Intro to JavaScript Visualizations

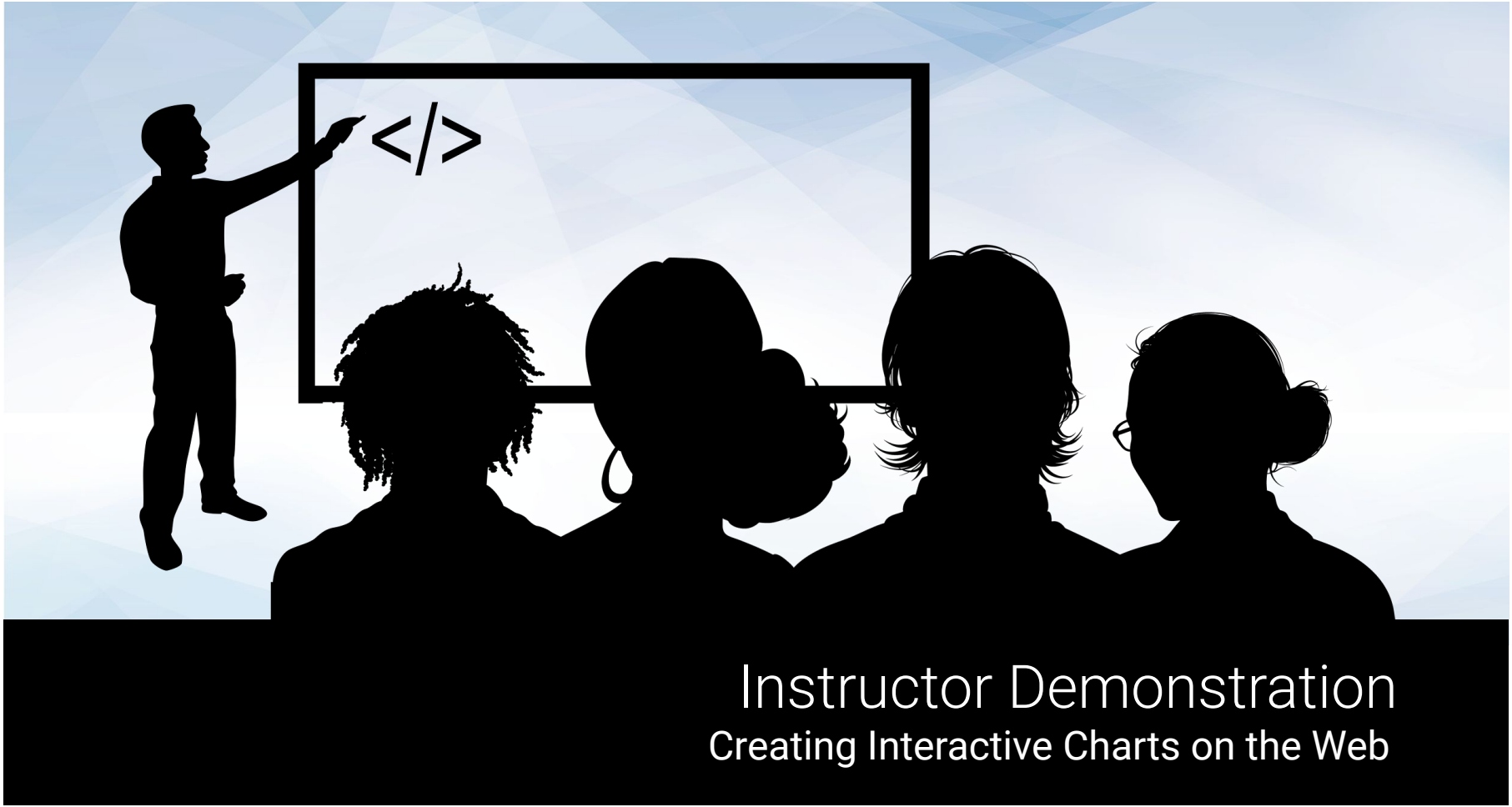
Data Boot Camp

Lesson 14.1

Class Objectives

By the end of this lesson, you will be able to:

- Describe JavaScript variables, arrays, data types, and statements.
- Implement basic JavaScript control flow (functions, loops, if/else statements).
- Create functions in JavaScript.
- Create, update, and iterate JavaScript Objects.
- Create basic charts, including nar charts and line charts using Plotly.
- Use Plotly’s layout object to customize the appearance of their charts.
- Annotate charts with labels, text, and hover text.



