

# JS #3

## JS avanzado, JSON y navegación de DOM

**NOTA:** los ejercicios ~~tachados~~ se podrán hacer después de la segunda parte de JS #2 (semana del 23/05).

1. ¿Qué es JSON y cómo es su estructura? ¿Para que se utiliza en JS?
2. ¿Cómo se obtiene un único elemento del DOM? ¿Cómo se obtienen múltiples elementos del DOM?
3. ~~¿En JS qué diferencias existen al asignar variables con let y var?~~
4. ~~¿Qué son las arrow functions?~~

### BÁSICO

5. Utilizando este **objeto JSON** para almacenar un equipo y sus datos en la tabla de posiciones (puntos, partidos jugados, ganados, empatados, perdidos, puntos):

```
let equipo = {  
  "nombre": "Barcelona FC",  
  "puntos": 98,  
  "pj": 35,  
  "pg": 32,  
  "pe": 2,  
  "pp": 1  
}
```

- a. Analizar lo que muestra `console.log(equipo)`
- b. Mostrar por consola las estadísticas del equipo

6. Crear un **arreglo de objetos JSON** que sea la **tabla entera** de posiciones del fútbol español que incluya los equipos. Pueden ser solo los primeros 5 equipos.

```
let tabla = [  
  {  
    "nombre": "Barcelona FC",  
    "puntos": 98,  
    "pj": 35,  
    "pg": 32,  
    "pe": 2,  
    "pp": 1  
  },  
  {  
    "nombre": "Atlético de Madrid",  
    "puntos": 74,  
    "pj": 35,
```

```

        "pg": 22,
        "pe": 8,
        "pp": 5,
    },
    { ... },
    { ... },
    { ... }
];

```

- a. Analizar lo que muestra `console.log(tabla)`
  - b. Mostrar por consola las estadísticas del **primer equipo**
  - c. Mostrar por consola la cantidad de equipos de la tabla.
  - d. Imprimir usando el DOM la diferencia de puntos entre el primer y segundo equipo de la tabla.
7. Crear tres botones que permitan contar elementos en una página.
- El primero debe contar el total de párrafos "`<p>`"
  - El segundo la cantidad de **items** de una lista con id "lista-menu"
  - El tercero el total de divs con la clase "noticia"

Investigue las diferencias entre: `getElementById()`, `getElementsByClassName()`, `getElementsByAttribute()` vs `querySelectorAll()`;

Ver: [slides a partir de la 36](#)

8. Implementar un botón que cada vez que sea clickeado borre un párrafo de la página, empezando desde el último al primero.

## INTERMEDIO

9. ~~Modificar el **ejercicio 8**, para que ahora los párrafos cambien de color de fondo cuando clickean sobre ellos mismos.~~
- ~~— Investigue el objeto `this` en el contexto de los event listeners~~
  - ~~— Investigue el elemento `event.target` y `event.currentTarget` en el contexto de los event listeners~~
10. Para el **ejercicio 5**:
- a. Hacer un botón que recorra e imprima la tabla de posiciones en el HTML.
  - b. Insertar un nuevo equipo (con todos sus datos) desde un formulario. Una vez ingresado se debe actualizar la tabla en la página.
11. Realice otra vez el ejercicio 5 y 11 del TP4. ¿Qué cambios haría para poder simplificar y reutilizar más el código?

Investigue con el material de la cátedra cómo se recorre el DOM y como se puede localizar un elemento cercano dentro del árbol.

Ver: [slides a partir de la 36](#)

Crear un tipo de botón **Ver más** que despliegue más contenido de texto. El botón debe ser tipo toggle que muestre y esconda la información expandida. [...]

#### TP 4 - Ejercicio 11

*Modifique la lista de tareas para que cada tarea debe poder "completarse" desde un botón.*

### AVANZADO

12. Hacer una galería de imágenes que incluya varias imágenes en miniatura. Cuando se posa el mouse sobre cada imagen en miniatura deberá marcarse con un recuadro y cuando sea cliqueada debe aumentar su tamaño para poder verla mejor. Esta funcionalidad debe ser aplicable a cualquier imagen dentro de la galería.

13. Hacer un menú principal que permita desplegar submenús con las siguientes características:

- Al parar el mouse sobre cada subitem se debe resaltar.
- Una vez que se **clickee un ítem**, se despliega el submenú con varios subitems.
- El código debe poder reutilizarse para más de un ítem del menú principal, y para múltiples niveles de subitems.

Nota: este tipo de ejercicios se realiza combinando JS (navegando el DOM para buscar elementos hijos) y aplicando clases CSS. Es importante entender la estructura del menú para poder realizarlo con múltiples niveles sin necesidad de repetir código.

## Challenge

**NOTA: esperar semana del 23/05**

Los challenge no entran dentro de la evaluación de la materia. Son cosas más avanzadas que en el ámbito laboral a veces un usuario la pide, pero no con gran frecuencia. Implican como su nombre lo dice *desafíos técnicos*. En algún momento el TPE cubría parte de estos temas, pero era demasiado trabajo durante la materia para ustedes. Este año queremos recuperar ese extra solo para los que tengan ganas/tiempo de llevárselo.

14. Challenge Eventos:

Hacer un div pequeño que al hacer click se pueda arrastrar manteniendolo apretado (drag & drop) y soltarlo en otro lugar cambiando su posición. No es necesario que se vea durante el arrastrado.

Recomendación:

- Investigar los diferentes [tipos de eventos](#) que existen en JS para poder realizar esto.
- Investigar el objeto [Event](#) y las distintas propiedades que tiene.
- En internet van a encontrar muchísimos ejemplos de esto. Se recomienda primero intentar hacer una versión por ustedes mismos.

### 15. Challenge JSON:

A la tabla de equipos anterior (**ejercicios 5 y 10**) agregarle un botón a cada equipo que permita ver el listado de partidos. Cada partido dice la cantidad de goles que hizo cada equipo, quien fue el oponente y si se jugaba de local o visitante.