How do you make input (e.g. a password) show up as those dots (so no one else can read them)?

```
<input type=“password”></input>
```

"=== compared to == , when to use either.”

=== type comparison
== no type comparison

“string to integer, string to float, number to string"

string to integer = parseInt();
string to float = parseFloat();
number to string = toString();

I want to practice more! Is there any website that has quizzes on what we've learned so far?

code academy

What's the best way to approach writing pseudo-code?

plan out each step you have to take to make something happen *still not sure on how strings work*

strings are just words

Why use one Method over another, the conceptual reasoning. Go!

usually newer is better

FINAL PROJECTS

AGENDA

- Review
- Functions — What are functions?
- Functions — Syntax
- Functions — Scope
- Functions — Return Values
- Lab Time — Temperature Converter

LEARNING OBJECTIVES

- Define a function with one or more parameters
- Execute a function within a program
- Given a function and a set of arguments, predict the output of a function

FUNCTIONS

REVIEW

JAVASCRIPT — VARIABLES

Declaring a variable →

```
var age;
```

← Semicolon!

Assigning a variable →

```
age = 29;
```

← Semicolon!

Both in one step →

```
var age = 29;
```

← Semicolon!

JAVASCRIPT — VARIABLES

```
var champion = “Sarah”;
  champion = “Christine”;
```

WHAT CAN BE STORED IN VARIABLES?

DATA TYPES:

STRINGS

`"Today is Monday"`

Letters and other characters enclosed in quotes

NUMBERS

`10` `22.75`

- Positive numbers
- Negative numbers
- Decimals

BOOLEANS

`true` `false`

Can have one of two values:

- True
- False

* Note: we'll meet some more data types later on down the road, too!

JAVASCRIPT — COMPARISON OPERATORS

>=	Greater than or equal to
<=	Less than or equal to
>	Greater than
<	Less than
Equal to	===
Not equal to	!==

ASSIGNMENT VS. COMPARISON — DON'T GET THEM CONFUSED!

ASSIGNMENT

=

```
var number = 7;
```

COMPARISON

===

or

==

```
if (number === 8) {
  // Do something
}
```

JAVASCRIPT — IF/ELSE IF/ELSE

```
if (answer === 38) {
    // Do something if first condition is true
} else if (answer === 30) {
    // Do something second condition is true
} else {
    // Do something if all above conditions are false
}
```

