- Color Safe palette



- Semantic HTML

<em>... to emphasize

<span style="font-style: italic;">

- Visual clues



- Let screen reader read out hyperlinks
- "Visit Wikipedia.org for more info."
- "Click here for more info."



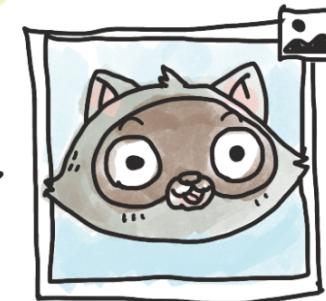
Accessible links = Better SEO

Accessible Rich Internet Applications  
a set of HTML attributes  
<div role="progressbar" aria-valuenow="75" ...>



## Images

- alt attribute

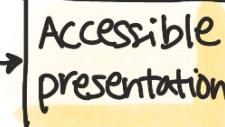
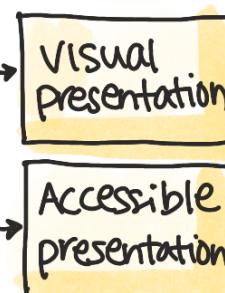
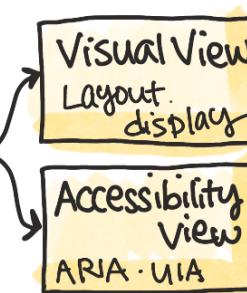
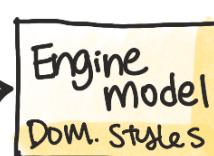


alt="Bit the Raccoon"

## Keyboard



- Keyboard navigation



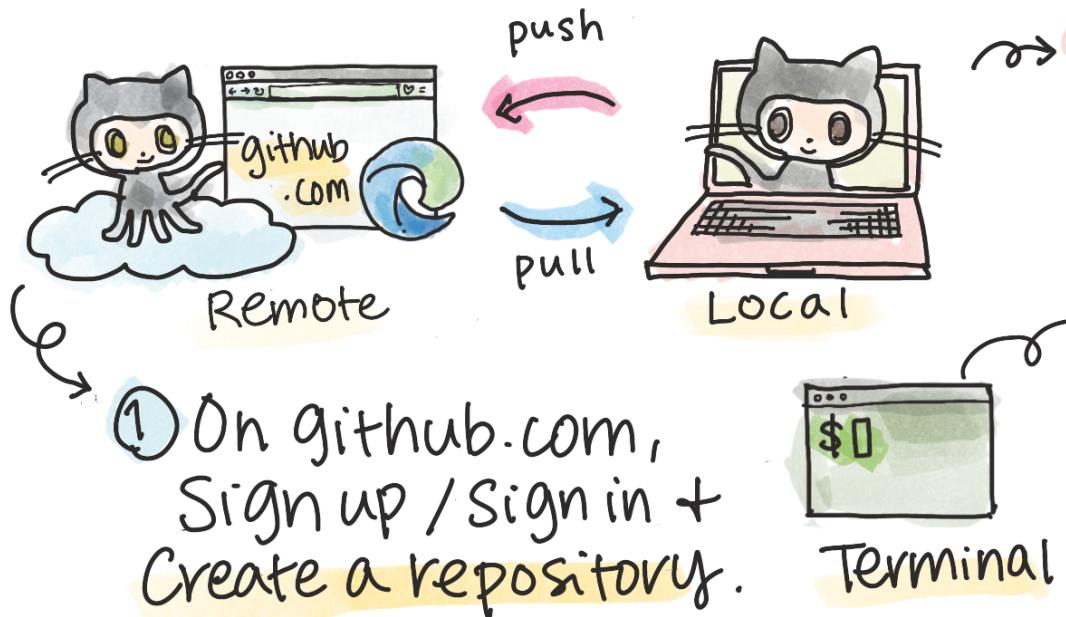
 Git = a distributed version control system for tracking changes in your code history

