SNACKS & DESIGN

TODAY

ERIC

(GOOGLE SHEET IS PINNED IN SLACK)

Q & A

Hardest Classes in Course

Office hours: 2pm - 5pm, Sunday @ GA

“Can we do more practice if/then statements next class?”

“CAN WE DO ANOTHER LAB POR FAVOR”

“Can you review the importance of converting data types again? In what instances would you use these?”

“I didnt understand a single thing in today's lesson. I need to see actual applications on websites.”

“Is there an advantage to writing vanilla js? Is it just so you don't need to link the jquery and be connected to the internet.”

“I'll need to review. Brain shut off around 7:30”

"How do you only allow numbers or letters in an answer field?

Booleans?"

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!*

DATA TYPE CONVERSION

STRING TO INTEGER:

```
var intString = "4";
var intNumber = parseInt(intString, 10);
```

STRING TO FLOAT:

```
var floatString = "3.14159";
var floatNumber = parseFloat(floatString);
```

NUMBER TO STRING

```
var number = 4;
number.toString(); => "4";
```

HTML Boilerplate

lesson13_starter_code > [-] final_project_boilerplate

LAB

LAB — TEMP CONVERTER

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

Until 8:45

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

LAB — TEMP CONVERTER — PART 2 (NEXT CLASS)

EXERCISE

KEY OBJECTIVE

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

EXECUTION

Until 7:20

1. Write .js to make the temperature converter functional.
2. **Bonus #1**: Change the background-color depending on what temperature the user enters
3. **Bonus #2**: Add error styles if the user doesn't enter a value in the form
4. **Bonus #3**: Add your own styles to the temperature converter

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

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
});
```

GENERAL ASSEMBLY

FUNCTIONS

Eric Boyer

LEARNING OBJECTIVES

- Describe arguments as they relate to functions.
- Predict values returned by a given function.

AGENDA

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

REVIEW

JAVASCRIPT — VARIABLES

Declaring a variable → `var age;` ← Semicolon!

Assigning a variable → `age = 29;` ← Semicolon!

Both in one step → `var age = 29;` ← Semicolon!

JAVASCRIPT — VARIABLE RE-ASSIGNMENT

```
var name = “Matt”;

name = “Ana”;
```

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

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

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

REVIEW EXERCISE — CONDITIONALS

EXERCISE

KEY OBJECTIVE

- Review and practice using variables and conditionals

TYPE OF EXERCISE

- Individual/paired

EXECUTION

12 min

1. Follow the instructions in lesson13_starter_code > [0] conditionals > main.js (Part 1)
2. If you finish early, work on the **bonus** section

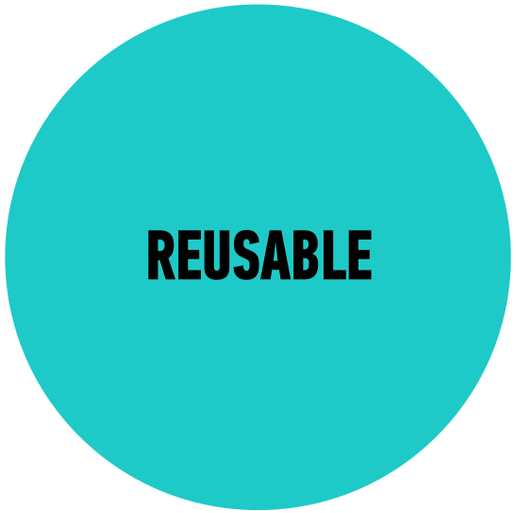
FUNCTIONS

WHAT ARE FUNCTIONS?

FUNCTIONS

FUNCTIONS

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

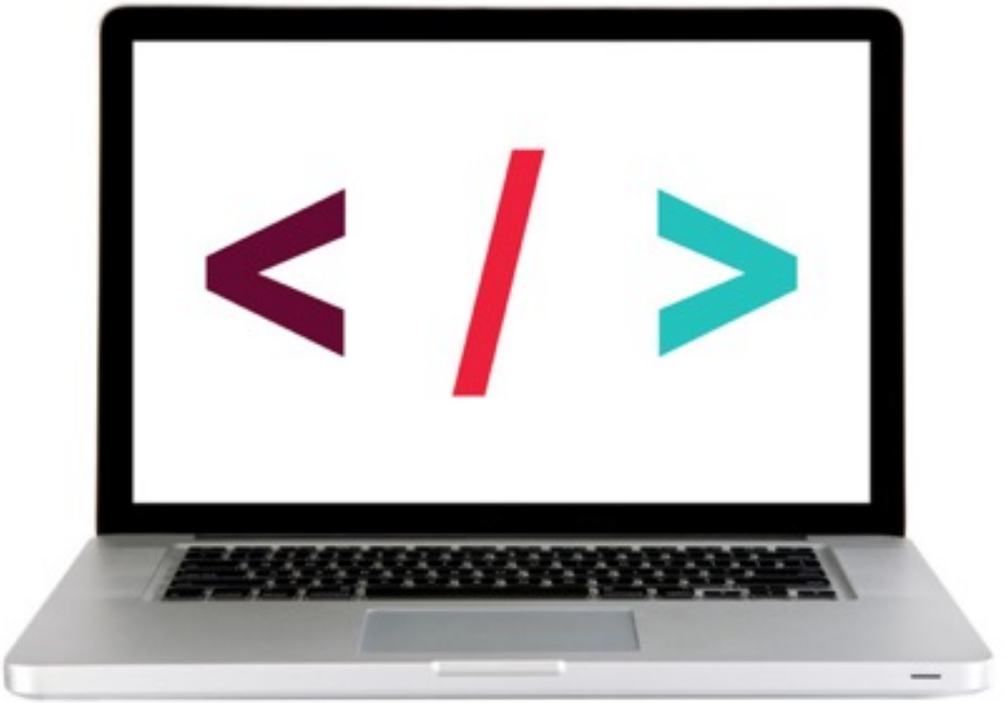


We can use the same function multiple times

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

LET'S TAKE A CLOSER LOOK



jQuery Traffic Light

SYNTAX

SYNTAX — DECLARING A FUNCTION

Keyword

Name

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

Code block

SYNTAX — CALLING A FUNCTION

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

