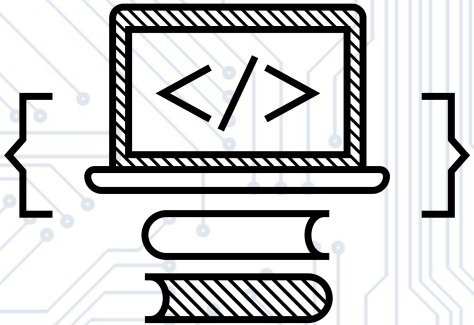
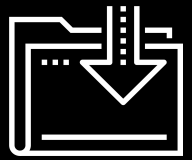


JavaScript Juggernauts

Web Development Boot Camp

Lesson 3.3



Today’s Class

Objectives

In today's class, we'll cover:

- JavaScript Recap
- JavaScript Functions & Objects
- Browser Window & Scope

JavaScript Recap

JavaScript Definition

JavaScript is one of the three fundamental programming languages of the modern web (the others are HTML and CSS).

HTML	CSS	JavaScript
Used to write content.	Used to format content.	Used to create dynamic web applications that take in user inputs, change what's displayed to users, animate elements, and much more.

Variable Basics: Syntax

Var Keyword	Variable name	Assignment	Value	Termination
`var`	`name`	`=`	`"Snow White"`	`;`

Be sure to notice the quotes (""),
which convey that Snow White is a string.

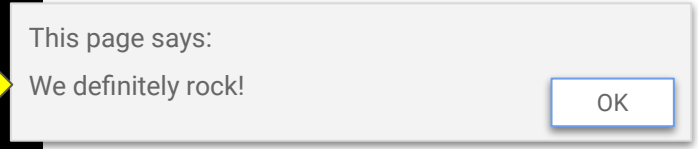
Alerts, Prompts, Confirms

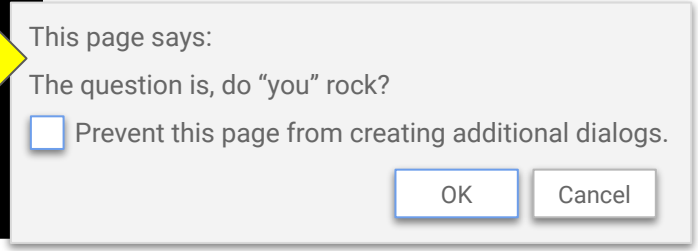
Alerts, prompts, and confirms create a popup in the browser when run.
These are also useful for development and debugging.

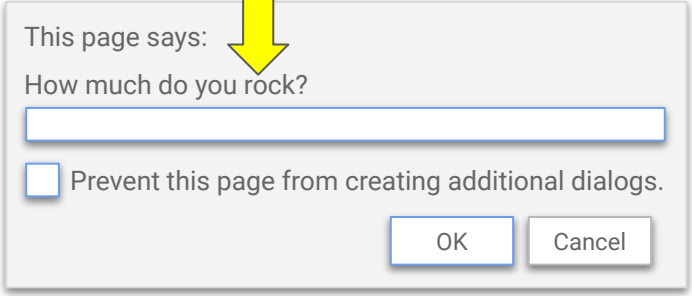
```
// Alert
alert("We definitely rock!");

// Confirm
var doYouRock = confirm("The question is, do *you* rock?");

// Prompt
var howMuchRock = prompt("How much do you rock?");
```







Console.log

`console.log` is a quick expression that prints content to the debugger—very useful during development and debugging!

```
var quick = "Fox";
var slow = "Turtle";
var numbers = 121;

// The console.log() method is used to display data in the the browser's console.
// We can log strings, variables, and even equations.

console.log("Teacher");
console.log(quick);
console.log(slow);
console.log(numbers + 15);
```

Arrays

Arrays are a type of variable that are *collections*.

These collections can be made up of strings, numbers, Booleans, other arrays, objects … anything.

Each element of the array is marked by an index. Indexes always start with `0`.

```
var nickCharacters = ["Tommy", "Doug", "Oblina"];

var diceNumbers = [1, 2, 3, 4, 5, 6,];

var mixedArray = ["Zoo", 12, "Carrot", 3];
```

Arrays: Indices

To recover the value at any specific index, include the name of the array with a square bracket `[ ]` and inside the bracket is the element's index.

You can easily grab the number of elements in the array using the method `array.length`.

```
// Our array of zoo animals.
var zooAnimals = ["Zebra", "Rhino", "Giraffe", "Owl"];

// Prints 4 to the console because there are 4 items in our zooAnimals array.
console.log(zooAnimals.length);

// Prints Rhino to the console. Remember, the first item in an array has an index position of 0!
console.log(zooAnimals[1]);

// Prints undefined...because the last index ("Owl") is 3.
console.log(zooAnimals[4]);
```

If/Else Statements Are Critical

Each statement is composed of an if, else-if, or else (keyword), a condition, and the resulting code in { } curly brackets.

```
// If the user likes sushi (confirmSushi === true), we run the following block of code.
if (confirmSushi) {
  alert("You like " + sushiType + "!");
}
// If the user likes ginger tea (confirmGingerTea === true), we run the following block of code.
else if (confirmGingerTea) {
  alert("You like ginger tea!!");
}
// If neither of the previous condition were true, we run the following block of code.
else {
  alert("You don’t like sushi or ginger tea.");
```

for Loops

for loops are **critical** in programming. We use them to run **repeated blocks of code** over a set period.