Instructor Demonstration

Creating Interactive Charts on the Web

Instructor Do: Creating Interactive Charts on the Web

index.html

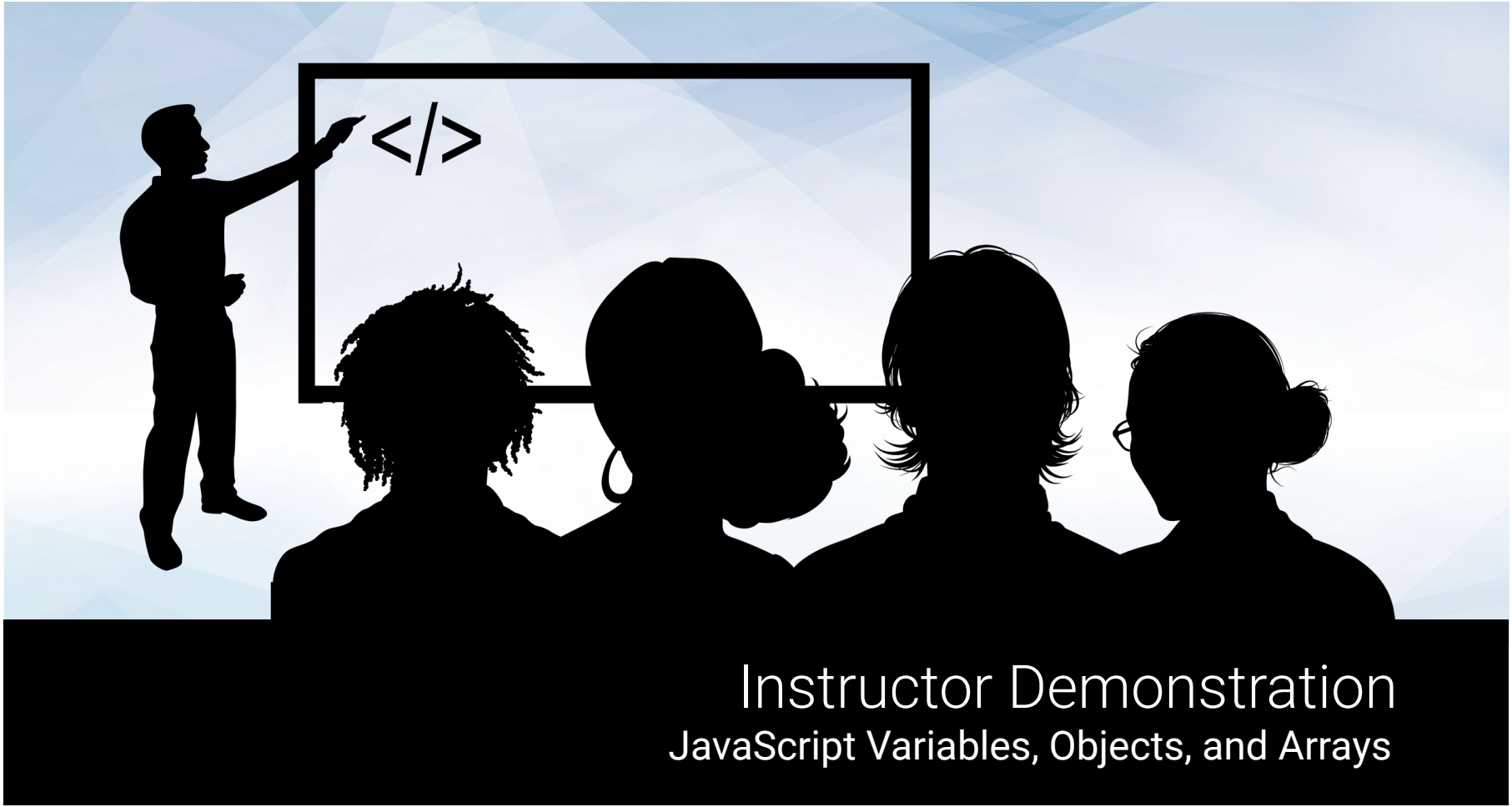
Instructor Do: Creating Interactive Charts on the Web

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Basic Charts</title>
   <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
<body>
   <div id="plot"></div>
   <script>
        let xData = [1, 2, 3, 4, 5];
        let yData = [1, 2, 4, 8, 16];
   </script>
   <script src="plots.js"></script>
</body>
</html>
```

➔ Loads the Plotly library

➔ JavaScript is written directly into the html file.

➔ Links to an external file.



Instructor Demonstration

JavaScript Variables, Objects, and Arrays

Instructor Do: Javascript Variables, Objects, and Arrays

JavaScript and Python variables are similar, however...

- In JavaScript they must be initialized.

Instructor Do: Javascript Variables, Objects, and Arrays

JavaScript and Python variables can be assigned to string values:

```
<variable name> = <Value>
```

```
name = "Homer Simpson"
```

```
let <variable name> = <Value>;
```

```
let name = "Homer Simpson";
```

Instructor Do: Javascript Variables, Objects, and Arrays

Can be assigned to booleans values:

```
<variable name> = true or false;
```

```
is_employed = True
```

```
let <variable name> = true or false;
```

```
let isEmployed = true;
```

Instructor Do: Javascript Variables, Objects, and Arrays

Can be assigned to numerical values:

```
<variable name> = integer or float
```

```
age = 39
hourly_wage = 11.99
```

```
let <variable name> = number;
```

```
let age = 39;
let hourlyWage = 11.99;
```

Instructor Do: Javascript Variables, Objects, and Arrays

Can be assigned in expressions using other variables.

```
<variable name> = <another variable name> ( +, -, /, *) integer or float;
```

```
daily_wage = hourly_wage * 8
weekly_wage = daily_wage *
```

```
let <variable name> = <another variable name> ( +, -, /, *) number;
```

```
let dailyWage = hourlyWage * 8;
let weeklyWage = dailyWage * 5;
```

Instructor Do: Javascript Variables, Objects, and Arrays

Template Literal:

```
# Python f-string
```

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

```
// JavaScript template literal
```

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

Instructor Do: Javascript Variables, Objects, and Arrays

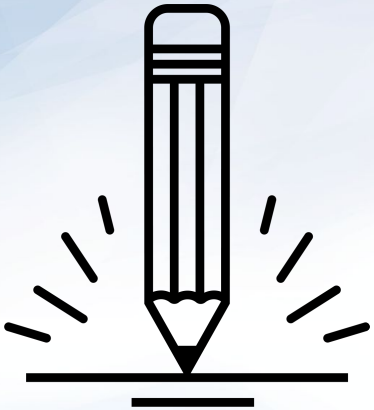
- Objects are collections of properties.
  - Properties are key-value relationships (pairs).

- There are two ways to access a property from JSON (JavaScript Object Notation).
  - Bracket notation, similar to Python.
  - Dot notation.

- A Syntax for storing and exchanging data.
- Is similar to a Python dictionary in many ways:
  - Organize information in `key` and `value` pairings.
  - They are unordered.
  - `key` is used to access the `value`.

Instructor Do: Javascript Variables, Objects, and Arrays

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



Activity: My Variables, Objects, and Arrays

In this activity, you will create variables and console logging strings with template literals.

Suggested Time:
10 Minutes

Instructions:

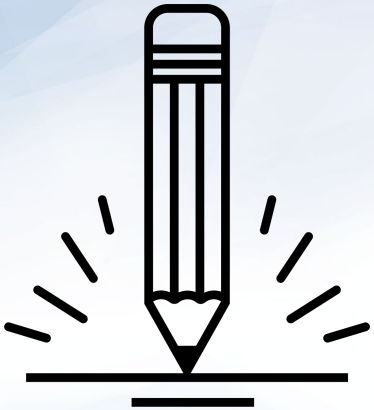
Activity: My Variables, Objects, and Arrays

- Open index.html in your browser to run your code in HelloVariableWorld.js. Check your code by refreshing your browser.
- Create two variables called `name` and `country` that will hold strings.
- Create two variables called `age` and `hourlyWage` that will hold integers.
- Create a variable called `satisfied` which will hold a Boolean.
- Create a variable called `dailyWage` that will hold the value of `hourlyWage` multiplied by 8.
- With template literals, print the `name`, `country`, `age`, `hourlyWage`, `dailyWage` and `satisfied` variables.

- **Hint:**

  - For more information about template literals, see MDN's reference page for template literals.

Let's Review



Activity: My First Plotly Chart

In this activity, you will create your first Plotly bar chart using the variables you created in the previous activity to show three books you've read as well as the number of times you've read them.

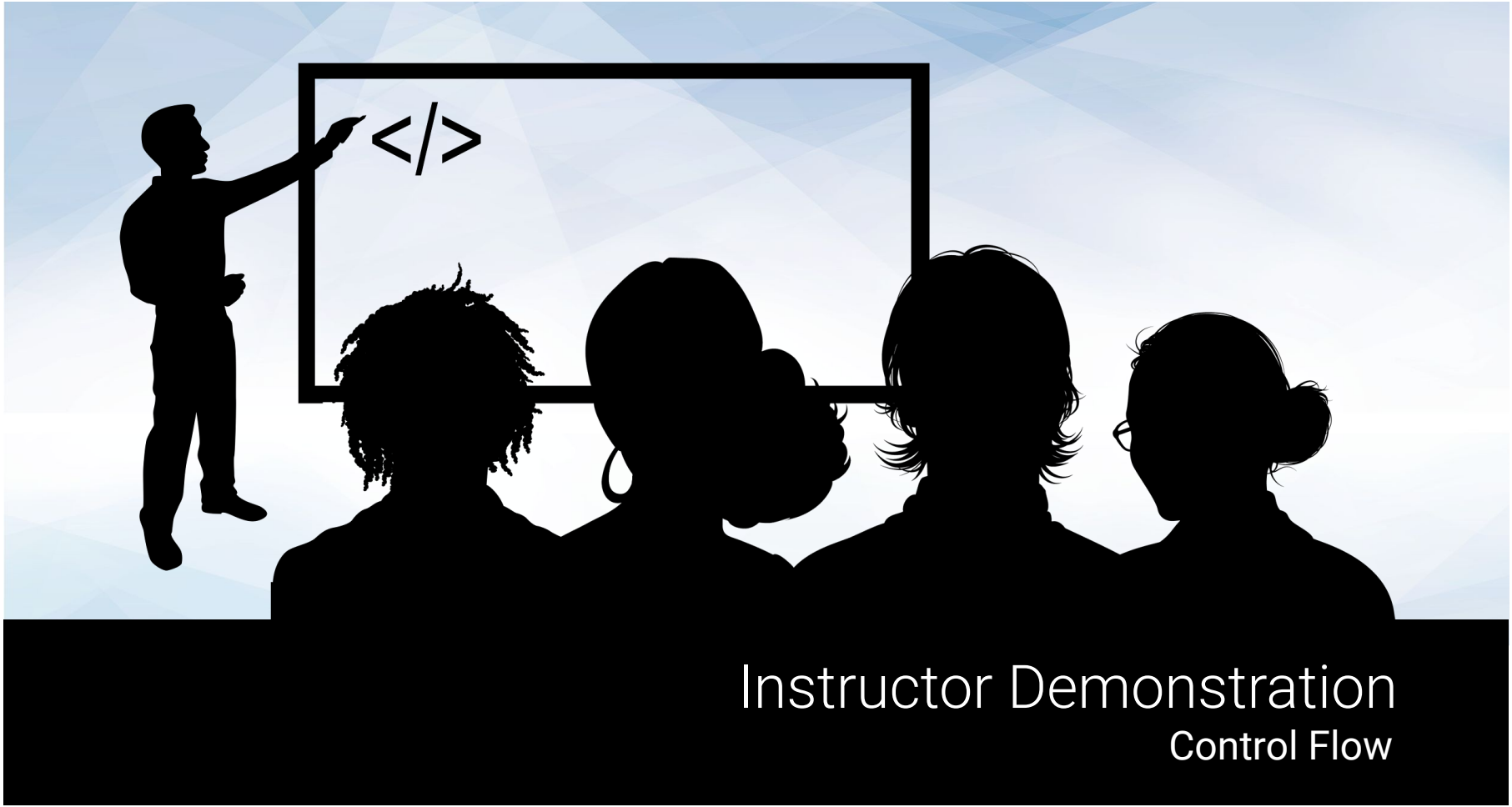
Suggested Time:
5 Minutes

Instructions:

Activity: My First Plotly Chart

- In the plots.js file, copy the following variables you created from the previous activity:
  - `name`
  - `title`
  - `books`
  - `timesRead`
- In the Plotly `trace1` object, assign the correct `x` and `y` values for a vertical bar chart.
- Open the index.html file to see your first Plotly bar chart.

Let’s Review



Instructor Demonstration

Control Flow

Instructor Do: Control Flow

- `for` loops in JavaScript.

➔ Start
➔ End condition
➔ Increment

```
for (var i = 0; i < 10; i++) {
  console.log("Iteration #", i);
}
```

Instructor Do: Control Flow

Conditionals

```
# and
```

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

```
// &&
```

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

Instructor Do: Control Flow

Conditionals

```
# or
```

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

```
// ||
```

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

Instructor Do: Control Flow

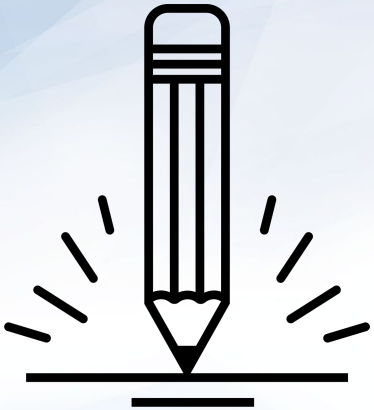
Conditionals

