# Intro to JavaScript

Data Boot Camp

Lesson 14.1

# Class Objectives

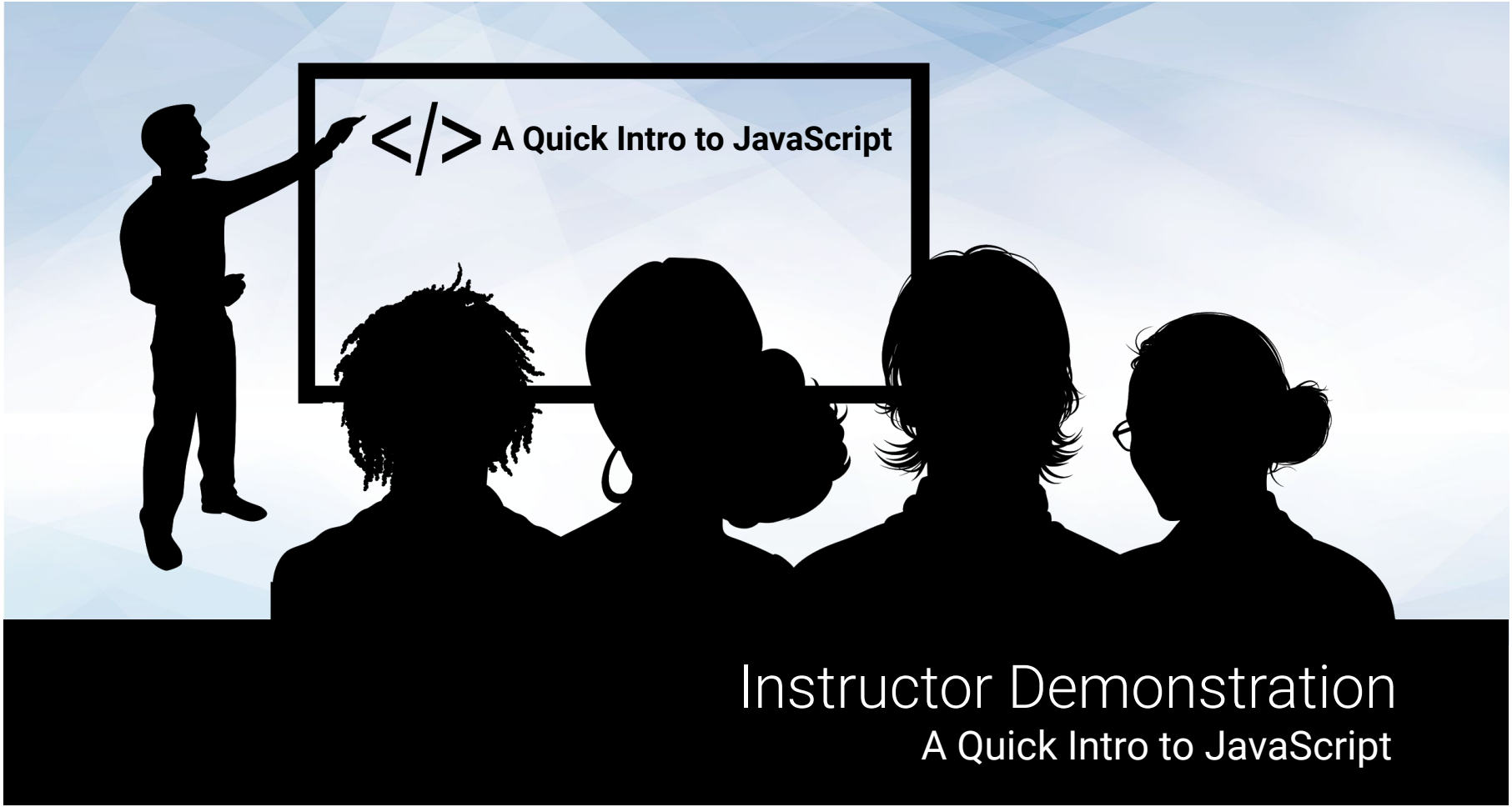
**By the end of today's class you will be able to:**

Understand JavaScript variables, data types, and statements.

Grasp basic JavaScript control flow (functions, loops, if/else statements).

Familiarize with JavaScript arrays.

Use and create functions in JavaScript, including built-in functions.




# Instructor Demonstration

## A Quick Intro to JavaScript

# Hello new language!

# A Quick Intro to JavaScript

## JavaScript fundamentals:

Arrays

Conditionals

Loops

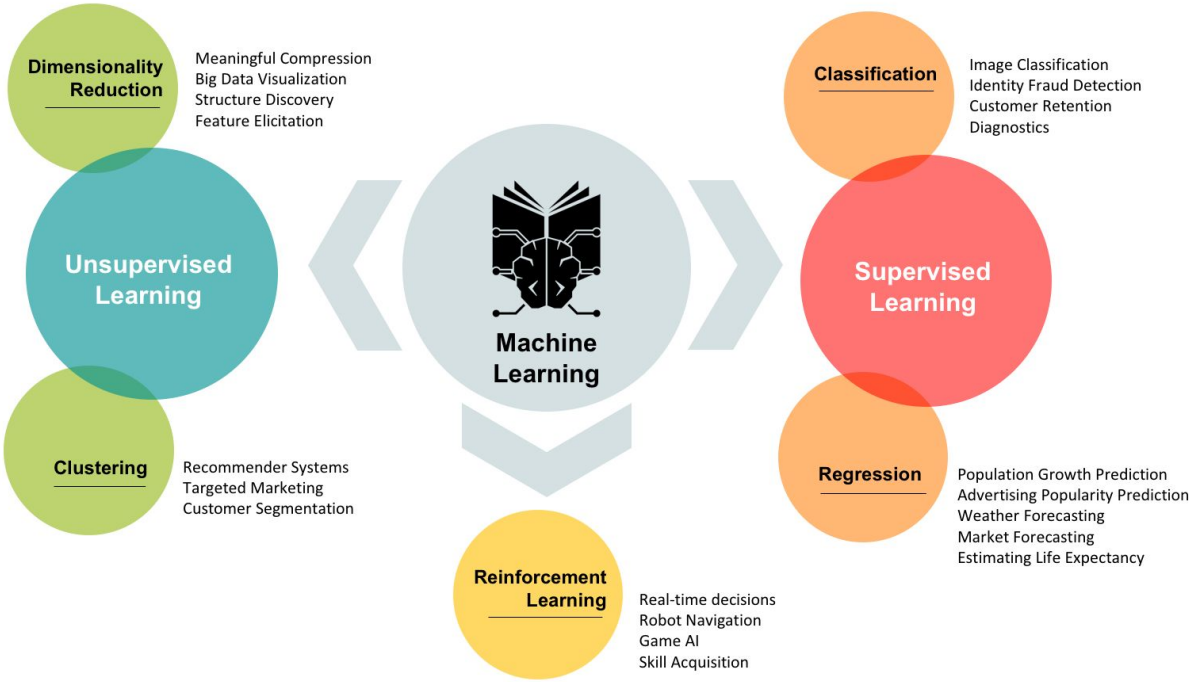
Functions

Objects


74.260
79.273
32.903
58.285
21.404
87.371
2.582
15.293
99.371
54.830
463
601
592
514
362
407
382
280
273
254
226
185
193
167
161
189
172
208
192