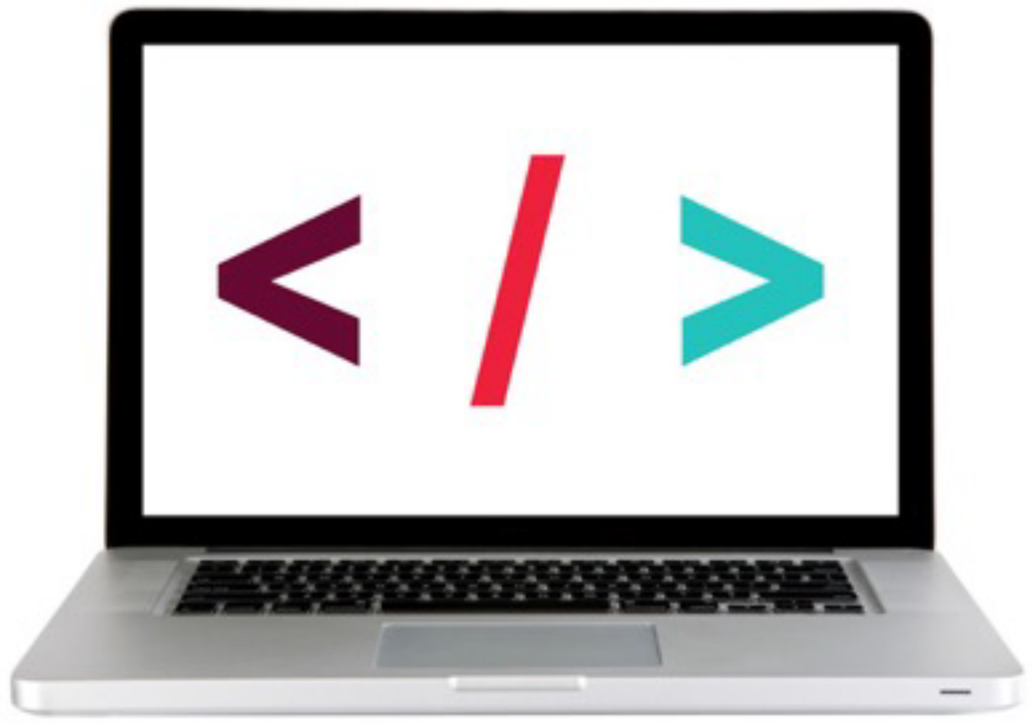
JAVASCRIPT — LOGICAL OPERATORS

&& and

|| or

! not

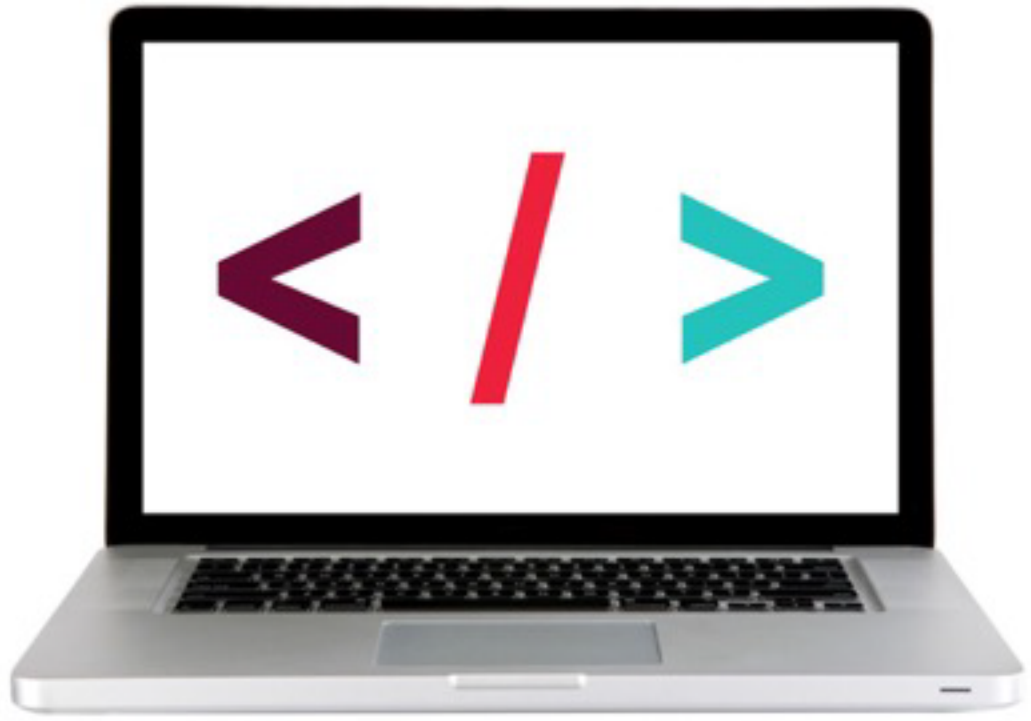
CLOSER LOOK



starter_code_lesson_10 > compare_two_numbers

CASH REGISTER PT. 1

CLOSER LOOK



Cash Register

FUNCTIONS

