



# Objects, ES6, and Tables

Data Boot Camp  
Lesson 14.2





# Class Objectives

---

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



Use `forEach` and callback functions and how to use them.



Create, update, and iterate JavaScript Objects.



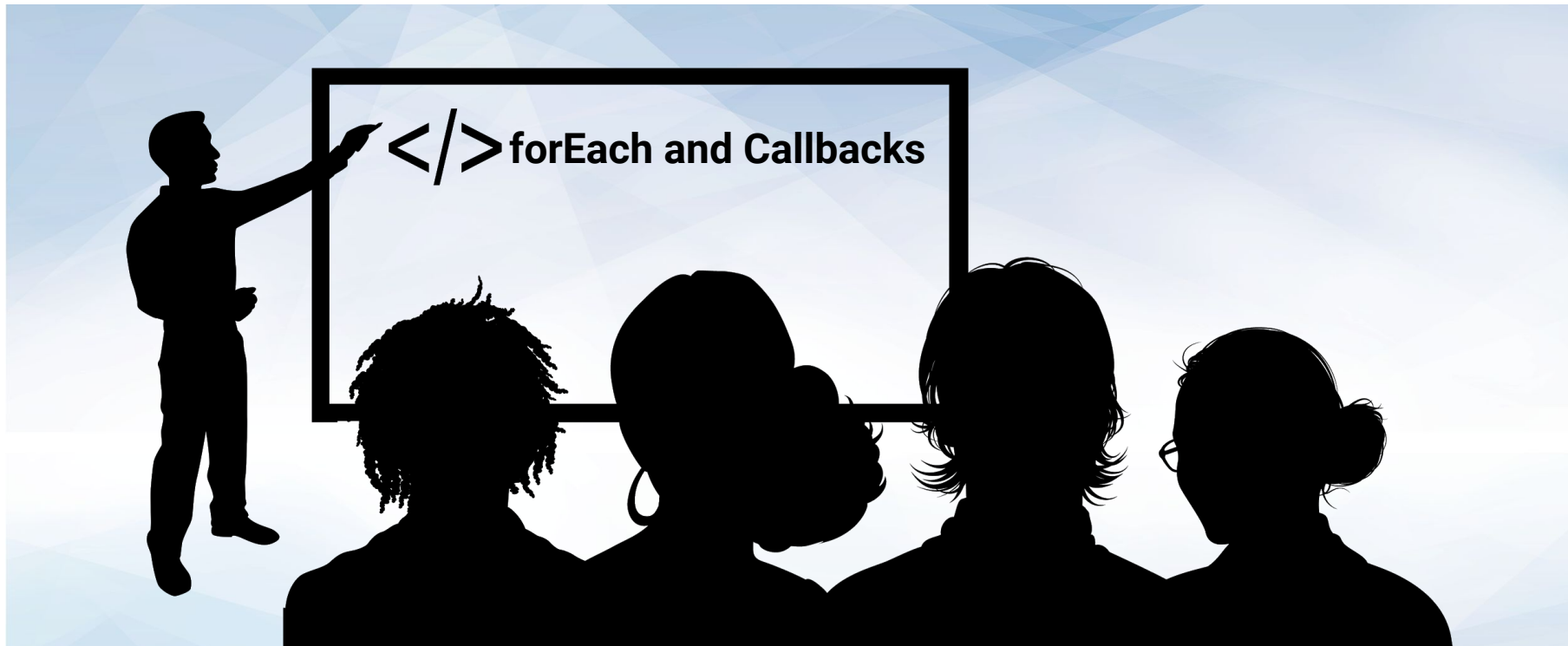
Understand how to apply `map` and `filter` to parse data.



Create and use arrow functions to simplify code.

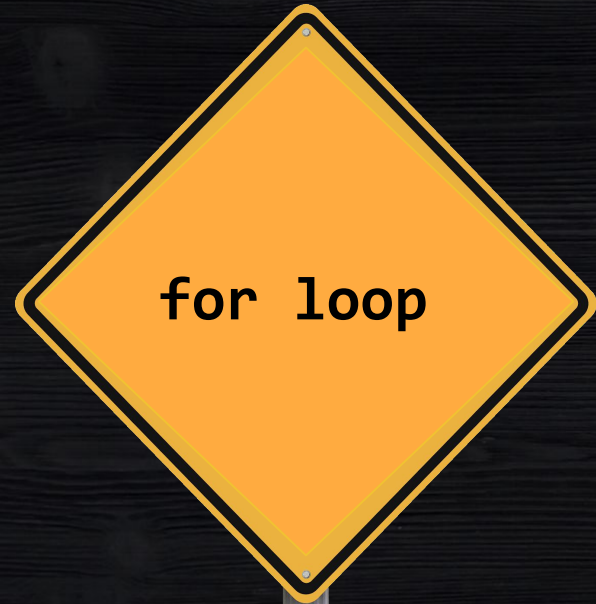


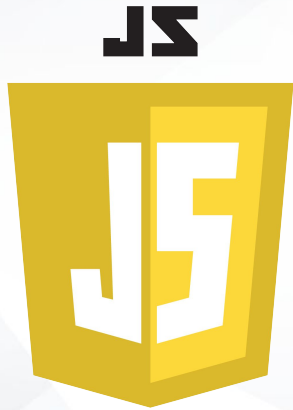
Understand the basic structure of a Bootstrap HTML table.