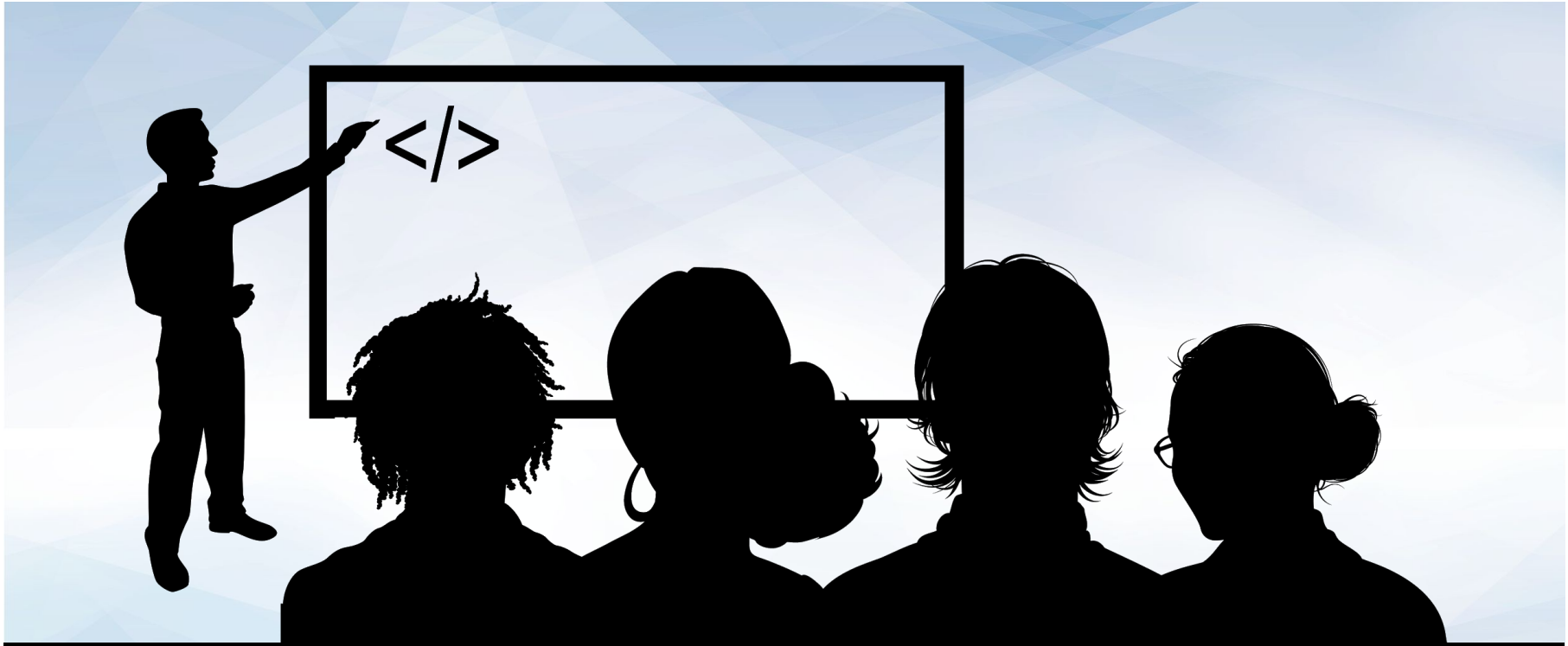
Each for loop is composed of a:

- Variable declaration or counter (iterator)
- Loop condition
- Iteration (addition)

```
// Start with an Array.
var vegetables = ["Carrots", "Peas", "Lettuce", "Tomatoes"];

// Loops through each index of the Array.
for (var i = 0; i < vegetables.length; i++) {
  console.log("I love " + vegetables[i]);
}
```

JavaScript Functions



Instructor Demonstration

Logging: No Functions

Mondo Repetitive

Who wants to maintain this?

Hint: No one.

