

Javascript Exercises

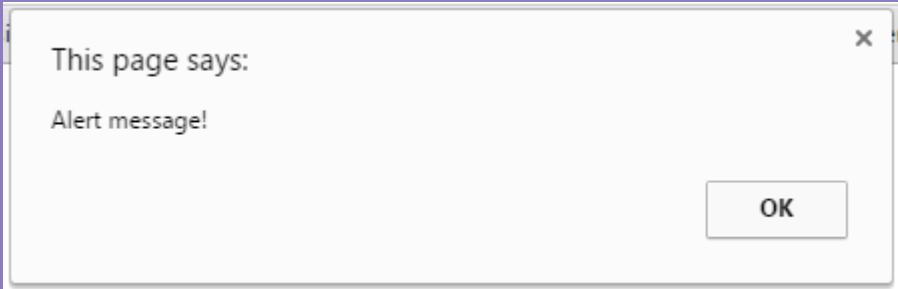
Exercise 1: Output using Javascript

Part A

Explanation

Alert

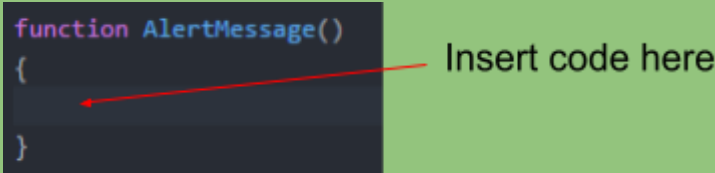
Alert messages will appear in a popup and have to be dismissed



The code looks like this:
`alert(“Alert message!”);`

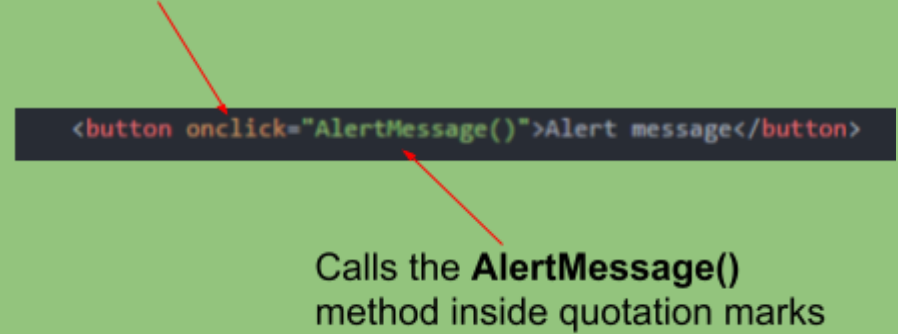
Do this

In the Javascript file, find the **AlertMessage()** function.
Write code to output an alert popup saying “**You have been Alerted!**”



In the HTML document, the code for the button is here:

