

Práctico N° 4 – JavaScript - Parte 1

Crear una carpeta que se llame practico-4. Luego dentro de ella grabe cada ejercicio con el nombre ej1, ej2, etc. Si necesita anexar imágenes guárdelas en una carpeta llamada “imágenes” dentro de “practico4”

Ejercicio 1:

Cree una nueva página web con el sencillo ejemplo en javascript:

```
let s = “Hola Mundo!!!”;
```

```
alert(s)
```

Incluir dichas sentencias en un script de todas las siguientes maneras

1. En el <head> de la página web
2. En el <body> de la página web
3. En un archivo externo colocando el vínculo en el <head>
4. En un archivo externo colocando el vínculo en el <body>

Ejercicio 2:

Cuales de los siguientes son nombres de identificadores válidos en javascript

- let Nombre1
- const _ar_1
- function calcularValor()
- var x
- let Edad Promedio
- let 1erValor
- let \$edad
- var Num#3
- function Nombre1()
- let Edad_Promedio

Ejercicio 3:

Pruebe los siguientes ejemplos con “const”. Utilice la solapa “console” en la sección del desarrollador de su navegador. (botón derecho / inspeccionar). Pruebe de comentar las líneas e ir descomentándolas de a una. Luego saque conclusiones

Prueba 1

```
const x = 234; console.log(x);  
x = 123; console.log(x); // luego cambiar esta línea por  
let x = 123; console.log;
```

Prueba 2

```
let x = 234; console.log(x);  
x = 123; console.log(x); // luego cambiar esta línea por  
let x = 123; console.log;
```

Prueba 3

```
var x = 234; console.log(x);  
x = 123; console.log(x); // luego cambiar esta línea por  
var x = 123; console.log;
```

Ejercicio 4:

Probar el siguiente ejemplo y escribir las conclusiones

```
if (true) {  
    var x = 10; // y luego con let
```

```
}  
console.log(x);
```

Ejercicio 5:

Pruebe el siguiente código

```
const ar_num1 = [23, 32, 4, 8];  
const ar_num2 = [14, 28, 3, 46];  
console.log(ar_num1);  
console.log(ar_num2);
```

realizar las siguientes pruebas

- `ar_num1 = ar_num2;`
`console.log(ar_num1);`
`console.log(ar_num2);`
- `// cambiar const por let en ambos arreglos volver a probar`
- `ar_num1[2] = 44`
`console.log(ar_num1);`
`console.log(ar_num2);`
Porque el 44 también se cambió en `ar_num2`?

Ejercicio 6:

Investigue como crear números aleatorios (random) entre el 1 y el 100. Genere un arreglo llamado “`ar_random`” con 1000 de esos números utilizando `ar_random.push(x)`. Luego muéstrelo por consola

Ejercicio 7:

Recorrer el arreglo anterior y cada uno de los valores los deberá concatenar en la variable string “`str_num_aleat`” separados por “;”. Al terminar mostrar en consola la variable string.

Hacer 4 versiones del algoritmo utilizando **while**, luego utilizando **for** (común), luego **for of** y por último **for in**

Ejercicio 8:

Con el arreglo generado en el ejercicio anterior

- Encontrar cual es el menor valor, cual es el mayor valor y en que posición se encuentra cada uno
- Mostrar la suma de todos los números y el promedio
- Mostrar cuantos números son mayores al promedio y cuantos son menores

Ejercicio 9:

Investigar como es la sintaxis y la forma de uso de las funciones **alert**, **prompt** y **confirm** en javascript. Hacer una ejemplo con cada una.

Ejercicio 10:

Hacer una función que reciba como parámetro un número entero y retorne true si el número es primo y false en otro caso. (Recuerde que un número es primo si solo es divisible por si mismo y por la unidad)

Ejercicio 11:

Utilizando la función “prompt”. Pedir un número entero al usuario n entre 1 y 10 (controlar que “n” sea un número y en ese rango, de lo contrario lo deberá pedir nuevamente mostrando el error con la función “alert”)
Luego utilizando la función del ejercicio anterior encontrar y guardar en un arreglo los n primeros números primos.

Ejercicio 12:

Hacer una función que reciba una cadena y retorne la misma pero con los caracteres en orden invertido. (ej “Hola” → “aloH”).
Pedir al usuario la cadena, y luego utilizando la función, pedir que la invierta y mostrarla con alert.

Ejercicio 13:

Hacer una función que reciba una cadena y devuelva la cantidad de vocales que tiene. Las letras pueden estar en mayúsculas o minúsculas.

Ejercicio 14:

Modificar el ejercicio anterior para que la función retorne un arreglo de 5 posiciones con la cantidad de letras “a, e, i, o y u”.

Ejercicio 15:

Hacer una función que reciba una cadena y devuelva true si es palindromo (es decir se le igual de izquierda a derecha que de derecha a izquierda) false en otro caso. Ej. “neuquen”. Agregue un parámetro más (puede ser booleano) para que sea sensible a mayúsculas y minúsculas o no.

Ejercicio 16:

Hacer una función que reciba un arreglo de números enteros y devuelva el segundo mayor. (Es decir el que le sigue al mayor, que puede ser el mismo en caso que se repita el número mayor). Para probarlo puedo utilizar el arreglo generado en el ej. N° 6.

Ejercicio 17:

Cree un arreglo con unos 20 nombres de pila inventados, otro con 20 apellidos y otro con 20 calles.

Hace una función generarPersona() que devuelva un objeto con los siguientes campos: dni, apellido, nombre, dirección, edad y email.

- El dni será un número aleatorio entre 10000000 y 99999999
- Escoger el apellido generando un número al azar para el arreglo de nombres
- Idem para el nombre
- Para el domicilio generar al azar el nombre de la calle y concatenarle un número entero generado al azar entre 1 y 2000
- La edad será un número al azar generado entre 0 y 100
- El email será la concatenación del apellido, un “.” (punto), el nombre, luego “@” y por último podrá ser gmail.com, yahoo.com o hotmail.com (generarlo al azar con un número del 1 al 3)

Ejercicio 18:

Usar el código del ejercicio anterior para generar un arreglo con los datos de n personas. El número n será solicitado al usuario.

Ejercicio 19:

Usando los datos del ejercicio anterior pedir al usuario un número de dni y buscar en el arreglo si existe la persona. Si existe mostrar los datos con un alert. Si no existe mostrar con alert un cartel que diga que no existe una persona con el dni nnnn ingresado

Ejercicio 20:

Similar al ejercicio anterior, pero en lugar de mostrar los datos, se deberá eliminar. Investigar el uso del método “array.splice()”.

El algoritmo deberá

- Pedir el dni
- Buscar la persona
- Si no existe, mostrar un alert diciendo que no existe
- Si existe, mostrar el apellido y nombre de la persona en un confirm y pedir la confirmación para eliminar.
En caso afirmativo proceder a eliminar al elemento