## Introduction to GitHub

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



### >Your first project with Git + GitHub



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



Terminal

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

→ ① Download + install Git  
Then, set up your local Git profile

git-scm.com

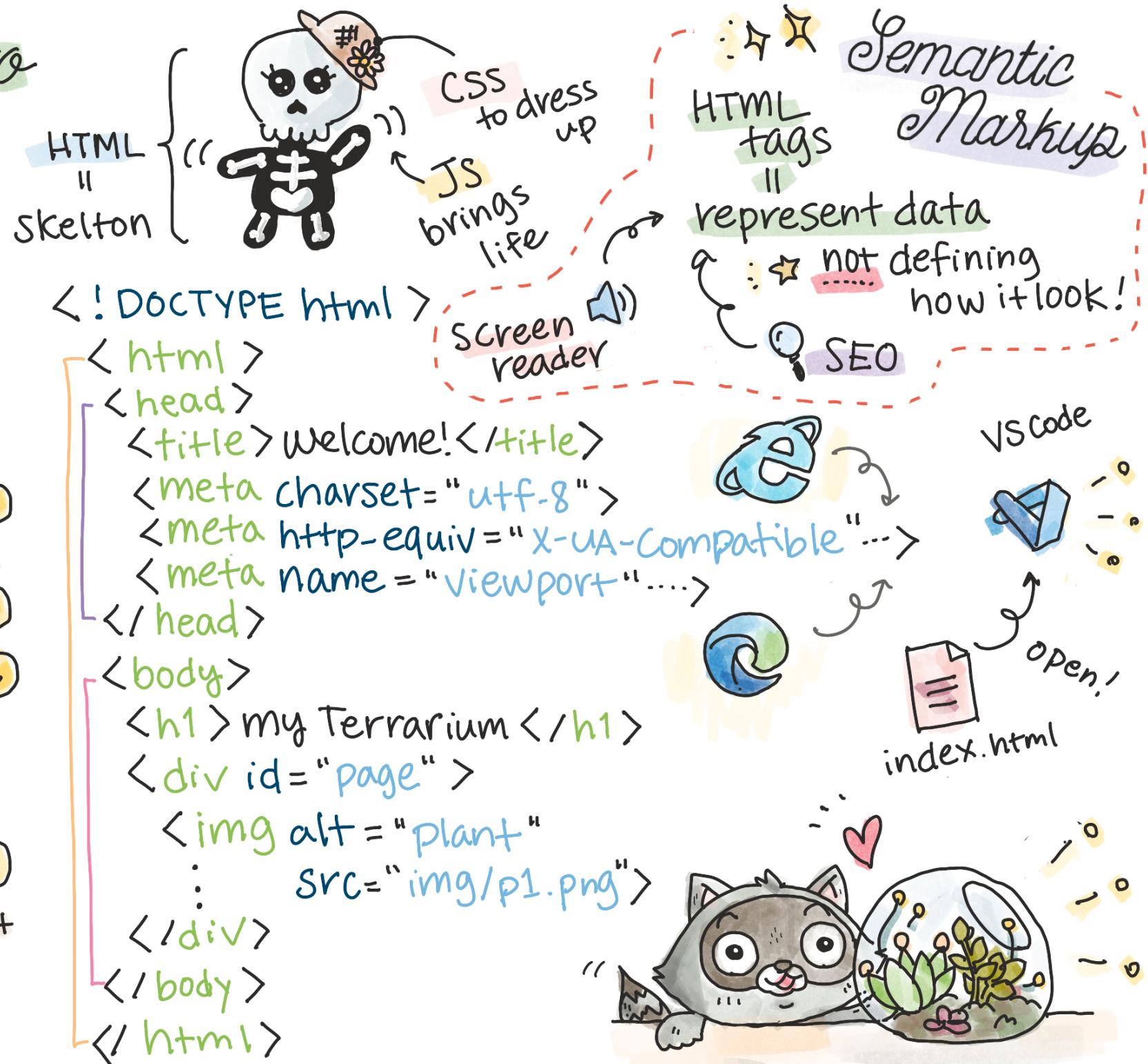
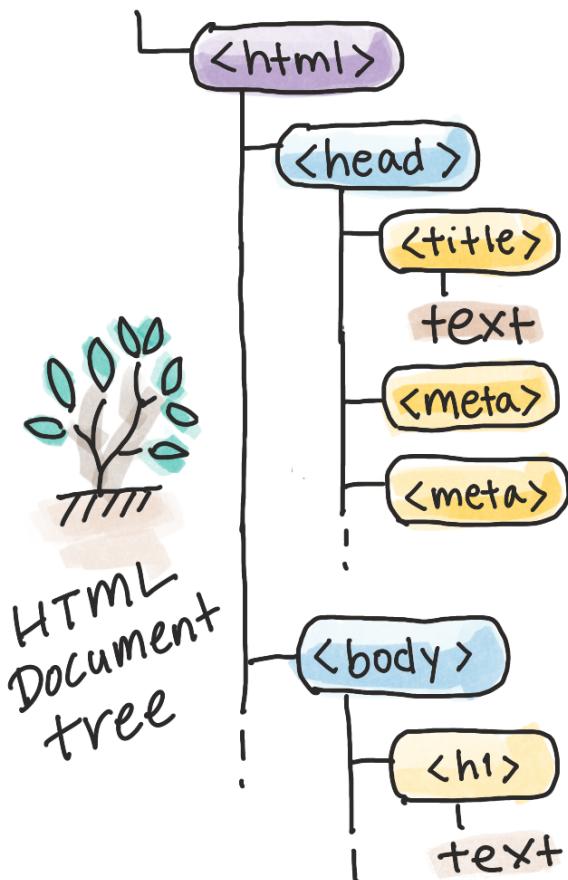