# Instructor Demonstration

## forEach and Callbacks





**JavaScript** is a language that will provide you different ways of doing something. For instance `forEach` and `for` loops.

# <Time to Code>







## Activity: Movie Scores - `forEach`

In this activity, you will revisit the activity Movie Scores from the previous class and refactor the code using the `forEach` method instead of a `for` loop.

**Suggested Time:**  
15 Minutes





# Activity: Movie Scores - forEach

---

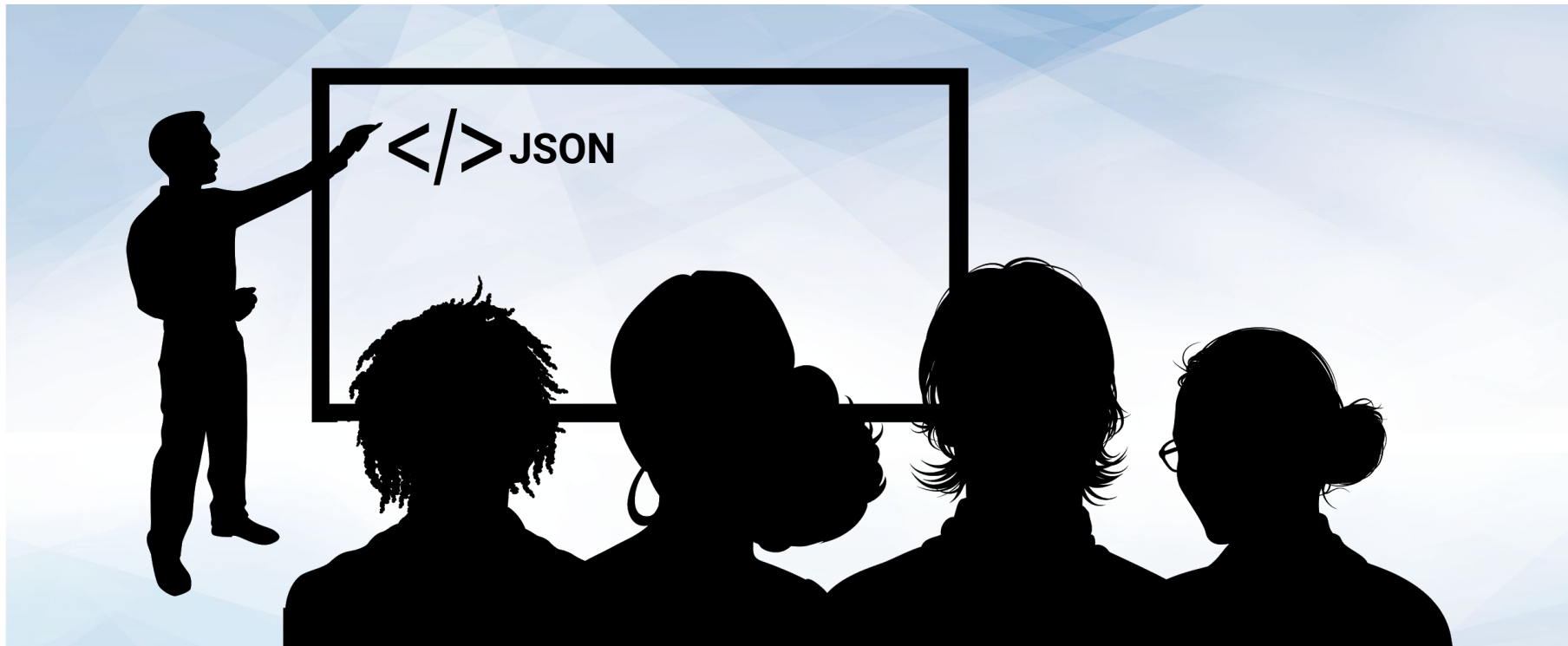
## Instructions:

- Revisit the [MovieScore](#) activity from the previous class.
- Refactor the code to use the `forEach` method instead of a `for` loop.
- **Hint:**
  - Remember that `forEach` will pass a function to each element in an array.