Uses the **onclick** attribute



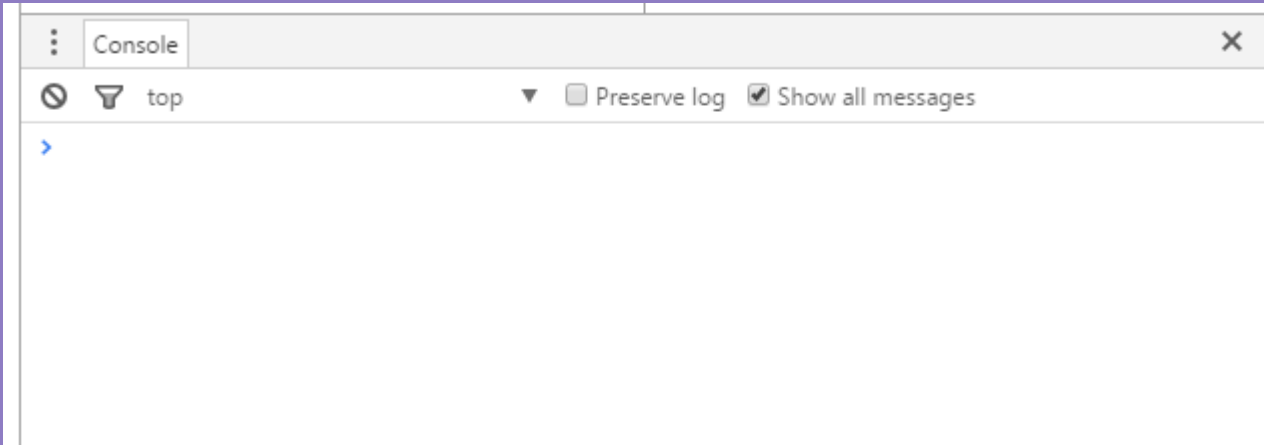
Links

Alert: https://www.w3schools.com/js/js_popup.asp

Explanation

Console output

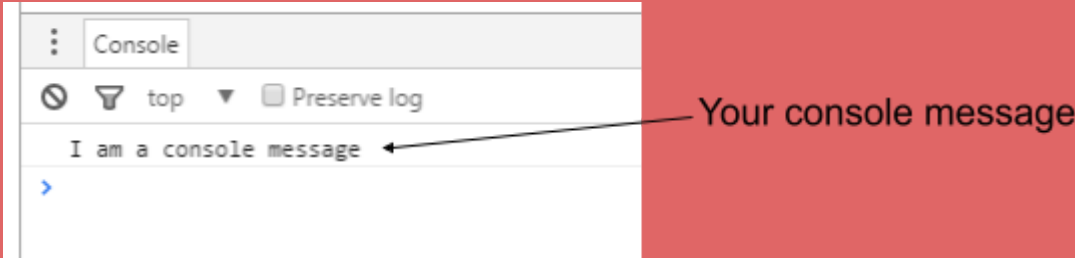
The console is part of the Google Chrome Web Dev tools.
Open the tools by pressing **F12** or **Ctrl + Shift + I**
With the tools open, press the Esc key to toggle the console open, it looks like this:



Check this

Leave the tools and the console open as you do this exercise.

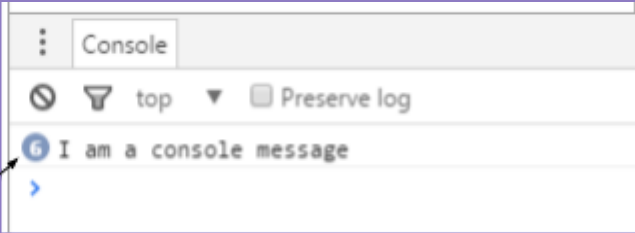
Console messages look like this:



Explanation

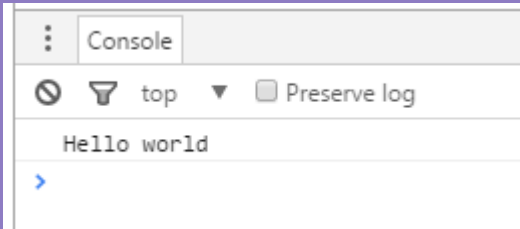
If more of the same message are sent to the console, it will show a count of the messages:

6 of these messages have been sent to the console



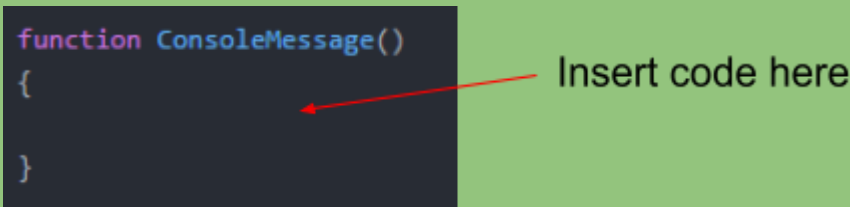
To write the message “Hello world” to the console, you would type this:
`console.log(“Hello world”);`

NOTE: Javascript uses a dot (.) to connect the console to the log function (e.g. `console.log()`)



Do this

In the Javascript file, find the **ConsoleMessage()** function.
Write code to output the message “**Console is active**”.



