JS INTRO

WITH GUY ROUTLEDGE

@GUYROUTLEDGE | #FEWD

OBJECTIVE

Learn the building-blocks of JavaScript so we can add more interaction to the page.

AGENDA

- Feedback
- The DOM
- Events
- Functions
- Variables
- Conditionals

WHAT IS JAVASCRIPT?

WHAT IS JAVASCRIPT?

- Programming language in the browser (mostly)
- Created in 10 days back in 1995 by Brendan Eiche
- Used to add interaction to the page
- Used to add complex functionality
- Often driven by user initiated events
- Very powerful
- Nothing to do with Java

JAVASCRIPT IS NOT JAVA

A SIMPLE EXAMPLE

http://codepen.io/guyroutledge/pen/aOEYgb

REAL WORLD EXAMPLES

- FAQs http://www.kallo.com/faqs/
- Menus http://www.chivas.com/en-gb/the-venture
- The Food Rush http://thefoodrush.com/
- Scroll effects http://www.eljimador.com/gb/our-tequilas/
- Modal popups https://www.yeovalley.co.uk/
- Selector tool http://lumix4k.panasonic.co.uk/

PROGRAMMING

Programming is a bit like a following a recipe. There's a series of steps to execute in a certain order that produce a particular outcome.

The process of programming takes an abstract concept and breaks it down into a series of small, logical, bite-sized steps which are then written in code and finally executed - usually by the browser in the case of JS.

THE DOM

THE DOM

JavaScript interacts with our HTML document. It can read info on the page and make changes to any element on the page.

To learn how to do this we need a common language to describe the construction of the document and the elements within it.

THE DOM

The **Document Object Model** is a representation of the page used by JS to interact with all the elements.

THE DOM

We can visualise the DOM as a tree like structure of each element and their heirarchy.

But the DOM is more than just a representation of the HTML. Each element on the page is an object that has certain characteristics and can perform certain actions.

THE DOM

Many programming languages use the concept of objects to model real world behaviour and JS is no different.

THE DOM

Imagine a pen. It is an object. It is made up of a lid, the pen tip and the ink catridge. This is an object that is made up of multiple parts - just like a web page.

THE DOM

A pen can be described in a couple of ways:

- The pen is red
- The pen can write

We can describe the object as having properties (like its colour) and methods (actions like writing)

THE DOM

We have the same thing in JavaScript.

The document is an object and has a method for finding elements within it.

Those elements are also objects which have properties like their style or inner text content.

Elements also have methods for actions like getting, setting or removing attributes.

THE DOM

When writing interactive JS we need to know 3 things

- The element(s) we want to interact with
- The event we want to kick off the interaction
- The execution: what happens next and what elements are involved

We can write this out in "normal" language first - a process called pseudo code.

EVENTS

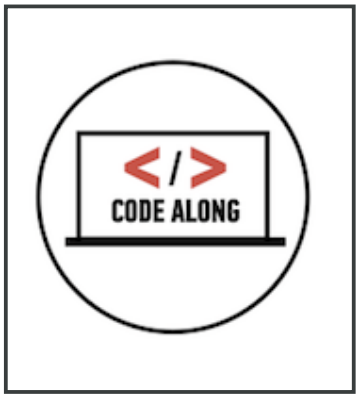
There are many events that occur in the browser as the user interacts with the page:

- click
- submit
- resize
- scroll
- hover
- touch
- drag and drop

EVENTS

We tell JavaScript to listen for these events and then trigger a bit of code (called a function) when the event occurs.

The function contains a series of instructions which allow us to build up more complex behaviour.



JS TRAFFIC LIGHTS

http://codepen.io/nevan/pen/GgMXEm

VARIABLES

VARIABLES

- We can tell our program to remember values for us to use later on.
- Their values can change over time (ie. they can vary)
- The action of setting a variable value is called assignment
- The action of getting the value from a variable is called accessing the variable

CREATING VARIABLES

To create a variable for the first time we use the **`var`** keyword and name the variable. We set its value with the **`=`** operator.

```
var score = 0;
```

To reference the variable we just call it by its name. Its value will be returned back to us.

```
score; // returns 0
```

VARIABLE RE-ASSIGNMENT

To change the value in an already created variable, we just reference it by name.

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

VARIABLE CONVENTIONS

- Variable names start with a lower case letter or _ or $ character
- Variables can't start with a number
- If they contain multiple words, subsequent words start with an upper case letter. This is known as camelCase.

```
var numberOfStudents = 10;
```

VARIABLES & DATA TYPES

We can store lots of different types of information in variables.

- `String` - for text
- `Integer` 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"
```

To have a string that contains apostrophes or air quotes, we need to use the opposite type of quote marks or the string will break into multiple parts.

STRING TO NUMBER

```
var amount = "4";
typeof( amount ); // returns "string"

amount = parseInt( amount, 10 );
typeof( amount ); // returns "number"

var pi = "3.14159";
typeof( pi ); // returns "string"

pi = parseFloat( pi );
typeof( pi ); // returns "number"
```

Sometimes we need to grab some text (a string) from the page and turn it into a number so we can do mathmatical operations with it.

NUMBER TO STRING

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

We can also convert from numbers to strings although this is less common.

NUMBERS

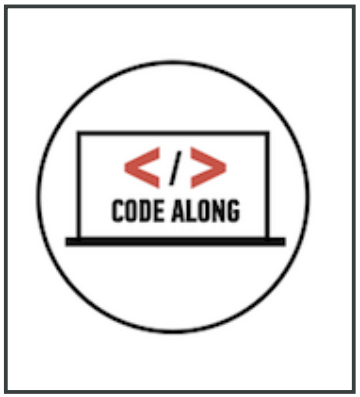
Represent numerical data in either whole numbers or decimals.