**Time's Up!** Let's Review.



# Instructor Demonstration

## JavaScript Objects



# What is JSON?



JavaScript Object 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.

# <Time to Code>





## Activity: Word Frequency Counter

In this activity, you will create a function in JavaScript that counts the number of occurrences of each word in a string.

**Suggested Time:**  
15 Minutes



# Activity: Word Frequency Counter

---

## Instructions:

- Create a function in JavaScript that counts the number of occurrences of each word in a string.
- The function should take in a string as its parameter.
- Use an object to hold word frequency in key-value pairs. For example, the following will be the frequency list for the string **"I yam what I yam and always will be what I yam"**

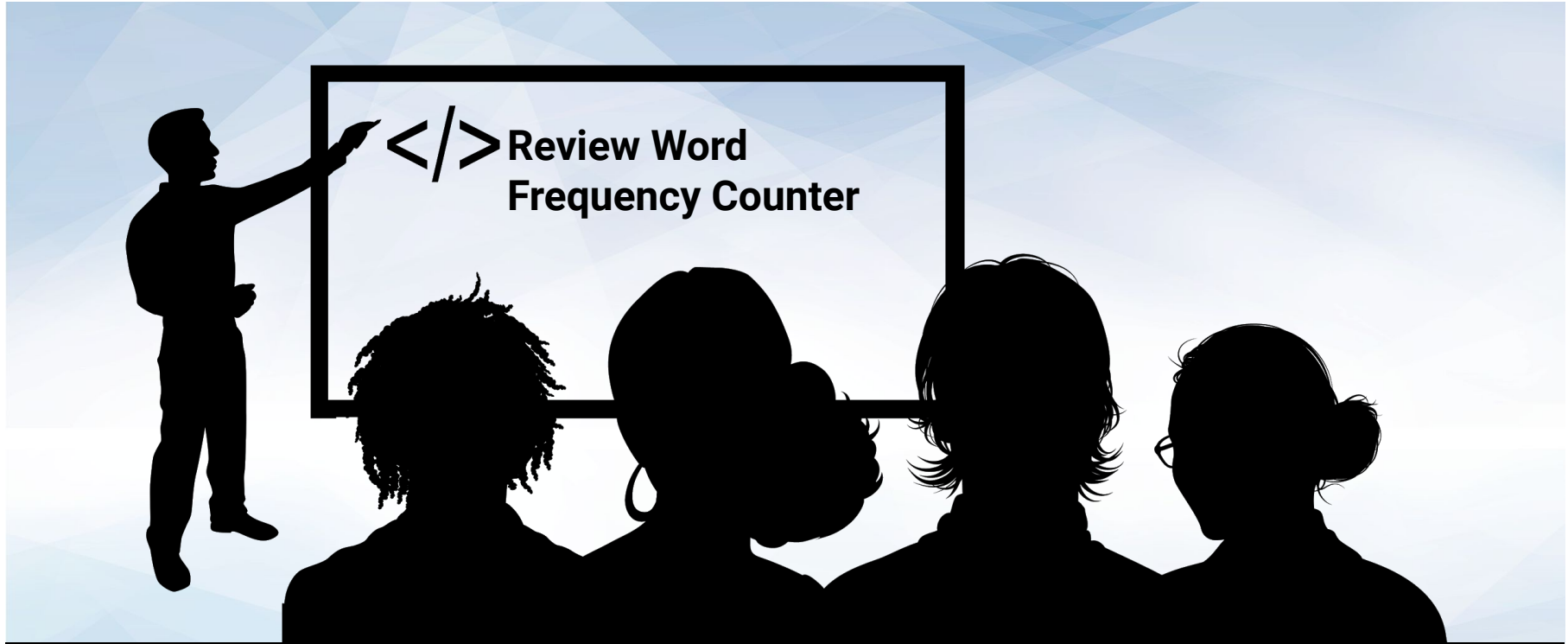
- **Hint:**



- How would you split a string into an array of words?
- Start the word frequency counter an empty object.
- How would you determine whether a word already exists in the object?

```
{  
  I: 3,  
  always: 1,  
  and: 1,  
  be: 1,  
  what: 2,  
  will: 1,  
  yam: 3  
}
```