JS
JavaScript
IoT & the Cloud

# Machine Learning in the Browser



# How to Learn JavaScript

# Your Brain on JavaScript

# A Quick Intro to JavaScript

**Advice: Start Sloooow...**

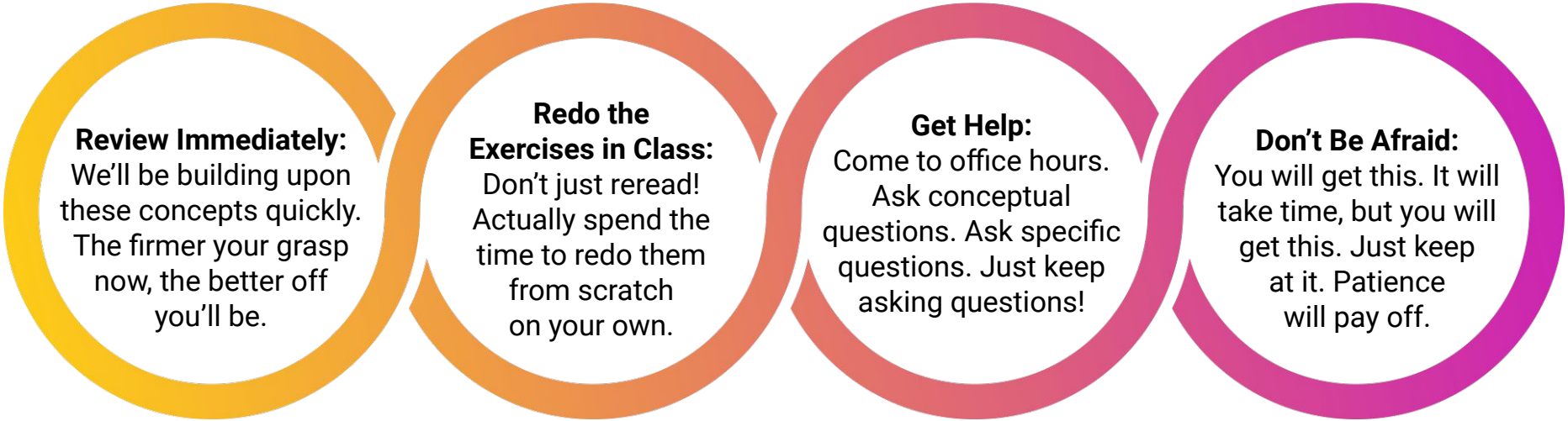
**and eventually, this will be you:**

# A Quick Intro to JavaScript

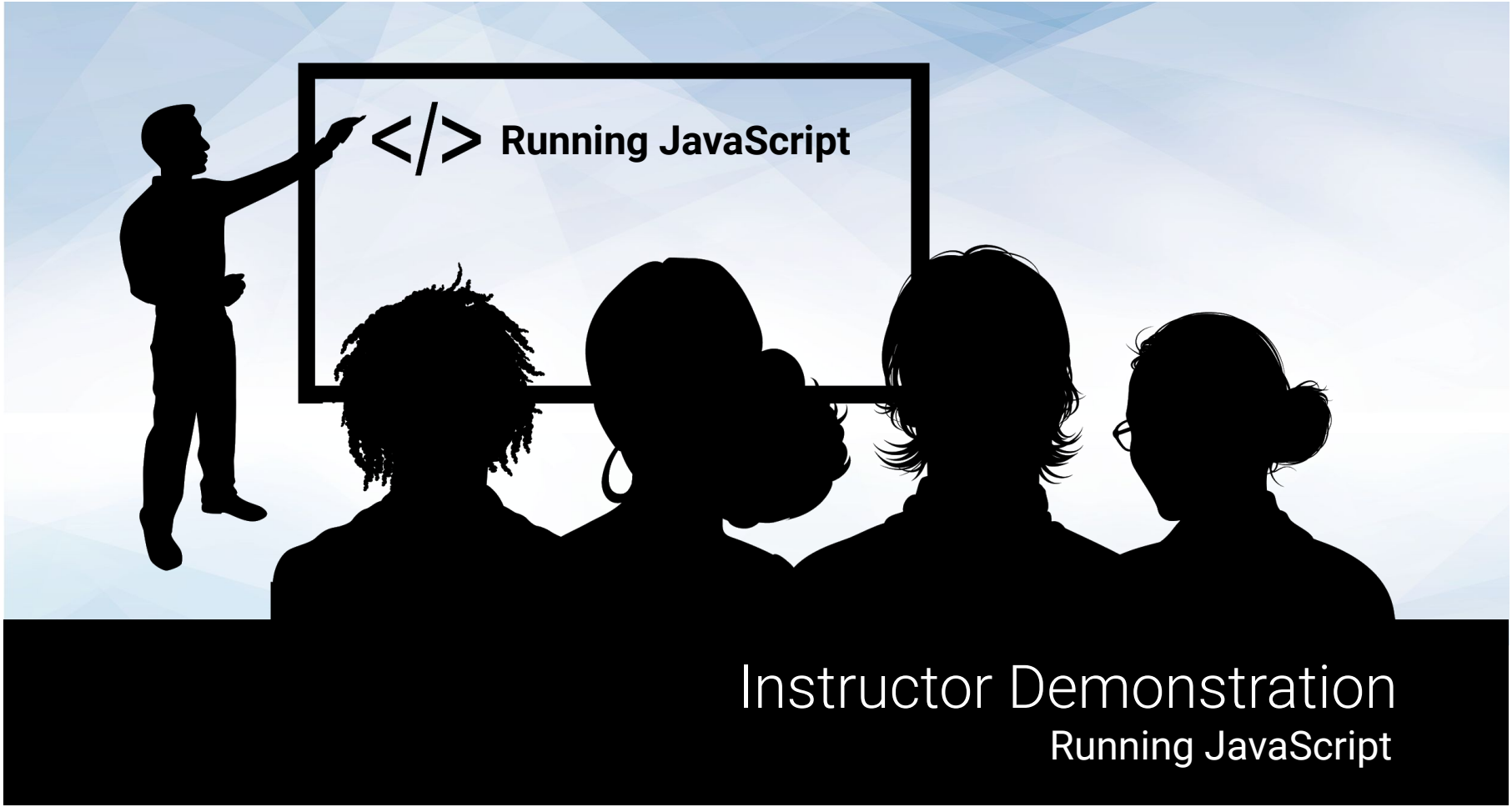
Talk to us if you need extra help!

