# Assignment 1 - JavaScript Refresher

Due Date: **10th April, 2022 23:59:59**

1. Define a variable called **count** with an original value of 0 and then increment it (add 1 to it) on the following line.
2. Define a variable **ageLimit** that cannot be re-assigned and give it a value of **18**.
3. Implement the function **canVote** such that it returns true whenever the age is 18 or above and false in all other scenarios.

```
function canVote(age) {
  // write your solution here
}

// sample usage
console.log(canVote(10))  // false
console.log(canVote(19)) // true
```

4. Complete the function **evenOrOdd** such that it returns the string "even" when the number parameter it receives is even and "odd" otherwise.

```
function evenOrOdd(age) {
  // write your solution here
}

// sample usage
console.log(evenOrOdd(-1)) // odd
console.log(evenOrOdd(10)) // even
```

4.1 Use the **ternary operator** for your conditional check in your function above.

5. In **sumGrades** function, use the **Array.reduce()** method to calculate and return the sum of all the grades it receives as a parameter.

```
function sumGrades(grades) {
  // write your solution here
}


// sample usage
console.log(sumGrades([15, 5, 10])); // 30
console.log(sumGrades([12, 10, 13, 19])); // 54
```

6. Complete the function **getCountriesDropdown** such that it returns the following HTML that will fill an existing **<select></select>** element. The HTML that's returned should look like the following:

    1. It should start with: **<option value="">Please select</option>**

    2. Then, for every country you should have the **<option>** for that. For example, for the country "The Gambia", you should have the following **<option value="the gambia">The Gambia</option>**

    Notice how the value contains the country name in lower case. The remaining countries will need to show up as well following the same criteria.

```
const getCountriesDropdown(countries) => {
  // put your solution here;
}

const countries = ['The Gambia', 'Senegal', 'Mali', 'Guinea Conakry',
'Nigeria', 'Sierra Leone'];

console.log(getCountriesDropdown(countries));

// Your output should look like below. Does not have to be formatted
like mine though:
/*
  <option value="">Select Country</option>
  <option value="the gambia">The Gambia</option>
  <option value="senegal">Senegal</option>
  <option value="mali">Mali</option>
  <option value="guinea conakry">Guinea Conakry</option>
  <option value="nigeria">Nigeria</option>
  <option value="sierra leone">Sierra Leone</option>
*/
```

7. Using an **Arrow** function and the **Array.filter()** method, convert and complete the function **getPositiveTemperatures** such that it returns an array containing the positive temperatures (the temperatures that are above 0).

```
function getPositiveTemperatures(temperatures) {

}

// Sample usage
console.log(getPositiveTemperatures([-5, 12, 3])); // [12, 3]
console.log(getPositiveTemperatures([1, -3, -2, 4, 10])); // [1, 4,
10]
```

8. Complete the function renderTableRows such that it returns the following HTML:

```
<tr>
  <td>label here</td>
  <td>value here</td>
</tr>
```

for every row that it receives in its rows parameter.

The rows parameters looks like the following:

```
[["Carbs", "17g"], ["Protein", "19g"], ["Fat", "5g"]]
```

This is a nested array. Every inner array contains 2 items, the first one referring to the label that you should replace instead of **label here** and the second one referring to the value which you should replace instead of **value here**.

Make sure to use console.log() every step of the way to visualize what you're working with.

```
function renderTableRows(rows) {
  // put your solution here
}

const rows = [["Carbs", "17g"], ["Protein", "19g"], ["Fat", "5g"]]
consoel.log(renderTableRows(rows));

// Sample output
/*
  <tr>
    <td>Carbs</td>
    <td>17g</td>
  </tr>
  <tr>
    <td>Protein</td>
    <td>19g</td>
  </tr>
  <tr>
    <td>Fat</td>
    <td>5g</td>
  </tr></tbody>
*/
```

9. Complete the function getProductDetails such that it returns an object with the following properties:

   - id: a number representing the id of the product.
   - title: a string representing the title of the product.

- inStock: a boolean representing whether the product is in stock or not.

Give these properties any value you'd like, as long as it fits the expected return type.

```
function getProductDetails() {
  // put your solution here
}

// sample usage
console.log(getProductDetails())  // the object you defined
```

10. Complete the function **getWeather** such that it returns the string:

```
It's currently X degrees in Y
```

Where **X** is replaced by the value and **Y** is replaced by the city name. These values are provided as a city parameter which is an object.

You are expected to use the **Template String Literal (Interpolation)** to build your returned string.

```
function getWeather(city) {
  // put your solution here
}

// Sample usage
console.log(getWeather({name: "Amsterdam", value: 3})); // "It's currently 3 degrees in Amsterdam"
console.log(getWeather({name: "Serrekunda", value: 28})); // "It's currently 28 degrees in Serrekunda"
```

By Omar Jasseh

Lead Instructor