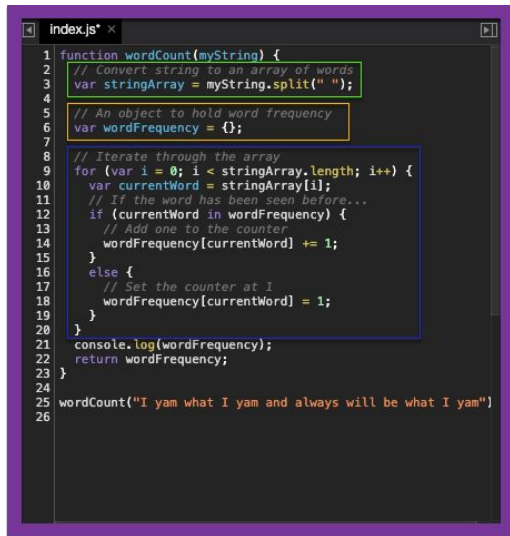
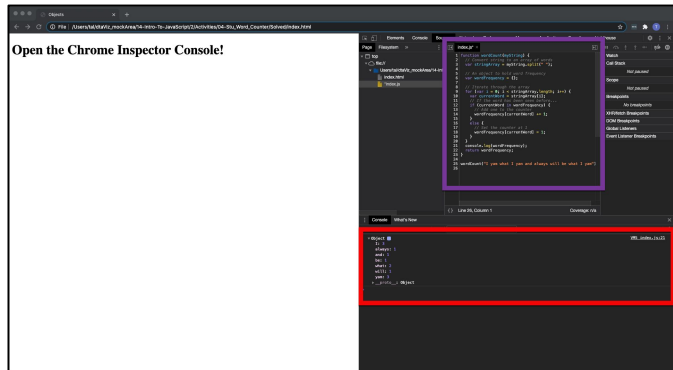




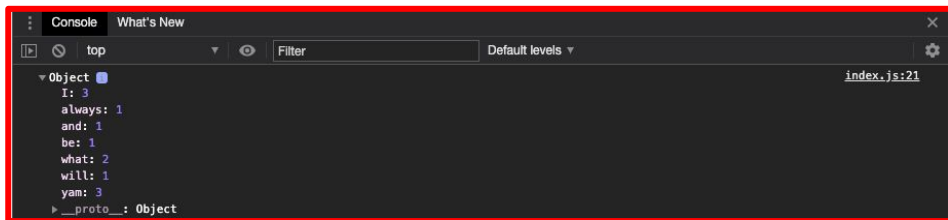
# Instructor Demonstration

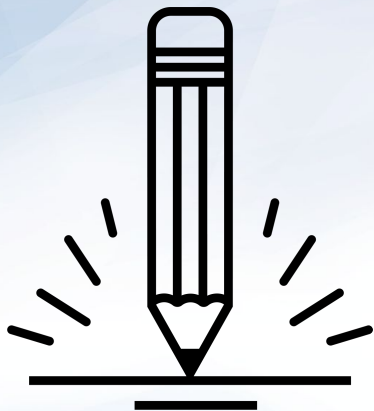
## Review Word Frequency Counter

# Review Word Frequency Counter



1. Convert string to an array of words.
2. An empty object is created to hold values.
3. Iterates through the array and increases the count by 1 for repeated words.
4. The function outcome.





## Everyone do: Map

In this activity, everyone will have the opportunity to cover the use of the `map` function.

**Suggested Time:**  
15 Minutes





# What is the `map()` method?

## `map()`

- A method that creates a new array with the results of calling a function for every element array.
- A method that calls the provided function in order and once for each element in an array.

# <Time to Code>





## Everyone do: Arrow Functions

In this activity, everyone will be presented to the `arrow functions`.

**Suggested Time:**  
15 Minutes





# What is the **arrow functions** ?



- The arrow function is an alternative way to write functions in JavaScript.
- It was introduced in ES6 and allow us to write shorter functions syntax.