# A Quick Intro to JavaScript

## General Tips




your
slack




# Instructor Demonstration

## Running JavaScript

# Running JavaScript

- JavaScript is known to be the language of the web, hence is associated with HTML, or inside HTML.
- The code is placed between a pair of `script` tags.
- The `console.log()` function prints out a message to a web browser's built-in console.


```
!<!DOCTYPE html>
<html lang="en">
  <head>
    <title></title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>

    <script type="text/javascript">

      console.log("My script is stored within the HTML!")

    </script>

  </body>
</html>
```

# Running JavaScript

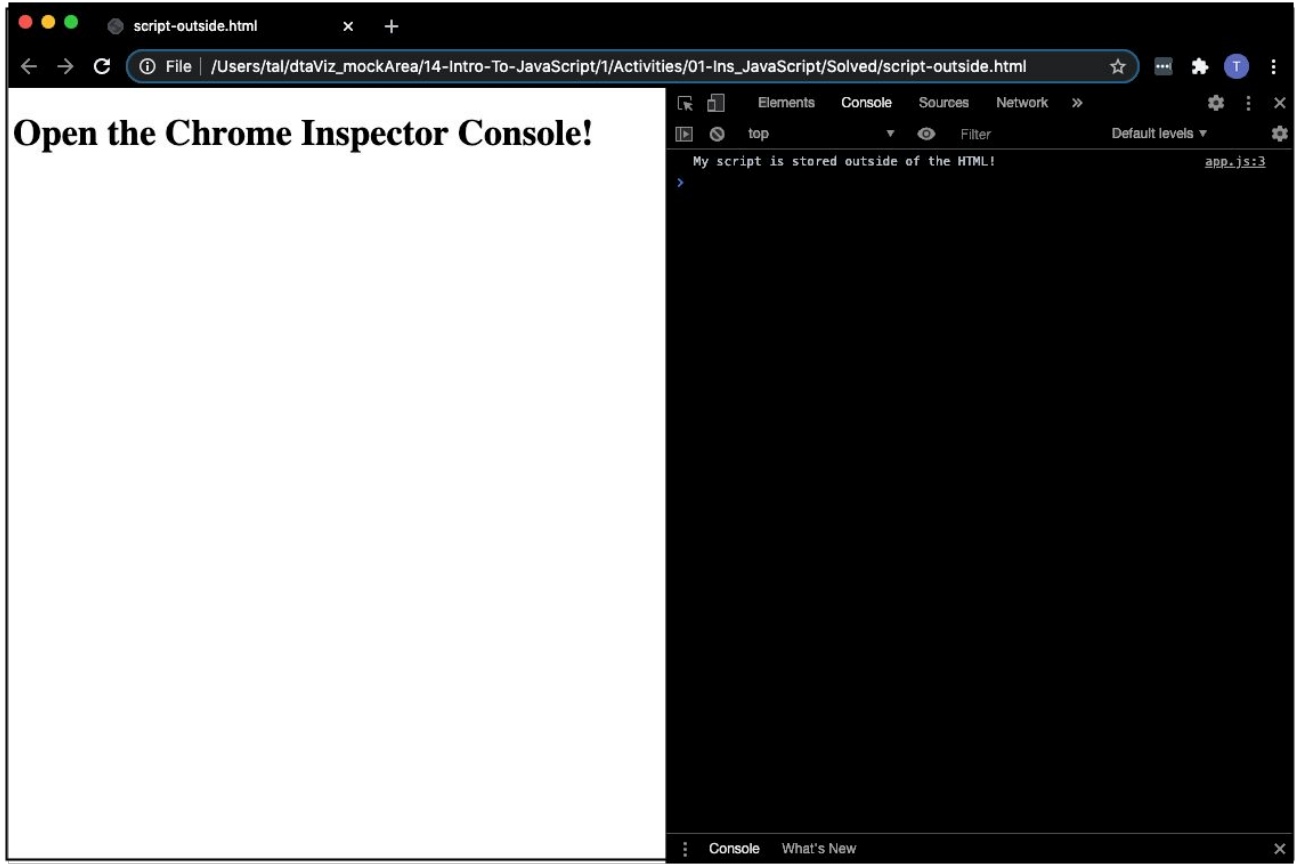
- src=”app.js”

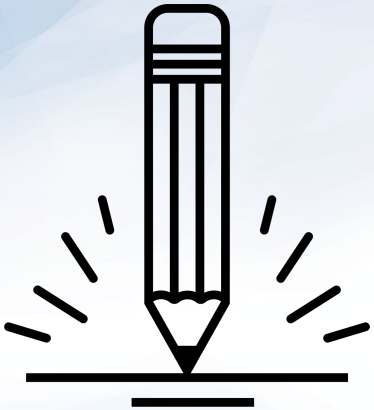
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <title></title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1">
  </head>
  <body>

    <script type="text/javascript" src="app.js"></script>

  </body>
</html>
```

- app.js

```js
// This code will run when linked to in HTML
console.log("My script is stored outside of the HTML!")
```

# Running JavaScript





# Everyone Do: From Python to JavaScript

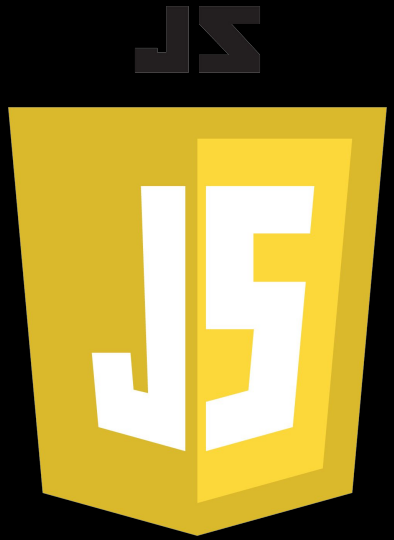
In this activity, everyone will work our way through some of the introductory Python scripts and translate them into JavaScript code.

**Suggested Time:**
25 Minutes


FEW THINGS
TO NOTE
BEFORE OUR
ACTIVITY

# **Everyone Do:** From Python to JavaScript

**JavaScript and Python are logically and syntactically similar.**




# **Everyone Do:** From Python to JavaScript

## Variable:

```
var <variable Name> = <Value>;
```

```
<variable Name> = <Value>
```

```
var name = "Mad Max";
```

```
name = "Mad Max"
```

# **Everyone Do:** From Python to JavaScript

## Booleans:

```
var <variable Name> = true;
```

```
var satisfied = true;
```

```
<variable Name> = True
```

```
satisfied = True
```

# **Everyone Do:** From Python to JavaScript

## String Template + note:

```
// JavaScript template literal
```

```
console.log(`Hello ${name}!`);
```

```
# Python f-string
```

```
print(f"Hello, {name}!")
```

# **Everyone Do:** From Python to JavaScript

## String format converted into numeric format:

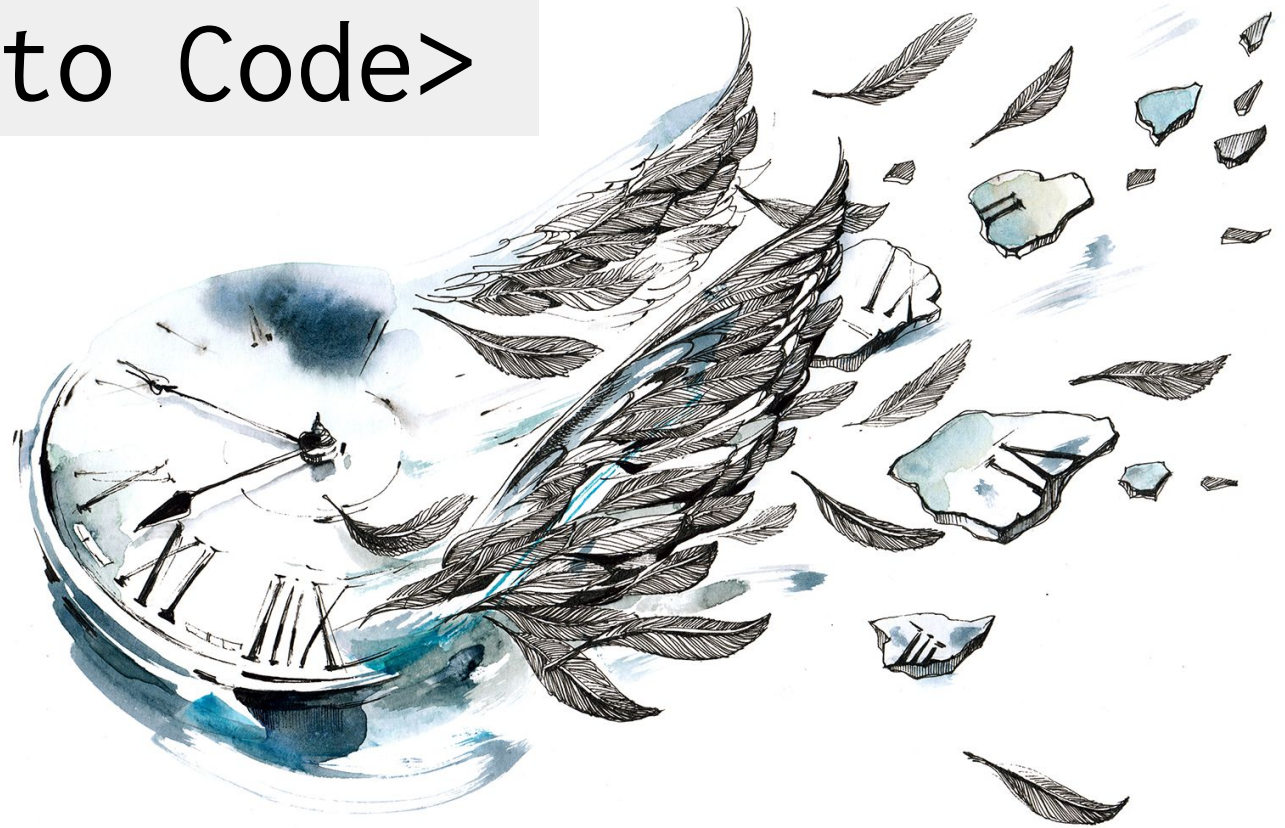
```
// parseInt() function
```

```
var weeklyWage = hourlyWage * parseInt(weeklyHours);
```

```
# int() function
```

```
weekly_wage = hourly_wage * int(weekly_hours)
```

# <Time to Code>




FEW THINGS
TO NOTE
BEFORE OUR
SECOND LIVE
CODE

# **Everyone Do:** From Python to JavaScript

## Curly braces, === vs. Whitespace, indentation and ==

```
// Check is one value is equal
// to another
```

```
if (x === 1) {
 console.log("x is equal to 1");
}
```

```
# Check is one value is equal to
# another
```

```
if x == 1:
    print("x is equal to 1")
```

# **Everyone Do:** From Python to JavaScript

## Conditionals

```
// &&
```

```
if (x === 1 && y === 10) {
 console.log("Both values
returned true");
}
```

```
# and
```

```
if x == 1 and y == 10:
   print("Both values
returned true")
```

# **Everyone Do:** From Python to JavaScript

## Conditionals

// ||

```
if (x < 45 || y < 5) {
 console.log("One or the other
statements were true");
}
```

# or

```
if x < 45 or y < 5:
   print("One or the other
statements were true")
```

# **Everyone Do:** From Python to JavaScript

## Conditionals

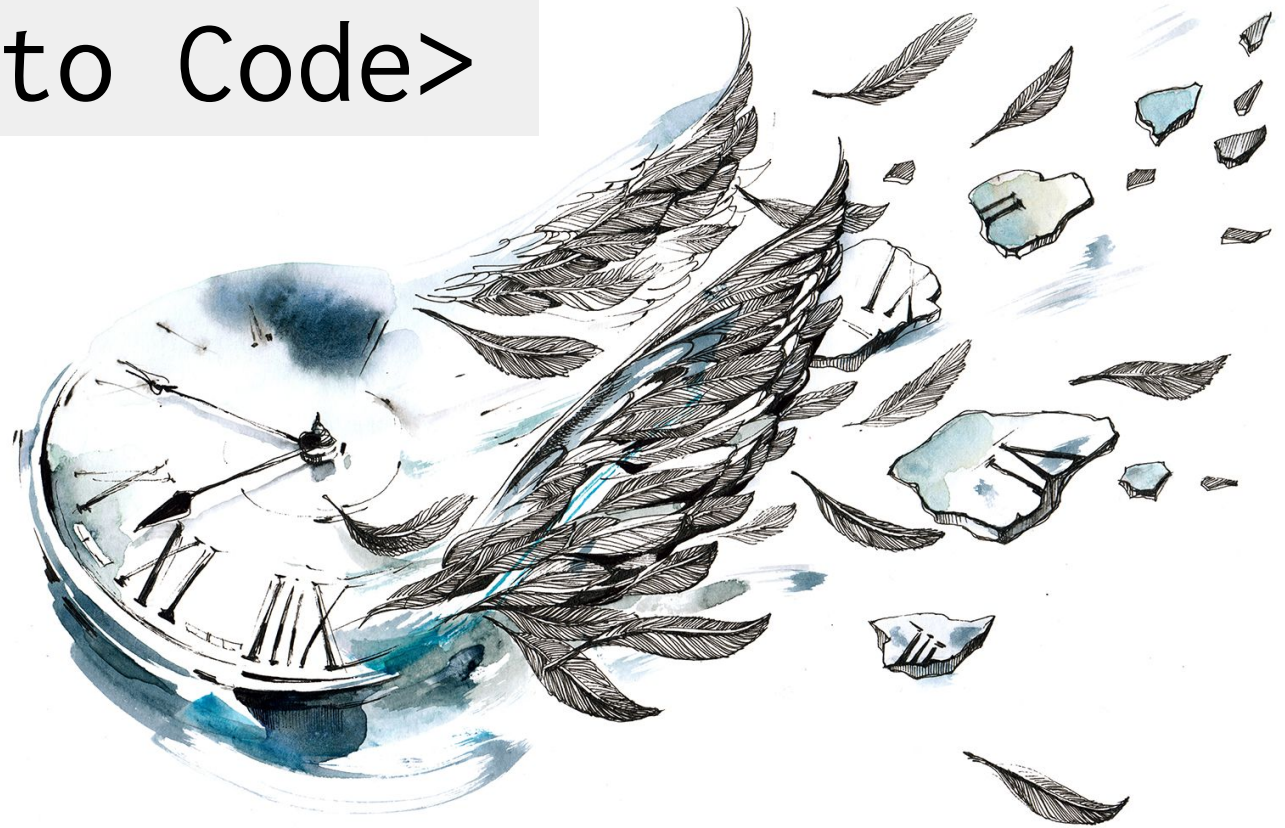
```
// Nested if...else if...else
```

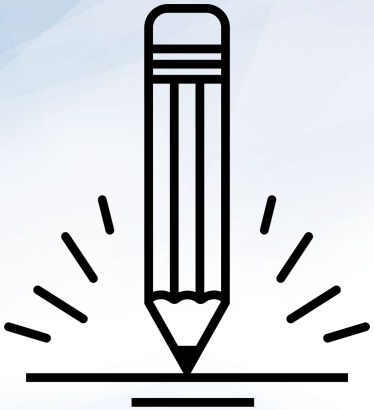
```
if (x < 10) {
  if (y < 5) {
   console.log("x < 10 && y < 5");
  }
  else if (y === 5) {
   console.log("x is less than 10 and y is equal to 5");
  }
  else {
   console.log("x < 10 and y >= 5");
  }
}
```

