LESSON 09 - VARIABLES & CONDITIONALS

AGENDA

- Learning Objectives
- Intro To jQuery Review
- Variables
- Conditionals
- Lab Time

LEARNING OBJECTIVES:

AFTER TODAY, YOU SHOULD BE ABLE TO...

- Define variables and identify best cases to use them.
- Differentiate between strings, integers and floats.
- Apply conditionals to change the programs control flow.

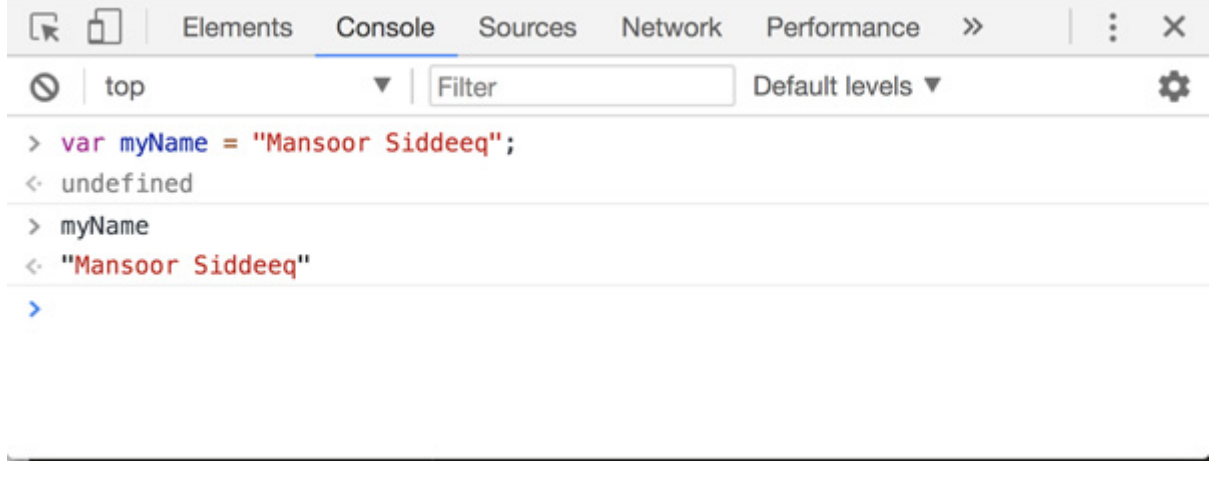
Today, we will be diving back into the meat of Javascript

But first, let's briefly review jQuery with a Code Along...

REVIEW: JQUERY DOM SELECTORS REVIEW

THE CONSOLE

The Console is your friend for testing basic JS functionality



GETTING STARTED WITH VARIABLES

WHAT ARE VARIABLES?

Variables are essentially containers for storing data values in your JS code.

We declare the name of the container, and then we put something in it for our program to use later on.

DECLARING VARIABLES

The act of creating/naming the variable is called a **declaration**

```
var myAge;
```

ASSIGNING VARIABLES

The action of saving a value to this variable is called **assignment**

```
myAge = 30;
```

The declaration and assignment of a variable can be done at the same time:

```
var legalAge = 18;
```

PULSE CHECK - VARIABLE BASICS

Help me record some important information about my car...

My Car information to capture:

- My car make
- My car model
- My car year
- My car type
- My car color

ACCESSING VARIABLES

The action of getting the value from a variable is called **accessing the variable**

```
myAge;
```

RE-ASSIGNING VARIABLES

You can re-assign variables as many times as you want

```
var myName = "Mansoor";

myName = "Sudi";
```

VARIABLE CONVENTIONS

- you should never use a reserved word (more on the next slide)
- they should always start with a lower case letter
- if they contain multiple words, subsequent words should start with an upper case letter

```
var numberOfStudents = 10;
```

Javascript has some reserved words you cannot use as variables or function names...

List of Reserved Words

VARIABLES & DATA TYPES

You can store all different types of data in your variables

Though, each one comes with their own special ways to be handled in JS

DATA TYPES

The types of different values we support include:

- **String** (text data)
- **Int/Float** (number data)
- **Boolean** (true or false)

WE DO: SCORE KEEPER

DATA TYPES: STRINGS

String variables store textual information.

The actual value of a string will always be surrounded by quotes:

```
"How is the weather today?"
```

```
'Warm'
```

There are times when you might need to include them in the actual string literal itself. In that case, do this:

```
'They "purchased" it'
```

```
"It's a beautiful day"
```

ESCAPING STRINGS

The backslash escape character \ turns special characters into string characters:

```
"They \"purchased\" it"
```

```
'It\'s a beautiful day'
```

PULSE CHECK - WHAT'S A STRING?

