HTML = Hypertext Markup Language
Uses tags to define content
 (content)

Brackets
Modern text editor for markup and code

! Tab is format and p tab is paragraph

World Wide Web
Standard based
Published by W3C
World Wide Web consortium
W3Schools.com provides lessons and reference

<h1-6> titles and subtitles they order the importance
Only one h1
<b> is bolding
<i> is italics
 looks the same as bold
 is emphasize looks like italics
Nested is next to each other

Cascading style sheets
Three location external sheet, style tag, or in html style class
Starts with a selector and ends with a soft bracket
55 different selectors
Inside the soft bracket you change properties
Last rule takes precedence
Never use inline styles
Select the html tag or select by class or an id which has to be unique to the page

CSS selectors
:Hover defines how the element is styled and how it looks
:Active used to style and select active link used with hover
:Checked marks or identifies the boxes or circles that the person has clicked on
: default selects the default form of a group of related elements

:Attribute select elements with a specified attribute
:Enabled matches every enabled element
:First-child used to select the specified selector only if its the first
:only-child matches every element that is only child
Element element everything in the first tag will be affected
:link would style links to pages you haven't visited
:not selects an element excludes it to being applied
::placeholder selects elements with placeholder text
*selects all elements
#id selects a specific element with the id tag
: :selection matches a portion of an element selected by the user
:lang used to select elements with lang attribute
:first-of-type matches every element that is the first child
:last-of-type matches every element of the last child
:empty selects every tag that is empty
:nth-of-type selects every element that is the nth child
:nth-last-child used to match the elements with color
:nth-child every nth child regardless of type
:First-of-type the first of that type
::first-line every first line of p
:optional selects form elements which are optional
:only-of-type selects element that is an only child
:required only works with form elements
:target element link jumps to a targeted element
:in-range specifies a value within a range
:indeterminate when a box is undetermined
:disabled targets disabled elements (elements that are not intractable with)
:last-child targets the last element of a parent tag
Element selects all elements with a specified name
element , element selects different elements simultaneously
:focus selects elements that have a focus
:valid selects the elements with a value
:invalid selects elements that do not validate
attribute$=value selects certain elements with an attribute at the end.

http-equiv=”X-UA-Compatible” content=”ie=edge” for backwards compatibility
name=”viewport” content=”width=device-width, initial-scale=1.0”
Responsive design
@media only screen and (max-width: ___px) 1024 common tablet 640 common phone
@media come after the normal css and in decreasing size
Max-width means up to the px

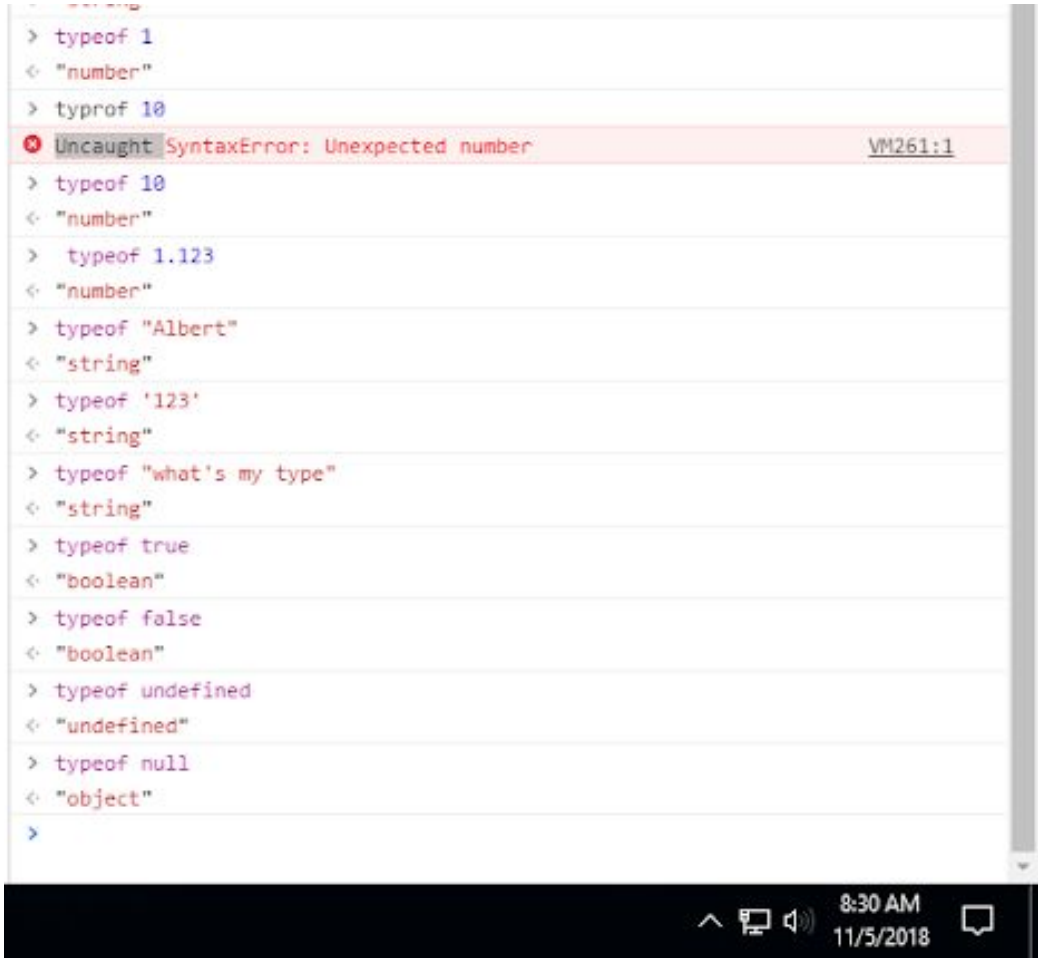
Form tag gets info from the visitar
Items go inside
Method get prints all info
Use method post
Impust is self closing and has a type
Must have labels for all inputs

