



## Programación II Guía de JS

### Introducción

1. Realizar un programa que muestre su nombre y su edad en una página HTML. Emplear el comando write del objeto document para imprimir. Tener en cuenta que si queremos que cada dato quede en una fila distinta de la página debemos insertar la etiqueta HTML `<br>` (salto de línea en HTML), es decir debemos disponer:  
`document.write('<br>')`

### Variables.

1. Confeccionar una programa en JavaScript que defina e inicialice una variable de tipo cadena de caracteres donde almacenemos el nombre de un empleado y otra variable de tipo entera donde almacenar el sueldo. Imprimir cada variable en una línea distinta en pantalla.

### Entrada de datos por teclado.

1. Confeccionar un programa que permita cargar el nombre de un usuario y su mail por teclado. Mostrar posteriormente los datos en la página HTML.

### Estructuras secuenciales de programación.

1. Realizar la carga del lado de un cuadrado, mostrar por pantalla el perímetro del mismo (El perímetro de un cuadrado se calcula multiplicando el valor del lado por cuatro)
2. Escribir un programa en el cual se ingresen cuatro números, calcular e informar la suma de los dos primeros y el producto del tercero y el cuarto.
3. Realizar un programa que lea cuatro valores numéricos e informar su suma y producto.
4. Se debe desarrollar un programa que pida el ingreso del precio de un artículo y la cantidad que lleva el cliente. Mostrar lo que debe abonar el comprador (Ingresar por teclado un precio sin decimales, es decir un entero: 2, 7, 90 etc.)

### Estructuras condicionales simples.

1. Se ingresan tres notas de un alumno, si el promedio es mayor o igual a siete mostrar el mensaje 'Promocionado'. Tener en cuenta que para obtener el promedio debemos operar  $\text{suma} = \text{nota1} + \text{nota2} + \text{nota3}$ ; y luego hacer  $\text{promedio} = \text{suma} / 3$ ; Cuando cargamos una nota y queremos convertir inmediatamente el valor ingresado a entero podemos hacer:  
`nota1=prompt('Ingrese primer nota:','');`  
`nota1=parseInt(nota1);`  
O en forma más corta:  
`nota1=parseInt(prompt('Ingrese primer nota:',''));`
2. Solicitar que se ingrese dos veces una clave. Mostrar un mensaje si son iguales (tener en cuenta que para ver si dos variables tienen el mismo valor almacenado debemos utilizar el operador `==`)



## Programación II Guía de JS

### Estructuras condicionales compuestas.

1. Realizar un programa que lea por teclado dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.
2. Se ingresan tres notas de un alumno, si el promedio es mayor o igual a 4 mostrar un mensaje 'regular', sino 'reprobado'.
3. Se ingresa por teclado un número positivo de uno o dos dígitos (1..99) mostrar un mensaje indicando si el número tiene uno o dos dígitos (recordar de convertir a entero con parseInt para preguntar posteriormente por una variable entera). Tener en cuenta qué condición debe cumplirse para tener dos dígitos un número entero.

### Estructuras condicionales anidadas.

1. Realizar un programa que pida cargar una fecha cualquiera, luego verificar si dicha fecha corresponde a Navidad (se debe cargar por separado el día, el mes y el año)
2. Se ingresan tres valores por teclado, si todos son iguales se imprime la suma del primero con el segundo y a este resultado se lo multiplica por el tercero (tener en cuenta que puede haber tres condiciones simples)
3. Se ingresan por teclado tres números, si todos los valores ingresados son menores a 10, imprimir en la página la leyenda ' Todos los números son menores a diez'.
4. Escribir un programa que pida ingresar la coordenada de un punto en el plano, es decir dos valores enteros x e y.  
Posteriormente imprimir en pantalla en qué cuadrante se ubica dicho punto. (1º Cuadrante si  $x > 0$  Y  $y > 0$  , 2º Cuadrante:  $x < 0$  Y  $y > 0$ , etc.)
5. De un operario se conoce su sueldo y los años de antigüedad. Se pide confeccionar un programa que lea los datos de entrada e informe
  - a) Si el sueldo es inferior a 500 y su antigüedad es igual o superior a 10 años, otorgarle un aumento del 20 %, mostrar el sueldo a pagar.
  - b) Si el sueldo es inferior a 500 pero su antigüedad es menor a 10 años, otorgarle un aumento de 5 %.
  - c) Si el sueldo es mayor o igual a 500 mostrar el sueldo en la página sin cambios.