```
# if...elif...else
```

```python
if x < 10:
    if y < 5:
        print("x is less than 10 and y is less than 5")
    elif y == 5:
        print("x is less than 10 and y is equal to 5")
    else:
        print("x is less than 10 and y is greater than 5")
```

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

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



Activity: Iterations and Conditionals

In this activity, you will create a for loop, append values into arrays based on the movie's decade, calculate the average rating of all movies, and print out how many of the top 10 movies came from each decade.

Suggested Time:
10 Minutes

Instructions:

Activity: My Variables, Objects, and Arrays

- In this activity, given a list of movie objects, you will:
  - Append the movies into arrays based on the movie's decade.
  - Calculate the average rating of all movies.
  - Print how many of the top 10 movies came from each decade.

- **Hints:**

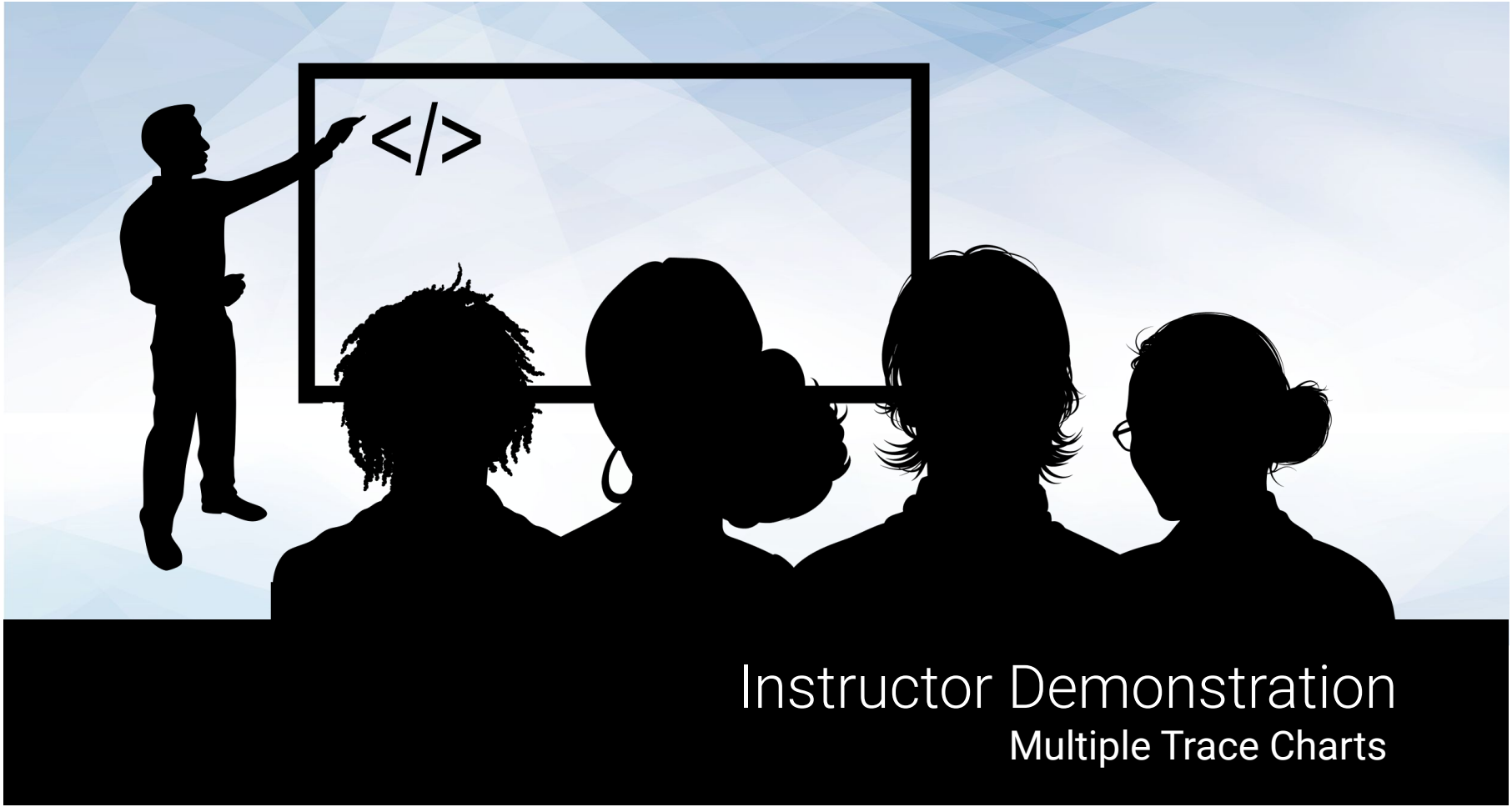
