JavaScript
Async Await

Jose Alba Arrufat

Index

Actividad 1

Hacer una calculadora que saque la letra del DNI con función de try catch para depuración de errores.

HTML:

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>DNI Calculator</title>
</head>
<body>
<form name="formulario">
 DNI Calculator:<br>
<input type="text" name="dni" maxlength="8" id="dni"><br><br>
<input type="button" value="Submit" id="calc">
</form>
<script src="script.js"></script>
</body>
</html>
```

JavaScript:

```javascript
function getDNInumber(dni) {
let word = "TRWAGMYFPDXBNJZSQVHLCKET";
let position = dni % (word.length - 1);
if(word[position] !== undefined){
return alert(dni + word[position]);
}
}

document.getElementById('calc').addEventListener('click', function() {
let dni = document.getElementById('dni');
onlyNumbers(dni);
let word = getDNInumber(dni.value);
})

function onlyNumbers(dni){
try{
if (isNaN(dni.value)) {
throw new InputMismatchError('Input is not a number');
}
}catch(err){
return alert('That isn\'t a valid DNI');
}
}
```

Actividad 2

Hacer un formulario con checks de que producto quiere hacer el usuario y sacar el precio total que ha de pagar este

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Generate client budget</title>
<style>
table {
border-collapse: collapse;
}
td, th {
border: 1px solid #dddddd;
text-align: left;
padding: 8px;
}
tr:nth-child(even) {
background-color: #dddddd;
}
</style>
</head>
<body>

<form name="form">
<h3>Processors:</h3>
<input type="radio" name="pro" value="300" checked/>Intel i7 (300€)<br><br>
<input type="radio" name="pro" value="200"/>AMD Ryzen 5 (200€)<br><br>
<input type="radio" name="pro" value="250"/>ARM Cortex-A76 (250€)<br><br>
<h3>Graphic Card:</h3>
<input type="radio" name="gp" value="400" checked/>Nvidia GTX 1660 (400€)<br><br>
<input type="radio" name="gp" value="600"/>Nvidia RTX 3060 (600€)<br><br>
<input type="radio" name="gp" value="500"/>AMD Radeon RX 5700 (500€)<br><br>
<h3>Box:</h3>
<input type="radio" name="box" value="50" checked/>NZXT H510 (50€)<br><br>
<input type="radio" name="box" value="100"/>Corsair 275R (100€)<br><br>
<input type="radio" name="box" value="80"/>Fractal Design Meshify C (80€)<br><br>
<h3>Refrigeration:</h3>
<input type="radio" name="refr" value="80" checked/>Cooler Master Hyper 212 (80€)<br><br>
<input type="radio" name="refr" value="150"/>NZXT Kraken X62 (150€)<br><br>
<input type="radio" name="refr" value="120"/>Corsair H100i (120€)<br><br>
<h3>Monitor:</h3>
<input type="radio" name="moni" value="150" checked/>Dell 24" (150€)<br><br>
```

```
<input type="radio" name="moni" value="300"/>LG 27" 4K (300€)<br><br>
<input type="radio" name="moni" value="200"/>ASUS 27" 144Hz (200€)<br><br>
<input type="button" value="Submit" id="calc">
</form>
<script src="script.js"></script>

<div id="cost"></div>
</body>
</html>
```

JavaScript:

```
document.getElementById('calc').addEventListener('click', function() {
// Get the selected values
let proValue = document.querySelector('input[name="pro"]:checked').value;
let gpValue = document.querySelector('input[name="gp"]:checked').value;
let boxValue = document.querySelector('input[name="box"]:checked').value;
let refrValue = document.querySelector('input[name="refr"]:checked').value;
let monitorValue = document.querySelector('input[name="moni"]:checked').value;

// Convert values to numbers and calculate the total
let total = Number(proValue) + Number(gpValue) + Number(boxValue) +
Number(refrValue) + Number(monitorValue);

// Display the total
document.getElementById('cost').innerHTML = `
<br>
<table>
<tr>
<th>Processor</th>
<th>Box</th>
<th>Refrigeration</th>
<th>Monitor</th>
<th>Graphic Card</th>
</tr>
<tr>
<td>${proValue}</td>
<td>${boxValue}</td>
<td>${refrValue}</td>
<td>${monitorValue}</td>
<td>${gpValue}</td>
</tr>
<tr>
<td colspan="2"> <strong>Total</td>
<td colspan="3"> ${total}
</table>`;
});
```

Actividad 3

Crear un array donde se muestren libros y que nada más habrás el navegador, haya un delay de 2 segundos para que se muestre.

Usar promesas.