- ② `Cd my project`
- ③ `git init`
- ④ `git status`
- ⑤ `git add .`

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

# Introduction to HTML

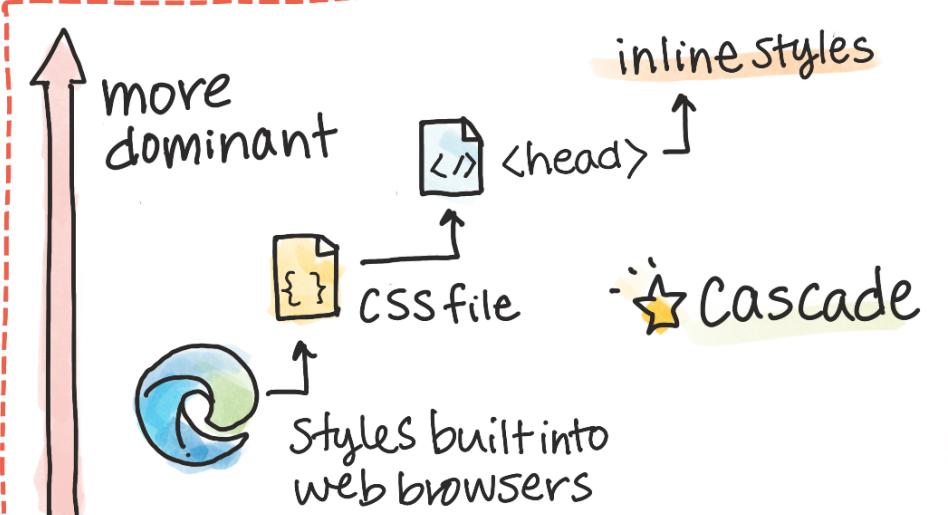
## Document



# Introduction to CSS

## CSS

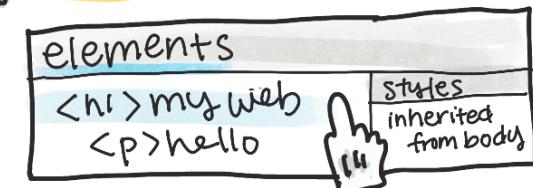
### Cascade



RWD makes web render nicely on a variety of screen sizes!

