Introduction 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 bar charts and line charts using Plotly.
- Use Plotly’s layout object to customize the appearance of charts.
- Annotate charts with labels, text, and hover text.

Instructor Demonstration

Creating Interactive Charts on the Web

Creating Interactive Charts on the Web

`index.html`

```
<!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

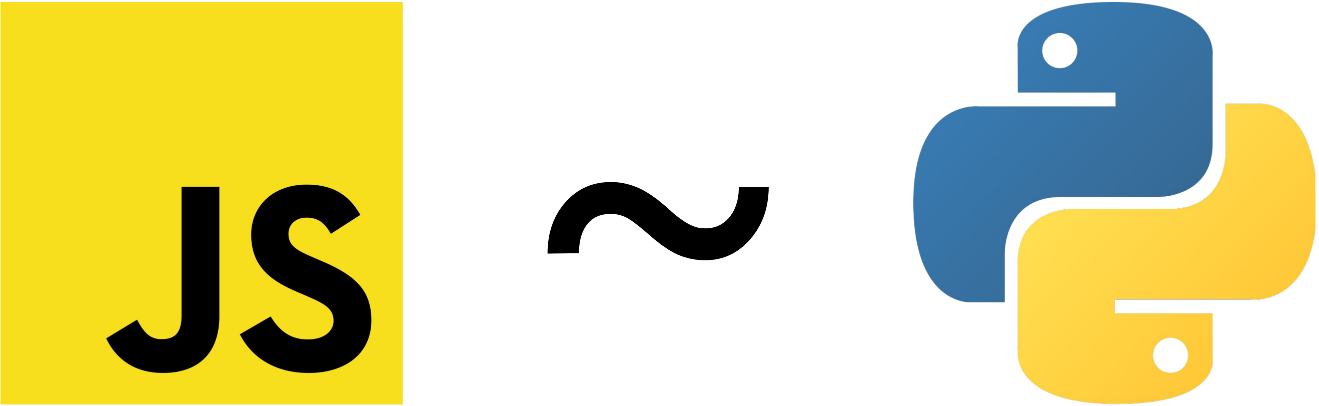
Links to an external file

Instructor Demonstration

JavaScript Variables, Objects, and Arrays

Javascript Variables, Objects, and Arrays

JavaScript and Python variables are **similar**, however...



...in JavaScript, variables must be initialized.

Javascript Variables, Objects, and Arrays

JavaScript and Python variables can be assigned to **string values**:

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

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

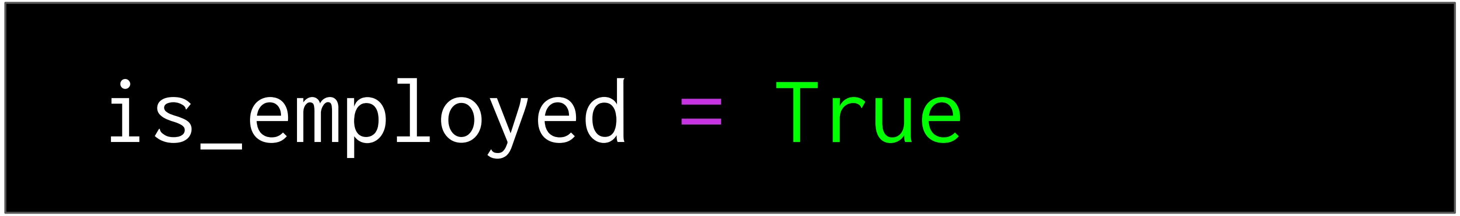
```
let <variable name> = <Value>;
```

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

Javascript Variables, Objects, and Arrays

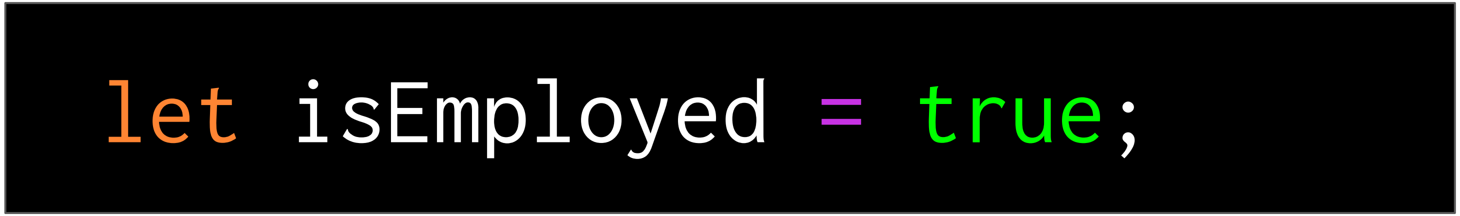
Can be assigned to **Boolean values**:

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