In the HTML document, the code for the button is here:

Onclick attribute

```
<button onclick="ConsoleMessage()">Console message</button>
```

Calls the ConsoleMessage() function

Links

Console log: https://www.w3schools.com/jsref/met_console_log.asp

Part C

Explanation

Javascript is written like a set of instructions.
Each line is a different instruction
Each instruction ends with a semicolon (optional, but good practice as it's not optional in C#):

```
var a = 1;  
var b = 3;  
console.log(a + b);
```

The code above creates 2 variables (data containers), then adds them together and logs the answer to the console

Explanation

Variables

Variables are containers for data, that can be used anywhere in your program after they are created.
They are created using the “var” keyword.

```
var a = 1;
```

A few types of variables are:

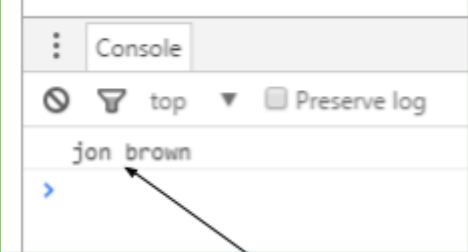
- Numbers (whole and decimal)
- Strings (text, contained in **single** or **double** quotes)
 - **var name = “bob”;**
 - **var surname = ‘brown’;**
- Booleans (TRUE or FALSE)
 - **var gamesRule = true;**

Do this

Create 2 variables called **firstName** and **lastName** with the values **Jon** and **Brown**
Each variable needs to have **NO SPACES** in the **value**, like this:

- **var firstName = ‘jon’;**

Output both names with a space in between to the **console**
NOTE: the space needs to be inserted with the “+” operator



Space in between the names

Links

String operators: https://www.w3schools.com/js/js_operators.asp

Exercise 2: Accessing the document using Javascript

Part A

Explanation

Replacing text with innerHTML

Javascript can change text within a tag.
To get the tag (also called an element), Javascript needs to get the whole HTML document, using the code “**document**”
Javascript can then search through the HTML document by:

- Tag name
 - `document.getElementsByTagName();`
- Class name
 - `document.getElementsByClassName();`
- Id
 - `document.getElementById()`
- CSS selectors
 - `document.querySelectorAll()`
- And more

Our code to get a tag by ID would look like this:
`document.getElementById(“myid”);`

Do this

- In the Javascript code, find the **ReplaceText()** function
Write code to change the text of the <p> tag in the HTML file:
1. Get the <p> tag using the getElementById function and giving it the id name of the <p> tag
 2. Get the text using **innerHTML**
 3. Set the text to “**My new text!**”

```
function ReplaceText()
{
}

```

Insert code here

Function call

```
<button onclick="ReplaceText();">Replace the text below</button>
<p id="placeholderText">placeholder text!</p>

```

Id of placeholder tag Text to be replaced

Links

Accessing elements using document: https://www.w3schools.com/js/js_htmlDOM_document.asp

Explanation

Adjusting styles dynamically

You can change a tag’s **CSS** styles using javascript.
We can get the tag by id using **document.getelementByld()**.
We can add the word **style** to access the CSS styles for that tag - **document.getElementById(“myid”).style**
A CSS style can then be accessed, like colour - **document.getElementById(“myid”).style.color**
We can output the colour by using the code above inside our console.log() function or other means
Or we can change it using the **assignment operator “=”** - **document.GetElementByld(“myid”).style.color = “green”**
NOTE: put the new style inside of quotation marks

Do this

- In the Javascript code, find the **MakeTextSmaller()** function
Write code to change the text of the <p> tag in the HTML file:
1. Get the <p> tag using the getElementByld function and giving it the id name of the <p> tag
 2. Get the CSS styles using **style**
 3. Add the style fontSize to the end
 4. Assign the new size using the “=” operator
 5. Put the new size inside of double quotes
 6. Make sure the size has a “px” on the end for pixels:
 - a. “50px”

```
function MakeTextSmaller()
{
}

```

Insert code here

Function call

```
<button onclick="MakeTextSmaller()">Make the text smaller</button>
<p id="textSize">Make me smaller!</p>