Los libros han de tener los siguientes valores → | title | year | isbn | author |

HTML:

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Library</title>
<style>
table {
border-collapse: collapse;
}
td, th {
border: 1px solid #dddddd;
text-align: left;
padding: 8px;
}
tr:nth-child(even) {
background-color: #dddddd;
}
</style>
</head>
<body>
<table id="booksTable">
<tr>
<th>Title</th>
<th>Year</th>
<th>ISBN</th>
<th>Author</th>
</tr>

</table>

<script src="script.js"></script>
</body>
</html>
```

JavaScript:

```javascript
function TheLibrary(){
return new Promise((arrayOfBooks) => {
setTimeout(() => {
arrayOfBooks([
{ title: "The Great Gatsby", year: 1925, isbn: "978-0743273565", author: "F. Scott Fitzgerald" },
{ title: "To Kill a Mockingbird", year: 1960, isbn: "978-0060935467", author: "Harper Lee" },
{ title: "1984", year: 1949, isbn: "978-0451524935", author: "George Orwell" },
{ title: "The Catcher in the Rye", year: 1951, isbn: "978-0316769488", author: "J.D. Salinger" },
{ title: "Moby-Dick", year: 1851, isbn: "978-1503280786", author: "Herman Melville" }
]);
}, 2000);
});
}

TheLibrary().then((arrayOfBooksValues) => {
let bookList = document.getElementById('booksTable');
let tableContent = ``;
for (const item of arrayOfBooksValues) {
tableContent += `
<tr>
<td>${item.title}</td>
<td>${item.year}</td>
<td>${item.isbn}</td>
<td>${item.author}</td>
</td>`
}
bookList.innerHTML += tableContent;
}).catch((error) => {
console.error('Error when u try to fetch the books :(', error);
});
```

Actividad 4

Hacer lo mismo que anteriormente pero con async / await

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Library</title>
<style>
table {
border-collapse: collapse;
}
td, th {
border: 1px solid #dddddd;
text-align: left;
padding: 8px;
}
tr:nth-child(even) {
background-color: #dddddd;
}
</style>
</head>
<body>
<table id="booksTable">
<tr>
<th>Title</th>
<th>Year</th>
<th>ISBN</th>
<th>Author</th>
</tr>

</table>

<script src="script.js"></script>
</body>
</html>
```