```
is_employed = True
```

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



```
let isEmployed = true;
```

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;
```

Javascript Variables, Objects, and Arrays

Can be assigned in expressions using **other variable**:

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

```
daily_wage = hourly_wage * 8
weekly_wage = daily_wage * 5
```

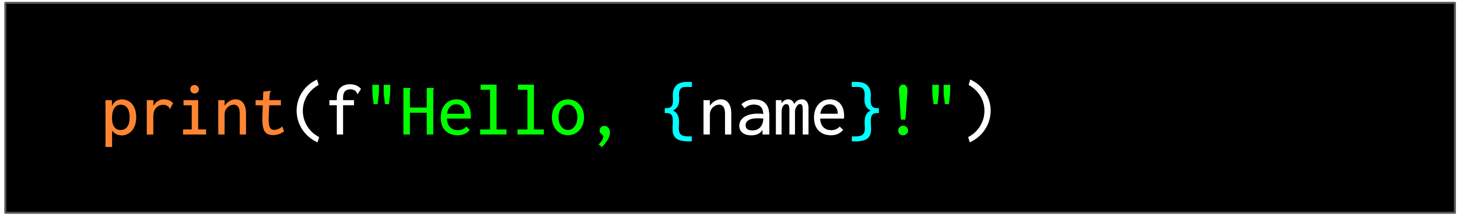
```
let <variable name> = <another variable name> ( +, -, /, *) number;
```

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

Javascript Variables, Objects, and Arrays

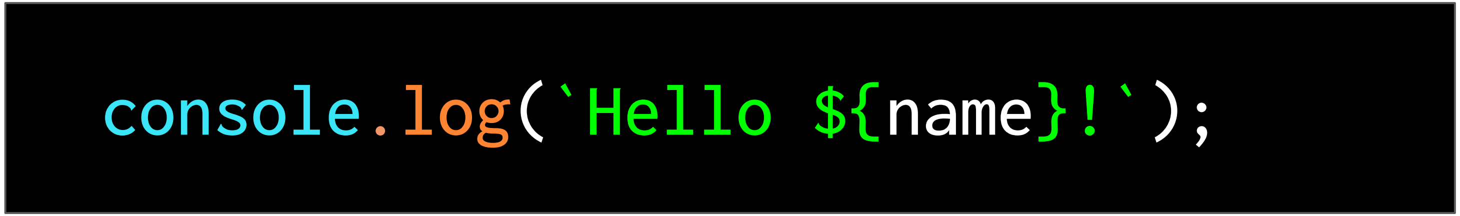
Template Literal:

```
# Python f-string
```