### Operadores lógicos || (o) en las estructuras condicionales.

1. Se ingresan por teclado tres números, si al menos uno de los valores ingresados es menores a 10, imprimir en la página la leyenda 'Alguno de los números es menor a diez'.

### Estructuras switch.

1. Solicitar el ingreso alguna de estas palabras (casa, mesa, perro, gato) luego mostrar la palabra traducida en inglés. Es decir, si se ingresa 'casa' debemos mostrar el texto 'house' en la página.



## Programación II Guía de JS

### Estructura repetitiva (while)

1. Realizar un programa que imprima 25 términos de la serie 11 - 22 - 33 - 44, etc. (No se ingresan valores por teclado).
2. Mostrar los múltiplos de 8 hasta el valor 500. Debe aparecer en pantalla 8 -16 -24, etc.

### Concepto de acumulador.

1. Escribir un programa que lea 10 notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores.
2. Se ingresan un conjunto de 5 alturas de personas por teclado. Mostrar la altura promedio de las personas.
3. En una empresa trabajan 5 empleados cuyos sueldos oscilan entre \$100 y \$500, realizar un programa que lea los sueldos que cobra cada empleado e informe cuántos empleados cobran entre \$100 y \$300 y cuántos cobran más de \$300. Además el programa deberá informar el importe que gasta la empresa en sueldos al personal.
4. Realizar un programa que imprima 20 términos de la serie 5 - 10 - 15 - 20, etc. (No se ingresan valores por teclado)
5. Mostrar los múltiplos de 10 hasta el valor 1500.  
Debe aparecer en pantalla 10 - 20 -30 etc.
6. Realizar un programa que permita cargar dos listas de 3 valores cada una. Informar con un mensaje cual de las dos listas tiene un valor acumulado mayor (mensajes 'Lista 1 mayor', 'Lista 2 mayor', 'Listas iguales')  
Tener en cuenta que puede haber dos o más estructuras repetitivas en un algoritmo.
7. Desarrollar un programa que permita cargar 5 números enteros y luego nos informe cuántos valores fueron pares y cuántos impares.  
Emplear el operador "%" en la condición de la estructura condicional.  

```
if (valor%2==0)
```

  
El operador "%" retorna el resto de la división valor / 2. Por ejemplo: 12 % 2, retorna 0; 13 % 2, retorna 1, porque el resto de dividir 13 en 2 es 1.





## Programación II

### Guía de JS

#### Estructura repetitiva (do/while)

1. Realizar un programa que acumule (suma) valores ingresados por teclado hasta ingresa el 9999 (no sumar dicho valor, solamente indica que ha finalizado la carga). Imprimir el valor acumulado e informar si dicho valor es cero, mayor a cero o menor a cero.
2. En un banco se procesan datos de las cuentas corrientes de sus clientes. De cada cuenta corriente se conoce: número de cuenta, nombre del cliente y saldo actual. El ingreso de datos debe finalizar al ingresar un valor negativo en el número de cuenta.  
Se pide confeccionar un programa que lea los datos de las cuentas corrientes e informe:
  - a) De cada cuenta: número de cuenta, nombre del cliente y estado de la cuenta según su saldo, sabiendo que:  
Estado de la cuenta      'Acreedor' si el saldo es  $>0$ .  
                                     'Deudor' si el saldo es  $<0$ .  
                                     'Nulo' si el saldo es  $=0$ .
  - b) La suma total de los saldos acreedores.
3. Se realizó un censo provincial y se desea procesar la información obtenida en dicho censo. De cada una de las personas censadas se tiene la siguiente información: número de documento, edad y sexo ('femenino' o 'masculino')  
Se pide confeccionar un programa que lea los datos de cada persona censada (para finalizar ingresar el valor cero en el número de documento) e informar:
  - a) Cantidad total de personas censadas.
  - b) Cantidad de varones.
  - c) Cantidad de mujeres.
  - d) Cantidad de varones cuya edad varía entre 16 y 65 años.