WHAT ARE FUNCTIONS?

CHRIS
NBA ALL-STAR
CODED IN COLLEGE
0:00 / 1:37

FUNCTIONS

GROUP STEPS

Allow us to group a series of statements together to perform a specific task

REUSABLE

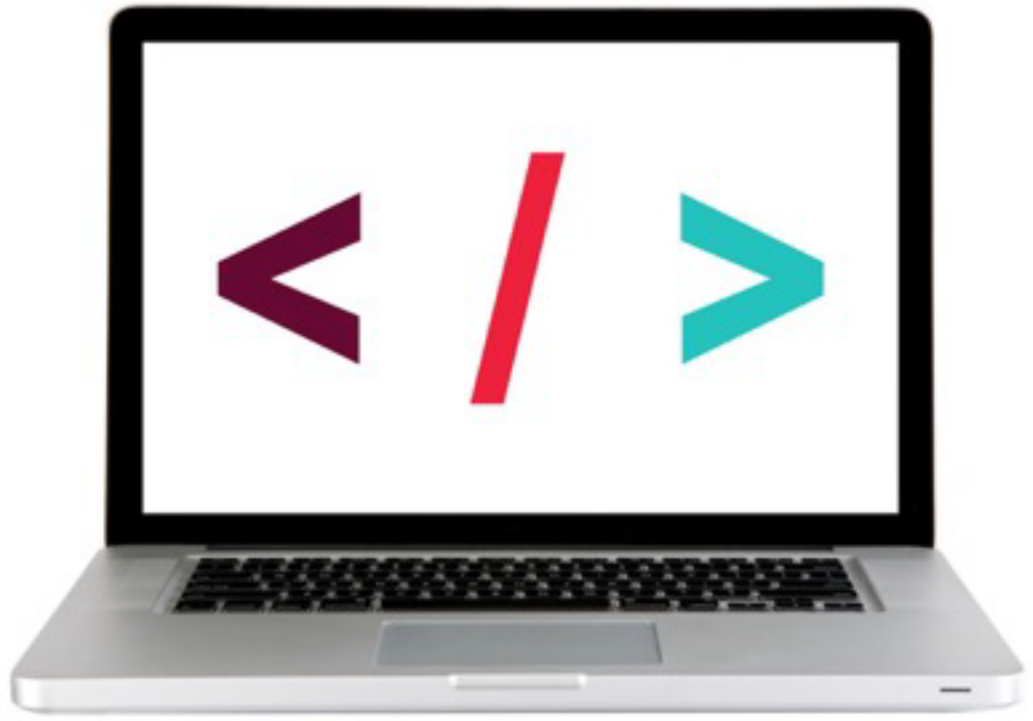
We can use the same function multiple times

