JavaScript

BASIC JavaScript

BASIC Syntax:

```
<script>
//JavaScript Code goes here
</script>
```

CAN MIX JavaScript & HTML

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My first PHP page</h1>
    <script>
       alert("Hello World!");
    </script>
  </body>
</html>
```

SEMICOLON

Put a semicolon after each line.

```
alert("hi");
var name = "Someone";
```

COMMENTS

Start with a // for single line and then start with /* and end with */ for multiple line comments.

```
// I am a single line comment
var animal = "cat";
/* I am a multiple line comment
that goes for several lines because
I have a lot to say */
```

VARIABLES

NAMING CONVENTIONS

Variable names:

- Contain letters, numbers and underscores
- Case matters. "Cat" & "cat" are 2 different variables.
- No reserved words like "prompt" or "alert".
- No spaces!

DECLARE A VARIABLE – states that it exists

```
var number_kids;
var class_name;
```

DECLARE & INITIALIZE A VARIABLE give a variable it's value when you create it.

```
var number_kids = 10
var class_name="PHP"
```

ASSIGN A VARIABLE – change the value it holds

```
number_kids = 5;
class_name = "JavaScript";
```

OUTPUT & INPUT

OUTPUT

- `alert` will create a popup box that gives a message to the user– can output text and variablesc

```
alert("Hello World!");
```

- `document.write()` will print out a string – can concatenate, can display HTML Markup. Output text and variables

```
document.write("Hello World!");
```

INPUT

- `prompt` will create a popup box that asks for input from the user. Returns a string value!

```
var name = prompt("What is your name?");
```

- `confirm` asks the user for their okay before procedeeding. It returns two values: **True** if they click OK and False if they click Cancel.

```
var more = prompt("Do you want to try again?");
```

MATH

+	Addition	3 + 2	5
–	Subtraction	4 – 5	1
/	Division	10 / 5	2
*	Multiplication	4 * 3	12
**	Exponentiation	2 ** 3	8
%	Remainder	5 % 3	2

ORDER

If no `()` – does multiplication, then division, addition and subtraction. Use `()` to change order of operations.

Converting

`parseInt("2")` – converts string 2 to number 2

`.tostring()` – converts number to string value

```
var num = 11
var snum =
num.toString()
```

CONDITIONALS

```
if (condition is true){
   codeline1
   codeline2
}elseif (condition1 is true){
   codeline3
}elseif ( condition2 is true ){
   codeline4
}else{
   codeline5
}
```

STRINGS

CONCATENATE– add strings together with + operator

```
phrase = "hello" + " "+"you" =>  "hello you"
```

DATA TYPES

```
Number
String
Boolean: true or false
Array
Object
```

WEB PAGE TEMPLATE w/ JavaScript

```
<!DOCTYPE html>
<head>
  <title>title</title>
  <link rel = "stylesheet"
href="style.css"></link>
  <script
src="script.js"></script>
</head>
<html>
<body>
   <!-- CONTENT GOES HERE
→
   <script>
     alert("Hi all!");
   </script>
</body>
</html>
```

OPERATORS

LOGICAL OPERATORS

==	Equal To
===	Equal To & Equal Type
! Or <>	Not Equal To
!===	Not Equal To or equal type
>	Greater Than
<	Less Than
>=	Greater than or Equal to
<=	Less than or Equal to

BOOLEAN OPERATORS

&&	and	all conditions are True
||	or	any one of the conditions is true
!	Not	if a single logical statement is not true

JavaScript 2 Sheet

FINDING DOM ELEMENTS

By ID: This allows you to access an element by it's CSS ID selector using document.getElementById. If you have the following HTML.

```
<h1 id="topHeader">My First Web Page</h1>
<p>A simple paragraph</p>
```

Access the element by:

```
var topHeader =
document.getElementById('topHeader');
```

By Tag Name: Access it using an HTML tag such as <p>,<h1>...

Access the element by:

```
var i = document.getElementByTagName('li');
```

By Class Name: This allows you to access an element by is class name. If you have the following HTML.

```
<h1 id="topHeader">My First Web Page</h1>
<p class="top">A simple paragraph</p>
<p>Another paragraph</p>
```

Access the element by:

```
var top = document.getElementByClassName('top');
```

CHANGING HTML

Once you have found an element – you can change it using the `innerHTML` property. Basic syntax: `document.getElementByid(id).innerHTML = "new text"`

```
<h1 id="topHeader">My First Web Page</h1>
<p>A simple paragraph</p>
```

Change header by using this bit of JavaScript

```
<script>
  document.getElementById('topHeader').innerHTML="New Header";
</script>
```

CHANGING ATTRIBUTES

Once you have found an element – you can change it using the `style` property. Basic syntax: `document.getElementByid(id).style.attribute = "new value"`

Change color of header by using this bit of JavaScript

```
<script>
  document.getElementById('topHeader').style.color="#FF5722";
</script>
```

NOTE: If the style is one word – like `color`; `margin`; `padding` – you just use that word. If they style has dashes in it – like `background-color`; `font-size` – you use the camel case version so `backgroundColor` or `fontSize`.

EVENTS

Events in HTML that JavaScript can react to. These include a web page loading, button or element is clicked., form is submitted, mouseover.

Simple onclick() Event

```
<h1 id="header" onlick="document.getElementById('header').innerHTML = 'New Header'">My Header</h1>
```

Using a Button with an onclick() Event

```
<h1 id="header">My First Web Page</h1>
<button onlick="document.getElementById('header').style.color='#FF5722';">Click Me</h1>
```

CONDITIONALS

```
if (condition is true){
   codeline1;
   codeline2;
}else if (condition1 is true){
   codeline3;
}else if (condition2 is true ){
   codeline4;
}else{
   codeline5;
}
```

FOR LOOPS

BASIC SYNTAX

```
for (start_value, condition, step_value){
   //run code
}
```

BASIC LOOP EXAMPLE

```
for (var i=0; i<3; i++){
    document.write(i);
}
#prints 0, 1, 2
```

FOR LOOP & ARRAYS

```
#prints out each value in array
var bugs=['fly', 'ant', 'moth'];
for(i=0; I < bugs.length, i++){
    document.write(bugs[i]);
}
```