# <Time to Code>



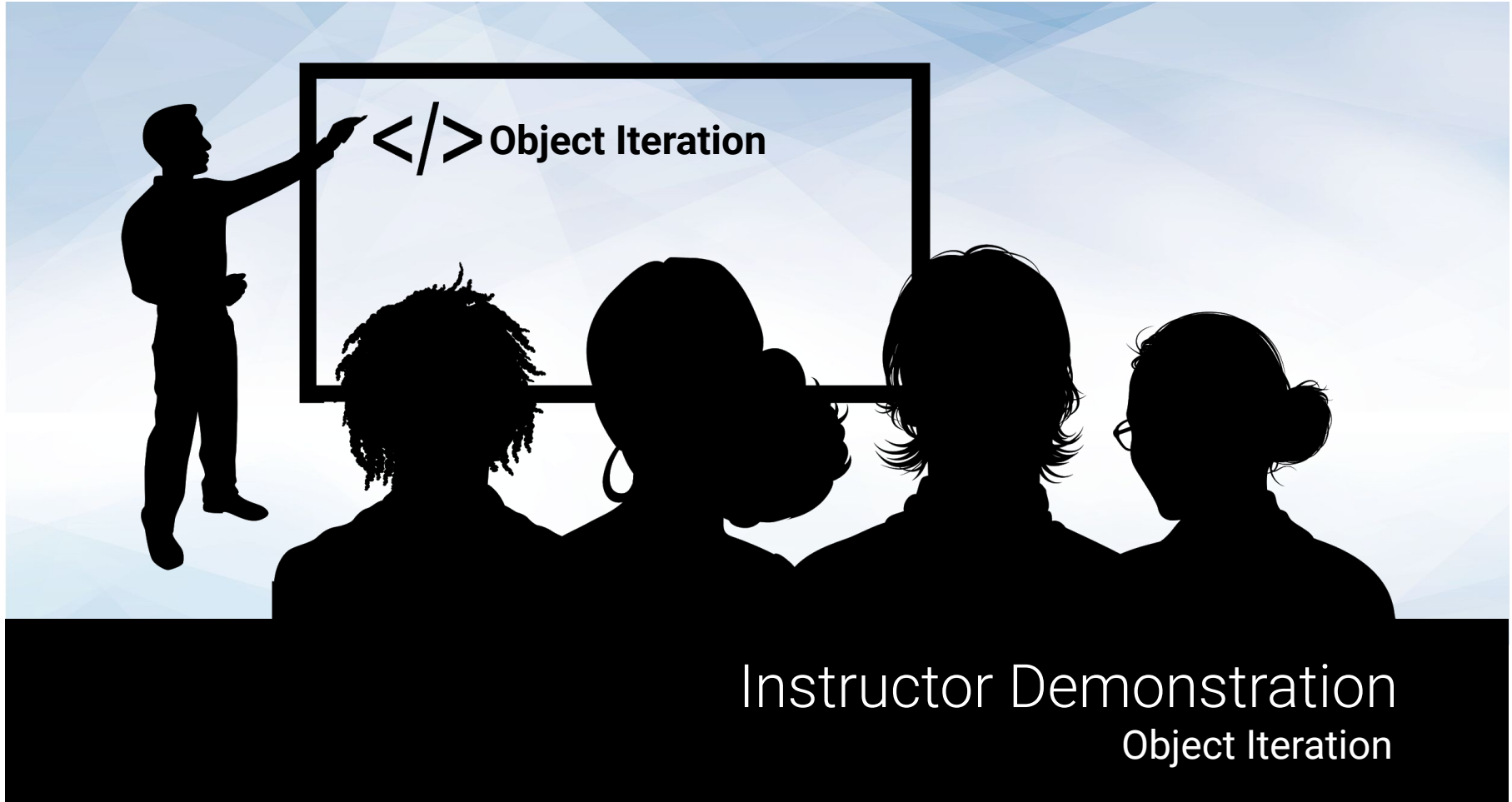




Countdown timer

**15:00**

(with alarm)



# Instructor Demonstration

## Object Iteration



What is the `forEach()` method?

`forEach()`

- A method used with `Object.keys()` that calls a function to iterate over `Object` keys and values, in order.



## Activity: Recipe Iteration

In this activity, you will practice iterating over arrays of objects.