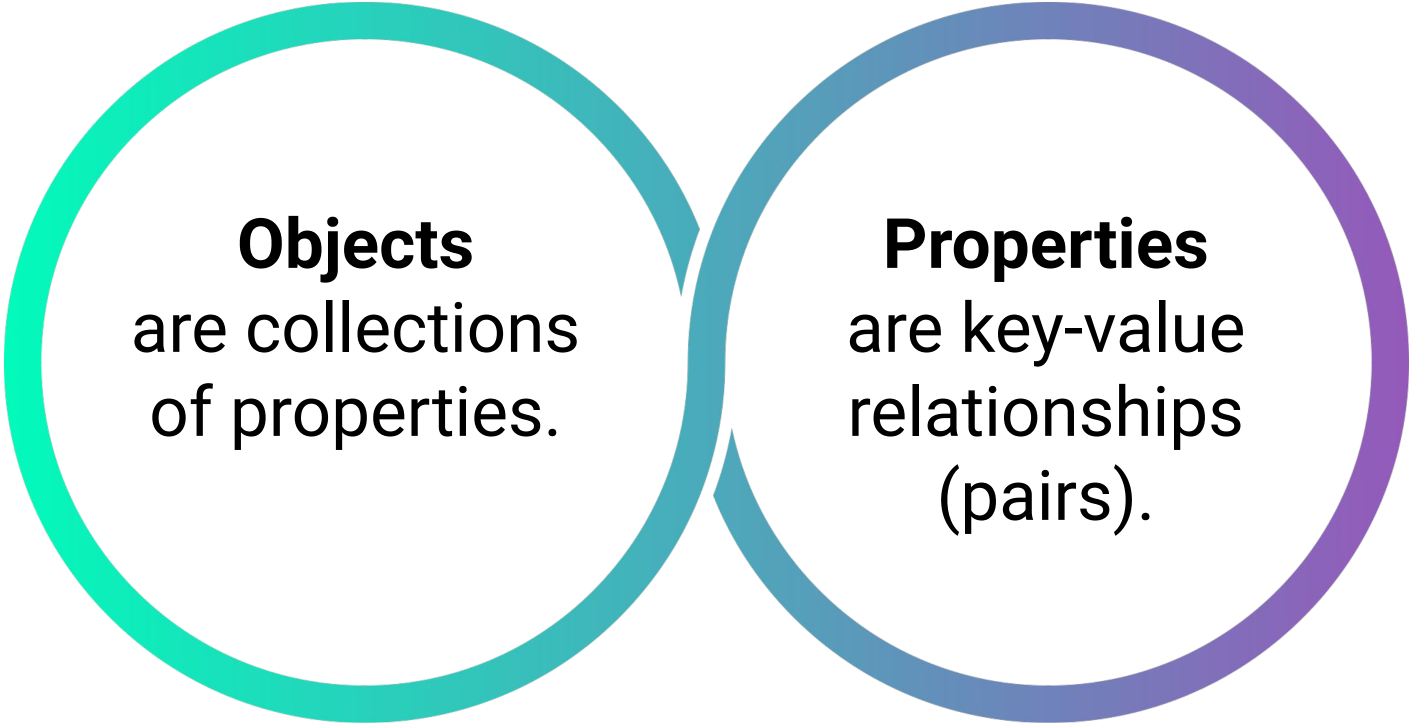
```python
print(f"Hello, {name}!")
```

```
// JavaScript template literal
```



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

Javascript Variables, Objects, and Arrays



Objects
are collections
of properties.

Properties
are key-value
relationships
(pairs).

Javascript Variables, Objects, and Arrays

JSON (**J**ava**S**cript **O**bject **N**otation) is a syntax for storing and exchanging data.
It’s similar to a Python dictionary in many ways:

{json}

- Organize information in `key` and `value` pairings.
- They are unordered.
- key is used to access the value.

There are two ways to access a property from JSON.

01 Bracket notation, similar to Python.

02 Dot notation.

JavaScript arrays are similar to Python lists.

Questions?

Activity: My Variables, Objects, and Arrays

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

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Questions?

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.

The chart will show three books you’ve read as well as the number of times you’ve read them.

Suggested Time:

5 Minutes

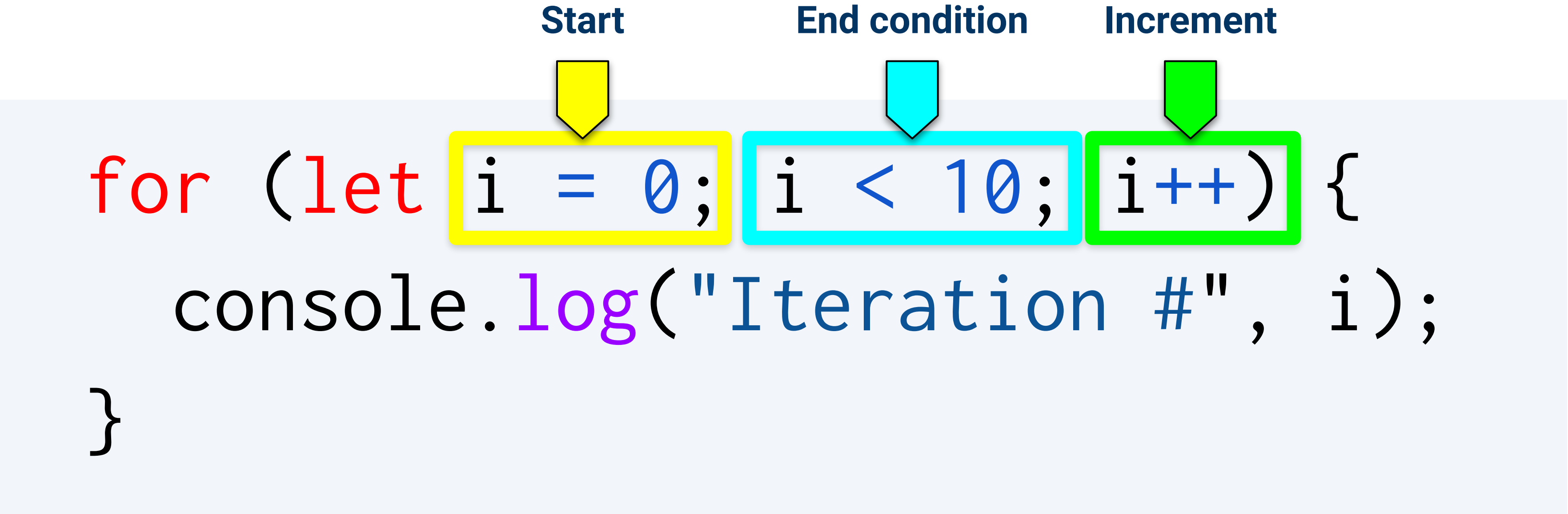
Time’s Up! Let’s Review.

