

# Variables & Conditionals

*Nayana Davis*

---

## Variables & Conditionals

---

# LEARNING OBJECTIVES

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

---

## Variables & Conditionals

---

# REVIEW

---

## **Variables & Conditionals**

---

What is programming?

---

## Variables & Conditionals

---

Programming is the task of writing instructions in a language that the computer can understand.

---

## **Variables & Conditionals**

---

What is pseudocode?

---

## Variables & Conditionals

---

Pseudocode is the process of writing a program without using the syntax of a programming language. It is a mixture of natural language and high-level programming constructs.

---

## **Variables & Conditionals**

---

What is Javascript?



---

## Variables & Conditionals

---

JavaScript is an object-oriented computer programming language commonly used to create interactive effects within web browsers.

---

## Variables & Conditionals

---

What is jQuery?

---

## Variables & Conditionals

---

jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. It allows for document traversal, CSS manipulation, event handling and more.

---

## Variables & Conditionals

---

How do you add jQuery to your site?

---

## Variables & Conditionals

---

`<script src="js/jquery-1.8.3.min.js"></script>` -  
Adding the file.

`<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>` - CDN

---

## Variables & Conditionals

---

Write out an example of jQuery syntax

---

# Variables & Conditionals

---

```
$(".class").click();
```

---

## Variables & Conditionals

---

Jot down a list of functions you've seen so far



---

# Variables & Conditionals

---

.click  
.slideToggle()  
.hide()  
.show()  
.slideUp()  
.slideDown()  
.children()  
.attr()

---

## Variables & Conditionals

---

# Variables

---

# Variables & Conditionals

---

- A variable is a bucket. You can put data in it. Then you can re-use it as many times as you want.
- "Data" just doesn't mean numbers -- it means literally anything in Javascript.

---

# Variables & Conditionals

---

## Primitive Data Types

- Boolean - True or False
- Null - assigned a value of “no value”
- Undefined — variable has been declared, but not assigned a value
- Number (integers, floats)
- String - anything between single or double quotes

---

# Variables & Conditionals

---

## Variable Declaration and Assignment

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

---

# Variables & Conditionals

---

Variable Re-Assignment

```
var name = "Tor";
```

```
name = "Nayana";
```

Name is now "Nayana"

---

# Variables & Conditionals

---

Reusability:

Now instead of having to write \$('#player') everywhere, we can write player.

[http://www.w3schools.com/js/js\\_variables.asp](http://www.w3schools.com/js/js_variables.asp)

```
var player = $('#player');
```

---

# Variables & Conditionals

---

## Variable Conventions:

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

```
var numberOfStudents = 10;
```



---

# Variables & Conditionals

---

## Data Types can be converted

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

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

---

## Variables & Conditionals

---

# Conditionals

# Variables & Conditionals

## Comparison Operators:

Operator	Description	Examples returning true
<a href="#">Equal (==)</a>	Returns true if the operands are equal.	<pre>3 == var1 "3" == var1  3 == '3'</pre>
<a href="#">Not equal (!=)</a>	Returns true if the operands are not equal.	<pre>var1 != 4 var2 != "3"</pre>
<a href="#">Strict equal (===)</a>	Returns true if the operands are equal and of the same type. See also <a href="#">Object.is</a> and <a href="#">sameness in JS</a> .	<pre>3 === var1</pre>
<a href="#">Strict not equal (!==)</a>	Returns true if the operands are of the same type but not equal, or are of different type.	<pre>var1 !== "3" 3 !== '3'</pre>
<a href="#">Greater than (&gt;)</a>	Returns true if the left operand is greater than the right operand.	<pre>var2 &gt; var1 "12" &gt; 2</pre>
<a href="#">Greater than or equal (&gt;=)</a>	Returns true if the left operand is greater than or equal to the right operand.	<pre>var2 &gt;= var1 var1 &gt;= 3</pre>
<a href="#">Less than (&lt;)</a>	Returns true if the left operand is less than the right operand.	<pre>var1 &lt; var2 "2" &lt; 12</pre>
<a href="#">Less than or equal (&lt;=)</a>	Returns true if the left operand is less than or equal to the right operand.	<pre>var1 &lt;= var2 var2 &lt;= 5</pre>

---

# Variables & Conditionals

---

## Conditionals

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

---

# Variables & Conditionals

---

## Conditionals

```
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");  
  
}
```

---

# Variables & Conditionals

---

## Logical Operators

Operator	Usage	Description
Logical AND ( <code>&amp;&amp;</code> )	<code>expr1 &amp;&amp; expr2</code>	Returns <code>expr1</code> if it can be converted to <code>false</code> ; otherwise, returns <code>expr2</code> . Thus, when used with Boolean values, <code>&amp;&amp;</code> returns <code>true</code> if both operands are true; otherwise, returns <code>false</code> .
Logical OR ( <code>  </code> )	<code>expr1    expr2</code>	Returns <code>expr1</code> if it can be converted to <code>true</code> ; otherwise, returns <code>expr2</code> . Thus, when used with Boolean values, <code>  </code> returns <code>true</code> if either operand is true; if both are false, returns <code>false</code> .
Logical NOT ( <code>!</code> )	<code>!expr</code>	Returns <code>false</code> if its single operand can be converted to <code>true</code> ; otherwise, returns <code>true</code> .

---

# Variables & Conditionals

---

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

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

---

## Variables & Conditionals

---

# Guided Practice: Blackout



## Variables & Conditionals

---

# Group Activity: DOM Selectors Practice

## Variables & Conditionals

---

You Do: Compare  
That & Score  
Keeper