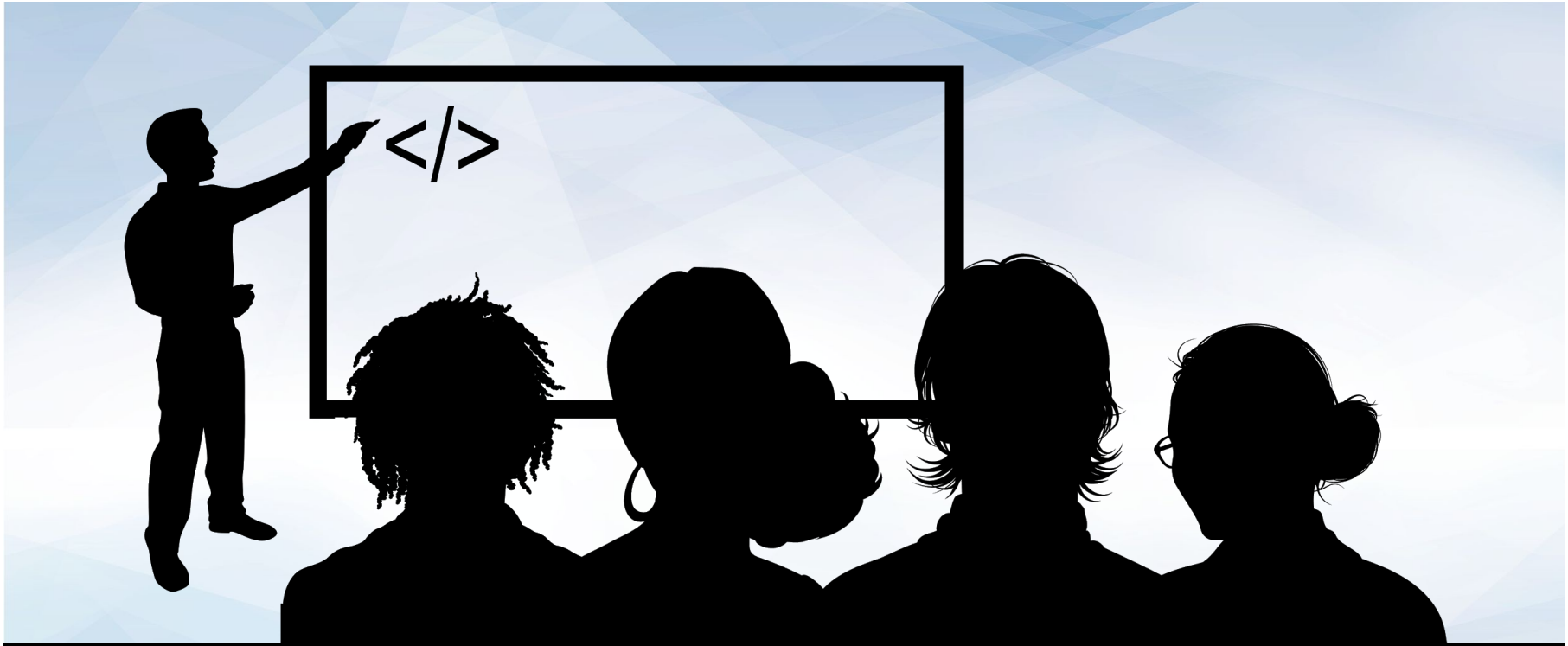
```javascript
// For Loop for Brands
for (var i = 0; i < brands.length; i++) {
  console.log(brands[i]);
}
console.log("---------");

// For Loop for Heroes
for (var i = 0; i < heroes.length; i++) {
  console.log(heroes[i]);
}
console.log("---------");

// For Loop for booksOnMyShelf
for (var i = 0; i < booksOnMyShelf.length; i++) {
  console.log(booksOnMyShelf[i]);
}
console.log("---------");

// For Loop for thingsInFrontOfMe
for (var i = 0; i < thingsInFrontOfMe.length; i++) {
  console.log(thingsInFrontOfMe[i]);
}
console.log("---------");

// For Loop for howIFeel
for (var i = 0; i < howIFeel.length; i++) {
  console.log(howIFeel[i]);
}
console.log("---------");
```



Instructor Demonstration

Logging: With Functions

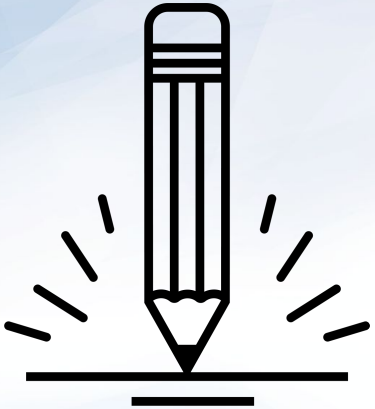
Much Better with Functions!

Squeaky clean code. Minimal repetition.

```
// Here we create a "Function" that allows us to "call" (run) the loop for any array we wish.
// We pass in an array as an "argument".
function consoleInside(arr) {

  // We then loop through the selected array.
  for (var i = 0; i < arr.length; i++) {

    // Each time we print the value inside the array.
    console.log(arr[i]);
  }
  console.log("---------");
}
```