Html content car model: the frame or chassis
Css style and layout car model: body and paint
Javascript functionality, interactivity, animation car model: motor
Many names mocha livescript jscript **EMCAScript**
Most misunderstood programing language toy language simple to start powerful once mastered
Difference between a scripting language and programing language is scripting are interpreted and programming are compiled
Made in **1995** Brendan **Eich** at **Netscape**
Released in early 1996
Livescript was renamed due to **java** **not the same**
A few months after microsoft released the functionally equivalent jscript language with Internet explorer 3
Netscape submitted javascript to ECMA international
ECMAScrit is the official name

Console.log displays requested info in the console useful for debugging code
Be more explicit with a back slash

Explain javascripts data types
Use the typeof operator to explore data types
All data is bits 1 and 0
We use data types for describing different bits of info
We can develop some expectations as to how the data can be used
2 is a number “2” is a string
The + changes from an add to a concatenate which pushes together
Order does not affect
2345 we think a sequence of numbers or 2345
Javascript can't do that it can only see numbers
Anything is “ ” is a string
we provide context
Typeof 2 // ”number” typeof “2” //”string” typeof ‘2’ // “string”
Typeof is a shortcut

Primitive types boolean(true or false) undefined(not set) null(nothing)



Javascript treats all numbers as if they have decimal points even if they don't have one
Strings are straightforward they are collections of characters
We have to escape quotation marks when they're inside of a string
When we wrap a string in double quote marks we don't need to escape single quotes
Adding strings is concatenation
We can insert strings in strings- interpolation
Template literals act like strings but wrapped in backticks
The whole string is wrapped in backticks
Part that is interpolate is wrapped in ${}
Signals that the interpreter should evaluate what's inside

Using the console is good for testing functionality
Functions can be run again and again
Written with the function keywords
Function keyword followed by a name followed by parentheses then curly braces
Run by typing the name followed by ()

Argument to pass info to a function
Parameters are placeholders between parentheses
When we call the function we can pass arguments to it
Doesn't matter what it is called it only matters where it is placed
Only logging info in their local scope
By wrapping the return in quotes its letting us know it's a string