STORE STEPS

Not always executed when a page loads. Provide us with a way to 'store' the steps needed to achieve a task.

DRY — DON'T REPEAT YOURSELF

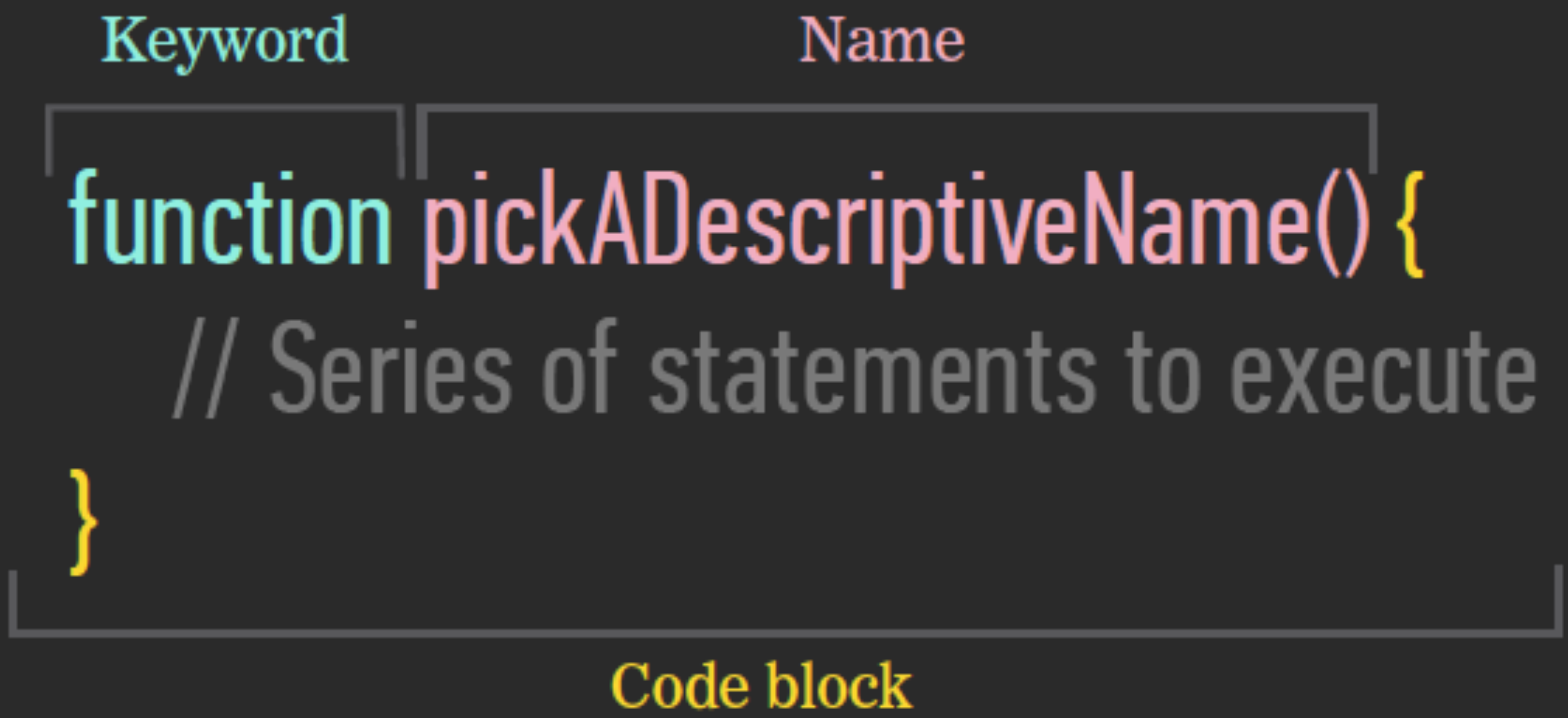
CLOSER LOOK



jQuery Traffic Light

SYNTAX

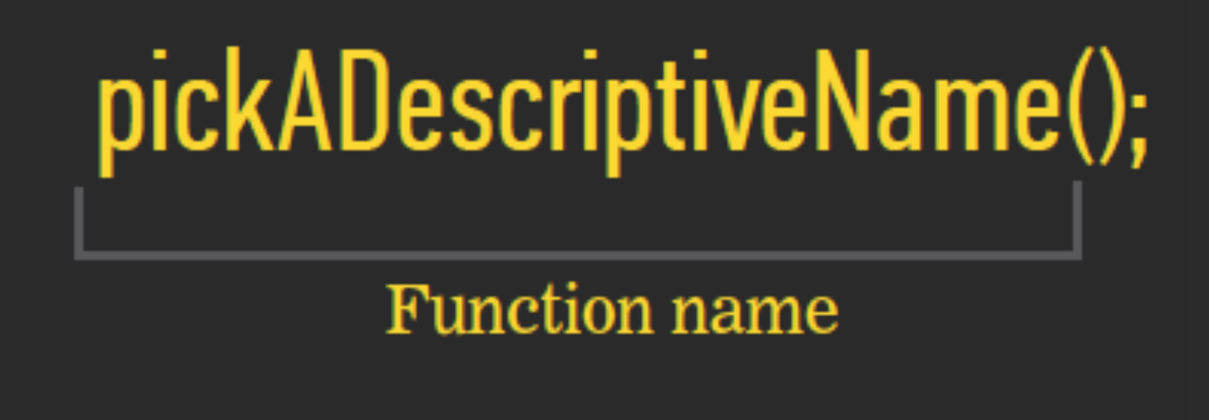
SYNTAX — DECLARING A FUNCTION



```
function pickADescriptiveName() {
    // Series of statements to execute
}
```

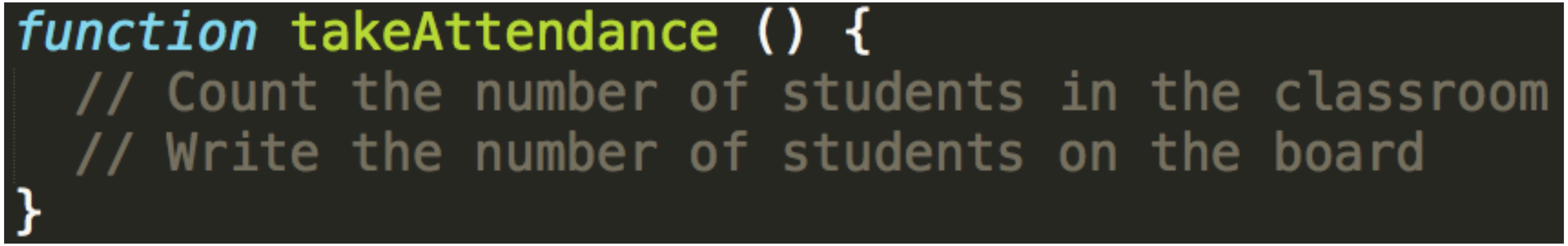
SYNTAX — CALLING A FUNCTION

- To run the code in a function, we 'call' the function by using the function name followed by parenthesis.



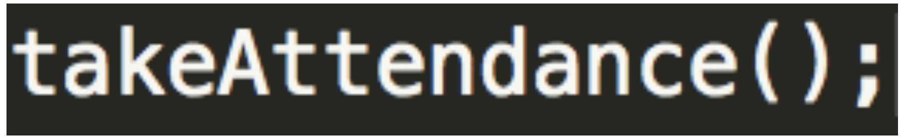
```
pickADescriptiveName();
```

FUNCTIONS — TAKING ATTENDANCE



```
function takeAttendance () {
  // Count the number of students in the classroom
  // Write the number of students on the board
}
```