Breakout Activity:
My First Functions

`27-MyFirstFunctions`

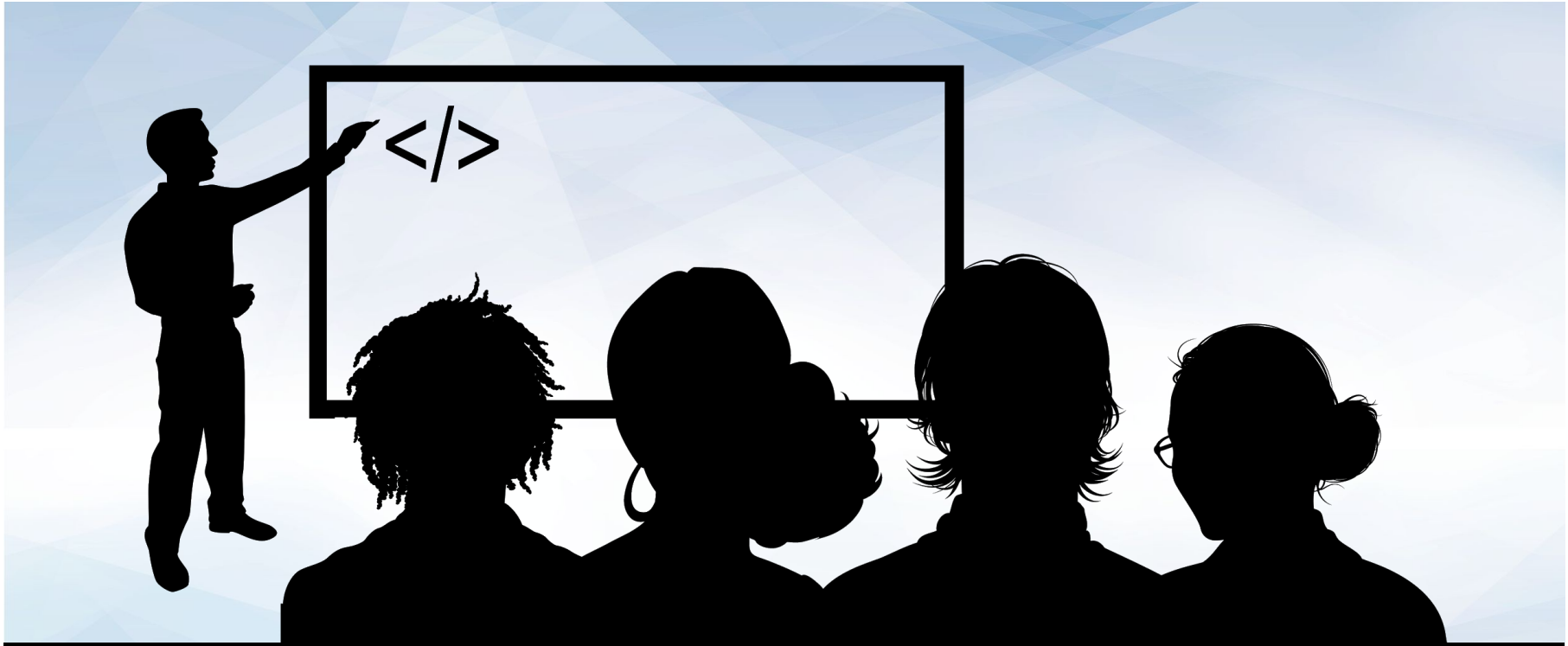
Breakout Activity: My First Functions

Working in breakout rooms, fill in the missing functions and function calls in `27-MyFirstFunctions`.

Note: Try to finish all four functions if you can, but don't worry if you only get one or two. The important thing is that you completely finish at least one function.