```
# Nested if...elif...else NESTED
```

```
if x < 10:
    if y < 5:
        print("x < 10 and y < 5")
    elif y == 5:
        print("x < 10 and y is equal to 5")
    else:
        print("x < 10 and y >= 5")
```

# <Time to Code>





# Activity: Javascript Loan Approver

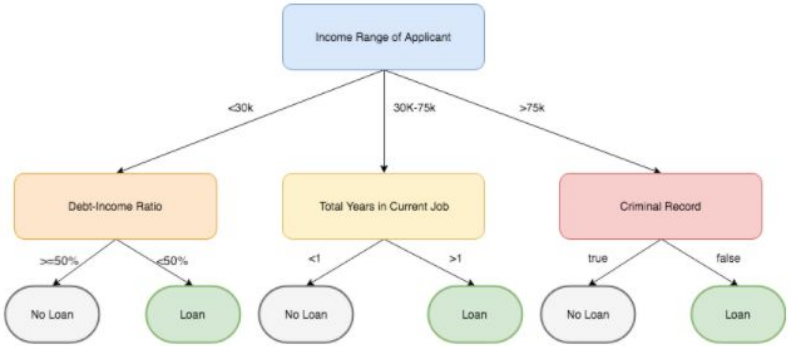
In this activity, you and your partner will use JavaScript to make a loan decision for a variety of conditions.

**Suggested Time:**
10 Minutes

# Javascript Loan Approver

## Instructions:

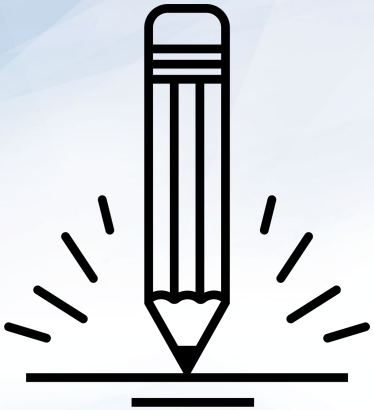
- Use the starter files and complete the logic to calculate a loan decision.
- Use the decision tree (right) to code the JavaScript conditional statements needed to make a loan decision.





- **Bonus:**
  - Use the advanced decision tree and write the logic to make a loan decision.
- **Hints:**
  - Start at the top of the tree and write the `if`, `else if`, `else` statement for income level. Then, fill in each of those with the `if`, `else` statements needed for the final decisions.
  - Consider writing the code in Python first and then translate it to JavaScript.

# **Time’s Up!** Let’s Review.



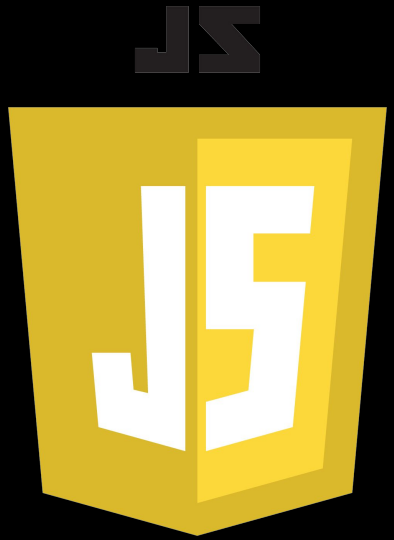
# Everyone Do: JavaScript Arrays

In this activity, we will be introduced to arrays in JavaScript.

**Suggested Time:**
15 Minutes

# **Everyone Do:** JavaScript Arrays

**JavaScript arrays are pretty similar to Python lists.**




# **Everyone Do:** JavaScript Arrays

## Arrays:

JavaScript array

```
var lettersArray = ["a", “b” , “c”, “d”];
```

index

```
var firstLetter = lettersArray[0];
var secondLetter = lettersArray[1];
a
b
```

# **Everyone Do:** JavaScript Arrays

## Arrays:

.push()

```
lettersArray.push("e");

var letterArray = ["a", “b” , “c”, “d”, “e”];
```

.slice()