FUNCTIONS — TAKING ATTENDANCE

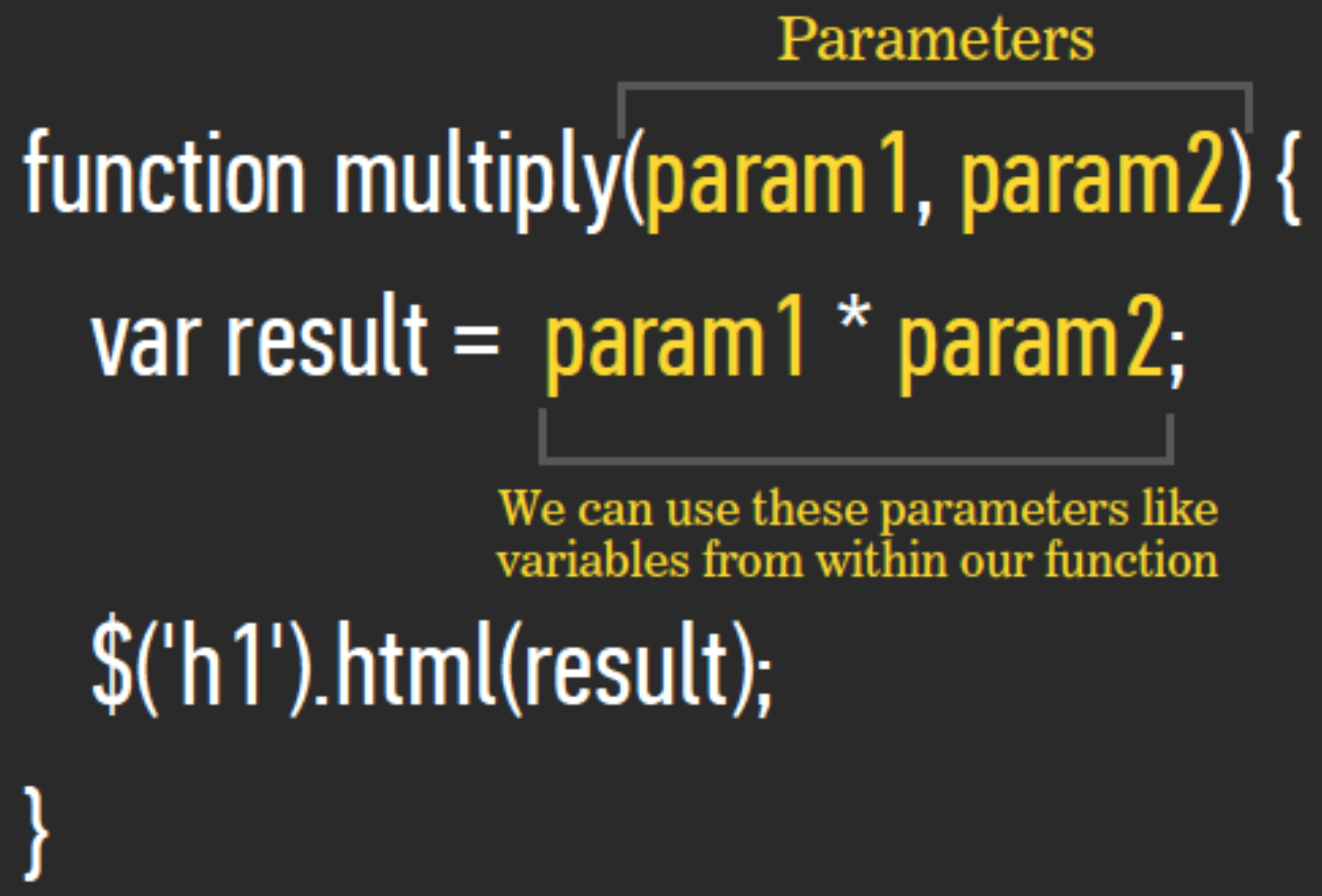


```
takeAttendance();
```

CODE ALONG — FUNCTIONS

Let's code! lesson10_starter_code > functions (part 1)

SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

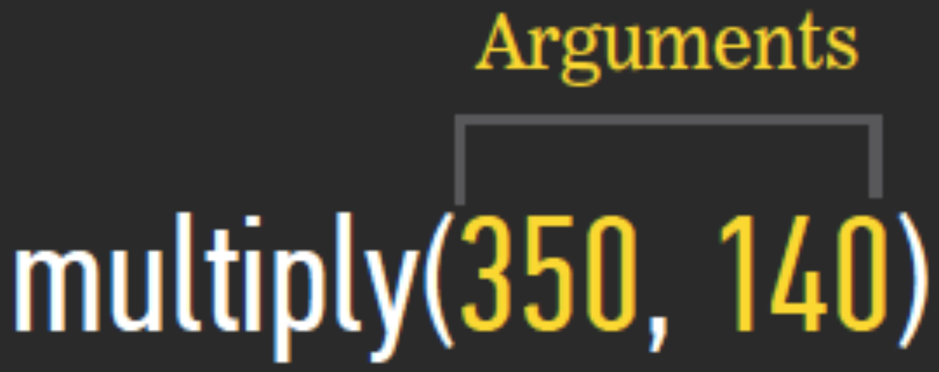


```
function multiply(param1, param2) {
   var result =  param1 * param2;

   $('h1').html(result);
}
```

