# **WELCOME TO**

# JS FUNDAMENTALS

# WHAT IS THIS?

 An experimental meetup to bring together people who are interesting in learning to code / enhance coding abilities

# WHO AM I?

- Been coding professionally for 5 years
- Data analyst -> Web developer
- Attended a boot camp in 2017 stayed on to teach afterwards
- Work remotely here in CT for a startup in NYC

#### WHY AM I DOING THIS?

- For fun:]
- I found my passion for computer programming later in life, maybe this will help you to find the same
- I'm not affiliated with any company, non-profit, organization, govt. entity, yada yada yada

# YOUR TURN 😂

Who are ya and what brings ya here?

# WHY JAVASCRIPT?

- An easier language to work with than many
- Powers a software application we all use everyday the web

# WHAT IS JAVASCRIPT?

- Web apps are largely powered by 3 languages: HTML,
   CSS, and JS
- If a web app was a body:
  - ▶ The bones are the HTML a frame of sorts
  - The clothes are the CSS styles the frame
  - The brain is the JS manipulates the HTML, CSS and more

# **USING JAVASCRIPT**

- We'll be talking about:
  - Variables
  - Data types
  - Functions

#### **VARIABLES**

- Hold references to values
- Declare using the var, let, and const keywords
  - Let's just use let for now

```
let foo = 1
let bar = 2
let fooBar = foo + bar
```

#### DATA TYPE: STRING

- Strings: characters
  - Denoted by wrapping the value in quotes

```
let foo = 'foo'
let bar = "bar"
let fooBar = foo + bar //'foobar'
```

#### **TYPE: OBJECT**

- "Containers" for data values
- Each value has an associated key. You can reference this key to access the value

```
let obj = { foo: 'bar' }
obj.foo //'bar'

obj.randomKey = 'abc'

let newKey = 'newKey'
obj[newKey] = 'new value'
obj.newKey //'new value'
```

#### **TYPE: ARRAY**

- A container for values, but without keys
  - You reference values by their position, or index, within the container
  - Indices start at 0

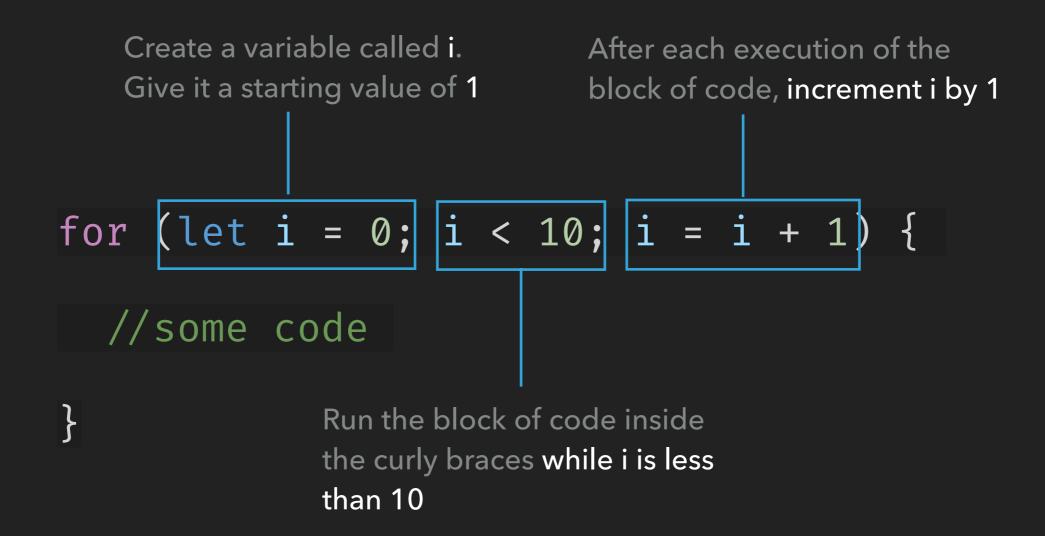
```
let arr = ['a', 'b', 1, 'foo']
arr[0] //'a'
arr[2] //1
arr[3] //'foo'

arr.push('bar')//['a', 'b', 1, 'foo', 'bar']
arr[4] //'bar'
```

#### **LOOPS**

 A chunk of code that is run over and over again, until a specified condition is met

```
let number = 0
for (let i = 0; i < 10; i = i + 1) {
  number = number + 1
}
number //10</pre>
```



#### **FUNCTIONS**

- A chunk of code that can be executed on demand
- Often returns a value
- Can "take in" values as arguments

```
function gimmeOne() {
  return 1
let one = gimmeOne() //1
let two = gimmeOne() + gimmeOne() //2
let three = two + gimmeOne() //3
function gimmeOneMoar(n) {
  return n + 1
let four = gimmeOneMoar(3) //4
let five = gimmeOneMoar(four) //5
let wut = gimmeOneMoar(gimmeOne())// ???
```

#### **CHALLENGES TIME!**

- We are going to use the technique of pair programming
  - A driver role and a navigator role
  - The driver writes the code
  - The navigator guides the driver (doesn't mean dictating code)
- Why?
  - Two minds are better than one when facing an unknown

#### **GUIDELINES FOR CHALLENGES**

- Google smartly!
- Keep your code tidy
- Use console.log in moderation
- Seek other groups for help! It's not a contest

# **REPL.IT**

# THE ONE RULE:

Don't be an asshole:1

# https://repl.it/@jsct/fundamentals