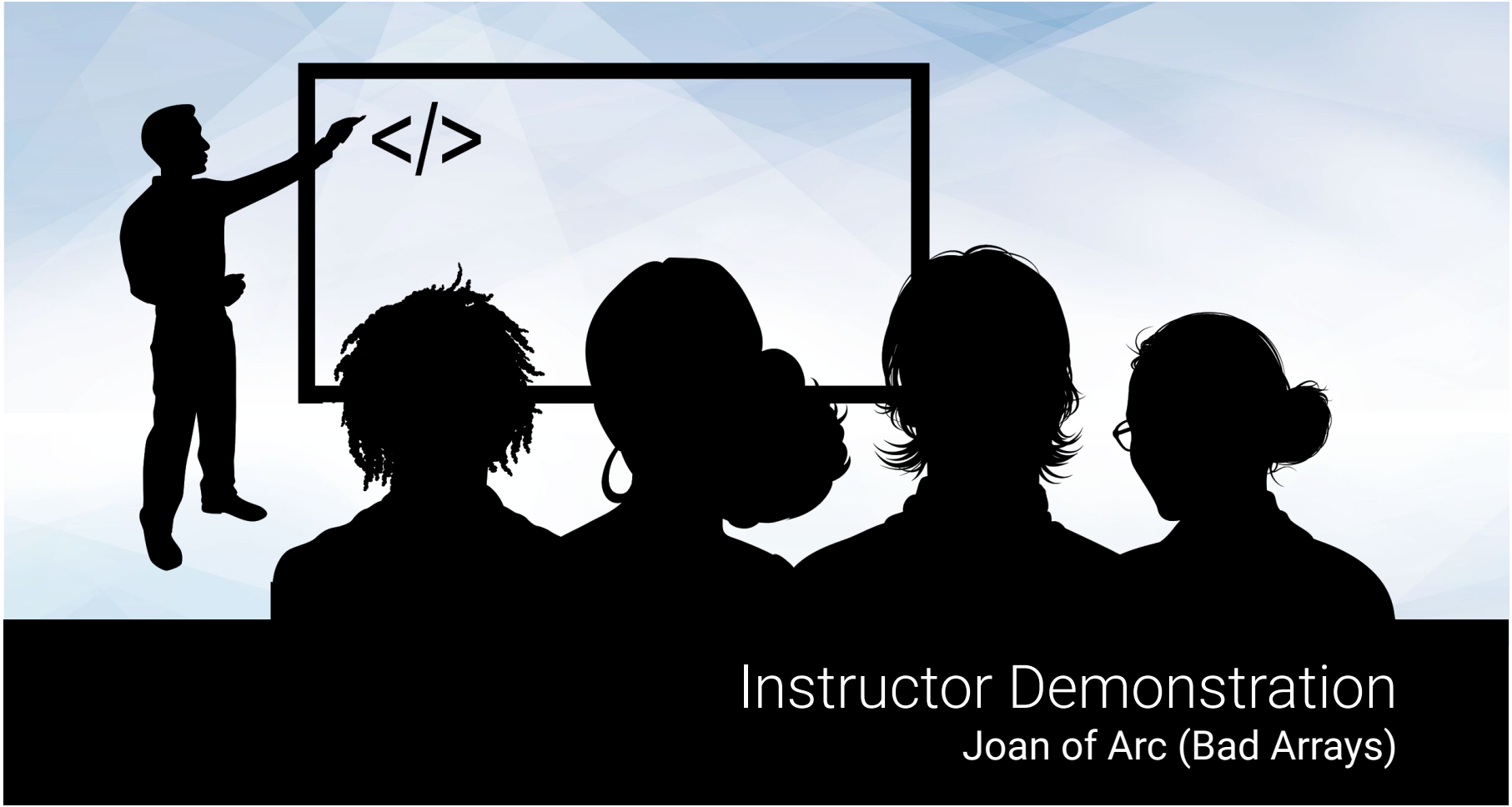
Suggested Time: 20 minutes

JavaScript Objects



Instructor Demonstration

Good Arrays

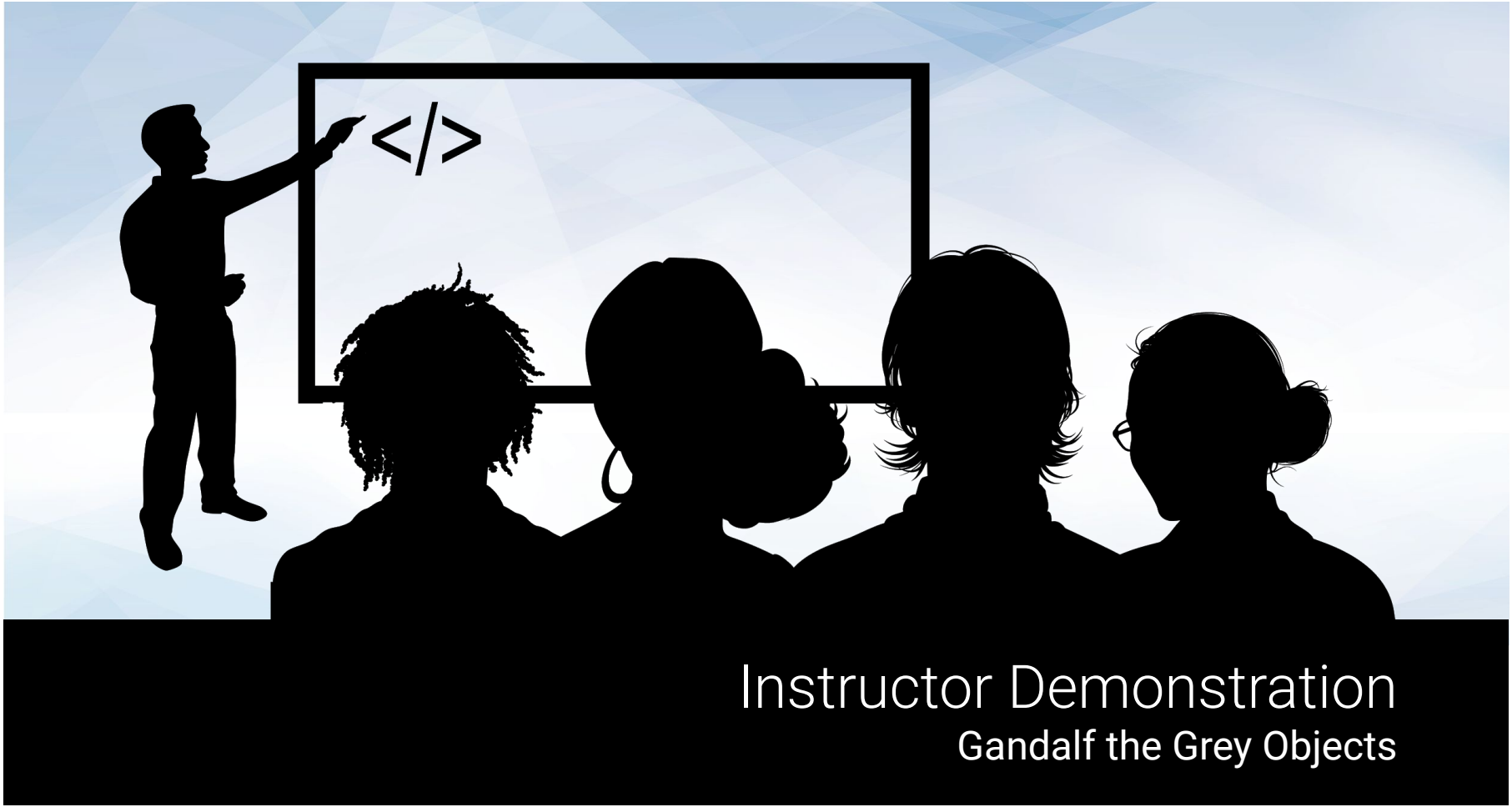

Instructor Demonstration
Joan of Arc (Bad Arrays)

Associated Data ==/== Arrays

Relating two separate arrays is not fun.

```
var joanOfArcInfoParts = ["Real Name", "Grew Up Where", "Known For", "Scars", "Symbolism"];


var joanOfArcInfoValues = ["Jehanne la Pucelle.", "Domremy, a village in northeastern France.",
  "Peasant girl, daughter of a farmer, who rose to become Commander of the French army.",
  "Took an arrow to the shoulder and a crossbow bolt to the thigh while trying to liberate Paris.",
  "Stands for French unity and nationalism."];
```