```
integer:    42
decimal:    3.14159265
```

Whole numbers are called integers and decimals are called floating point numbers.

ARITHMETIC IN JAVASCRIPT

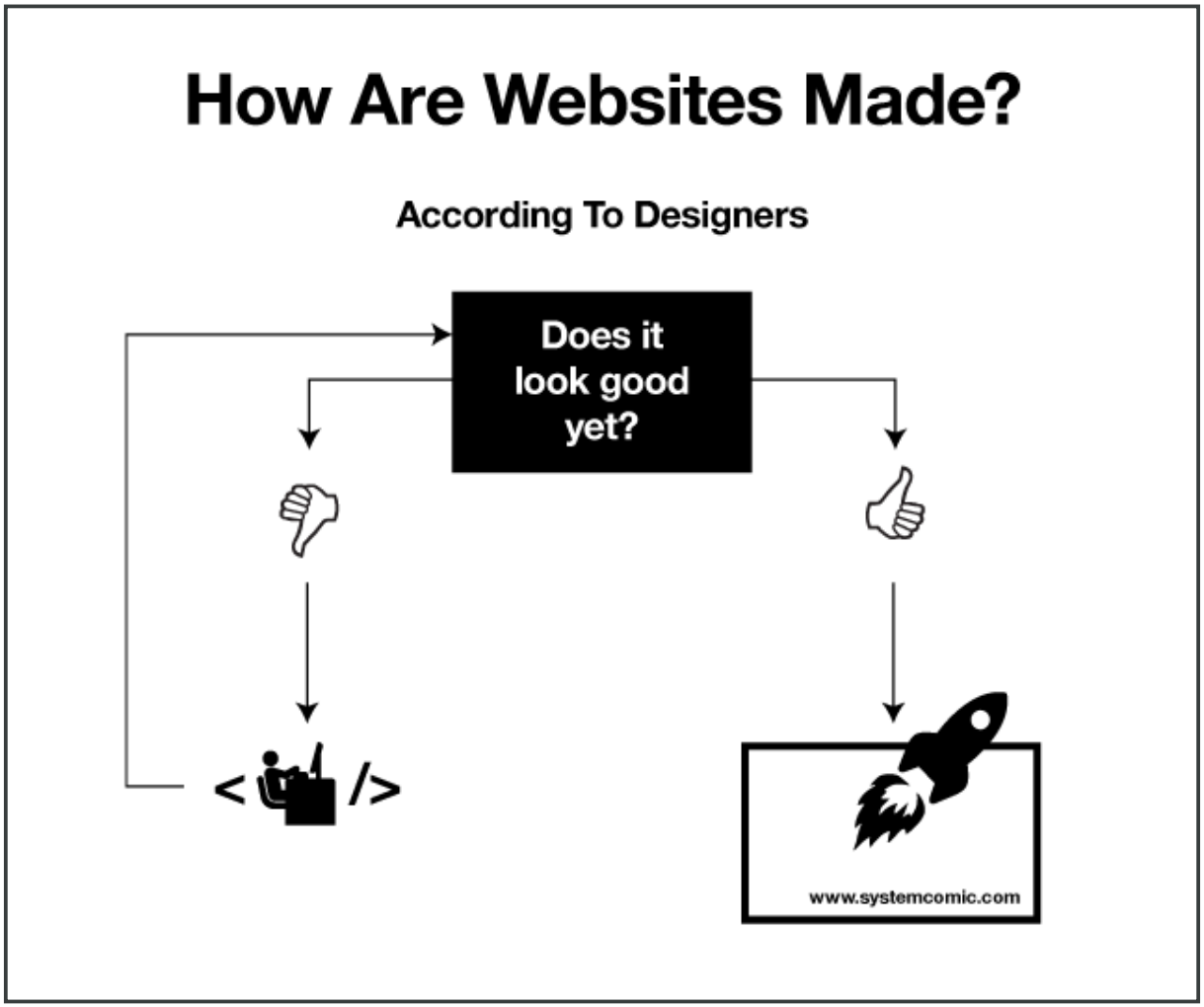
```
var a = 43;
var b = 10;

a + b; // returns 53
a - b; // returns 33
a * b; // returns 430
a / b; // returns 4.3
a % b; // returns 3
```



SCORE KEEPER

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

CONDITIONALS



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 // returns true
10 === 5  // returns false
```

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

We can make the condition more strict with a logical **AND** operator. For the whole if statement to be true, both conditions must be true.

```
if ( name == "GA-Guest" && password == "schooldinner"
    // Allow access to internet
}
```

MULTIPLE CONDITIONS

We can make the condition more relaxed with a logical **OR** operator. For the whole if statement to be true, only one of the conditions must be true.

```
if ( day === "Tuesday" || day === "Thursday" ) {
    // Exercise
}
```

TEMP CONVERTER