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

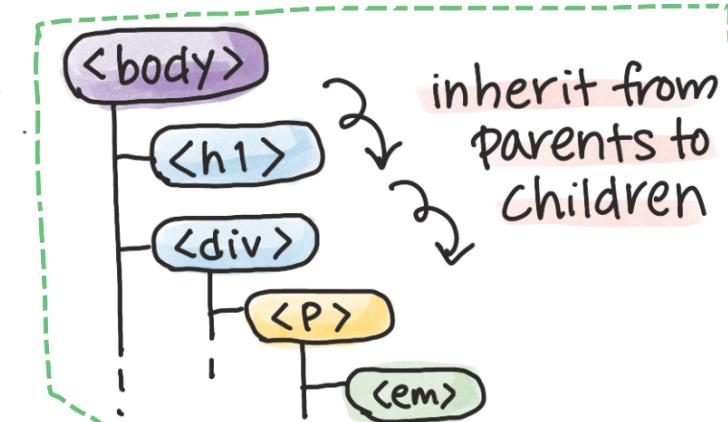
### Inspect in browser dev tools



H1's font is inherited from the body!

### Selectors

- tag
- id
- class



`<div id="Plant1" class="plant">`

`#plant1 { ... }`

### Layouts

- Positioning
- display rules
- flows etc.

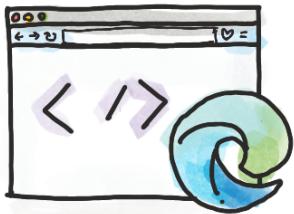
`.plant { position: absolute; width: 150%; z-index: 2; }`

