

CURSO PROGRAMACIÓN WEB FULL- STACK NIVEL 2

MÓDULO 1 - Nivelación y fundamentos de la Programación

Ejercicio 1:

- Crear una carpeta con la siguiente nomenclatura: “TP1_Apellido_Nombre” (Ej: TP1_Lopez_Jose).
- Dentro de dicha carpeta, crear un archivo llamado index.html con la estructura básica de una plantilla HTML y pegar el siguiente código dentro del en su interior:

```
<body>
  <script src="ejercicio_1.js"></script>
</body>
```

Dentro de la misma carpeta, se irán creando otros archivos con la extensión “.js”, cada uno de estos archivos corresponderá a uno de los ejercicios que se les presentará más adelante en este trabajo. Por ejemplo, el ejercicio N°5 deberá ser guardado en el archivo “ejercicio_5.js”.

- Crear el primer archivo llamado “ejercicio_1.js” y pegar el siguiente texto en su interior:

```
document.write("Hola Mundo!");
```

- agregar un comentario debajo que explique que hace esa primer linea de código

Ejercicio 2:

Copiar el siguiente código dentro de un archivo, luego guardarlo con el nombre: “ejercicio_2.js”.

```
let A, B;
A = 9;
B = 7;
resultado = A + B;
alert (`La suma de A+B es: ${resultado}`);
alert (`La diferencia de A-B es: ${A-B}`);
alert (`La multiplicación de AxB es: ${A*B}`);
alert (`La división de A/B es: ${resultado=A/B}`);
```

Para poder ejecutarlo en el navegador, deberá cambiar el contenido del archivo index.html donde dice:

`<script src="ejercicio_1.js"></script>` por
`<script src="ejercicio_2.js"></script>`

Ejercicio 3:

Copiar el siguiente código dentro de un archivo, luego guardarlo con el nombre: “ejercicio_3.js”. Sin embargo, dicho código tiene un error, para corregirlo se debe depurar el código en el navegador.

```
let a, b;  
a = 3; b = 7;  
resultado = a * b;  
document.write(`Variable a contiene ${a} <br> ` );  
document.write(`Variable b contiene ${b} <br> ` );  
document.write(`El producto de a por b es ${result} <br> `);  
document.write("Los puntos de interrupcion son fundamentales para programar");
```

Ejercicio 4:

Completar las condiciones de los **IF** del siguiente script para que los mensajes se muestren siempre de forma correcta, guardándolo con el nombre: “ejercicio_4.js”.

```
var num1 = 3;  
var num2 = 7;  
if(...) {  
    console.log("num1 no es mayor que num2");  
}  
if(...) {  
    console.log("num2 es positivo");  
}  
if(...) {  
    console.log("num1 es negativo o distinto de cero");  
}  
if(...) {  
    console.log("Incrementar en 1 unidad el valor de num1 no lo hace  
    mayor o igual que num2");  
}
```

Ejercicio 5:

Crear el archivo correspondiente a “ejercicio_5.js”, Ejecutarlo y luego agregar líneas de comentario explicando que hace el programa en cada línea de código

```
let dato, resultado;  
val1 = window.prompt("Introduce tu nombre", "...");  
val2 = window.prompt("Introduce tu apellido", "...");  
resultado = `Concatenado tu nombre y apellido es: ${val1} ${val2} ` ;  
document.write(resultado);
```

Ejercicio 6:

Crear el archivo “ejercicio_6.js”. Declarar un array llamado “meses” y que tenga guardado los meses del año. Luego el usuario debe poder ingresar en el navegador un número de 1 al 12 y este le devolverá el nombre del mes. Por ejemplo, escribir un 5 debe devolver el mes mayo.

Ejercicio 7:

A partir del siguiente array que se proporciona:

```
var valores = [true, false, 2, "hola", "mundo", 3, "char"];
```

1. Determinar cuál de los “**elementos de texto**” es mayor, es decir el que contenga más letras.
2. Imprimir estos elementos de menor a mayor cantidad de letras.
3. Determinar el resultado de las cuatro operaciones matemáticas básicas (suma, resta, multiplicación y división) realizadas con los dos **elementos numéricos**

Ejercicio 8:

Escribir un script que muestre la posición de la primera vocal de un texto introducido por teclado. Por ejemplo: perro = “e” es la letra N°2; árbol = “a” es la letra N°1.