## Programación II Guía de JS

### Estructura repetitiva (for)

1. Confeccionar un programa que lea 3 pares de datos, cada par de datos corresponde a la medida de la base y la altura de un triángulo. El programa deberá informar:
  - a) De cada triángulo la medida de su base, su altura y su superficie.
  - b) La cantidad de triángulos cuya superficie es mayor a 12.
2. Desarrollar un programa que solicite la carga de 10 números e imprima la suma de los últimos 5 valores ingresados.
3. Desarrollar un programa que muestre la tabla de multiplicar del 5 (del 5 al 50).
4. Confeccionar un programa que permita ingresar un valor del 1 al 10 y nos muestre la tabla de multiplicar del mismo (los primeros 12 términos)  
Ejemplo: Si ingreso 3 deberá aparecer en pantalla los valores 3, 6, 9, hasta el 36.
5. Realizar un programa que lea los lados de 4 triángulos, e informar:
  - a) De cada uno de ellos, qué tipo de triángulo es: equilátero (tres lados iguales), isósceles (dos lados iguales), o escaleno (ningún lado igual)
  - b) Cantidad de triángulos de cada tipo.
  - c) Tipo de triángulo del que hay menor cantidad.
6. Escribir un programa que pida ingresar coordenadas (x,y) que representan puntos en el plano. Informar cuántos puntos se han ingresado en el primer, segundo, tercer y cuarto cuadrante. Al comenzar el programa se pide que se ingrese la cantidad de puntos a procesar.
7. Se realiza la carga de 10 valores enteros por teclado. Se desea conocer:
  - a) La cantidad de valores negativos ingresados.
  - b) La cantidad de valores positivos ingresados.
  - c) La cantidad de múltiplos de 15.
  - d) El valor acumulado de los números ingresados que son pares.

### Funciones

1. Confeccionar una función que solicite la carga de dos enteros (primero el menor y luego el mayor) y nos muestre desde el menor hasta el mayor de uno en uno.  
Por ejemplo si ingresamos los valores 4 y 10 luego se debe mostrar por pantalla:  
4 5 6 7 8 9 10

### Funciones con parámetros.

1. Elaborar una función a la cual le enviemos tres enteros y muestre el menor.
2. Confeccionar una función a la cual le envíe tres enteros y los muestre ordenados de menor a mayor.



## Programación II Guía de JS

Funciones que retornan un valor.

1. Confeccionar una función a la cual le envíe tres enteros y retorne el mayor de ellos.
2. Elaborar una función a la cual le envíe el valor del lado de un cuadrado y me retorne su perímetro.
3. Desarrollar una función que retorne la cantidad de dígitos que tiene una variable entera positiva de hasta 5 dígitos.
4. Elaborar una función que reciba tres enteros y retorne el promedio.
5. Confeccionar una función que solicite la carga de 5 valores por teclado y retorne su suma.

### **Programación orientada a objetos.**

#### Clase Date

1. Confeccionar un programa que muestre en que cuatrimestre del año nos encontramos. Para esto obtener el mes.
2. Confeccionar una función que nos retorne un string con el siguiente formato:  
Hoy es Lunes 9 de Agosto de 2021  
  
Para poder recuperar el día de la semana debemos llamar al método:  
`let diaSemana=fecha.getDay();`  
El método `getDay()` devuelve el día de la semana de la fecha especificada, siendo 0 (Domingo) el primer día.