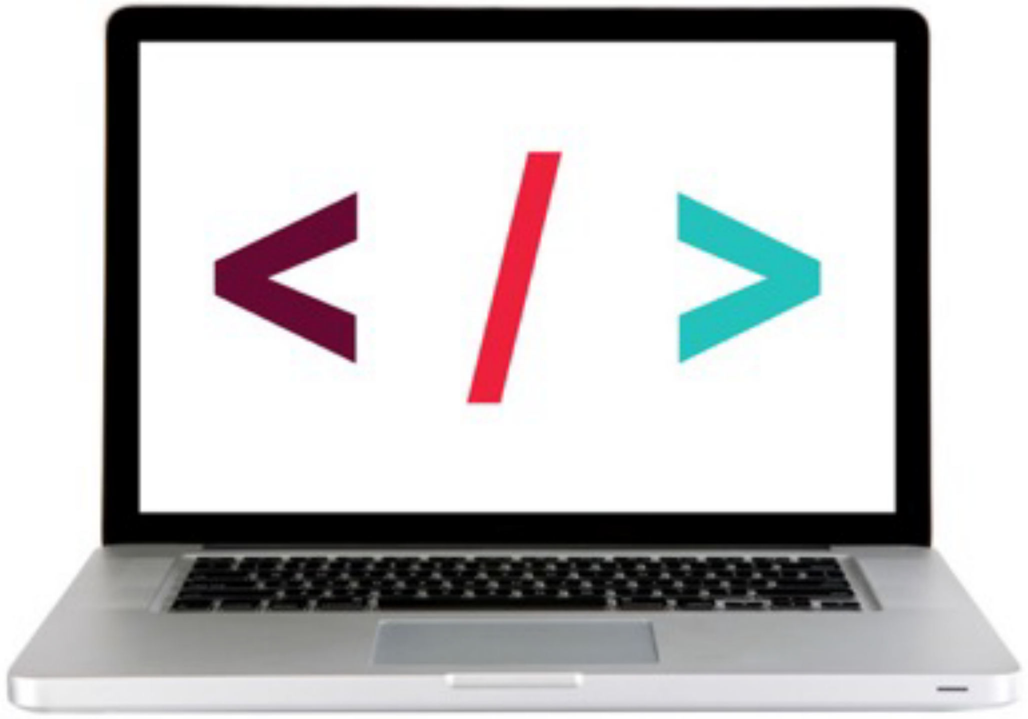
We can use these parameters like variables from within our function

SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)



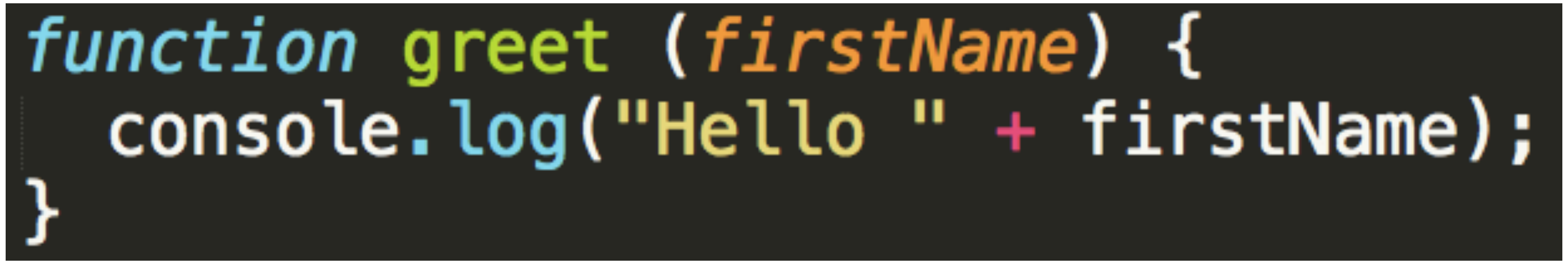
```
multiply(350, 140)
```

CLOSER LOOK



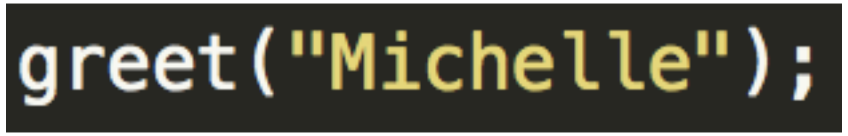
Multiply

FUNCTIONS — GREET



```
function greet (firstName) {
  console.log("Hello " + firstName);
}
```