# JavaScript

## DOM

manipulation

### Document Object Model



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

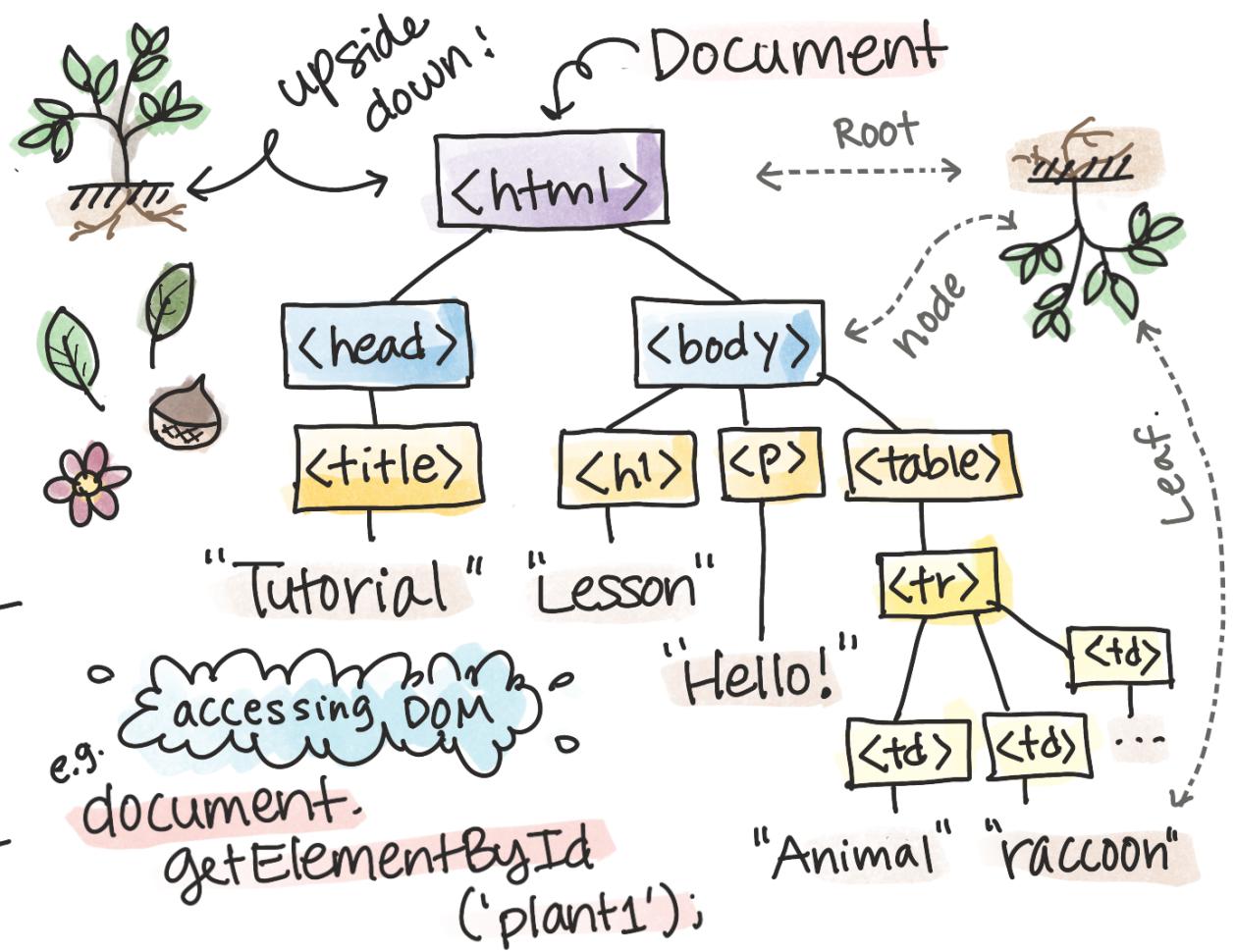
### Closure

↳ an outer function that encloses an inner function



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

@AzureAdvocates  
@girlie-mac



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

```
function pointerDrag(e) {  
  e.preventDefault();  
  pos3 = e.clientX;  
  ...  
}
```

# 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

keyword      variable name  
const greeting = "Hello";  
Declaring a variable      assigned value

# Data types

String

a set of characters that reside between

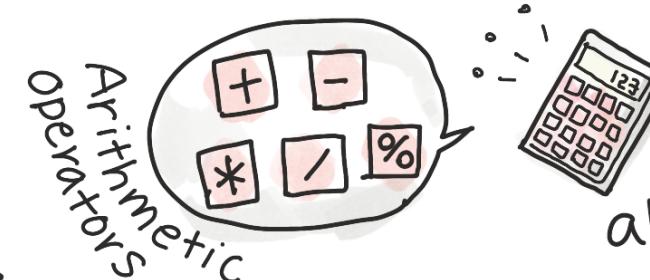
single or double quotes. ☺☺

Number

let donut = 32;

can be:  
integer, negative  
decimals, etc.