Ejercicio 9:

Crear un programa que recibe 5 (cinco) notas de un alumno, las mismas deben ser ingresadas por el usuario y ser naturales < 11. Debe calcular el promedio e imprimir en pantalla si el valor de media es < 5 “**Reprobado**”, entre 6 y 8 “**Aprobado**” y si es mayor a 8 “**Sobresaliente**”

JSON

JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

Los tipos de datos disponibles con JSON son:

- **Números:** Se permiten números negativos y opcionalmente pueden contener parte fraccional separada por puntos. Ejemplo: 123.456
- **Cadenas:** Representan secuencias de cero o más caracteres. Se ponen entre doble comilla y se permiten cadenas de escape. Ejemplo: "Hola"
- **Booleanos:** Representan valores booleanos y pueden tener dos valores: true y false
- **null:** Representan el valor nulo.
- **Array:** Representa una lista ordenada de cero o más valores los cuales pueden ser de cualquier tipo. Los valores se separan por comas y el vector se mete entre corchetes. Ejemplo ["juan","pedro","jacinto"]
- **Objetos:** Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena entre comillas dobles. El valor puede ser de cualquier tipo.

El siguiente enunciado es solamente un ejemplo orientativo: Supongamos que estás desarrollando un sistema de gestión de libros para una biblioteca. Además de la información básica del libro, como el título y el autor, también necesitas almacenar información sobre las copias disponibles y prestadas de cada libro.

En este JSON, hemos creado un objeto "libro" con propiedades como "titulo", "autor" y "genero". Además, hemos incluido un objeto anidado "copias" que contiene las propiedades "disponibles" y "prestadas", que indican cuántas copias del libro están disponibles y cuántas están actualmente prestadas.

```
{ "biblioteca": [  
  
  {  
    "titulo": "El Alquimista",  
    "autor": "Paulo Coelho",  
    "genero": "Ficción",  
    "copias": {  
      "disponibles": 5,  
      "prestadas": 3  
    }  
  },  
  {  
    "titulo": "Cien años de soledad",  
    "autor": "Gabriel García Márquez",  
    "genero": "Realismo mágico",  
    "copias": {  
      "disponibles": 7,  
      "prestadas": 1  
    }  
  },  
  {  
    "titulo": "1984",  
    "autor": "George Orwell",  
    "genero": "Ciencia ficción",  
    "copias": {  
      "disponibles": 3,  
      "prestadas": 2  
    }  
  }  
]  
}
```

Para dar formato y validar se pueden emplear herramientas online: <https://jsonformatter.org/>

Ejercicio 10:

Crear un archivo JSON llamado ejercicio_10.json que contenga los siguientes datos:

Nombre	Apellido	DNI	Telefono	Localidad
Pedro	Guzman	22773450	3755121314	L. N. Alem
Maria	Nieves	35928532	3764131517	Posadas
Roberto Carlos	Guerra	40726384	3758171819	Apóstoles
Julieta	Lopez	29384710	3751222324	Eldorado
Ramon	Rodriguez	31749223	3765334455	Posadas

Ejercicio 11:

Crear un archivo JSON llamado ejercicio_11.json que contenga la siguiente información:

- id_producto: 283; nombre: Gafas de sol; Marca: Acme; disponible: 20; precio: 9800; temporada: Verano 2023;
 - caracteristicas: color: negro; marco: metal;
- id_producto: 191; nombre: Reloj deportivo; Marca: Timer; disponible: 2; precio: 21400;
 - caracteristicas: color: plateado; material: goma y plástico;
- id_producto: 148; nombre: Juego de medias; Marca: Piecito; disponible: 38; precio: 500;
 - caracteristicas: color: varios; material: algodón;
- id_producto: 82; nombre: Camiseta selección Argentina; Marca: Argento; disponible: 11; precio: 6800;
 - talle: L
- id_producto: 81; nombre: Camiseta selección Argentina; Marca: Argento; disponible: 15; precio: 6800;
 - talle: M

PROGRAMACIÓN ORIENTADA A OBJETOS

Ejercicio 12:

Crear un archivo con el nombre “ejercicio_12.js” que contenga el código JavaScript donde cree una clase base y luego hereden de ella para crear una clase hija. La clase representará algo general, como un **vehículo**, y la clase hija podría ser un tipo específico de vehículo, como un **automóvil**. En este ejercicio deberán realizar los siguientes pasos

- Definir una clase base Vehiculo con tres atributos (marca, modelo y año) y una función (*obtenerInformacion*) para mostrar información del vehículo.
- Crear una clase hija Automovil que hereda de Vehiculo y añadir dos nuevos atributos (color y precio). También se sobrescribe el método *obtenerInformacion* para incluir la información específica del automóvil.
- Crear instancias de ambas clases y mostrar la información en la consola del navegador.