Instructor Demonstration
Gandalf the Grey Objects

Gandalf: The Object

Gandalf's **properties** and **values** are associated in object form, making it easy to recall specific data.

```
11  var gandalf = {
12    "real name": "Gandalf",
13    "age (est)": 11000,
14    "race": "Maia",
15    "haveRetirementPlan": true,
16    "aliases": [
17      "Greyhame",
18      "Stormcrow",
19      "Mithrandir",
20      "Gandalf the Grey",
21      "Gandalf the White"
22    ]
23  }
24
25  // Object properties can be accessed with "bracket notation"
26  alert("My name is " + gandalf["real name"]);
27
28  // Or with "dot notation" if the property has no spaces
29  if (gandalf.haveRetirementPlan) {
30
31    // Or with a variable that matches the name of the property
32    var ageProperty = "age (est)";
33    var years = gandalf[ageProperty];
34    alert("My 401k has been gathering interest for " + years + " years!");
35  }
```

Objects Visualized

This is Gandalf. According to code, Gandalf is an **object**.

var gandalf	=	{

“real name”	:	“Gandalf”	,
“age (est)”	:	11000	,
“race”	:	“Maia”	

}

Objects Visualized

These are Gandalf's **properties** (like descriptors).

var gandalf	=	{

"real name"	:	"Gandalf"	,
"age (est)"	:	11000	,
"race"	:	"Maia"	

}

Objects Visualized

These are the **values** of Gandalf's properties.

var gandalf	=	{

"real name"	:	"Gandalf"	,
"age (est)"	:	11000	,
"race"	:	"Maia"	