DATA TYPES: NUMBERS

Numbers in Javascript represent numerical data.

Numbers can either be **integers** or **floats**

```
42
```

```
3.14159265
```

Number data types are special in that you can perform arithmetic on them.

Here are some basic arithmetic operations you can perform in JS:

Operator	Meaning	Example
+	Addition	8 + 10
-	Subtraction	10 – 8
*	Multiplication	12 * 2
/	Division	10 / 5
%	Modulus	10 % 6

MODULUS

The modular operator `%` returns the division remainder.

```
10 % 2
```

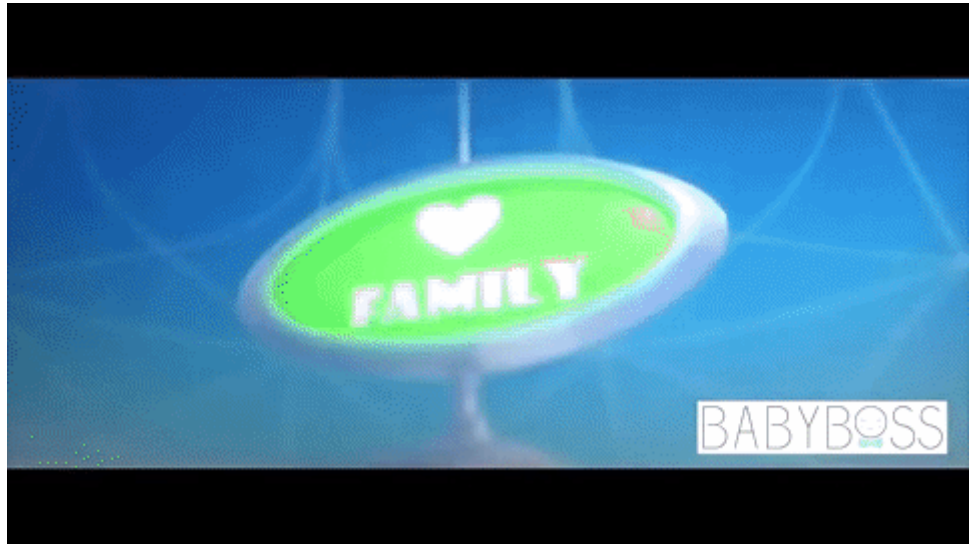
PULSE CHECK - JS MATHEMATICS POP QUIZ

Using the browser's Console, let's do some basic arithmetic using JS

- 196 + 2,567
- 25 % 2
- 50 % 5
- 342 / 67
- 71 * 42

5 MINUTE BREAK

CONDITIONAL STATEMENTS



To understand conditionals, we first need to understand what "control flow" is in the context of your JS code...

CONTROL FLOW

The control flow is the order in which the computer executes statements in a script.

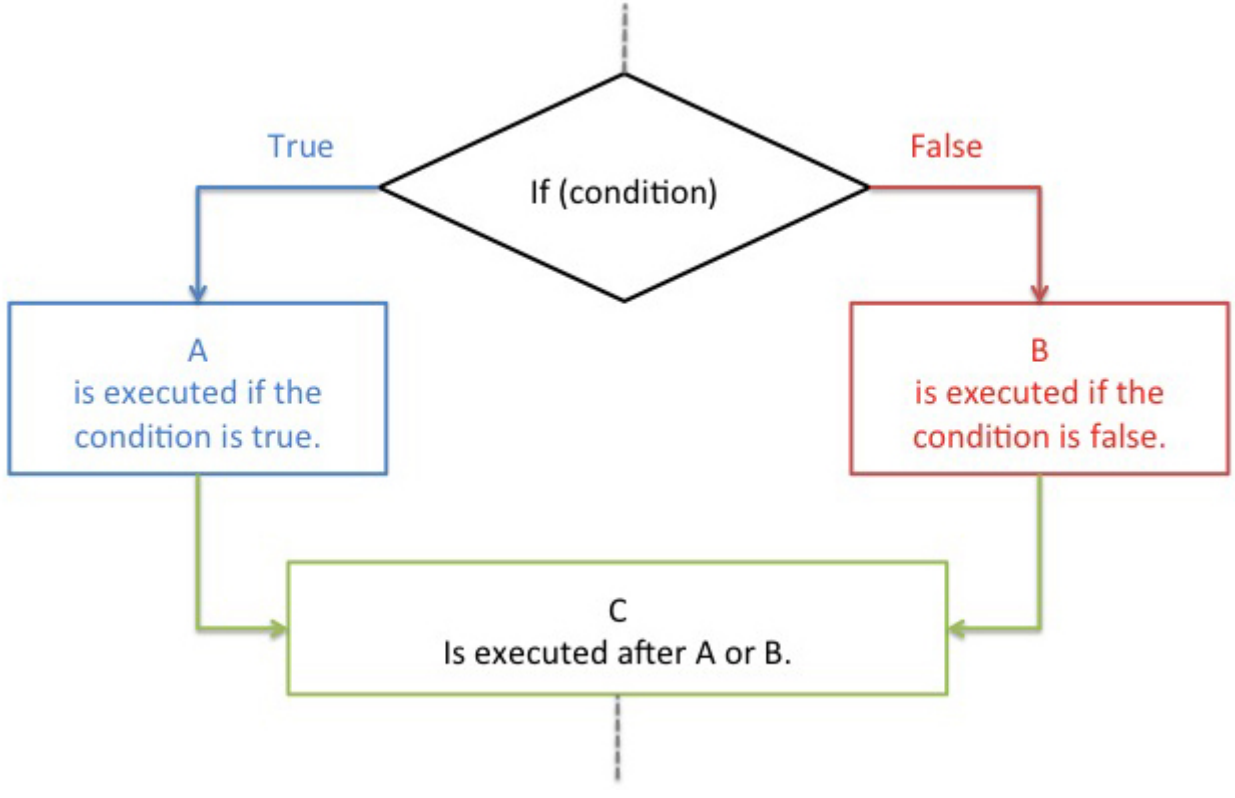
Typically, this is from left to right, top to bottom.