  - For more information about how to push elements to an empty array, review 02-Ins_JavaScript-Variables_and_Data_Structures.
  - To find the length of the array, review the JavaScript Array documentation from W3Schools.

Let's Review

Countdown timer

15:00

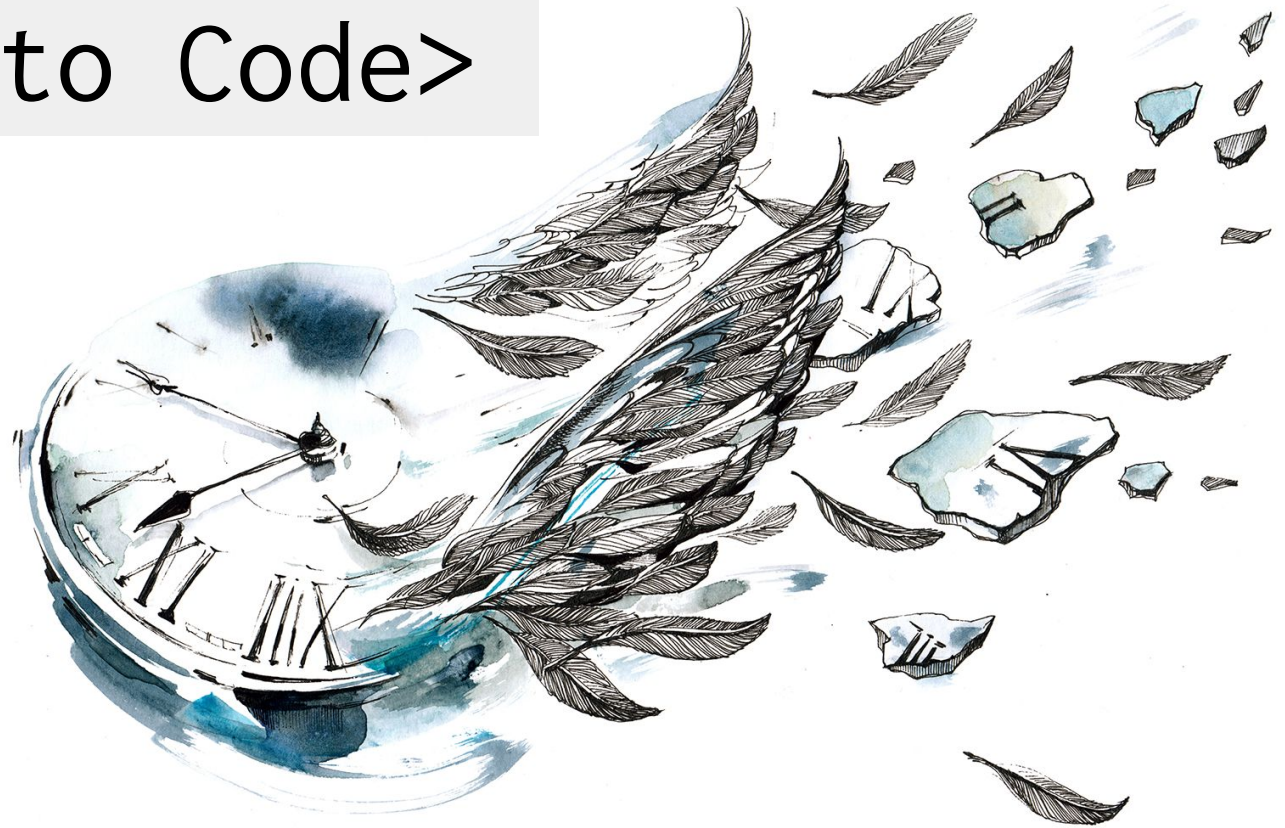
(with alarm)

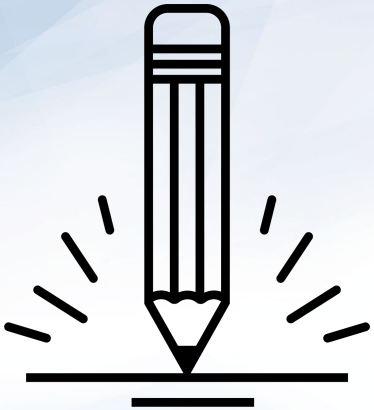


Instructor Demonstration

Multiple Trace Charts

<Time to Code>





Activity: Multiple Traces

In this activity, you will compare search results between Greek and Roman mythology to see which god is the most popular.

Suggested Time:
15 Minutes

Instructions:

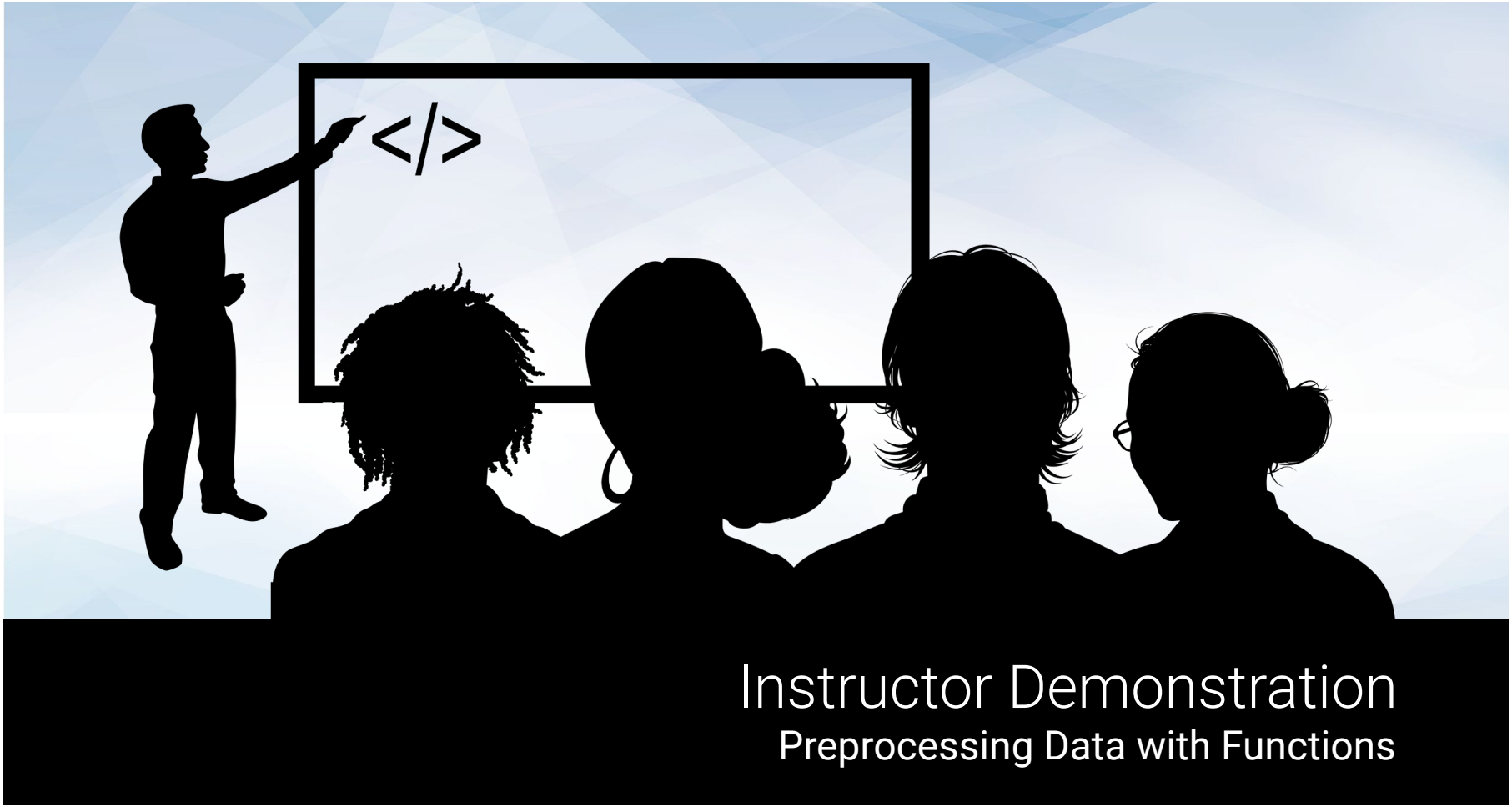
Activity: Multiple Traces

Ancient Roman gods were often counterparts to or imports of Greek gods. For example, the Greek god Zeus became the Roman god Jupiter via an etymological transformation from Zeus to Zeus Pater ("Father Zeus") to Jupiter. (Classical Latin lacked a "J" consonant.) The question is, in today's world, are these gods better known by their Roman names or Greek names?