}

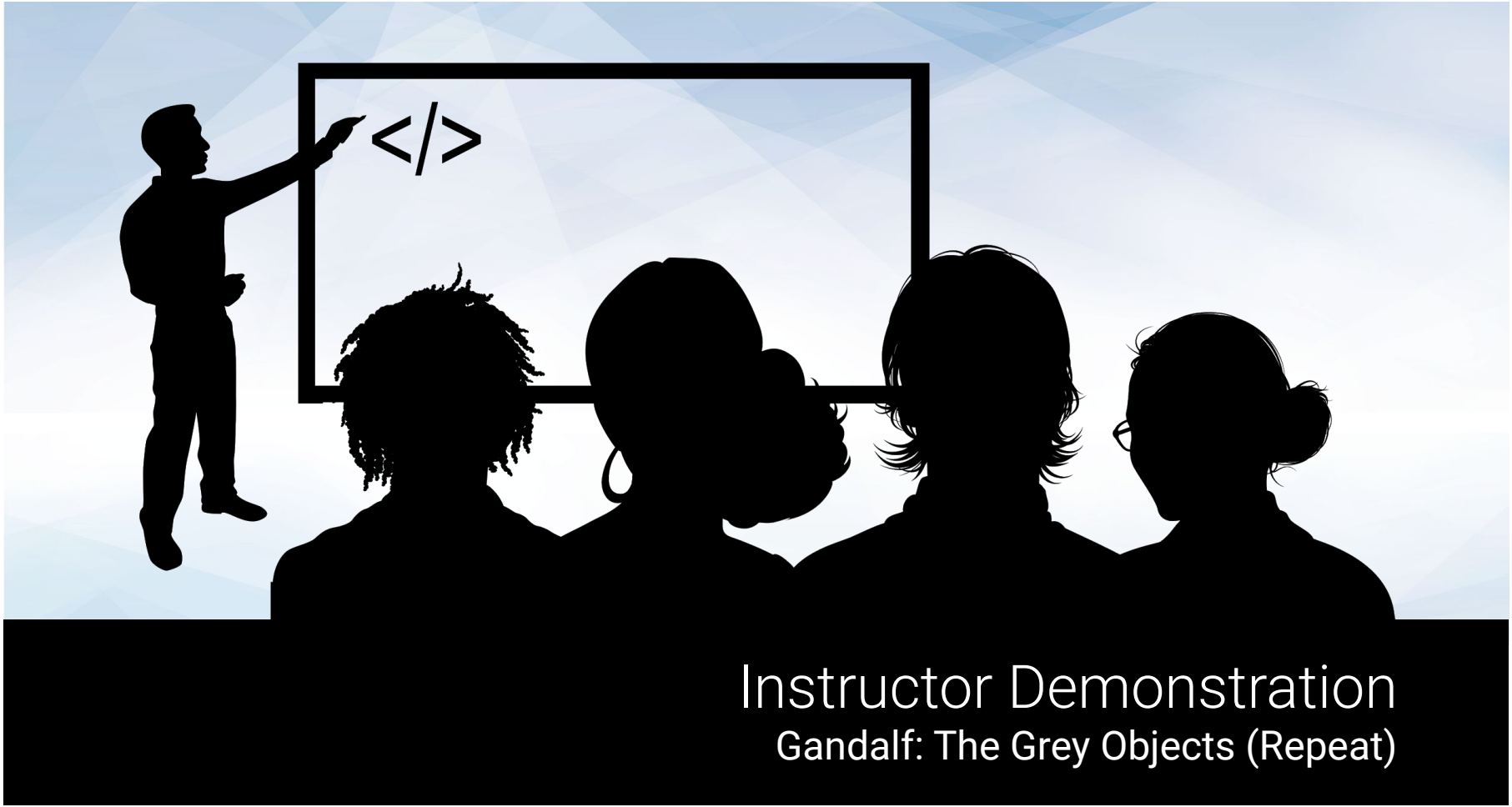
Objects Visualized

Thus: `gandalf["race"] = "Maia`

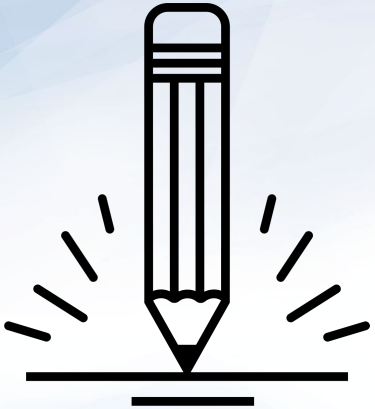
var gandalf	=	{

"real name"	:	"Gandalf"	,
"age (est)"	:	11000	,
"race"	:	"Maia"	

}


Instructor Demonstration
Gandalf: The Grey Objects (Repeat)


Break



Breakout Activity:

Basic Objects

`31-MyFirstObject`

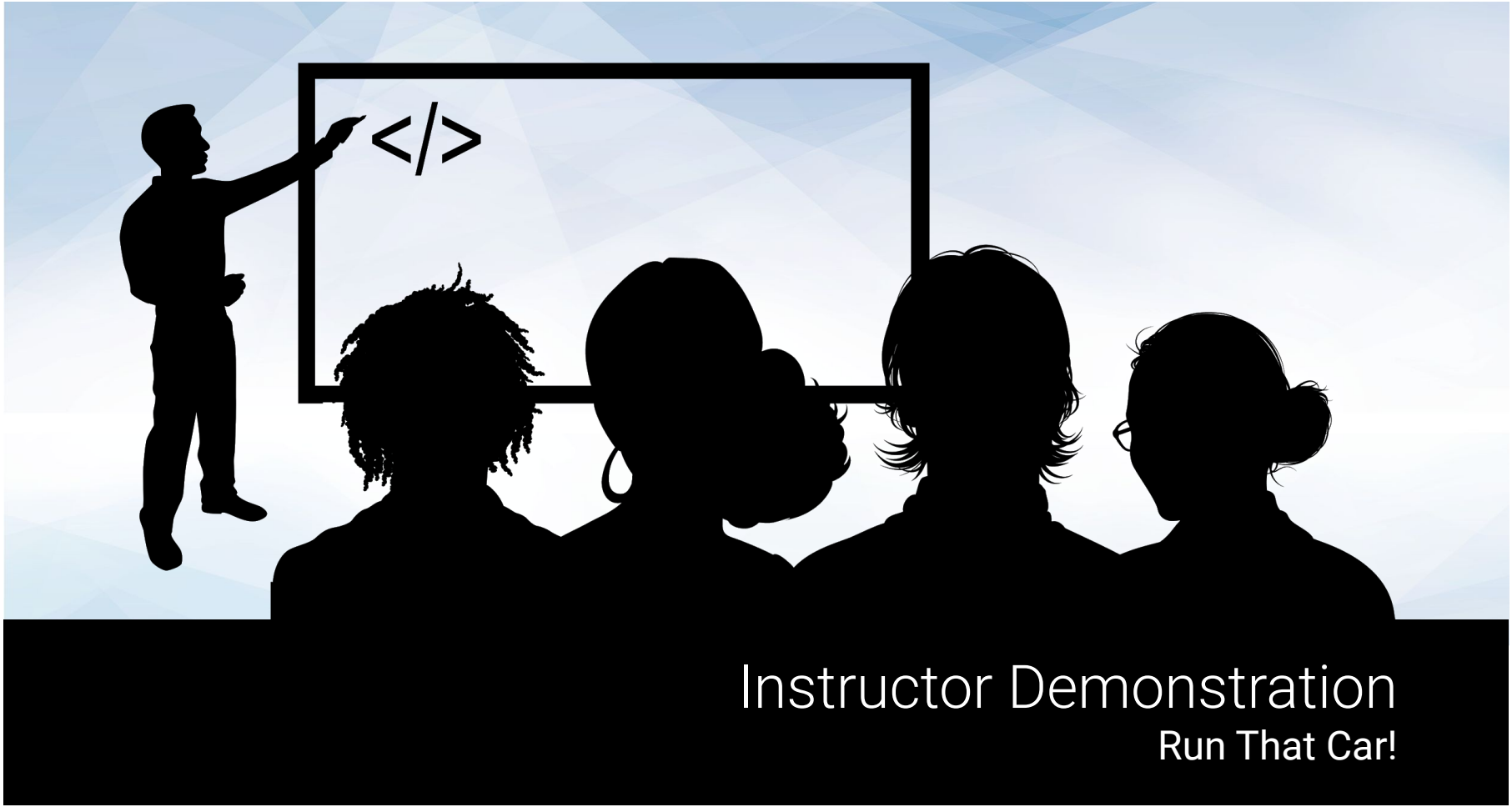
Group Activity: Basic Objects

With a partner, spend a few minutes studying the code in `31-MyFirstObject`.