Instructor Demonstration

Control Flow

Control Flow

`for` loops in JavaScript



Start

End condition

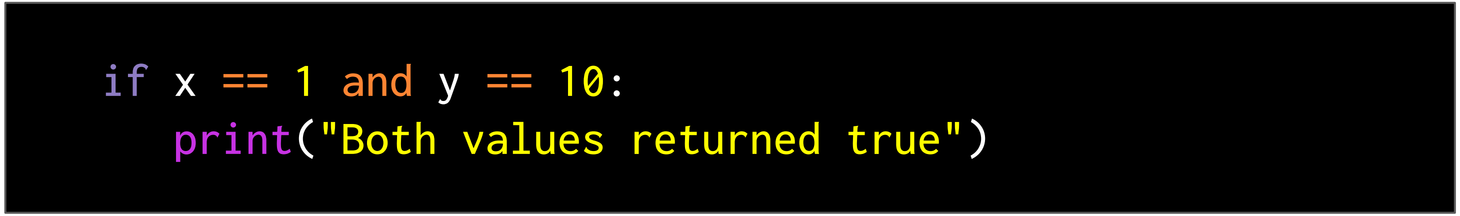
Increment

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

Control Flow

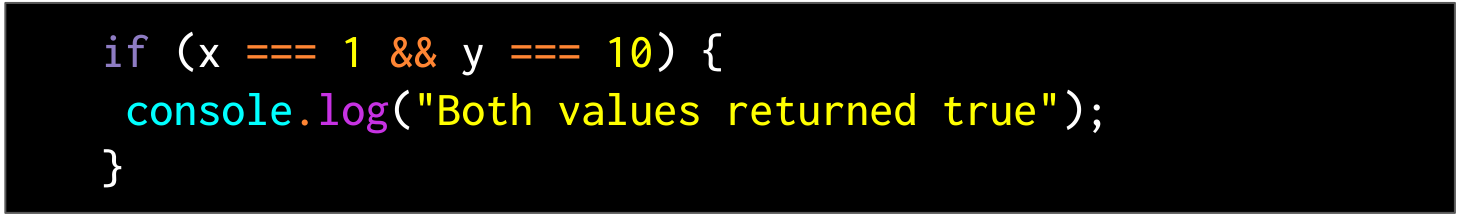
Conditionals

# and



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

// &&



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

Control Flow

Conditionals

```
# or
```

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

```
// ||
```

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

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 a movie's decade, calculate the average profit of all movies, and print out how many of the top 10 movies came from each decade.

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Break

Instructor Demonstration

Multiple Trace Charts

Questions?

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

Time’s Up! Let’s Review.

Instructor Demonstration

Preprocessing Data with Functions

Preprocessing Data with Functions

def

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

function

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

Preprocessing Data with Functions

def

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

function

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

Preprocessing Data with Functions

def

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

function

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

Preprocessing Data with Functions

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

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

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

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

Preprocessing Data with Functions

// Functions can call other

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

    return total
```

// Accepts a parameter and iterates through an array

```
function doubleAddition(c, d) {
 let total = addition(c, d) * 2;

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

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

```
let longDecimal = 112.34534454;
let rounded Decimal = Math.round(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

Time's Up! 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

Time’s Up! Let’s Review.

Questions?

The End