

Tota la documentació ha de tenir portada, cos del treball i bibliografia (si passa d'una pàgina). El document es lliurará sempre en **format pdf**.

Tarea3. Promeses & Async-Await

Entrega

El lliurament serà per mitjà d' un document pdf o en format zip a la plataforma moodle del centre, segons especifiqui el professor. També és possible compartir-ho a través de Google Drive, en aquest cas, el document ha de tenir l'opció de "Compartir amb qualsevol que tingui l'enllaç". NO CAL QUE COMPARTEIXI EL DOCUMENT AMB L' ADREÇA DE CORREU DEL PROFESSOR. S'ha de lliurar el document/s amb la següent nomenclatura: Apellido1_Nombre_Task3. En cas d' especificar-se una altra nomenclatura s' ha de prioritzar la indicada en l' activitat.

1. Qualsevol pràctica que estigui copiada d' un company o d' Internet **automàticament en tindrà un 0**.
2. El lliurament s' ha de lliurar dins del límit temporal fixat, si no és així i sense tenir una deguda justificació **tindrà automàticament un 0** en la nota de l' esmentada pràctica.
3. Qui no lliuri el document amb el format que es demana (nom i estil) **tindrà una penalització del 25%** de la nota final que obtingui.

Valoració

Es valorarà a nivell de presentació:

- El format
- L' ortografia
- La informació aportada

Es valoraran les activitats de la següent manera:

1. Activitat 1: 20%
2. Activitat 2: 20%
3. Activitat 3: 20%
4. Activitat 4: 20%
5. Activitat 5: 20%

Es valorarà a nivell de preguntes:

- Resultats
- Explicació

Activitats

S'han d'adjuntar tots els elements en un arxiu amb la nomenclatura:

Apellido1_Apellido2_Nombre_T3Practica.zip.

Internament s'han de distingir el codic segons l'activitat, per exemple: a l'activitat 1 → T3Activitat1.* (definim l'extensió segons correspongue).

1. Es te que crear un calculador de DNI (per calcular la lletra del DNI), la funcionalitat consisteix en: Introduir la part numèrica del DNI per a obtindre la lletra corresponen a eixe DNI.

PD: Haureu d'utilitzar la gestió d'errors (try/catch) per poder gestionar errors dins del codic, podeu consultar el PDF del aules associat a "Errores Halding". Per exemple, en cas que en l'entrada sols es puguem valors numèrics i s'introdueix una cadena amb lletres, es deu gestionar el error en eixida.

2. Generar a un usuari un pressupost final per la compra d'un equip informàtic:

Per a poder generar el pressupost final el usuari te que poder seleccionar 3 opcions dins de les següents 5 categories:

- Processador: Selecció mitjançant un radio button entre 3 opcions (de diferents marques). Un exemple:

Selecciona el procesador

- ☒ Intel i7 11700k - 380€
- ☐ AMD Ryzen 7 5800x - 400€
- ☐ ARM - 200€

- Caixa: Selecció mitjançant un radio button entre 3 opcions (cada una amb un preu diferent).
- Refrigeració: Selecció mitjançant un radio button entre 3 opcions (cada una amb un preu diferent).
- Monitor: Selecció mitjançant un radio button entre 3 opcions (cada una amb un preu diferent).
- Gràfica: Selecció mitjançant un radio button entre 3 opcions (cada una amb un preu diferent).

PD: Les 3 opcions com el seu valor es deixa a elecció del alumne.

El pressupost final es mostrarà mitjançant una tabla:

Presupuesto final

Procesador	Caja	Refrigeración	Monitor	Gráfica
0	0	0	0	0
Total		0		

El total del pressupost te que mostrar el total calculat en cada moment segons les opcions elegides per el propi usuari.

Podeu revisar la part de “Activitat 2” en el annexo.

3. Crearem un array de objectes per gestionar un llistat de llibres. Una vegada estigue creat es vol mostrar per pantalla amb un retràs de 2 segons (setTimeout). Es te que tindre en compte el us de les promeses, per poder retornar un valor correcte per eixida y estar preparat per poder retornar el resultat correcte (then → resolve) en cas de que sigue necessari.

PD: Cada objecte que crearem tindra les següents propietats: title, year, isbn (codic del llibre), author. Aquestos valors es poden assignar directament en cada propietat objecte amb valor inventats i al menys es deuen crear 4 objectes.