Then below each comment, write code to log the relevant information about the provided `car` object.

Bonus: If you finish early, create a new object of your own. Slack out a snippet of the code to the class when you are done. Be creative!

Suggested Time: 15 minutes


Instructor Demonstration
Run That Car!

Code Challenge:

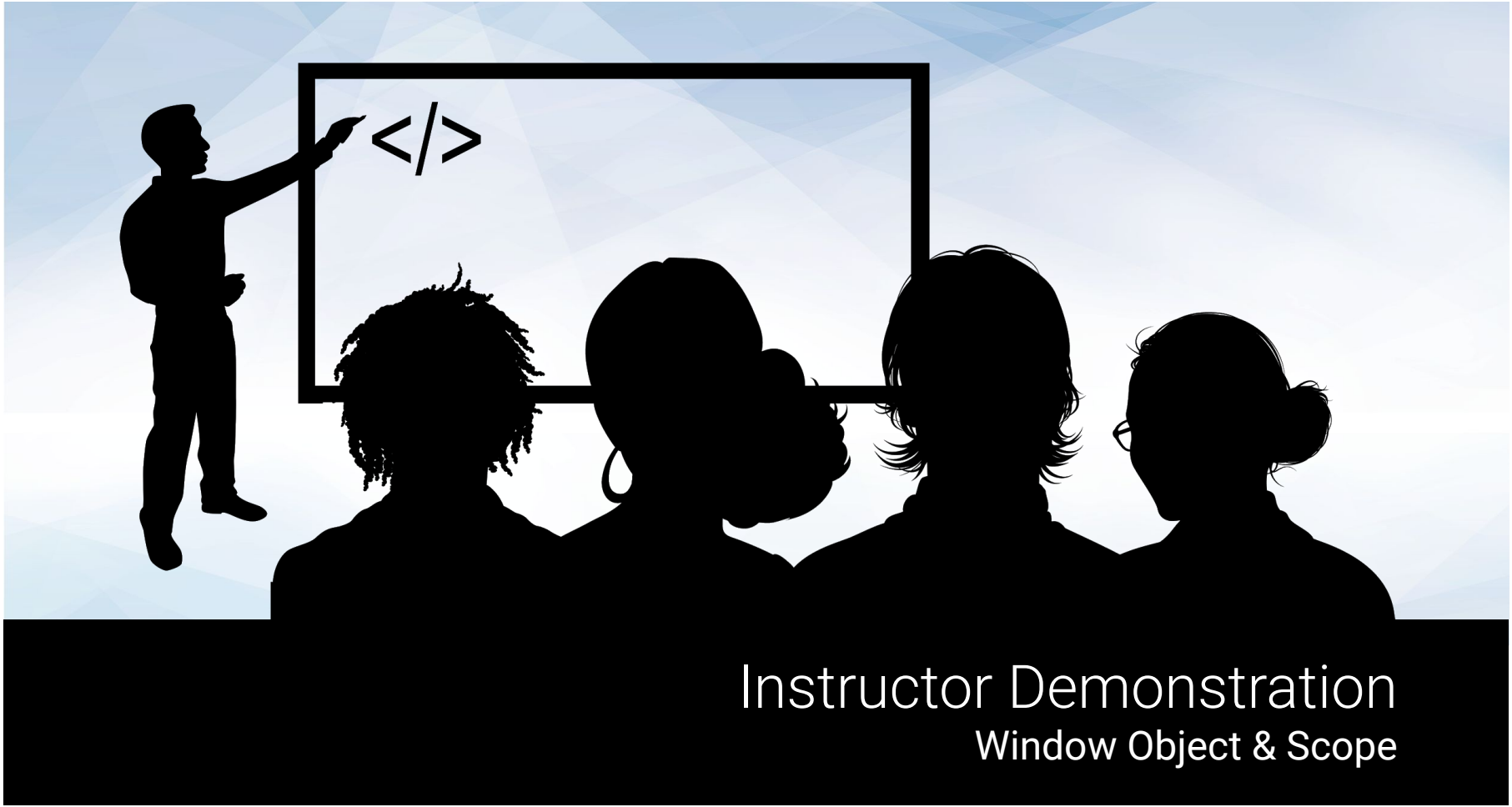
Run That Car!

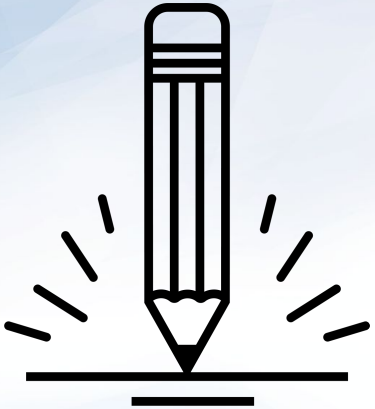
`32-CarGame`

Challenge: Run That Car!

Using the code from the previous activity as a starting point, create a complete application that fulfills the following requirements:

- Users can enter keyboard input (letters).
- Each of the car's methods are assigned to a key.
- When the user presses a key, it calls the appropriate function.
- These letters also trigger a global function called `rewriteStats()` that logs the car's make, model, color, mileage, and `isWorking` status to the console.


Instructor Demonstration
Window Object & Scope



CodeAlong Activity:

Location Redirect

`35-LocationRedirect`

Homework 3

Questions?