Run code under certain conditions
When we check for this statement we check for true or false
Var light = prompt ("is the light green, yellow, or red?")
If (L== G){
go()
}Else if(L==Y){
slow()
}Else{
stop()
}
If the thing is the () is true then it runs what's in the {}
Most comparisons come straight from math
<
>
<=
>=
Can't use = because that a setting for the value of a variable
Exact comparison is ===
Match value and type
== tries to coerce values and my not compare what you expect
Use ===
&&(and) and ||(or)
With && both statements must be true in order for the entire expression to be true
With || only one need to be true
Reads from left to right
Returns last statement
Only evaluates as many as necessary
In && if first is false won't look at second
In || if first is true then won't look at second
5===5&&1 returns 1
5===4&&0 returns false because it stops evaluteing
200<100|| 'alphabet' returns 'alphabet'
200>100||'treasure' returns true doesn't check the right side
Lets us control what blocks of code to execute using if else if and else

if(something){
//(B.O.B) do something
}
If Truthy (so the boolean true or anything other than the empty string 0 false null or undefined) the code runs' if not it's skipped
Ifs are used with an else clause
Else will execute if all other statements are false
Else if this is like an else but will only run if the condition is true and the previous is false
You can add an else after all statements
Anything after return won't get executed
We can use this to make code terser
Ternary is a shortcut for if-else
Tests a condition if true evaluates the left side of the : otherwise it evaluates the right
conditionToTest ? valueToBeReturnedIfTrue : valueToBeReturendIfFalse
Switch statements are bif if/else if/else chains
The value of the expression is compared to the values of each case
switch(expression){
Case n://if case n is true
break //will stop processing
Case m:// if case m is true
Default: //all other cases
Return will halt execution
Return exits the black and returns a value break exits a block and does not have a value

How we select and manipulate html with javascript
It changes what we see not the actual html
Making changes to the dom we can change the way it displays even without changing the html
Current view of the browser can be manipulated without reloading a page
Html never changes after rendered
We can select a specific piece using javascript
I could erase my website.
Have draggable things

When a page loads its not javascript
The document is in tree form
Nodes
The highest level is the window next is the document
The window object has a large number of properties
innerWidth and innerHeight
The document object represents any page in the browser

Contains all the nodes
Use it to traverse the html and manipulate elements
All returns all nodes
contentType returns type of content
Add using appendChild
Remove using removeChhild
Can be called on any node
Node called element element.attributes access its attributes
Anc remove with removeAttribute
Modify the nodes style property
Listen for key presses or mouse evnents
addEventListener
getElementByTagName
All tags on page
querySelectorAll
All queries on the page
getElementById
the element with that id
getElementByClassName
Gets the elements with that class
Html class is className in javascript

```
.flex-container{
Display: flex;
}
```

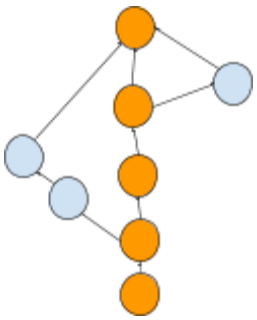
A script is a series of instructions
Like a recipe or handbook or manual
Recipes some are simple other are more complex and a lot of terminology
Hand books include steps for different scenarios
Manuals follow steps to check systems like an if statment
Scripts are made of instructions a computer can follow step-by-step
First state goal and list tasks needed to complete goal
Start with big picture and break into smaller parts
1 define the goal
2 design the script split the goal out into a series of tasks
3 code each step each step need to be written in a language the computer can understand
It pays to spend time designing
Need to get to grips with the vocab and syntax
Computers are logical and obedient
Need to be told every detail and will do it without question

Need to think like a computer

Git removes the need to copy files to and from the class share
Git is like taking a snapshot of your files at a specific point in time
Git is a checkpoint for you files
modify/ change/ break/ improve your code
A collaboration tool that allows different people to work on all parts of a project
Protects you from yourself and others
Creates a repository in the folder you ran the command on
Modified files that are new or changed that have not been saved
Staged the current version of the file and commit
Committed files that are safely stored by Git
Git init to start
Git tracks changes by git status
Git add (file)
Commit the box to storage and note what it contains
Until we commit there is no checkpoint
Moving into longer term storage
It does not move or remove files from the working directory

Remote repository
Push local files to remote files
Learn branches
Learn to merge
How to fix a merge conflict
Remote repository is a “cloud”

Do not push every time
Branches are smaller bits of info
They are different versions of code
Branches allow us to fix code without breaking the master
Fixes and new features should always start on a branch
Master is trunk
Should only contain clean code



Git branch makes a new branch
Git checkout changes branch
Git merge (branch) merges branches together
Merge conflict is when file has changed in both of the branches
To fix delete code you don’t want and then add and commit

It can help by being able to work on the same document at once. You can work on the same document and be able to work without messing each other up. You can also work on a file that's not the main file so if you mess something up you still have working code. You will also be able to see what work your partner has done and be able to implement similar lines of code into the master.

3

No questions