FUNCTIONS — GREET



```
greet("Michelle");
```

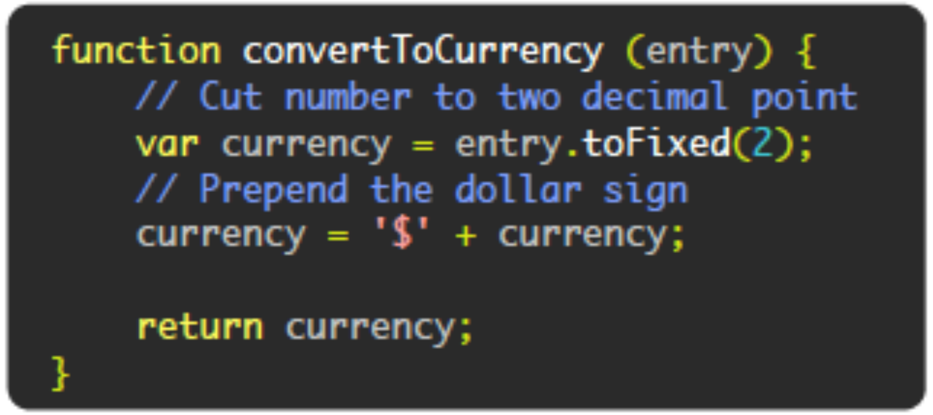
CODE ALONG — FUNCTIONS

Let's code! lesson10_starter_code > functions (part 2)

RETURN VALUES

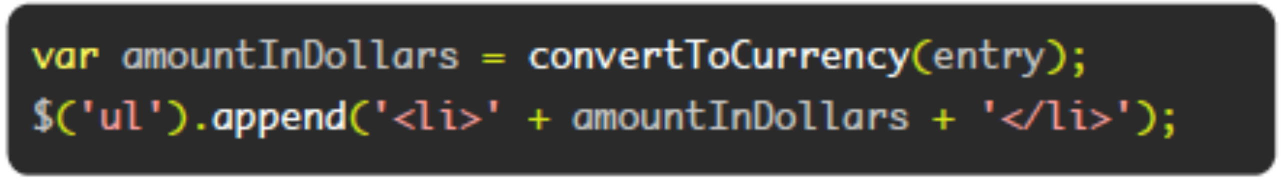
RETURNING VALUES FROM A FUNCTION

- To return a value from a function, we use the return keyword
- From within a function, the return keyword 'hands' a value back to the code that called the function
- We can then do something with that value, or store it in a variable for use later in the script



```
function convertToCurrency (entry) {
    // Cut number to two decimal point
    var currency = entry.toFixed(2);
    // Prepend the dollar sign
    currency = '$' + currency;

    return currency;
}
```



```
var amountInDollars = convertToCurrency(entry);
$('ul').append('<li>' + amountInDollars + '</li>');
```

SCOPE

FUNCTIONS — TAKING ATTENDANCE

LOCAL VARIABLES

- A **local** variable is a variable that is declared *inside* a function.
- It can **only be used in that function**, and cannot be accessed outside of that function

GLOBAL VARIABLES

- A **global** variable is a variable that is declared *outside* of a function.
- It can be used **anywhere in the script**.

LAB TIME!

LAB — TEMP CONVERTER — FORMULAS

Formula to convert fahrenheit to celsius: (fahrenheit - 32) / 1.8;

Formula to convert celsius to fahrenheit: 1.8 * celsius + 32;

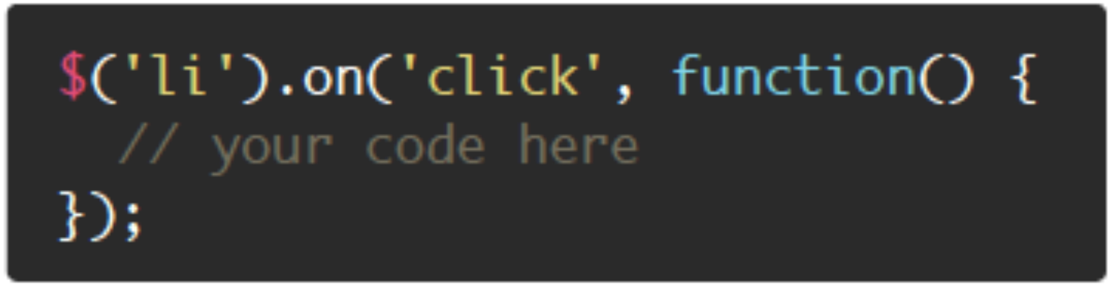
JQUERY METHODS — EVENTS!

CREATE EVENT LISTENERS

The .on() method is used to handle all events.