Your task is to plot the number of search results, of both Roman and Greek names, returned for each god in order to answer this question.

- Begin by examining the data in `data.js`. Note the names of properties in each data object.
- To accomplish this task, you will need to create two traces, one for Roman gods, and another for Greek gods.
- To define the data for each plot point in a trace, use a `for loop` on the dataset. For each trace:
  - For the x-axis, provide an array of the paired string of Greek and Roman god names, e.g., Amphitrite-Salacia.
  - For the y-axis, provide an array of search results for Greek and Roman god names in their separate traces.

Let's Review

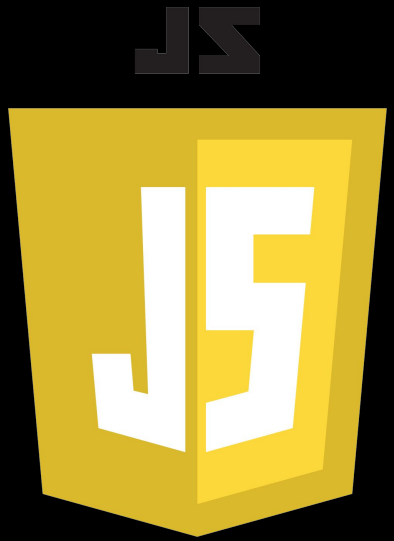


Instructor Demonstration

Preprocessing Data with Functions

Instructor Do: Preprocessing Data with Functions

Comparing functions in Python and JavaScript.



Instructor Do: Preprocessing Data with Functions

def