```
var slicedArray1 = lettersArray.slice(1);
// Return the first three items of an array
var slicedArray2 = lettersArray.slice(0, 3);
// Return the second and third items of an array
var sliced Array 3 = lettersArray.slice(1, 3);
```

# **Everyone Do:** JavaScript Arrays

## Arrays:

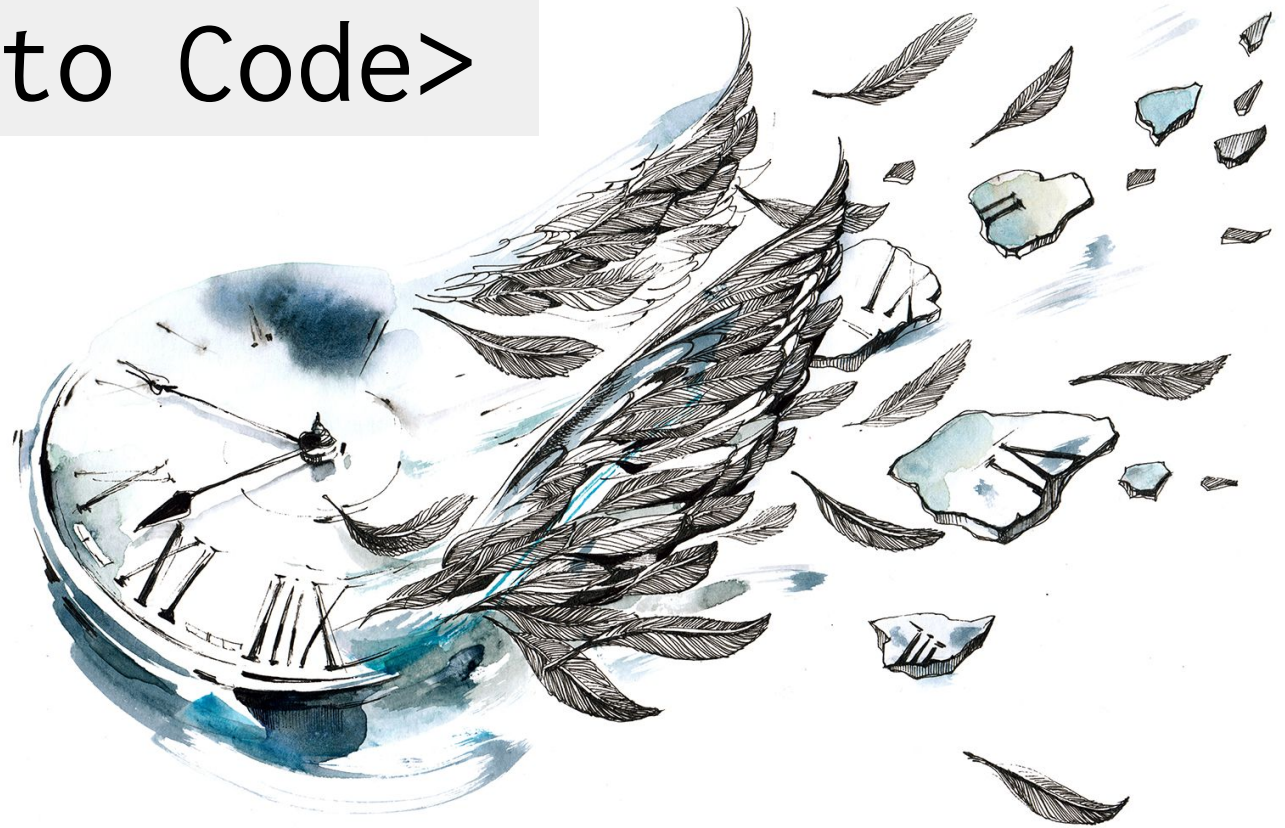
.join()

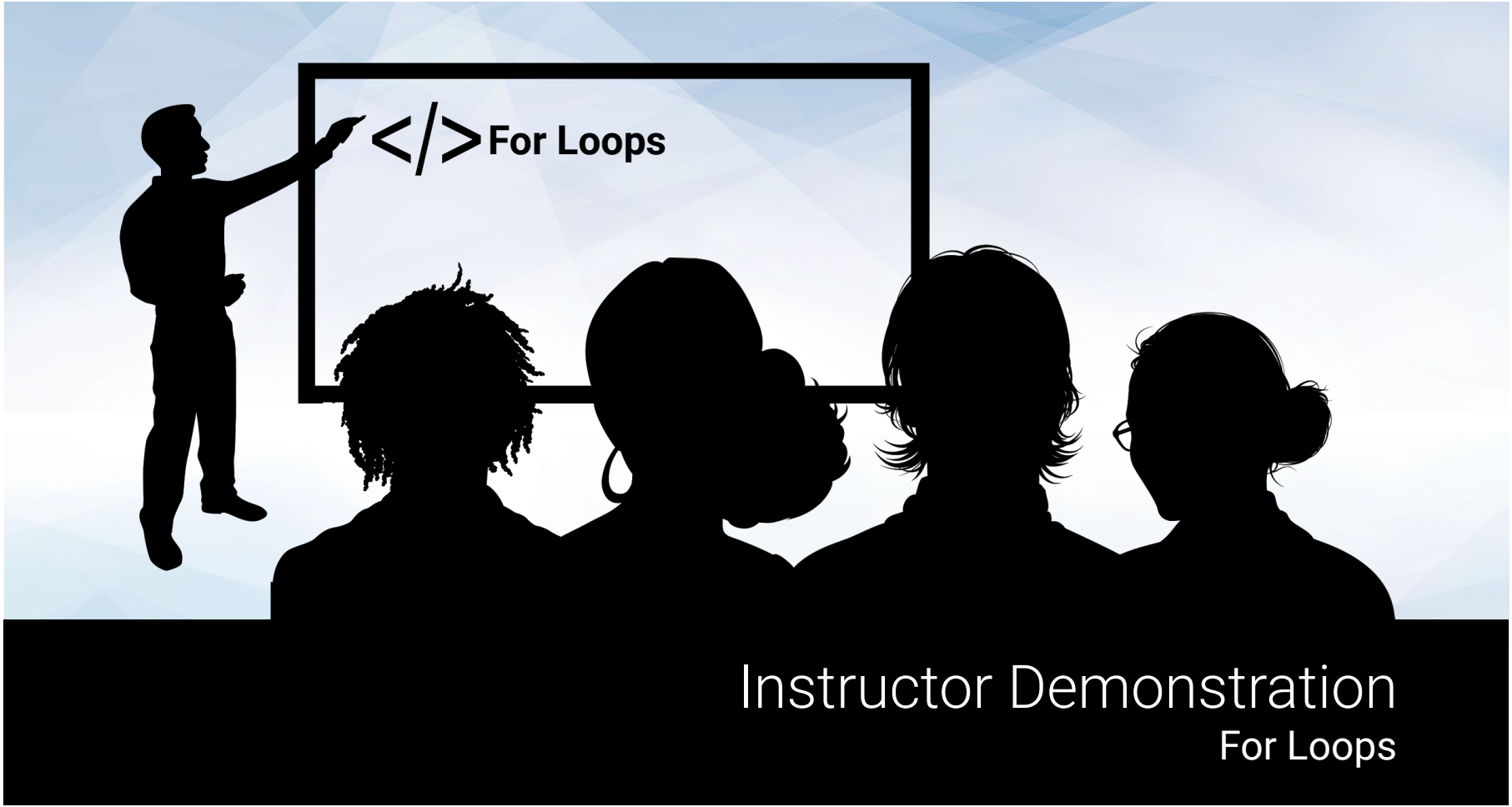
```
var joinedArray = lettersArray.join(", ");
```

.split()