Syntax: `$('selector').on('event', code_that_should_run);`

Example:



```
$('li').on('click', function() {
  // your code here
});
```

LAB — TEMP CONVERTER — PART 1

EXERCISE

KEY OBJECTIVE

- Build an application using HTML/CSS and JS that converts a temperature from Fahrenheit to Celsius

TYPE OF EXERCISE

- Groups of 3-4

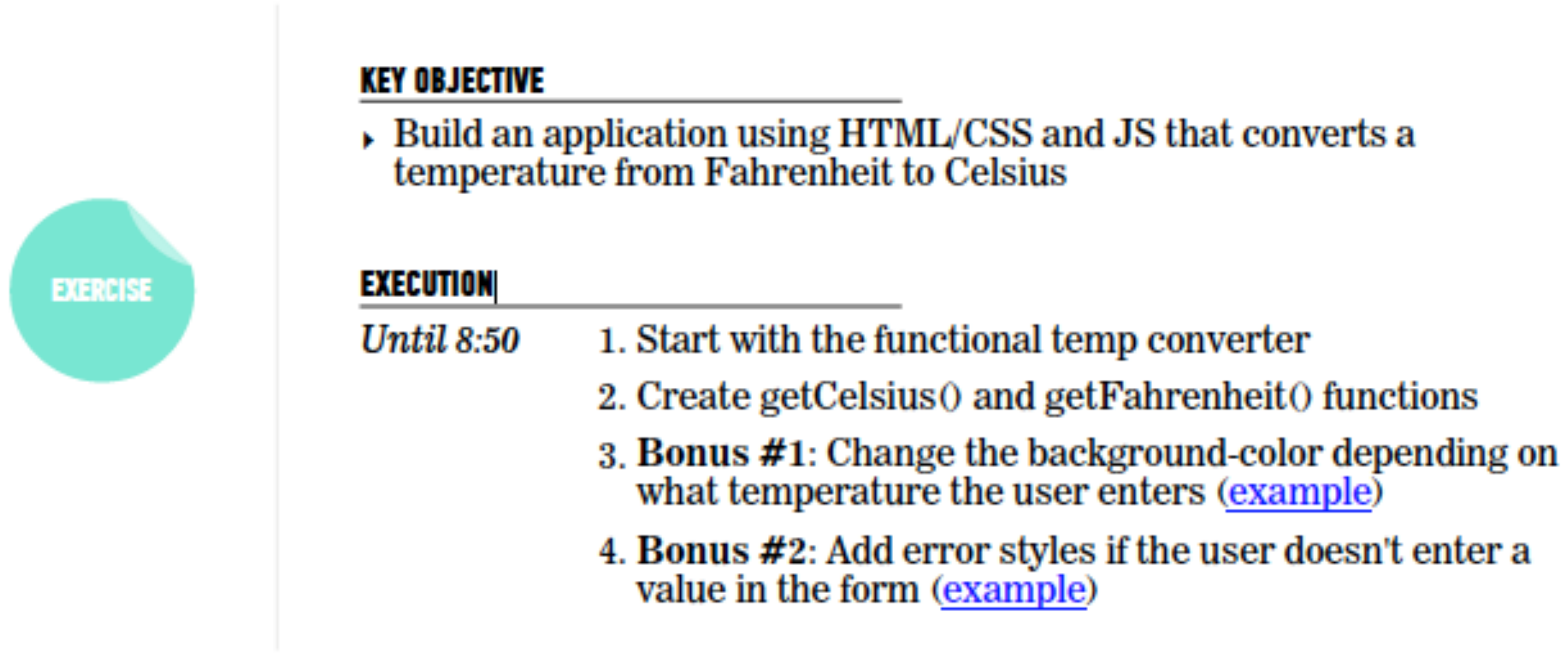
SMALL GROUP PLANNING

1. In groups of 3-4 test out the functional temperature converter and write pseudo code to convert a temperature from Fahrenheit to Celsius

CODE ALONG — FUNCTIONS

Let's code! lesson9_starter_code > [2] temp_converter

FUNCTIONS — TAKING ATTENDANCE



EXERCISE

KEY OBJECTIVE

- Build an application using HTML/CSS and JS that converts a temperature from Fahrenheit to Celsius

EXECUTION

Until 8:50

1. Start with the functional temp converter
2. Create getCelsius() and getFahrenheit() functions
3. **Bonus #1**: Change the background-color depending on what temperature the user enters (example)
4. **Bonus #2**: Add error styles if the user doesn't enter a value in the form (example)

**For reference, see the Compare Two Numbers and Score Keeper

HOMEWORK

CASH REGISTER

ADRIANA	JAMIE
Blanca	Daniel
Eva	Ben
Chris	Nancy
James	Connor
Amanda	Florencia
Federico	Lee
Teri	Madison
Jiha	Jermaine
ADRIANA.LCS316@GMAIL.COM	JAMIELEEPILGRIM@GMAIL.COM

FEWD

EXIT TICKETS