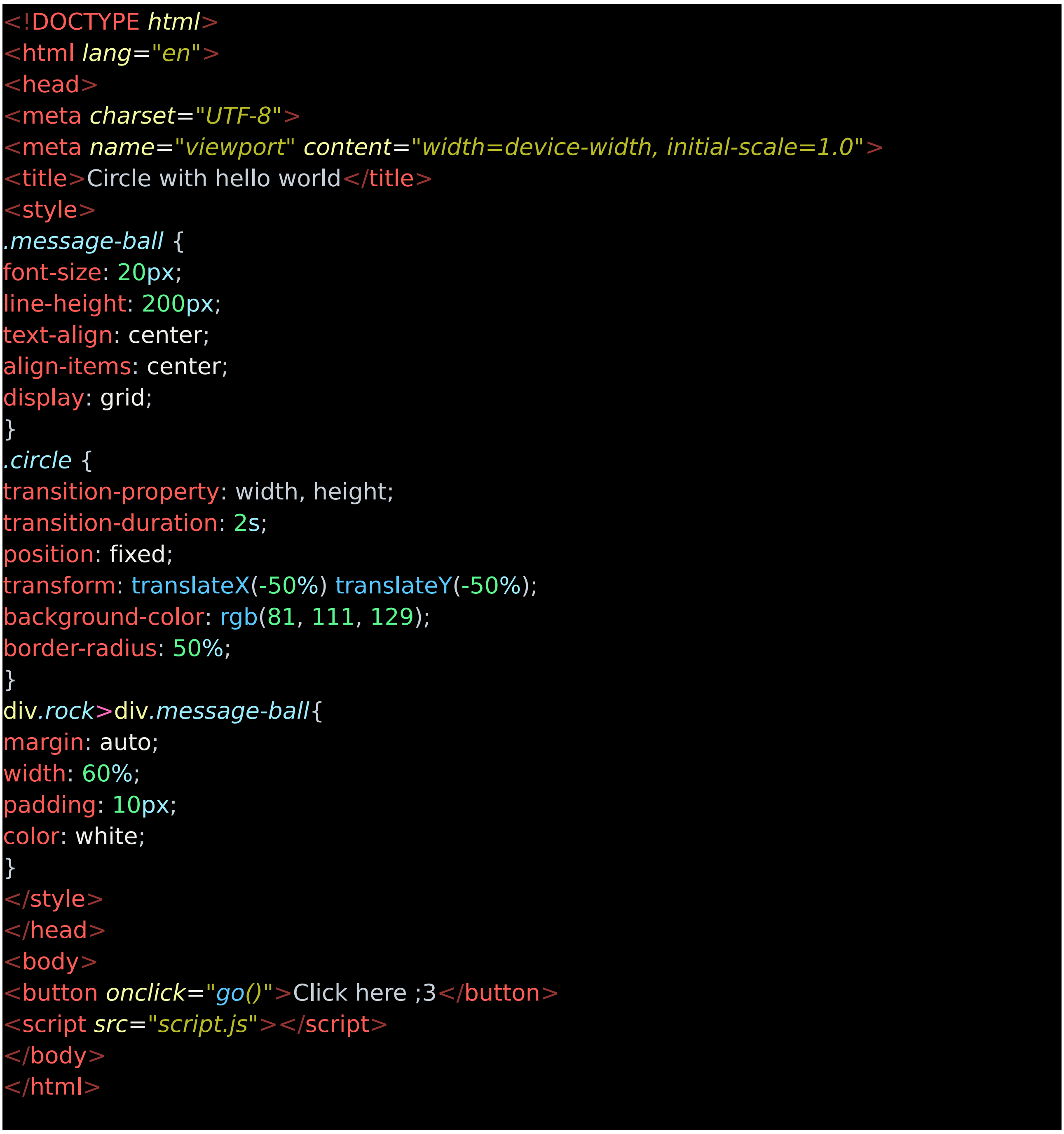
JavaScript:

```javascript
async function TheLibrary() {
return new Promise(arrayOfBooks => {
setTimeout(() => {
arrayOfBooks([
{ title: "The Great Gatsby", year: 1925, isbn: "978-0743273565", author: "F. Scott Fitzgerald" },
{ title: "To Kill a Mockingbird", year: 1960, isbn: "978-0060935467", author: "Harper Lee" },
{ title: "1984", year: 1949, isbn: "978-0451524935", author: "George Orwell" },
{ title: "The Catcher in the Rye", year: 1951, isbn: "978-0316769488", author: "J.D. Salinger" },
{ title: "Moby-Dick", year: 1851, isbn: "978-1503280786", author: "Herman Melville" }
]);
}, 2000);
});
}

async function showTheBooksOfTheLibrary() {
let bookList = document.getElementById('booksTable');
let tableContent = ``;
try {
// This is where yu use the await to wait to finish the execution of the TheLibrary code ;3
let arrayOfBooks = await TheLibrary();
for (const item of arrayOfBooks) {
tableContent += `
<tr>
<td>${item.title}</td>
<td>${item.year}</td>
<td>${item.isbn}</td>
<td>${item.author}</td>
</tr>`;
}

bookList.innerHTML += tableContent;
} catch (error) {
console.error('Error when u try to fetch the books :(', error);
}
}

showTheBooksOfTheLibrary();
```

Actividad 5

Hacer un botón que vaya aumentando su grandaría progresivamente hasta que dentro del circulo se muestre un mensaje.

HTML:



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Circle with hello world</title>
<style>
.message-ball {
font-size: 20px;
line-height: 200px;
text-align: center;
align-items: center;
display: grid;
}
.circle {
transition-property: width, height;
transition-duration: 2s;
position: fixed;
transform: translateX(-50%) translateY(-50%);
background-color: rgb(81, 111, 129);
border-radius: 50%;
}
div.rock>div.message-ball{
margin: auto;
width: 60%;
padding: 10px;
color: white;
}
</style>
</head>
<body>
<button onclick="go()">Click here ;3</button>
<script src="script.js"></script>
</body>
</html>
```

JavaScript:

```
function go() {
showCircle(1000, 450, 100).then(div => {
div.classList.add('message-ball');
div.append("SOY UN CACAHUETEEEEEEEE!!!!!1!111");
});
}
function showCircle(cx, cy, radius) {
let div = document.createElement('div');
div.style.width = 0;
div.style.height = 0;
div.style.left = cx + 'px';
div.style.top = cy + 'px';
div.className = 'circle';
document.body.append(div);
let containerOfMessageBall = document.createElement('div');
containerOfMessageBall.className = 'rock';
containerOfMessageBall.append(div);
document.body.append(containerOfMessageBall);
return new Promise(resolve => {
setTimeout(() => {
div.style.width = radius * 5 + 'px';
div.style.height = radius * 5 + 'px';
div.style.transition = 'width 0.5s ease, height 0.5s ease';
div.addEventListener('transitionend', function handler() {
div.removeEventListener('transitionend', handler);
resolve(div);
});
}, 0);
})
}
```