```

Id of placeholder tag Text to be reduced

Links

DOM CSS: https://www.w3schools.com/js/js_htmlDOM_css.asp

Explanation

Setting Attributes

Just like CSS styles, you can change a tags attributes using Javascript.
Same as before, using a tags id, you can use the code `document.getElementById(“myid”)` to get the tag
Add the name of the attribute you wish the change (like “width”) - `document.getElementById(“myid”).width`
Set the new value using the assignment operator “=” `document.getElementById(“myid”).width = “50px”;`
NOTE: make sure to put the new value in quotation marks, adding px on the end if the value is in pixels

Explanation

Linking to files

When using other files on a web page, like images, we need to link to the folder the file is in.
If an image file is in the same folder as the html file, just link using its name - `src=“myimage.jpg”`
If an image is in another folder, use the folders name - `src=“images/myimage.jpg”`

Do this

In the Javascript code, find the **TurnLightOn()** function
Write code to change the **src** of the tag in the HTML file:

1. Get the by id using the getElementById() function
2. Get the src attribute of the tag
3. Set the src attribute to the folder of the image, plus the **bulb on** image name, plus the image type
 - a. NOTE: check the type of image is correct (.png or .jpg)

Do this

Find the **TurnLightOff()** function

4. Get the by id using the getElementById() function
5. Get the src attribute of the tag
6. Set the src attribute to the folder of the image, plus the **bulb off** image name, plus the image type

Id of placeholder tag

Attribute to be changed

```

<br />
<button onclick="TurnLightOn()">Turn light on</button>
<button onclick="TurnLightOff()">Turn light off</button>
```

Function calls

Links

DOM CSS: https://www.w3schools.com/js/js_htmldom_css.asp

Exercise 3: Conditionals and functions

Part A

<h3>Explanation</h3> <p>Conditional statements: the If statement</p> <pre>if(condition) { Run this block of code }</pre> <p>NOTE: the “condition” part of an if statement needs to evaluate to true or false, (health equals zero, weapon is firing) NOTE: your notes for SOFT152 on conditionals will apply here, conditionals work the same way in Javascript!</p> <p>For more than one condition, you can use the else statement:</p> <pre>if(condition) { Run this block of code } else { Run this block of code instead }</pre> <p>For more than 2 conditions, you can use the else if statement NOTE: you don't have to use an else statement last if it isn't required</p> <pre>if(condition 1) { Run this block of code } else if (condition 2) { Run this block of code instead } else { Run this block of code if the first two conditions were false }</pre> <p>NOTE: the code will only run one of the blocks of code, depending on the condition</p>
<h3>Do this</h3> <p>Change the src attribute of the lightBulb image tag using an if statement inside of the ToggleLight function HINT: use the isLightOn variable to check if the lightBulb should be on or off HINT: read the previous examples on how to change the src attribute of an img tag</p> <pre>if(isLightOn == true) { Set lightBulb src to bulb-on } else { Set lightbulb src to bulb-off }</pre> <p>NOTE: you must also set the isLightOn variable to true or false to see results!</p> <div><p>Use this id</p><pre> <button onClick="ToggleLight()">Toggle Light</button></pre><p>Use this function</p></div>
<p>Use this variable in your if statement</p> <pre>var isLightOn = false; function ToggleLight() { }</pre> <p>Write your code in here</p>
<h3>Links</h3> <p>JavaScript if conditional: https://www.w3schools.com/js/js_if_else.asp</p>