Unless, the computer runs across a structure that changes this "control flow", such as functions, loops, or conditionals.

Similarly, often when you write code, you will want to perform different actions for different decisions.

You can use **conditional statements** in your code to do this.

This will often take the form of an "if/else" statement


True
If (condition)
False
A
is executed if the
condition is true.
B
is executed if the
condition is false.
C
Is executed after A or B.

IF/ELSE STATEMENT

This simply checks to see if something is either **TRUE** or **FALSE**

Then does something based on the outcome.

EXAMPLE

If you are older than 17, then you are an adult.

```
if (age > 17){
    console.log("You are an adult");
}
```

EXAMPLE

If you are older than 17, then you are an adult. Else, you are a child.

```
if (age > 17){
    console.log("You are an adult.");
}else{
    console.log("You are a child.");
}
```

We're making decisions with our code.

You can even chain multiple `if` statements for more conditions...

```
if(condition is true) {
    // Do thing A
}else if{
    // Do thing B
}else{
    // Do something else
}
```

COMPARISON OPERATORS

To check if something is true or not, we need **comparison operators** to compare the criteria

Operator	Description	Example	Result
==	Equal to	1 == 1	true
===	Equal in value and type	1 === '1'	false
!=	Not equal to	1 != 2	true
!==	Not equal in value and type	1 !== '1'	true
>	Greater than	1 > 2	false
<	Less than	1 < 2	true
>=	Greater than or equal to	1 >= 1	true
<=	Less than or equal to	2 <= 1	false

WE DO: COMPARE THAT

LOGICAL OPERATORS

We can also check multiple conditions in a single conditional statement like so:

```
if (name == "GA" && password == "YellowPencil"){
    //Allow access to internet
}
```

To do this, we use **logical operators**

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x == 5 || y == 5) is false
!	not	!(x == y) is true

THE LOGICAL AND (&&) OPERATOR

This checks to see if **ALL** of the conditions are true before running the code inside of the conditional statement.

```
if (name == "GA" && password == "YellowPencil"){
    //Allow access to internet
}
```

THE LOGICAL AND (&&) TRUTH TABLE

AND - &&	TRUE	FALSE
TRUE	true	false
FALSE	false	false

THE LOGICAL OR (||) OPERATOR

This checks to see if **AT LEAST ONE** of the conditions are true before running the code inside of the conditional statement.

```
if (day == "Monday" || day == "Wednesday"){
    //We have class today
}
```

THE LOGICAL OR (||) TRUTH TABLE

OR - ||	TRUE	FALSE
TRUE	true	true
FALSE	true	false

WE DO: BLACKOUT

LAB: TEMP CONVERTER - PART 1

- Get temperature in Celsius
- Convert temperature to Fahrenheit
- Check temperature to determine what color to change the background
  - if temperature is less than/equal to 65
    - set background to Blue
  - if temperature is greater than 65, but less that 85
    - set background to Yellow
  - if temperature is greater than/equal 85, but less than 95
    - set background to Orange
  - if temperature is greater than 95
    - set background to Red

Now you work on the code to build this. Don't forget your HTML Templating first.

EXIT TICKETS

Let's spend 5-10 minutes to fill out today's Exit Survey

LEARNING OBJECTIVES REVIEW

- We Defined variables and identify best cases to use them.
- We Differentiated between strings, integers and floats.
- We Applied conditionals to change the programs control flow.

No Homework over the weekend...