



JS Basics

with Guy Routledge

@guyroutledge | #FEWD

Objective

Learn the building-block syntax of JavaScript

Agenda

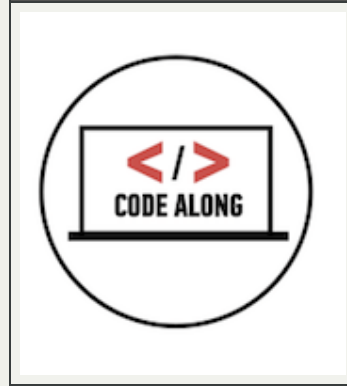
- Homework Review
- Review JS Syntax
- Variables
- Arithmetic
- Conditionals

JS Basics Syntax

Syntax: Spelling and grammar rules of a programming language.

JS Syntax

- Semicolons
- Curly Brackets
- Parentheses
- Square brackets
- Strings
- Function keyword
- Function arguments



JS Basics

Variables

What are variables?

Variables

- We can tell our program to remember values for us to use later on.
- The action of setting a variable value is called assignment

Variables

- The action of getting the value from a variable is called accessing the variable
- Variable assignment and accessing is often referred to "setting" and "getting"

Creating variables

```
var score = 0;
```

Variable Re-Assignment

```
var score = 0;  
score = 10;
```

Variable Conventions

- Variables start with a lower case letter
- Variables can start with an or \$ as well
- If they contain multiple words, subsequent words start with an upper case letter.

```
var numberOfStudents = 10;
```

Variables & Data Types

What can you store in a variables?

- **String** - for text
- **int** or **float** - numbers
- **Boolean** - true or false values
- **Array** - collections of data
- **Objects** - contain series of **key:value** pairs
- **Functions**

Strings

- Stores text content
- 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"
```

Conversion: String To Number

```
var fourTheString = "4";  
var fourTheNumber = parseInt(intString, 10);  
var pi_as_a_string = "3.14159";  
var pi_to_five_decimal_places = parseFloat(floatString);
```


Conversion: Number To String

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

OR

```
number + "";
```

Numbers

Represent numerical data

```
int:      42  
float:    3.14159265
```

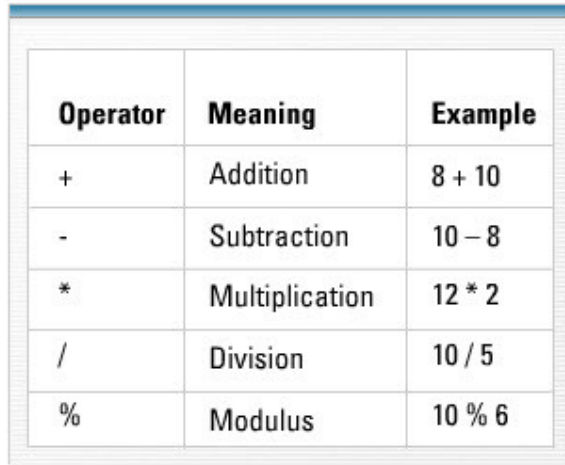
Numbers

Signed

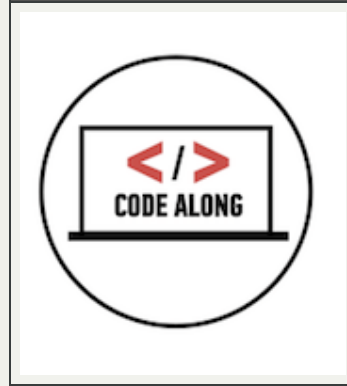
```
int:      +6  
float:    -8.2
```

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



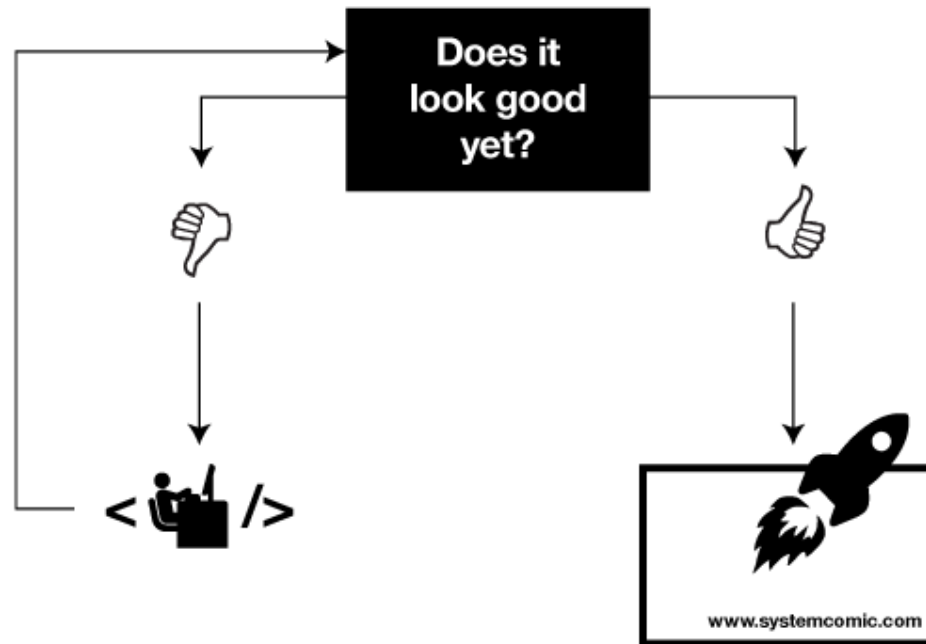
Score Keeper

<http://codepen.io/guyroutledge/pen/qKLxk>

Conditionals

How Are Websites Made?

According To Designers



How are websites made?

Making Decisions

It's either TRUE or FALSE

If you are greater than 18 you are an adult