4. Seguint la activitat anterior, es te que generar la mateixa funcionalitat però ara utilitzant les funcions pròpies de javascript → async/await.

5. Cercle anidat amb promesa: Mitjançant un esdeveniment de botó (en la part “Activitat 5” del annexo trobareu com generar-lo) es te que generar un cercle en la pantalla, que vage incrementant la seu grandària fins acabar mostrant un missatge dins estil: “Hola mundo”.

PD: En la part del annexo teniu tota la informació necessària.

Annexo

Activitat 2

En el cas de que vulguem generar una opció en varios radio buttons, aci es mostre un exemple:

```
<h3> Selecciona el procesador </h3>
<input type="radio" id="radio" name="procesador" value="380"> Intel i7 11700k - 380€<br>
<input type="radio" id="radio" name="procesador" value="400"> AMD Ryzen 7 5800x - 400€<br>
<input type="radio" id="radio" name="procesador" value="200"> ARM - 200€<br>
```

Per a poder generar una tabla podem utilitzar les següents definicions en javascript:

```
tabla = `
    <table id="tabla">
        <tr>
            <th>Procesador</th>
            <th>Caja</th>
            <th>Refrigeración</th>
            <th>Monitor</th>
            <th>Gráfica</th>
        </tr>`;
```

A mes a mes, una vegada definides les columnes que compondran la tabla, es mostrarán els valors de cada una de les columnes, segons les eleccions fetes als radio buttons. A continuació una manera de mostrar-ho:

```
tabla += `
    <tr>
        <td>${procesador}</td>
        <td>${caja}</td>
        <td>${ref}</td>
        <td>${moni}</td>
        <td>${graf}</td>
    </tr>
    <tr>
        <td colspan="2"> <strong>Total</td>
        <td colspan="3"> ${total}`;
```

Mostrem la manera de calcular el valor del processador:

```
if ( input[i].type == "radio"){
    if(input[i].checked){
        procesador = input[i].value;
    }
}
```

Activitat 3

Per a poder gestionar la funció `setTimeout` revisarem els apunts de javascript associats a “Promesas & Async/Await”.

```
1 let promise = new Promise(function(resolve, reject) {
2   // la función se ejecuta automáticamente cuando se construye la promesa
3
4   // después de 1 segundo, indica que la tarea está hecha con el resultado "hecho"
5   setTimeout(() => resolve("hecho"), 1000);
6 });
```

El primer valor retorna el resultat quan s'hage resolt la promesa.

El segon valor fa referencia al temps i en aquest cas 1000 es equivalent a 1 segon.

Per no mostrar valors incorrectes per eixida es necessari que la funció “`setTimeout`” estigue directament relacionada amb una promesa, per a que, la execució sigue asíncrona en el temps, poden retornar un valor correcte.

La creació d'una promesa va associada a 2 parametres: `resolve`, `reject`. A mes a mes, del us de les funcions → `then` & `catch`.

Activitat 5

Per poder generar un esdeveniment de botó, usarem la següent instrucció:

```
<button onclick="go()">Haz clic</button>
```

Un exemple per a definir el “`style`” del cercle i el missatge a mostrar:

```
<style>
  .message-ball {
    font-size: 20px;
    line-height: 200px;
    text-align: center;
  }
  .circle {
    transition-property: width, height;
    transition-duration: 2s;
    position: fixed;
    transform: translateX(-50%) translateY(-50%);
    background-color: red;
    border-radius: 50%;
  }
}
```

A continuació s'adjunte funcions que permeten tan generar, ampliar el tamany o com mostrar un missatge dins d'un cercle.

```
function go() {
  showCircle(150, 150, 100).then(div => {
    div.classList.add('message-ball');
    div.append("Hola, mundo!");
  });
}
```

```
}
```

```
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);
```

```
  return new Promise(resolve => {  
    setTimeout(() => {  
      ➔ Faltaria definir les sentencies per ampliar progressivament la  
        grandària del cercle (ràdio del mateix)  
  
      div.addEventListener('transitionend', function handler() {  
        div.removeEventListener('transitionend', handler);  
        resolve(div);  
      });  
    }, 0);  
  })  
}
```