```
pickADescriptiveName();
```

Function name

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! lesson13_starter_code > functions (part 1)

SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

Parameters

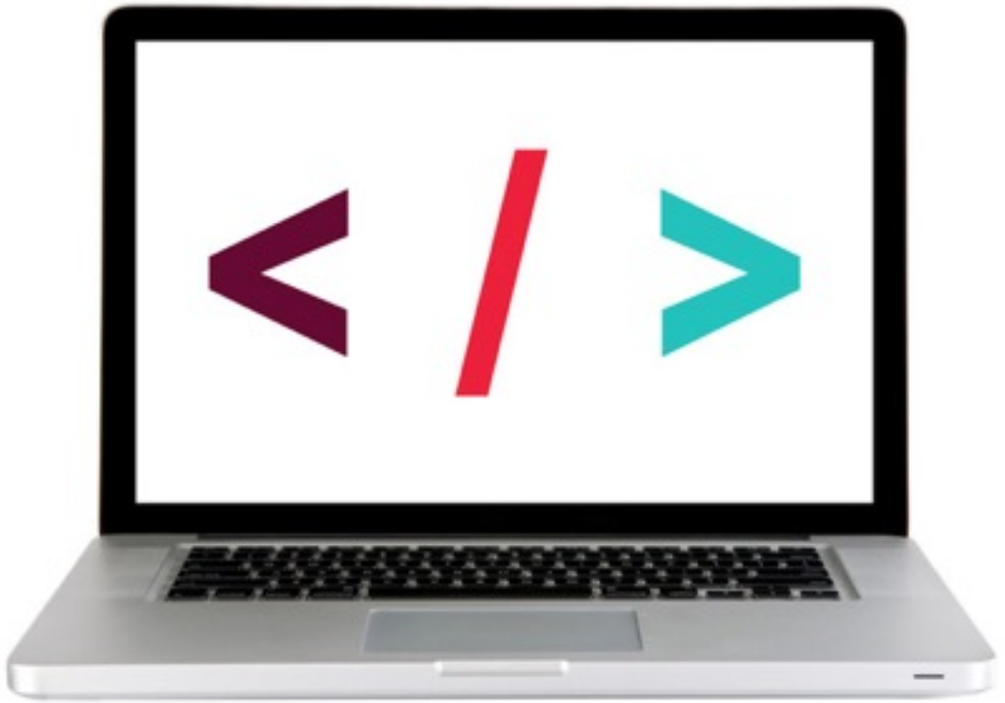
```
function multiply(param1, param2) {
  var result =  param1 * param2;

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

We can use these parameters like variables from within our function

SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)

LET'S TAKE A CLOSER LOOK



Multiply on CodePen

FUNCTIONS — GREET

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

FUNCTIONS — GREET

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

CODE ALONG — FUNCTIONS

Let's code! lesson13_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

VARIABLE SCOPE

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

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

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! lesson13_starter_code > [2] temp_converter

LAB — TEMP CONVERTER — PART 2

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*

LEARNING OBJECTIVES

- Describe arguments as they relate to functions.
- Predict values returned by a given function.

LAB

HOMEWORK: “CASH_REGISTER"

SNACKS & DESIGN

MONDAY

RYAN

(GOOGLE SHEET IS PINNED IN SLACK)

EXIT TICKETS