**Suggested Time:**  
15 Minutes



# Activity: Recipe Iteration

---

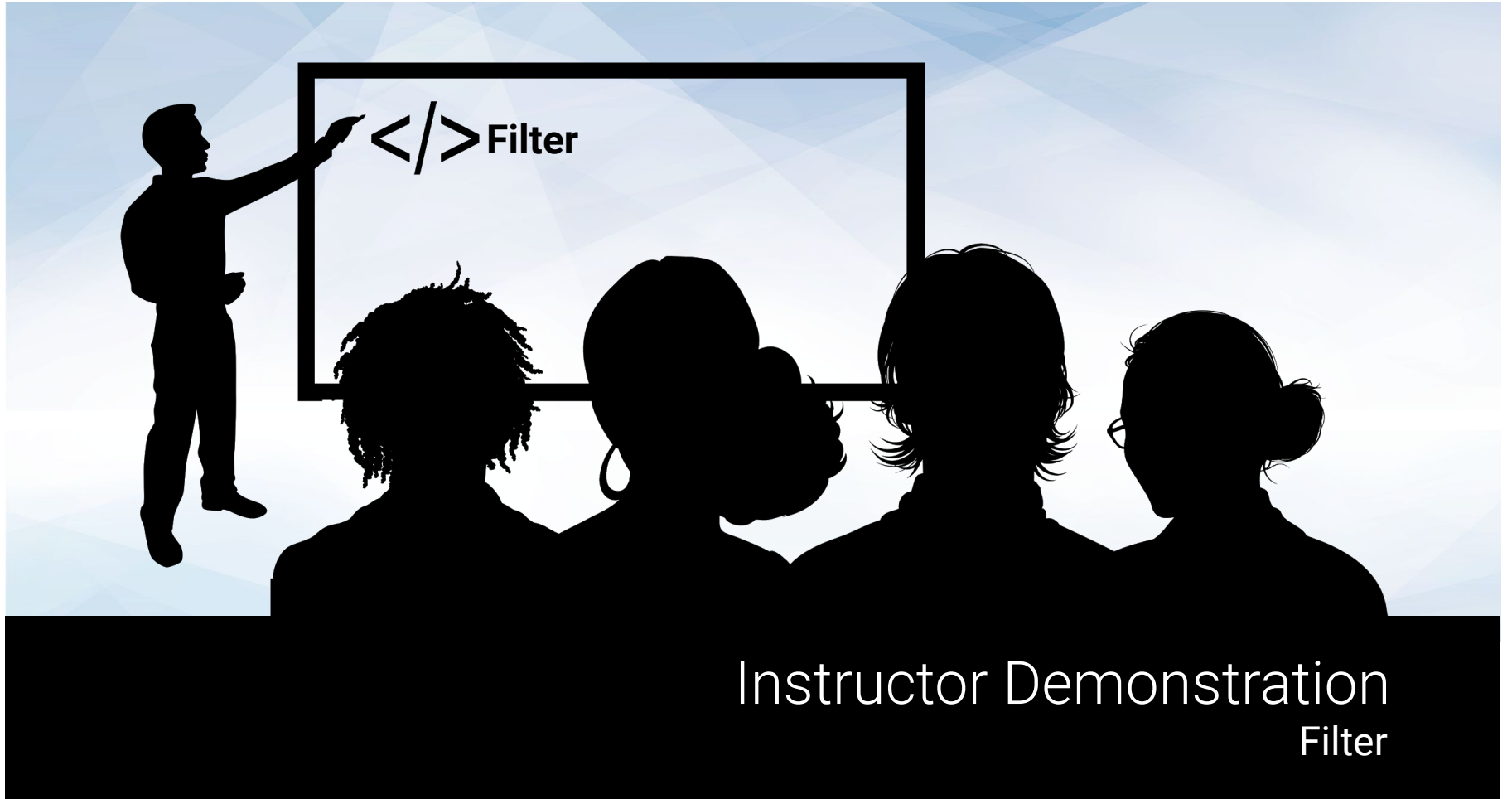
## Instructions:

- Create two empty arrays called `dishes` and `spices`.
  - Use `Object.entries` and `forEach()` to iterate over an array of recipe objects.
  - Push each dish into the `dishes` array.
  - Push each spice into the `spices` array.
  - Log each final array to the console.
- 
- **Bonus:**
    - Create both arrays using `map` instead of `forEach`.



**Time's Up!** Let's Review.





Instructor Demonstration  
Filter



What is the `filter()` method?

`filter()`

- A method that creates an array with elements that pass a test provided as a function.

```
// Create a custom filtering function  
function selectYounger(person) {  
  return person.age < 30;  
}  
  
// filter() uses the custom function as its argument  
var youngSimpsons = simpsons.filter(selectYounger);
```





## Activity: Filters

In this activity, you will use the `filter()` method to determine which players have made the team and which have not.

