



WEEK 5 / LESSON 9:

# VARIABLES & CONDITIONALS

Leonie Dunnett

# LEARNING OBJECTIVES

- Define variables and identify best cases to use them
- Differentiate between strings, integers and floats
- Apply conditionals to change control flow in a program

# TODAY'S SCHEDULE

- jQuery Basics Review
- Variables
  - Score Keeper Code Along
- Conditionals
  - Compare That Code Along
  - Blackout Code Along
- Weather Application - Part 1 Lab

# JQUERY BASICS REVIEW

- Questions?
- DOM Selector Practice **Review**

# VARIABLES

# WHAT ARE VARIABLES?

- We can tell our program to remember values for later use
- Saving a value to memory is called **assignment**
- The entity we store the value in is called a **variable**
- Getting a variable's value is called **accessing the variable**

We will use all of these techniques to store values into variables and generate new values using existing variables

# VARIABLES DECLARATION

Declaration: `var age;`

Assignment: `age = 21;`

Both at the same time: `var age = 21;`

# VARIABLE RE-ASSIGNMENT

```
var name = 'Jo';
```

```
name = 'Fred';
```

**name is now Fred**



# VARIABLE CONVENTIONS

- Variables start with a lower case letter
- If they contain multiple words, subsequent words start with an upper case letter - camel case

```
var numberOfStudents = 14;
```

# VARIABLES & DATA TYPES

What can you store in a variables?

The types of different values we support include:

- **string** (text)
- **number** (integers, floats)
- **boolean** (true or false)
- **objects** {...}
- **functions**



# SCORE KEEPER

Variables, data types and arithmetic

# STRINGS

- Store textual information
- String literal is surrounded by quotes

"How is the weather today?"

'Warm'

# STRINGS

Double vs single quoted strings:

'They "purchased" it'

"It's a beautiful day"

Escaping:

"They \"purchased\" it"

'It\'s a beautiful day'

# CONVERSION: STRING TO NUMBER

```
var intString = "4";  
var intNumber = parseInt(intString);  
var intNumber = +intString;  
  
var floatString = "3.14159";  
var floatNumber = parseFloat(floatString);  
var floatNumber = +floatString;
```

Why would you need to convert data types?

[Data Types Concept](#) via W3Schools

# CONVERSION: NUMBER TO STRING

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

OR

```
number + "";
```

# NUMBERS

Represent numerical data

```
int:      42
float:    3.14159265
int:      +6
float:    -8.2
```

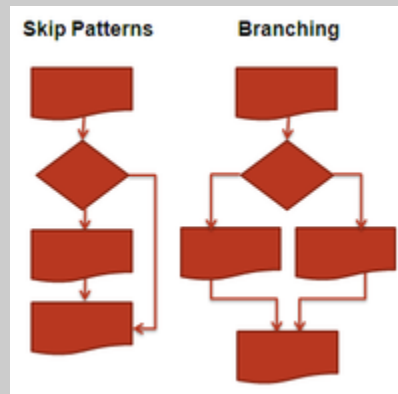
You can perform arithmetic on number data types



# ARITHMETIC IN JAVASCRIPT

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

# CONDITIONALS

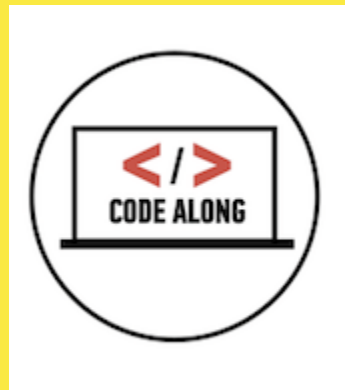


# MAKING DECISIONS

It's either TRUE or FALSE (like booleans)

If you are greater than 18 you are an adult

```
if (age > 18){  
    document.write("You are an adult");  
}
```



**COMPARE THAT**

# COMPARISONS - EQUALITY

Are two things equal?

```
10 === 10 //true
```

```
10 === 5 //false
```

```
"hi" === "hi" //true
```

# LOGICAL OPERATORS

**x = 3**

Logical Operators			
Operator	Description	Comparing	Returns
<b>==</b>	equal to	<b>x == 8</b>	<b>FALSE</b>
<b>===</b>	exactly equal to(value and type)	<b>x === "3"</b>	<b>FALSE</b>
		<b>x === 3</b>	<b>TRUE</b>
<b>!=</b>	is not equal	<b>x != 8</b>	<b>TRUE</b>
<b>!==</b>	is not equal(neither value nor type)	<b>x !== "3"</b>	<b>TRUE</b>
		<b>x !== 3</b>	<b>FALSE</b>
<b>&gt;</b>	greater than	<b>x &gt; 8</b>	<b>FALSE</b>
<b>&lt;</b>	less than	<b>x &lt; 8</b>	<b>TRUE</b>
<b>&gt;=</b>	greater than or equal to	<b>x &gt;= 8</b>	<b>FALSE</b>
<b>&lt;=</b>	less than or equal to	<b>x &lt;= 8</b>	<b>TRUE</b>

# CONDITIONAL SYNTAX

```
if(condition is true) {  
    //Do cool stuff  
}
```

# CONDITIONAL SYNTAX

```
if(condition is true) {  
    //Do cool stuff  
}else{  
    //Do other cool stuff  
}
```



# CONDITIONAL SYNTAX

```
var topic = "JS";  
if (topic == "JS") {  
    console.log("You're learning JavaScript");  
} else if(topic == "JavaScript") {  
    console.log("You're still learning JavaScript");  
} else {  
    console.log("You're learning something else");  
}
```

# MULTIPLE CONDITIONS

AND - **&&**

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

OR - **||**

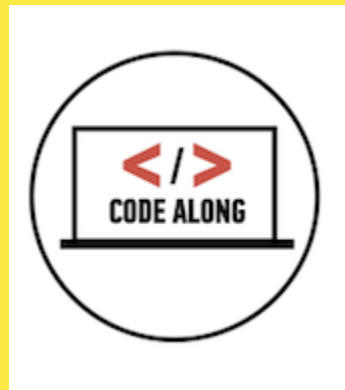
```
if (day == "Tuesday" || day == "Thursday"){  
    //We have class today  
}
```

# THE AND - && TRUTH TABLE

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

# THE OR - || TRUTH TABLE

OR -	TRUE	FALSE
TRUE	true	true
FALSE	true	false



# BLACKOUT



# WEATHER APPLICATION - PART 1

Temp Converter

# WEATHER APPLICATION - PART 1

- As a class, write feature requirements / user stories necessary to create a fully functional application that:
  - takes celsius temperature,
  - converts it to Fahrenheit,
  - changes the background to match the temperature.
- In groups of 3-4 write pseudo code for the application.
- In pairs write the code to convert Celsius into Fahrenheit, and display the result in the browser.

# REVIEW: LEARNING OBJECTIVES

- Define variables and identify best cases to use them
- Differentiate between strings, integers and floats
- Apply conditionals to change control flow in a program



# HOMework

- **Review:**

Slides and Resources

- **Code:**

Re-create the weather widget app we began in class as a city widget web application that takes a user's input and updates the page's background and images to reflect the input (images provided).

See Week 5 Assignment folder for details.

# JAVASCRIPT RESOURCES

- W3Schools:
  - [Variables](#), [Operators](#), [Conditionals](#)
  - [Numbers](#), [Number methods](#)
- Tutorials Point:
  - [Variables](#), [Operators](#), [Conditionals](#)