Part B

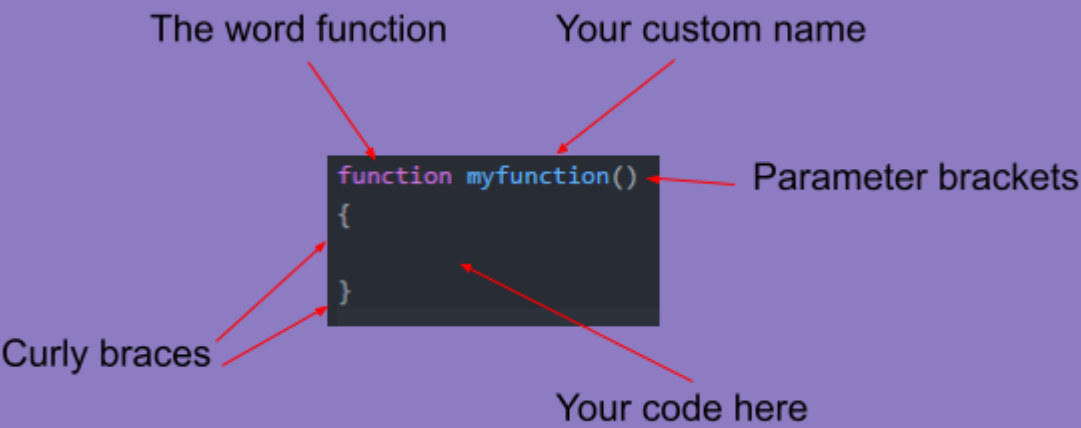
Explanation

Make a function to add numbers together and output the result

Functions

A function is a block of code that can be reused throughout your program.

Anatomy of a function



Function parameters

Functions can also take variables to use inside of the function, these are called **parameters**. Functions can have as many parameters as you want, but may get a little confusing if more than 5 or so are present. NOTE: make sure to name your parameters to something useful! (e.g. remaininghealth, damage etc)

Parameters can be named (nearly) anything you want

```
function myfunction(param1 param2 param3 param4)
{
  console.log(param1 + param2);
  myvar = param3;
  document.getElementById(param4);
}
```

Some ways of using parameters in a function

Do this

In the Javascript file, write a function called **AddNumbers**, with 2 parameters, called **numA** and **numB**

The function should have 2 parameters, matching the ones below

```
<button onClick="AddNumbers(10,20)">Add Numbers</button>
<br />
<p id="numbers">numbers added</p>
```

Output to this element using the id shown

The output should replace this text

Links

Functions: https://www.w3schools.com/js/js_functions.asp

Exercise 4

Explanation

Use the form elements to create a player for a game

- Player name
- Player colour
- Player health
- Player weapon

Player name

Uses an **input** element of type **text**

Player colour

Uses an **input** element of type **color**

Player health

Uses an **input** element of type **number**

Player weapon

Uses a **select** element with **option** tags for each weapon

Do this

Add code in the **CreatePlayer** function to output the values from the **form section** into the **player details section** of the web page

NOTE: Do not change the html file, only add code to the javascript file

Form section

Player name:

Player colour:

Player health:

Player weapon:

Crossbow of much hurting ▼

Submit

Player details section

Player details

player name

Player health

Player weapon

Player name

Get the the text typed in by the user from the input tag

Use this to get the tag

```
<input type="text" id="playername" />
```

The value attribute contains the user input text

```
document.getElementById('playername').value;
```

Output the player name text to the **player details section** into the element with an **id** of **name**

Player colour

Get the colour selected by the user from the input tag

Use this to get the tag

```
<input type="color" id="playercolour"/>
```

The value attribute contains the selected colour

```
document.getElementById('playercolour').value;
```

To show the colour in the **player details** section, set the **background-color** of the element with an **id** of **colour**

Player health

Get the health number entered by the user in the input tag

Use this to get the tag

```
<input type="number" id="playerhealth"/>
```

The value attribute contains the user entered health number

```
document.getElementById('playerhealth').value;
```

Output the player health number to the **player details** section into the element with an **id** of **health**

Player weapon

Get the selected weapon value

The selection will be a number, 1, 2 or 3

```
<option value="1">Crossbow of much hurting</option>
<option value="2">Broadsword of so slicing</option>
<option value="3">Wand of amaze magics</option>
```

Using an **if statement** output the name of the weapon to the element with an **id** of **weapon** in the player details section

```
if( weapon value == 1 ) {
    Output "Crossbow of much hurting"
}
else if( weapon value == 2 ) {
    Output "Broadsword of so slicing"
}
else {
    Output "Wand of amaze magics"
}
```

A Challenge

Instead of using an if statement how would you select a weapon using a switch statement?

[Switch statements in w3Schools](#)