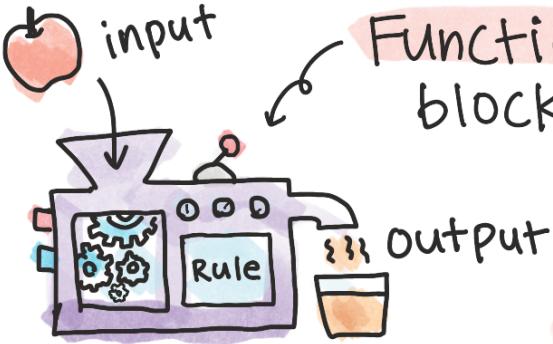
also: Infinity, BigInt



- + Addition
- Subtraction
- \* Multiplication
- / Division
- % Remainder

Boolean





**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

## ★ Passing info

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

# JavaScript Basics

## Functions

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

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

## ★ Default values

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

### Anonymous Function

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

## ★ Function as parameter

```
function displayDone() {
    console.log('3 sec. elapsed.');
}
```

### Fat Arrow Function

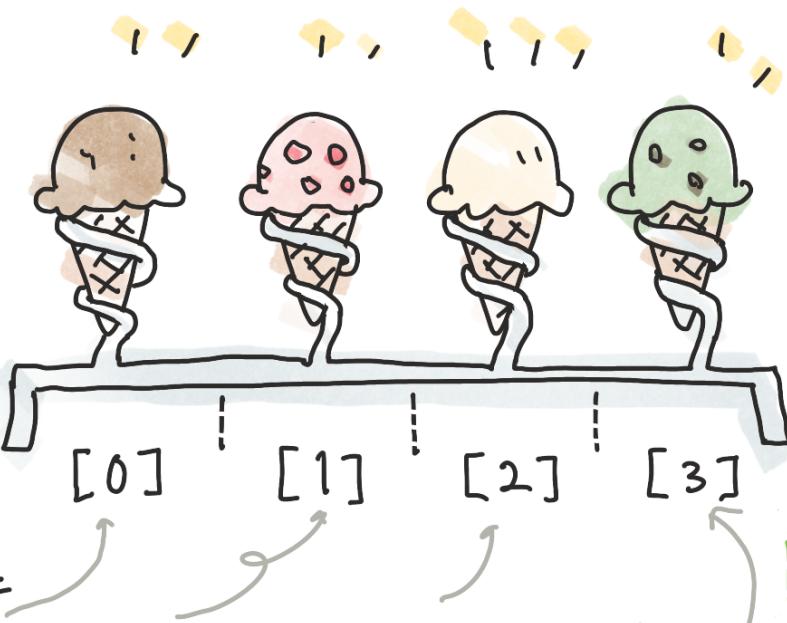
```
setTimeout(3000, () => {
    console.log(...);
});
```

```
setTimeout(3000, displayDone);
```



# Arrays

of  
ice cream



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

flavors[2]; // 'Vanilla'

flavors[4] = 'Rocky Road';

flavors[4]  
= 'Butter Pecan';

new!

Replaced!  
= Butter  
Pecan



@girlie-mac  
@AzureAdvocates

# JavaScript Basics

## Arrays [ and Loops ]

### For Loop

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

↳ Prints out each flavor  
after each iteration!

### While-Loop

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

↳ The loop will stop  
when the condition  
is met!

# JavaScript Basics

## Making Decisions

:= true Booleans false :=

let myStatement = true

let anotherStatement = false

if  
else If statements

if (status == 200) {

true → message = 'OK';  
Condition

} else {

false → message = 'Error!';

↓ Ternary

const message =  
(status == 200) ? 'OK' : 'Error!';

## Comparing values

> Greater than

>= Greater than or equal to

== equal values

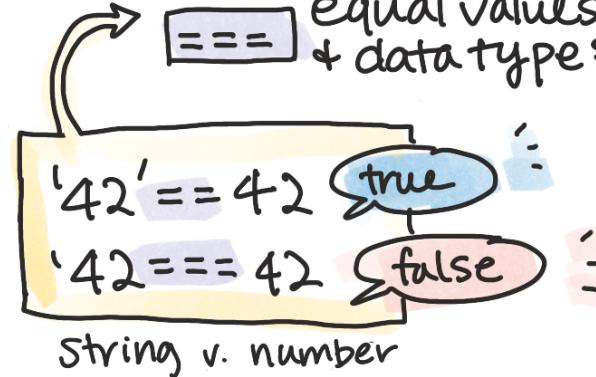
==+ data types

< Less than

<= Less than or equal to

!= not equal values

!== not equal values  
↓ datatypes



## Logical Operators

&&

logical AND

||

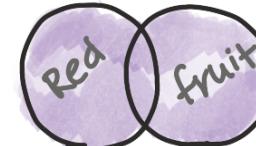
logical OR

!

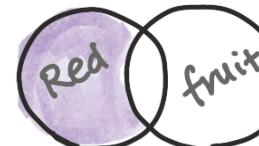
logical NOT



Red AND fruit



red OR fruit



red, not fruit

