

Introducción

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

1. Que puedas **integrar los diferentes conceptos de JavaScript** y los utilices en la **creación de un juego interactivo** que favorece a la escritura rápida y precisa.
 - o En este Proyecto trabajarás con Funciones, Arreglos, Eventos y manipularás también el DOM.
2. **Que aprendas algunas funcionalidades** como generar palabras *randoms* y actualizar oportunamente un temporizador.
3. Que logres una **herramienta muy útil** para mejorar tus *skills* para la redacción y el tipeo.

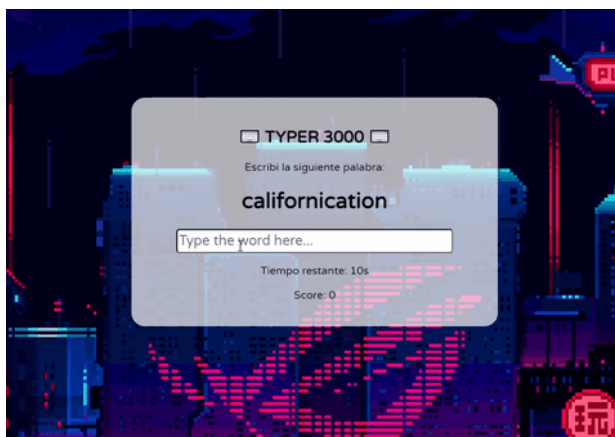
Metodología

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:



HTML

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

Para eso, seguí este paso a paso:

Creá tu archivo index.html en el VSC.

Copió y pegá este código.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <link rel="stylesheet" href="styles.css" />
</head>

<body>
  <div class="container">
    <div class="main">
      <h2>🎮 TYPED 3000 🎮</h2>
      <p>Escribí la siguiente palabra:</p>

      <h1 id="randomWord"></h1>

      <input type="text" id="text" autocomplete="off" placeholder="Type the
word here..." />

      <p class="time-container">Tiempo restante: <span
id="timeSpan">10s</span></p>

      <p class="score-container">Score: <span id="score">0</span></p>
    </div>
    <div id="end-game-container" class="end-game-container"></div>
  </div>
  <script src="script.js"></script>
</body>

</html>
```

3. Creá tus archivos .css y .js y asegúrate de que estén bien *linkeados* en tu código HTML.

Función randomWords

En esta sección del Proyecto crearás 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',  
  'unlimited',  
  'arcadium',  
  'love',  
  'getaway',  
  'stadium',  
  'quixoticelixir',  
  'quarter',  
  'snow',  
  'dylan',  
  'zephyr',  
  'funky',  
  'chili'  
];
```

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

PISTA

Podés agregarle las palabras que quieras.

Importante: referenciá los elementos HTML con tu JS. Tené en cuenta los nombres de las clases y de los id cuando declares las distintas variables.

A continuación seguí este paso a paso:

1. Creá la Función `randomWords` y ejecutala en la consola. Antes de avanzar, asegúrate que retorne una palabra aleatoria.
2. Declará la variables `palabraAleatoria`, `time` (que empiece en 10) y `score` (que empiece en 0) .
3. Escribí la función `addToDOM` que le agrega una palabra *random* al `h1` e invocala.

PISTA

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

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

 **TYPER 3000** 

Escribí la siguiente palabra:

californication

Type the word here...

Tiempo restante: 10s

Score: 0

Evento De Tipo Input

En esta sección tendrás 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. Escribí el Evento que se aplica al *input* referenciado de tipo *text*.
2. Creá la Función del Evento que tenga un parámetro "e" (que usarás más adelante).

PISTA

Si querés ver qué es este parámetro, hacé un `console.log(e)` y escribí algo en tu *input*.

3. Creá una variable que se llame *palabraIngresada*.
4. Compará las variables *palabraIngresada* y *palabraAleatoria*. Probá que todo funcione correctamente.

Si las palabras coinciden, hacé 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 vas a crear las funcionalidades para manejar el tiempo del juego.

Para eso, seguí este paso a paso:

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

PISTA

Aplicá esto al elemento con el `id timeSpan`. Recordá referenciar los elementos con tu HTML.

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

SOLUCIÓN

Condición

```
Solución: if (time === 0) {  
  clearInterval(timeInterval);  
}
```

Tendría que tener un parecido a esto. Prestá atención al contador:

 **TYPER 3000** 

Escribí la siguiente palabra:

aeroplane

Type the word here...

Tiempo restante: 7s

Score: 0

Funciones

En esta sección trabajarás en dos últimas Funciones: `updateScore` y `gameOver`

Función `updateScore`

Para implementar correctamente esta Función seguí este paso a paso:

1. Creá la Función `updateScore` la cual incrementara la variable `score` en 1 en 1.
2. Añadí tu variable `score` al elemento HTML referenciado con el id `score`.
3. Una vez terminada tu Función, invocala 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 seguí 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 muestre el puntaje final de la partida.
 - El siguiente botón:

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

 Si querés 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, vas a darle los estilos a tu página. Como habrás notado en el HTML tenés 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*.

Podés sumarle todo lo que quieras para personalizarlo a tu medida.

Conclusión

En este Proyecto implementaste todo lo visto hasta la Clase 14 inclusive. Se trató de un repaso práctico completo que te permite generar un juego muy útil y divertido.

Además de mejorar tus *skills* para el tipeo fuiste incorporando palabras en otro idioma. **¡Podés modificarlo en todos los idiomas que quieras para aprender a escribir correctamente!**



¡Felicitaciones!