```
var soundArray = soundOfMusic.split(" ");
```

# <Time to Code>





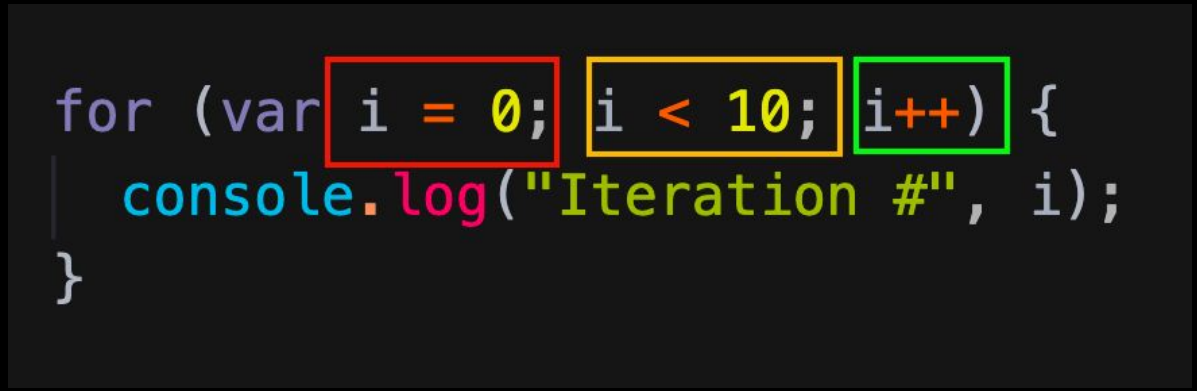

# Instructor Demonstration

## For Loops

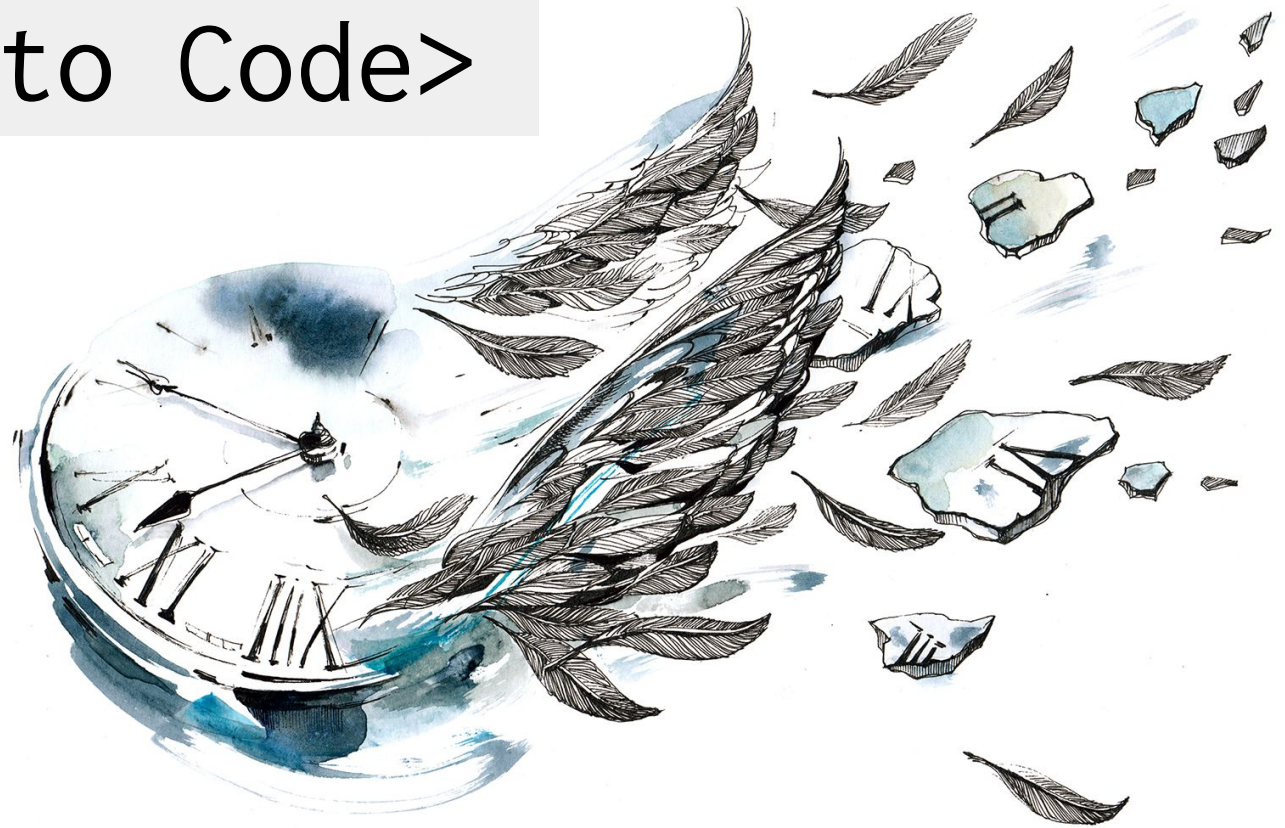
# For Loops

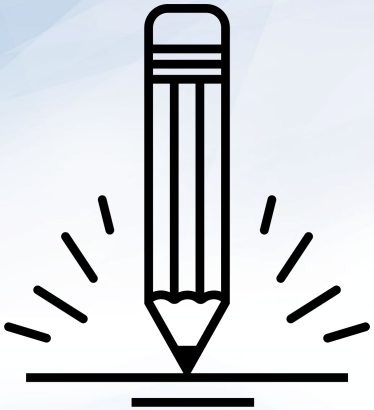
- `for` loops in JavaScript.

➔ Start
➔ End condition
➔ Increment



# <Time to Code>





# Activity: Movie Scores

In this activity, you will use conditionals and loops to iterate through an array of movie scores and sort scores into new arrays by their values.

**Suggested Time:**
20 Minutes

# Movie Scores

## Instructions:

- Given a list of movie scores, determine how many good, ok, and bad movies were there.
- Create a for loop to go through the `movieScore` list.
- Add scores over 7 to the `goodMovies` array.
- Add scores between 5 and 7 to the `okMovies` array.
- Add the rest of the scores to the the `badMovies` array.
- Also, calculate the average rating for all of the movies.
- Finally, print out how many good, ok, and bad movies there were and what the overall total score was.

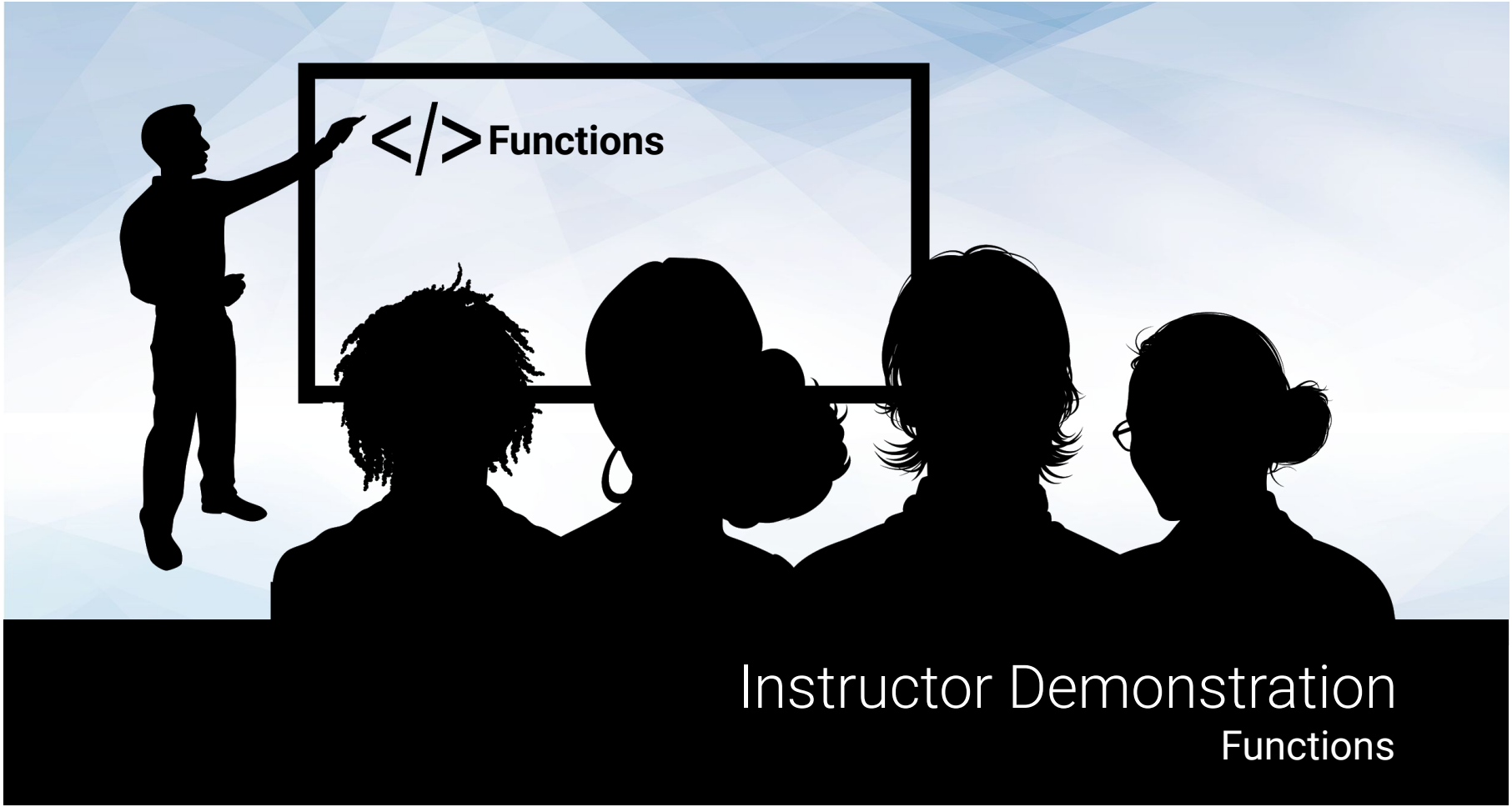
- **Hints:**
  - You will need to research how to push elements to an empty array.
  - Check your slack for the documentation to find the length of the array.

**Time’s Up!** Let’s Review.


Countdown timer

15:00

(with alarm)

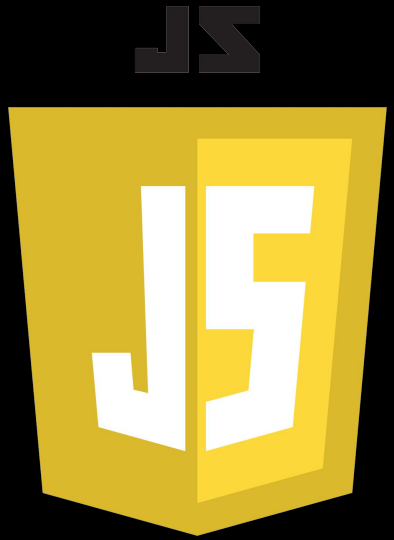



# Instructor Demonstration

## Functions

# **Everyone Do:** JavaScript Arrays

**Comparing functions in JavaScript and Python.**




# Functions

function

