

Iterations - Counter

(<https://github.com/witzakr/Counter.git>)

1. Preparation

The assignment was to create a website featuring a counter that starts from the number 1 and keeps counting upwards endlessly. Additionally, the website had to display random text alongside the counter. Our task was to ensure that both the counter and the text were visible and readable by adjusting their appearance using CSS.

For my website, I chose the phrase "hello world" as the text to display. To make it more challenging, I decided to translate this phrase into a different language every second. I used an API to handle the translations.

2. Process

a) HTML

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Counter</title>
7      <link rel="stylesheet" href="styles.css">
8      <script src="script.js"></script>
9  </head>
10 <body>
11     <section>
12         <h3>0</h3>
13         <p></p>
14     </section>
15 </body>
16 </html>
17
```

After adding the necessary elements, h3 would be the counter and the 0 would be replaced by +1 value and p would be replaced by hello world and its translations.

b) CSS

```
body {  
  margin: 0;  
  padding: 0;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
section {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  margin-top: 450px;  
}
```

```
section h3 {  
  font-size: 400px;  
  position: absolute;  
  animation: bounce 2s infinite;  
  text-shadow:  
    -2px -2px 0 #000,  
    2px -2px 0 #000,  
    -2px 2px 0 #000,  
    2px 2px 0 #000;  
}
```

```
section p {  
  font-size: 200px;  
  position: absolute;  
  text-shadow:  
    -2px -2px 0 #000,  
    2px -2px 0 #000,  
    -2px 2px 0 #000,  
    2px 2px 0 #000;  
}
```

```
@keyframes bounce {  
  0%, 100% {  
    transform: translate(0, 0);  
  }  
  25% {  
    transform: translate(50px, -50px);  
  }  
  50% {  
    transform: translate(100px, 0);  
  }  
  75% {  
    transform: translate(50px, 50px);  
  }  
}
```

Once I finished writing the HTML code for the website, it was time to style it using CSS. To ensure that both the numbers and the text were easy to read, I added an outline around them. Additionally, I made the numbers bounce in a clockwise direction to add some visual interest.

0

c) JS

```
1 document.addEventListener('DOMContentLoaded', function() { // Loading DOM content before executing
2
3   var counterElement = document.querySelector('h3');
4   var counterTextElement = document.querySelector('p');
5
6   // Initializing variables
7
8   var counterValue = 0;
9   var translations = []; // Translations array that will get data from API in the later stage
10  var translationIndex = 0; // Index to keep track of current translation
11
12  // Function to fetch translations from an API
13
14  function fetchTranslations() {
15    fetch('https://mocki.io/v1/27da8bc0-4c9f-4a18-812f-e07a6bca0441')
16      .then(response => response.json()) // Parsing the JSON response
17      .then(data => {
18        translations = data.translations.map(entry => entry.translation); // Storing translations in the translations array and choosing the right entry
19      })
20      .catch(error => {
21        console.error('Error fetching translations:', error);
22      });
23  }
24
25  // Function to update the counter and display random colors
26
27  function updateCounter() {
28    counterValue++;
29    counterElement.textContent = counterValue;
30
31    // Generating random colors for the counter and text
32
33    var randomColour = '#' + Math.floor(Math.random()*16777215).toString(16); // Generating random color hex code, 16777215 is equal to FFFFFFFF, so its able get every colour
34    counterElement.style.color = randomColour; // Setting the color of the counter element
35    counterTextElement.style.color = randomColour; // Setting the color of the text element
36
37    // Generating random background color
38
39    var randomBackgroundColour = '#' + Math.floor(Math.random()*16777215).toString(16);
40    document.body.style.backgroundColor = randomBackgroundColour;
41
42    // Displaying translations
43    counterTextElement.textContent = translations[translationIndex]; // Setting the text content to the current translation
44    translationIndex = (translationIndex + 1) % translations.length; // Updating the translation index for the next translation
45  }
46
47  // Call fetch of translations
48
49  fetchTranslations();
50
51  // Fetching translations every second
52
53  setInterval(fetchTranslations, 1000);
54}
```

I won't delve too much into the details of the JavaScript code here, as it's already explained in the readme file. But briefly, the code makes the counter functional, translates 'hello world' into 10 different languages (within a loop fetching data from an API), and changes counters, backgrounds, and text colors randomly every

second. Fixing text errors."

3. The result and final thoughts

Overall, the teacher and I were satisfied with the result. I received very positive feedback for it, and although using the API wasn't necessary, it was greatly appreciated. This assignment motivated me for the further assignments of this semester.



Bonjour 123 le monde