#### Clase Array

1. Desarrollar un programa que permita ingresar un vector de 8 elementos, e informe:  
El valor acumulado de todos los elementos del vector.  
El valor acumulado de los elementos del vector que sean mayores a 36.  
Cantidad de valores mayores a 50.
2. Realizar un programa que pida la carga de dos vectores numéricos. Obtener la suma de los dos vectores, dicho resultado guardarlo en un tercer vector del mismo tamaño. Sumar componente a componente.  
El tamaño del vector es a elección.

#### Clase Math

1. Confeccionar un programa que solicite el ingreso de un número y nos muestre dicho valor elevado a la tercera potencia.
2. Ingresar por teclado un valor y luego mostrar la raíz cuadrada de dicho valor.





## Programación II Guía de JS

### Clase String

1. Ingresar una serie de nombres por teclado hasta que se cargue la palabra Fin, y mostrar cuántos nombres se ingresaron.
2. Igual al anterior, pero que termine la aplicación sin contemplar mayúsculas ni minúsculas. Es decir que para salir se pueda teclear fin, Fin o FIN.
3. Realizar la búsqueda de un string clave en un string fuente. Se deberá ingresar una frase o texto (fuente) y luego la clave a buscar. En caso de encontrarla, imprimir la posición, de lo contrario una leyenda que indique que no está.
4. Ingresar una palabra o texto por teclado y determinar si es o no una palabra palíndromo. (Palabra que se lee de igual manera de adelante hacia atrás, que de atrás hacia adelante)
5. Realizar un programa que permita cargar una dirección de mail e implementar una función que verifique si el String tiene cargado el carácter @.
6. Cargar un String por teclado e implementar las siguientes funciones:
  - a) Imprimir la primera mitad de los caracteres de la cadena.
  - b) Imprimir el último carácter.
  - c) Imprimirlo en forma inversa.
  - d) Imprimir cada carácter del String separado con un guión.
  - e) Imprimir la cantidad de vocales almacenadas.
7. Codifique un programa que permita cargar una oración por teclado, luego mostrar cada palabra ingresada en una línea distinta.  
Por ejemplo si cargo:  
La mañana está fría.  
Debe aparecer:  
La  
mañana  
está  
fría.

### Formularios y Eventos.

1. Crear un formulario con tres botones con las leyendas "1", "2" y "3". Mostrar un mensaje indicando qué botón se presionó.

### Controles FORM, BUTTON y TEXT.

1. Crear un programa que permita cargar un entero en un text y al presionar un botón nos muestre dicho valor elevado al cubo (emplear la función alert).
2. Cargar dos números en objetos de tipo text y al presionar un botón, mostrar el mayor.
3. Cargar un nombre y un apellido en sendos text. Al presionar un botón, concatenarlos y mostrarlos en un tercer text (Tener en cuenta que podemos modificar la propiedad value de un objeto TEXT cuando ocurre un evento)



## Programación II

### Guía de JS

#### Control PASSWORD

1. Disponer dos campos de texto tipo password. Cuando se presione un botón mostrar si las dos claves ingresadas son iguales o no (es muy común solicitar al operador el ingreso de dos veces de su clave para validar si las escribió correctamente, esto se hace cuando se crea una password para el ingreso a un sitio o para el cambio de una existente).  
Tener en cuenta que podemos emplear el operador `==` para ver si dos string son iguales.

