TASK

# JavaScript Refresher

Visit our website

# Introduction

## WELCOME TO THE JAVASCRIPT REFRESHER TASK!

You've come a long way in such a short space of time -- well done! After a journey through HTML, CSS, Java, SQL, PHP and WordPress, we're now coming full circle; back to JavaScript. Before we get into the advanced concepts of using JavaScript to create full stack web applications, let's do a quick refresher on the fundamentals that you learned in level 1.

## BASIC JAVASCRIPT SYNTAX

This is a brief overview of the fundamentals of JavaScript syntax and structures as a quick reference guide for you for the upcoming tasks.

**Variables:**

```
let greeting = "Hello";       // 'let' means the variable can be reassigned
const question = "How are you?";    // 'const' means the variable cannot be reassigned
```

**Conditional statements:**

```
if (greeting === "Hello" && question === "What's the time?") {
      console.log("It is 10 'o'clock.");
}
else if (greeting === "Hello" || question === "What is the weather like tomorrow?") {
      console.log("It will be sunny tomorrow.");
}
else {
      console.log("I don't understand the question.");
}
```

**Arrays and loops:**

```
let classMarks = [62,88,90,70,69,85];

for (i=classMarks.length-1; i >= 0; i--) {
  console.log(classMarks[i]);
}   // outputs each element reversed

for (let number of classMarks) {
  console.log(number);
}     // outputs each element

classMarks.forEach(function(number) {
  console.log(number+2);
});        // adds 2 to each number and outputs it

let counter = 0;
let j = 0;

while (counter <= 4) {
  console.log(classMarks[j]);
  counter ++;
  j++
}   // outputs the first 5 elements in the array
```

Some array methods can be found below:

- *indexOf()* - Searches the array for the given element and returns its position
- *includes()* - Checks if the array contains a given element
- *remove()* - Removes an item from the list
- *pop()* - Removes the last element in the array
- *push()* - Adds an element to the end of the array and returns the new length
- *shift()* - Removes the first element in the array
- *sort()* - Sorts elements in the array in ascending order
- *splice()* - Adds or removes elements in an array
- *reverse()* - Reverses the order of elements in the array

**Maps:**

```
let teachers = new Map();
teachers.set("grade 1", "Mrs Johnson");
teachers.set("grade 2", "Mr Bennett");
for (let [key, value] of teachers) {
  console.log(key + ' = ' + value);
}
// grade 1 = Mrs Johnson
// grade 2 = Mr Bennett

for (let key of teachers.keys()) {
  console.log(key);
}
// grade 1
// grade 2

for (let value of teachers.values()) {
  console.log(value);
}
// Mrs Johnson
// Mr Bennett

for (let [key, value] of teachers.entries()) {
  console.log(`The ${key} teacher is ${value}.`);
}
// grade 1 = Mrs Johnson
// grade 2 = Mr Bennett
```

Some map methods can be found below:

- *has()* - checks to see if a key exists in the map and returns a boolean
- *get()* - returns the value of the given key
- *delete()* - deletes the given key-value pair from the map
- *clear()* - removes all key-value pairs in a map

**Built-in functions:**

Some common built-in functions include:

- *charAt()* - returns a character in a string at the given index
- *indexOf()* - returns the index of a the first occurrence of a character in a string
- *charCodeAt()* - returns the ascii number of the character at the given index in a string
- *fromCharCode()* - returns the character of the given ascii number
- *replace()* - replaces a matched substring with a new substring
- *split()* - splits a string into an array of substrings
- *toUpperCase()* - returns the given string all in upper case
- *toLowerCase()* - returns the given string in all lower case
- *join()* - opposite of *split()*. Joins all array elements into a string

- *pow()* - returns a base to the exponent power
- *min()* - returns the smallest valued element
- *max()* - returns the largest value element
- *round()* - returns a number rounded to the nearest integer

**User-defined functions:**

```
function doubleNumber(number) {
    return number * 2;
}
console.log(doubleNumber(10));  // outputs 20
```

## EVENTS

Some common events in DOM manipulation include:

- *onchange* - Some HTML element has been modified
- *onclick* - An HTML element has been clicked on
- *onmouseover* - An HTML element was hovered over
- *onmouseout* - Mouse cursor moves off HTML element
- *onkeydown* - A keyboard button is pressed
- *onload* - The HTML page has finished loading

These can be built into functions, for example:

```
function closeDoc(){
      alert("You are closing this page!");
      window.close();
}
```

## JQUERY

**Functions:**

```
$(function(){
   //jQuery events and effects go here
);
```

**Selectors:**

```
$("h1")            // element selector
```

```
$("#mainHeading")   // ID selector
$(".yellow")        // class selector
$("*")              // selects all HTML elements
$(this)             // selects current HTML element
$("button")         // selects button element
$("[href]")         // selects all link-based elements
```

**Creating and appending elements:**

```
let  newParagraph = $("<p></p>").text("This is a new paragraph.");
// creates a new paragraph
$("body").append(newParagraph);     // appends newParagraph to the end of
the body element
```

Common built-in functions in jQuery include:

- *after()* - Inserts content after the selected HTML element
- *append()* - Inserts content at the end of the selected element
- *before()* - Inserts content after the selected HTML element
- *empty()* - Removes the child elements from the selected element, but not the element itself
- *prepend()* - Inserts content at the beginning of the selected element
- *text()* - sets or returns the text content of the selected element
- *val()* - Sets or returns the value of a form field
- *remove()* - Removes the selected element, including its child elements

**Separate Scripting Files:**

```
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
</script>
    <script src="script.js"></script>
</head>
<!-- Used to create a scripting file incorporating jQuery -->
```

**Events and effects:**

```
$("p").click(function(){
    $("this").hide('slow');
});
```

Common events in jQuery:

- Mouse events:
    - Click
    - Double click
    - Mouse enter
- Keyboard events:
    - Key press
    - Key down
    - Key up
- Form events:
    - Submit
    - Change
    - Blur

**Effect chaining:**

```
$("button").css("color", "blue").slideDown(500).slideUp(500); // example of effect chaining
```

Common effects in jQuery:

- *Hide* - Hides the selected element
- *Show* - Reveals the selected element
- *Fade* - Fades the element in or out
- *Slide* - Slides the element
- *Animate* - Animates some property of the element
- *Stop* - Stops any effect of an element

Note that this is a summary of some of what you covered in level 1 and is, therefore, not an exhaustive list. Feel free to go back and brush up on anything you feel unsure about, and be sure to contact an expert code reviewer if you get stuck.

## Compulsory Task

Follow these steps:

- Create a folder called **Back to JS**.
- Using the information above (and anything else that you find useful) create an interactive website that forms a tutorial of JavaScript concepts.
- Your website should have separate web pages for each of the headings above, where each web page explains the concepts and allows the user to practise their knowledge.
- Be sure to style your website and make it user-friendly.

Rate us

## Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

**Click here** to share your thoughts anonymously.