```
function printHello() {
    console.log("Hello there!");
}
function addition(a, b) {
  return a + b;
}
```

def

```
def print_hello():
    print("Hello there!")
def addition(a, b):
    return a + b
```

# Functions

function

```
printHello();
console.log(addition(44, 50));
```

def

```
print_hello()
addition(44, 50):
```

# Functions

```
// Accepts a parameter and
iterates through an array
```

```
function listLoop(userList) {
 for (var i = 0; i < userList.length; i++) {
   console.log(userList[i]);
 }
}
var friends = ["Sarah", "Greg", "Cindy",
"Jeff"];
listLoop(friends);
```

```
# Takes in a list and loops
through
```

```
def list_loop(user_list):
   for i in user_list:
       print(i)
```

# Functions

```
// Functions can call other functions
function doubleAddition(c, d) {
  var total = addition(c, d) * 2;

  return total;
}
// Log results of doubleAddition function
console.log(doubleAddition(3, 4));
```

```
# Uses a previous declared function
def double_addition(c, d):
    total = addition(c, d) * 2

    return total
```

# Functions

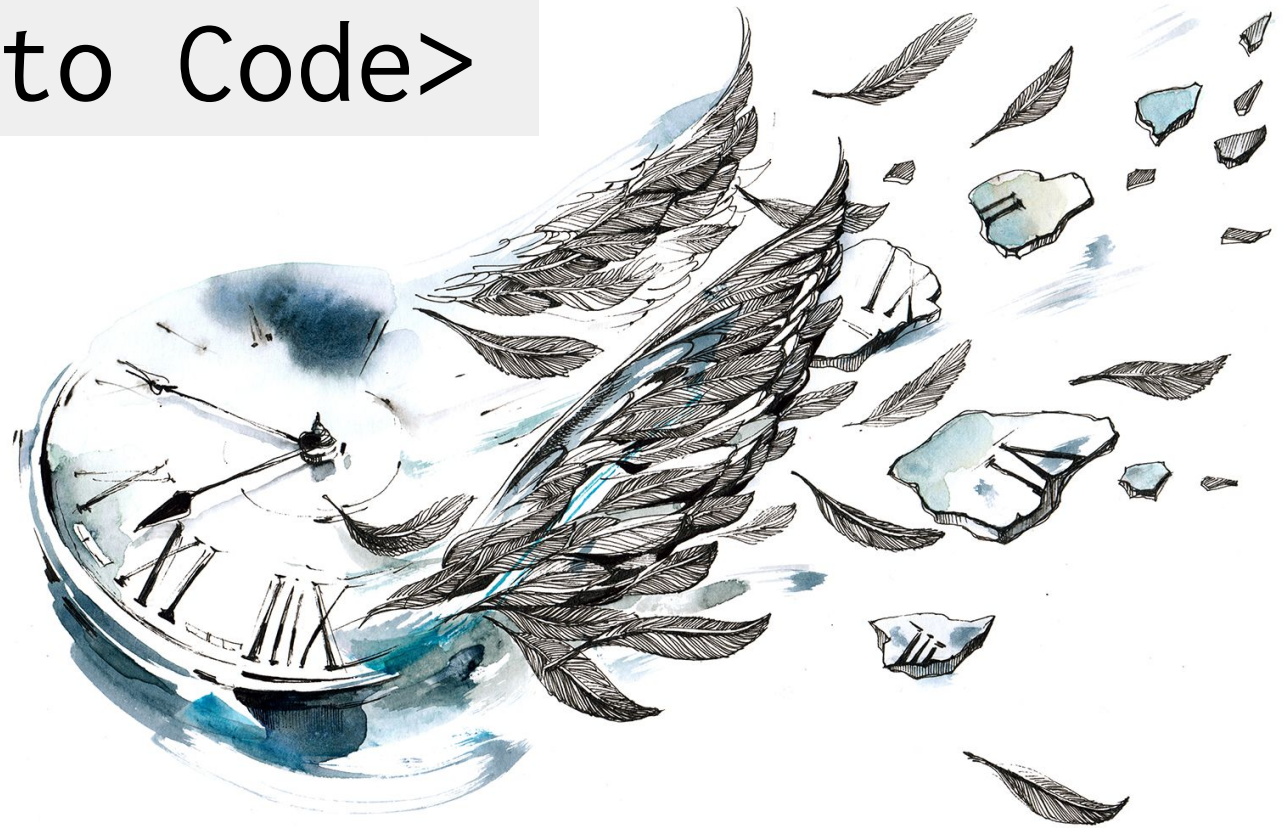
```
// JavaScript built in functions
```

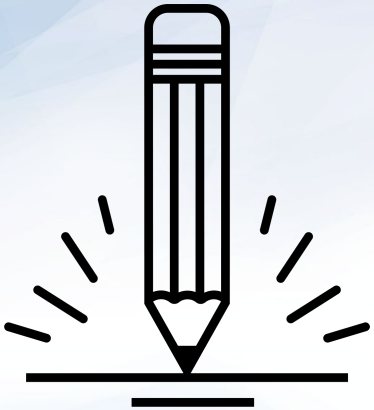
```
var longDecimal = 112.34534454;
var roundedDecimal = Math.floor(longDecimal);
console.log(roundedDecimal);
```

```
# Python built in function for rounding
```

```
long_decimal = 112.34534454
rounded_decimal = round(long_decimal)
print(rounded_decimal)
```

# <Time to Code>





# Activity: Statistics Functions

In this activity, you will create functions that returns statistical values from any given array of data.

**Suggested Time:**
20 Minutes

# Movie Scores

## Instructions:

- Using the movie array from earlier as a starting point, create functions that will return statistical values from any given array of data.
- Create functions that will find the following:
  - Mean
  - Variance
  - Standard Deviation
- Each function should `console.log` both the name of the statistic used and its value. For example "The Mean is: 33.3".
- The functions should be able to take any array of numbers and return the statistical value.
- After you have the functions working with movie data set run them on the following additional data points:
  - `monthlyAvgRainFall = [3.03, 2.48, 3.23, 3.15, 4.13, 3.23]`
  - `mileRunTimes = [5.06, 4.54, 5.56, 4.40, 7.10]`
- **Hints:**
  - Use the Javascript Math library to handle calculations needing exponents or square roots.
  - Check your slack to refresh how to calculate variance and standard deviation.

# **Time’s Up!** Let’s Review.


What?
Homework?