**Suggested Time:**  
10 Minutes



# Activity: Filters

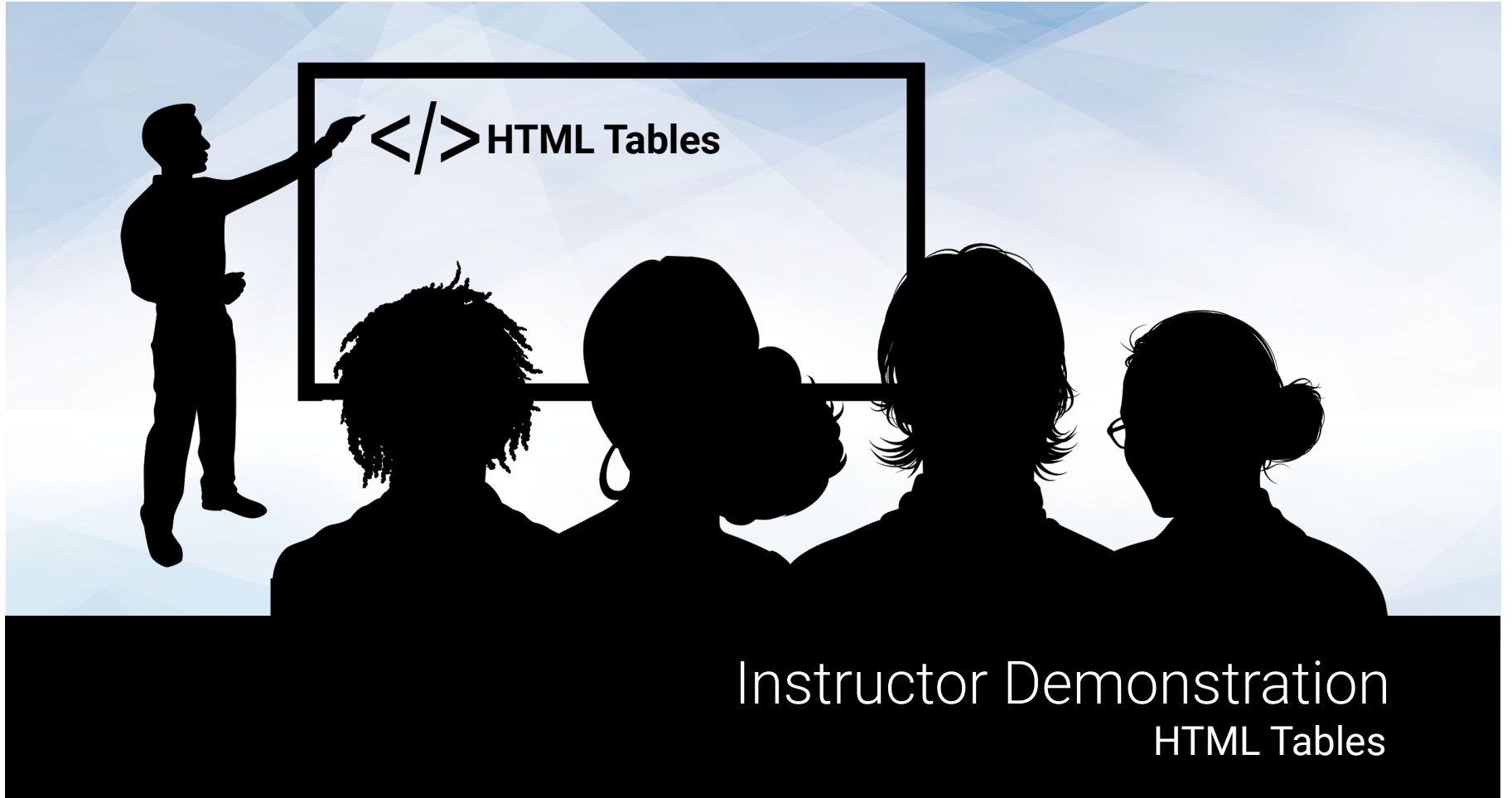
---

## Instructions:

- Given a roster of players, determine which players have made the team and which have not, using `filter()`.
- Print out all players who have made the team.
- Print out the number of players who made the team, and of players who did not.



