

Ejercicio final - evaluación M02 C09

JavaScript básico - Evaluación

Introducción:

El Proyecto **Typer D.** tiene varios objetivos:

1. Que pueda integrar los diferentes conceptos de JavaScript y los utilice en la creación de un juego interactivo que favorece a la escritura rápida y precisa.
 - En este Proyecto trabajará con Funciones, Arreglos, Eventos y manipulación de DOM.
2. Que aprenda algunas funcionalidades como generar palabras randoms y actualizar oportunamente un temporizador.
3. Que logre una herramienta muy útil para mejorar sus habilidades para la redacción y el tipeo.

Metodología:

A la vista del usuario, este juego interactivo se desarrolla de la siguiente manera: recibirá una palabra en inglés y la tendrá que tipear en un input en el menor tiempo posible.

Si la palabra tipeada coincide con la recibida, pasa a la siguiente y se le asigna un tiempo extra.

Al empezar la partida, el usuario tendrá 10 segundos para resolver las palabras que se le presentan. A medida que acierta, ese tiempo se modificará a su favor.

Veamos un ejemplo sobre cómo podría quedar este [proyecto](#):

Cuando comprendas las distintas reglas del juego avance a la siguiente sección.

HTML:

En este Proyecto le daremos la estructura de la página para que se concentre fuertemente en su funcionalidad. Más adelante podrás darle los estilos que prefieras.

Para eso, seguí este paso a paso:

1. Descargue el siguiente archivo [HTML](#) y trabaje con su Visual Studio Code.
2. Cree sus archivos `.css` y `.js` y asegúrese de que estén bien linkeados en su código `HTML`.

Función `randomWords`:

En esta sección del Proyecto creará una Función que devuelva las palabras randoms que necesitamos.

Para eso, vas a arrancar con este Arreglo en tu archivo `.js`:

```
const words = [  
  'californication',  
  'plataforma5',  
  'black',  
  'summer',  
  'flea',  
  'aeroplane',  
  'peppers',
```

OTEC PLATAFORMA 5 CHILE

```
'unlimited',  
'arcadium',  
'love',  
'getaway',  
'stadium',  
'quixoticelixer',  
'quarter',  
'snow',  
'dylan',  
'zephyr',  
'funky',  
'chili'  
];
```

Como se habrá dado cuenta, se trata de las palabras que aparecerán en el juego.

Importante: reference los elementos [HTML](#) con su [JS](#). Tené en cuenta los nombres de las `class` y de los `id` cuando declares las distintas variables.

A continuación seguí este paso a paso:

1. Cree la Función `randomWords` y ejecutala en la consola. Antes de avanzar, asegúrate que retorne una palabra aleatoria.

2. Declare la variables `palabraAleatoria`, `time` (que empiece en 10) y `score` (que empiece en 0) .

-
3. Escriba la función `addToDOM` que le agrega una palabra *random* al `h1` e invóquela.

PISTA: `palabraAleatoria` tiene que ser igual al retorno de la Función `randomWords`.

Revisá si, de momento, el Proyecto te está quedando así:

 **TYPYR 3000** 

Escribi la siguiente palabra:

californication

Tiempo restante: 10s

Score: 0

Evento de tipo `Input`:

En esta sección tendrá que crear un Evento que tome lo escrito por el usuario en el `input` y que, a su vez, chequee la coincidencia con la palabra random.

Para eso realizarás un Evento de tipo `input`. Seguí este Paso a Paso para realizarlo:

1. Escriba el Evento que se aplica al `input` referenciado de tipo `text`.
2. Cree la Función del Evento que tenga un parámetro "`e`" (que usarás más adelante).

PISTA: Si quiere ver qué es este parámetro, haga un `console.log(e)` y escriba algo en su `input`.

3. Cree una variable que se llame `palabraIngresada`.
4. Compare las variables `palabraIngresada` y `palabraAleatoria`. Pruebe que todo funcione correctamente.

Si las palabras coinciden, haga que suceda lo siguiente:

- Que la variable `time` aumente `3 segundos`.
- Que el `input` vuelva a quedar vacío.
- Volver a invocar la Función `addToDOM()`.

Manipular el tiempo:

En esta sección va a crear las funcionalidades para manejar el tiempo del juego.

Para eso, siga estos pasos:

1. Declare la Función `actualizarTiempo` en donde la variable `time` haga una cuenta regresiva de 1 en 1. *Esta acción deberá verse en pantalla.*

PISTA: Aplique esto al elemento con el `id timeSpan`. Recuerda referenciar los elementos con su `HTML`.

2. Cree la variable llamada `timeInterval` (por fuera de la Función `actualizarTiempo`) de la siguiente forma:

```
let timeInterval = setInterval(actualizarTiempo, 1000);
```

Antes de avanzar, ¿te das cuenta que está haciendo el método `setInterval`? Sino, podés leer su definición [acá](#).

3. Volvé a la Función `actualizarTiempo` y establezca la siguiente condición: si la variable `time` llega a 0 hacé uso del método `clearInterval` pasándole como parámetro la variable **Funciones** que corresponda. Esto posibilita que el contador corte en 0 y no continúe su acción con números negativos.

-----SPOILER-----

Solución:

```
if (time === 0) {  
  clearInterval(timeInterval);
```

-----SPOILER-----

Tendría que tener un parecido a [esto](#). Prestá atención.

Funciones:

Función `updateScore`:

Para implementar correctamente esta Función seguí este paso a paso:

1. Cree la Función `updateScore` la cual incrementa la variable `score` en 1 en 1.
2. Añada su variable `score` al elemento `HTML` referenciado con el `id score`.
3. Una vez terminada su Función, invoque cuando el usuario ingrese correctamente la palabra.

Si todo sale bien deberías ver cómo el `score` aumenta al acertar la palabra.

Función `gameOver`:

Para implementar correctamente esta Función siga este paso a paso:

1. Creá la Función `gameOver` que tiene como finalidad agregar elementos al contenedor con el `id end-game-container`.
2. A este contenedor le agregarás:
 - Un título que le indique al usuario cuando se haya quedado sin tiempo.
 - Un párrafo que muestra el puntaje final de la partida.
 - El siguiente botón:

```
<button onclick="location.reload()">Volvé a empezar</button>
```

Si quiere saber qué hace el método `location.reload()` consultá [acá](#).

3. Volvé a la Función `actualizarTiempo` y que dentro del `if` se invoque la Función `gameOver`. Además, pensá cómo podrías remover el contenedor con `class main` para que el estado de la partida se actualice.

Importante: ya tenés todas las funcionalidades de tu proyecto. Probalo antes de darle el estilo que quieras.

CSS:

En esta sección, y antes de terminar el Proyecto, va a darle los estilos a su página.

Como habrá notado en el `HTML` tiene las siguientes clases para definir:

-
1. **body** : donde podés pensar un `background`, un `text-align`, un `font-family` y un `font-size`.
 2. **container** : donde podés definir un `background`, un `width`, un `border-radius`, un `margin` y un `padding`.
 3. **input** : para determinar un `width`, `height` y un `font-size`.

Puede sumarle todo lo que quiera para personalizarlo a tu medida.

Conclusión:

En este Proyecto implementó todo lo visto en estas 18 clases. Se trató de un repaso práctico completo que le permite generar un juego muy útil y divertido.

Además de mejorar sus habilidades para el tipeo fue incorporando palabras en otro idioma. ¡Puede modificarlo en todos los idiomas que quiera para aprender a escribir correctamente!

¡Felicitaciones! 🎉

OTEC PLATAFORMA 5 CHILE