```
def print_hello():
    print("Hello there!")
```

function

```
function printHello() {
    console.log("Hello there!");
}
```

Instructor Do: Preprocessing Data with Functions

def

```
def addition(a, b):
    return a + b
```

function

```
function addition(a, b) {
    return a + b;
}
```

Instructor Do: Preprocessing Data with Functions

def

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

function

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

Instructor Do: Preprocessing Data with Functions

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

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

```
// 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
```

Instructor Do: Preprocessing Data with Functions

```
# Uses a previous declared function


def double_addition(c, d):
    total = addition(c, d) * 2

    return total
```

```
// 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));
```

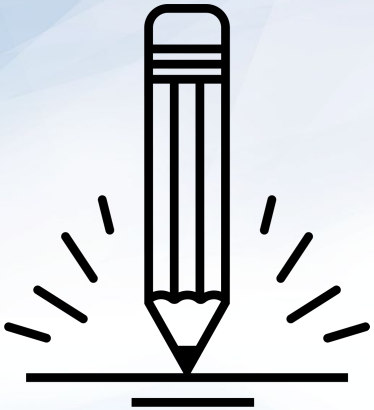
Instructor Do: Preprocessing Data with Functions

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

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

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

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



Activity: Creating Functions

In this activity, you will create functions that will calculate the mean, variance and standard deviation.

Suggested Time:
15 Minutes

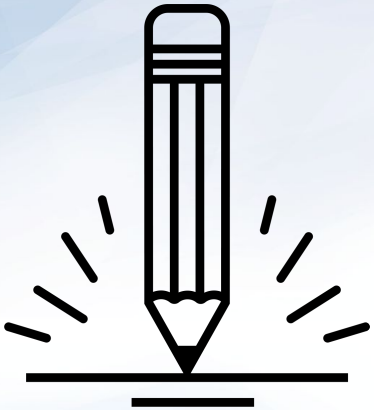
Instructions:

Activity: Creating Functions

- Using the movie array provided in the code, create functions that will return statical values from any given array of data.
  - `movieScore = [4.4, 3.3, 5.9, 8.8, 1.2, 5.2, 7.4, 7.5, 7.2, 9.7, 4.2, 6.9]`
- Create functions that will find the following:
  - Mean
  - Variance
  - Standard Variation
- 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 an 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.
  - If you want to review how to calculate variance and standard deviation, see the following pages:
    - https://stats.stackexchange.com/questions/212650/variance-explanation
    - https://www.mathsisfun.com/data/standard-deviation.html

Let's Review



Activity: Preprocessing Data for Plotly

In this activity, you will create functions that preprocess films from the Pagila database and create a bar chart of average values by age rating.

Suggested Time:
15 Minutes

Instructions:

Activity: Preprocessing Data for Plotly

- Using the films.js array of movies from the Pagila database, create a function that will calculate averages by age rating of one of the following metrics:
  - Rental rate (`rental_rate`)
  - Length (`length`)
  - Replacement cost (`replacement_cost`)
- With the calculated averages create a functions that will generate a Plotly bar chart with the average values.
- After you have the functions working try changing the metric to update the bar chart.

- **Hints:**
  - You will need to use conditionals, loops, and functions to complete this activity. You may find it helpful to review the following two activities:
    - 06-Stu_Iteration_and_Conditionals
    - 10-Stu_Creating-Functions

➔ **About the data**

- Devrim Gündüz (2021) *Pagila*. Retrieved from: https://github.com/devrimgunduz/pagila

Let's Review