```
if ( age > 18 ) {  
    console.log('You are an adult');  
}
```


Comparisons

Are two things equal?

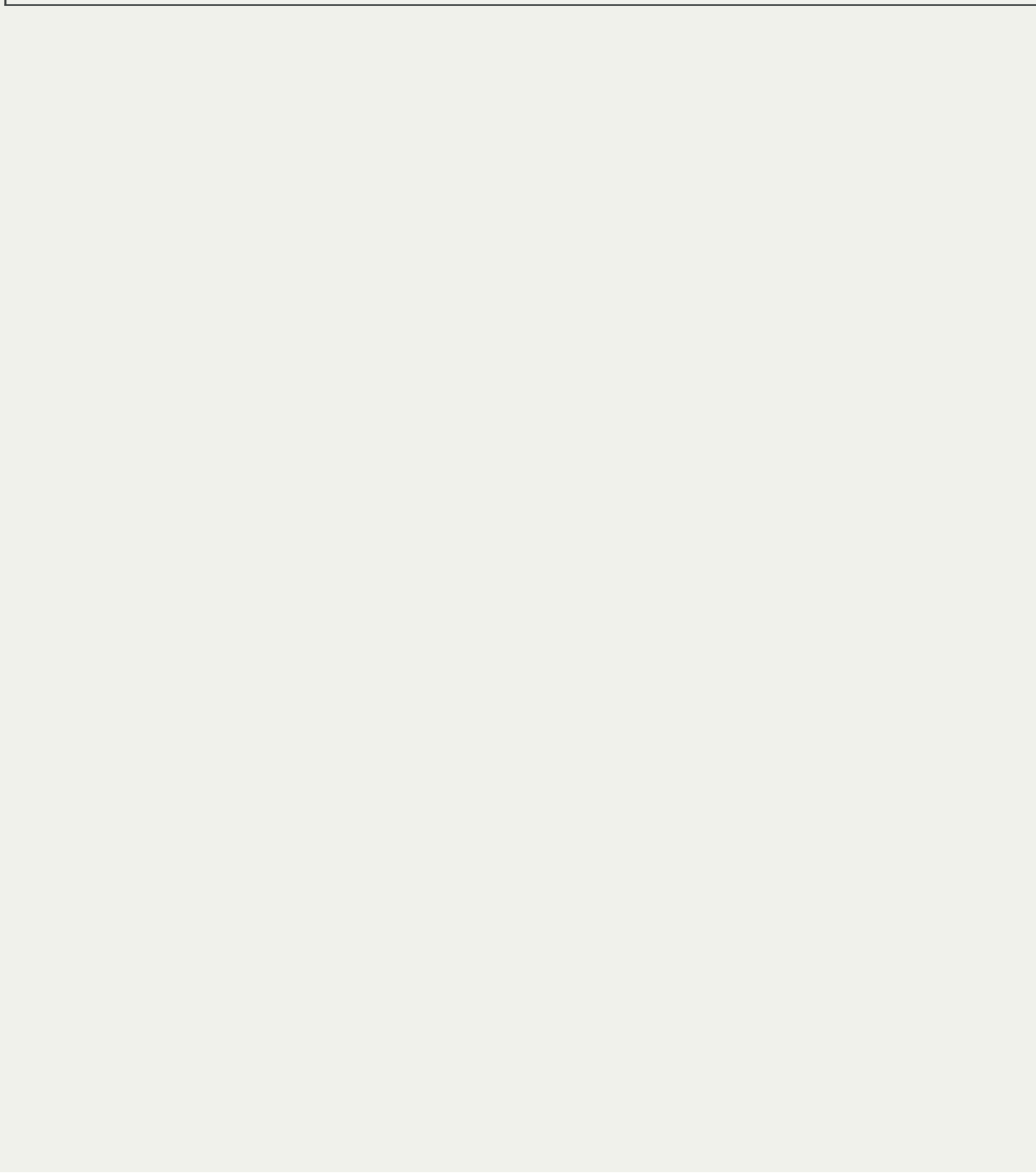
```
10 == 10 // true
10 == 5  // false
"hi" == "hi" // true
"hi" == 'hi' // true
```

Equality?

Would it be useful to check anything other than equality?

x = 3

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



Conditional Syntax

```
if ( conditionIsTrue ) {  
    // Do stuff  
}
```

Conditional Syntax

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

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


The Truth Table

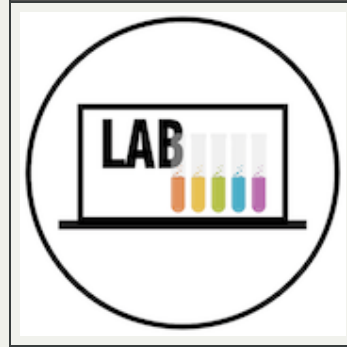
AND - &&	TRUE	FALSE
TRUE	true	false
FALSE	false	false

The Truth Table

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

The Truth Table

OR - 	TRUE	FALSE
TRUE	true	true
FALSE	true	false



Temp Converter