**Time's Up!** Let's Review.

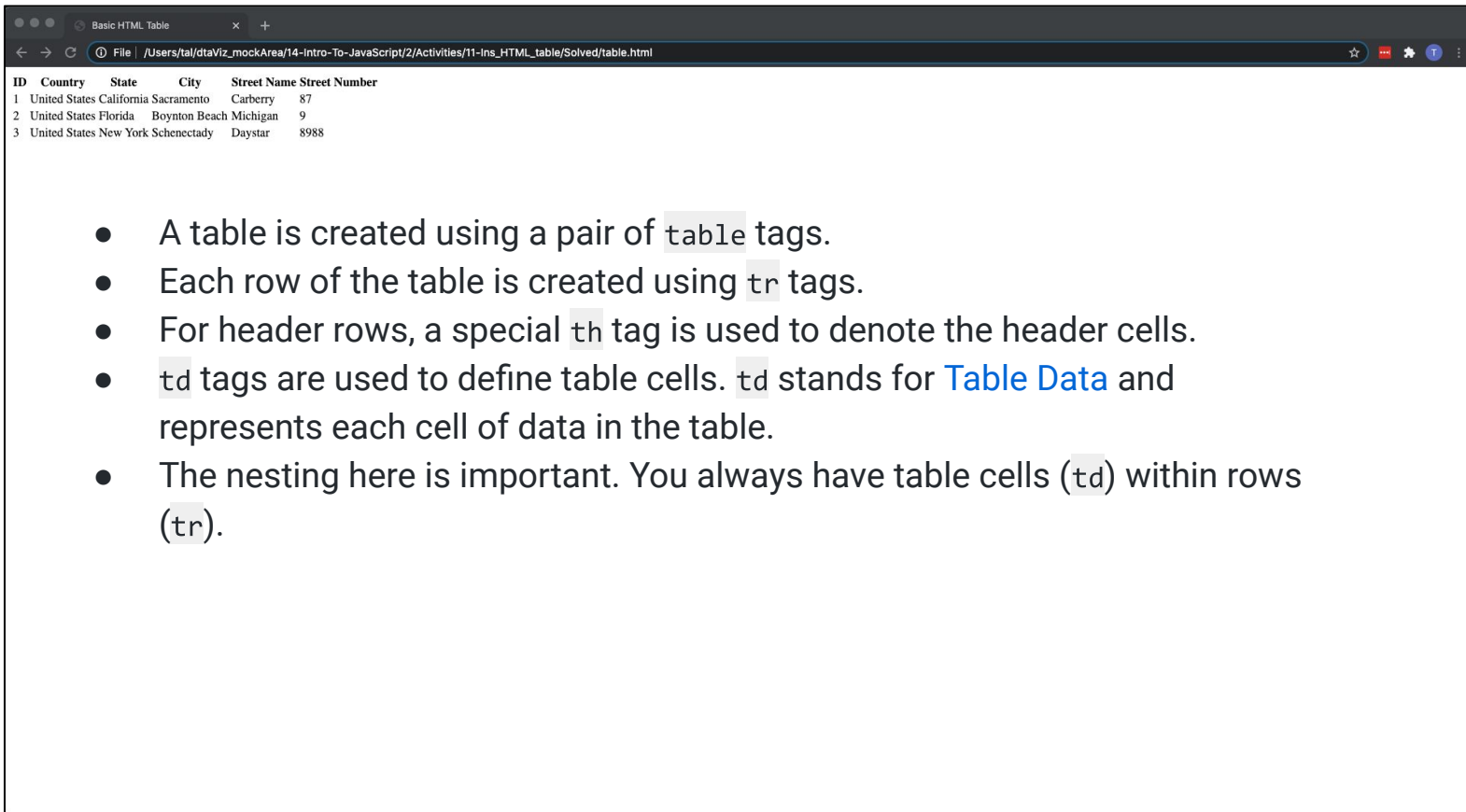


# Instructor Demonstration

## HTML Tables



# HTML Tables

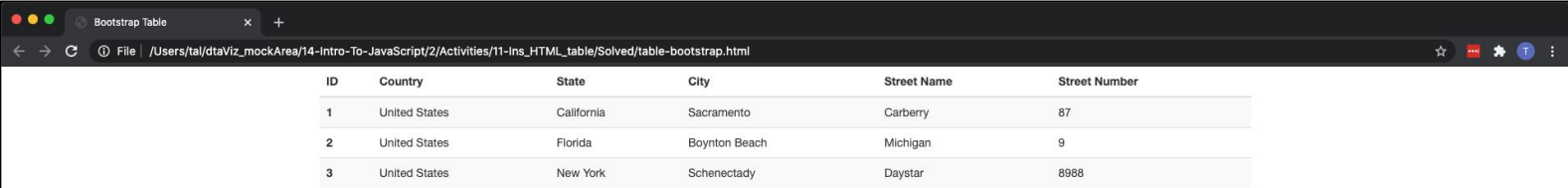


The screenshot shows a web browser window titled "Basic HTML Table". The address bar displays the file path: `/Users/tal/dtaViz_mockArea/14-Intro-To-JavaScript/2/Activities/11-Ins_HTML_table/Solved/table.html`. The browser content shows an HTML table with the following structure:

ID	Country	State	City	Street Name	Street Number
1	United States	California	Sacramento	Carberry	87
2	United States	Florida	Boynton Beach	Michigan	9
3	United States	New York	Schenectady	Daystar	8988

- A table is created using a pair of `table` tags.
- Each row of the table is created using `tr` tags.
- For header rows, a special `th` tag is used to denote the header cells.
- `td` tags are used to define table cells. `td` stands for **Table Data** and represents each cell of data in the table.
- The nesting here is important. You always have table cells (`td`) within rows (`tr`).

# HTML Tables



The screenshot shows a web browser window with the title "Bootstrap Table". The address bar displays the file path: `/Users/ta/dtaViz_mockArea/14-Intro-To-JavaScript/2/Activities/11-Ins_HTML_table/Solved/table-bootstrap.html`. The browser shows a table with the following data:

ID	Country	State	City	Street Name	Street Number
1	United States	California	Sacramento	Carberry	87
2	United States	Florida	Boynton Beach	Michigan	9
3	United States	New York	Schenectady	Daystar	8988

- Bootstrap allows you to create visually appealing data tables.
- Bootstrap adds additional table classes to the HTML tags to style the table.
- Bootstrap tables also use `thead` and `tbody` tags to separate the table header code from the table body.
- Within `thead`, the header is still defined using a `tr` row with `th` tags for header cells.
- `tbody` uses one `tr` tag per row with `td` tags per data cells.

*The  
End*