#### Control SELECT

1. Confeccionar una página que muestre un objeto SELECT con distintos tipos de pizzas (Jamón y Queso, Muzzarella, Morrones). Al seleccionar una, mostrar en un objeto de tipo TEXT el precio de la misma.
2. Generar un presupuesto de un equipo de computación a partir de tres objetos de tipo SELECT que nos permiten seleccionar:  
Procesador (Intel I3 - \$400, Intel I5 \$600, Intel I7 \$800).  
Monitor (Samsung 20' - \$250, Samsung 22' - \$350, Samsung 26' - \$550)  
Disco Duro(500 Gb - \$300, 1 Tb - \$440, 3 Tb - \$500)  
Para cada característica indicamos string a mostrar (Ej. Intel I3) y el valor asociado a dicho string (Ej. 400).  
Al presionar un botón "Calcular" mostrar el presupuesto en un objeto de tipo TEXT.
3. Confeccionar una página que permita seleccionar una pizza y la cantidad de unidades. Luego al presionar un botón calcular el importe a pagar.  
Utilizar un objeto de la clase SELECT para la selección de la pizza, pudiendo ser:  
Queso: \$ 4.  
Jamón y queso: \$ 6.  
Especial: \$ 10.  
A la cantidad de pizzas cargarla en un objeto de la clase TEXT y en otro objeto de la clase TEXT mostrar el importe total a abonar.
4. Confeccionar una página que permita tomar un examen múltiple choice. Se debe mostrar una pregunta y seguidamente un objeto SELECT con las respuestas posibles. Al presionar un botón mostrar la cantidad de respuestas correctas e incorrectas (Disponer 4 preguntas y sus respectivos controles SELECT)

#### Control CHECKBOX

1. Confeccionar una página que muestre tres checkbox que permitan seleccionar los deportes que practica el usuario (Fútbol, Baské, Tenís) Mostrar al presionar un botón los deportes que eligió.

#### Control RADIO

1. Confeccionar una página que muestre dos objetos de la clase RADIO solicitando que seleccione si es mayor de 18 años o no. Al presionar un botón mostrar un alert indicando si puede ingresar al sitio o no.

#### Control TEXTAREA

1. Confeccionar una página de visitas a un sitio, solicitar ingresar el nombre de una persona, su mail y los comentarios (TEXTAREA). Mostrar luego llamando a la función alert los datos ingresados.





## Programación II Guía de JS

### Eventos onFocus y onBlur

1. Solicitar que se ingrese el nombre y la clave de un usuario. Mostrar una ventana de alerta si en la clave se ingresan menos de 7 caracteres o más de 20 (capturar el evento onBlur)

### Eventos onMouseOver y onMouseOut

1. Modificar el segundo problema resuelto (las casillas de la tabla que cambian el color cuando ingresamos con el mouse) para permitir llamar mediante hipervínculos a distintos programas que administran web-mail (gmail, hotmail y yahoo)

### Evento onLoad

1. Confeccionar una página que muestre un mensaje con la función alert inmediatamente después que se cargue.

### El objeto window

1. Confeccionar una página que permita abrir otra ventana cuando se presiona un botón. Dicha ventana debe tener como ancho 600 pixeles y alto 300 pixeles.

### Propiedad location del objeto window

1. Confeccionar una página que tenga un botón. Cuando se presione dicho botón generar un valor aleatorio entre 0 y 2. Si se genera el 0 llamar al webmail de Outlook, si se genera un 1 llamar a gmail en caso de generarse un 2 llamar a yahoo.  
Para generar un valor aleatorio utilizar la función random del objeto Math.

```
let num;
num=Math.random()*3; //se guarda en num un valor comprendido entre 0.00001 y 2.99999
num=parseInt(num); //guardamos solo la parte entera de la variable num
if (num==0){
    window.location='https://www.outlook.com';
}
.....
```

### Propiedad history del objeto window

### Propiedad screen del objeto window

1. Confeccionar un programa que al presionar un botón aparezca una ventana que ocupe todo el ancho del monitor y la mitad de su altura.

### Propiedad navigator del objeto window

### Archivo JavaScript externo (\*.js)



## Programación II Guía de JS

Programación orientada a objetos en JavaScript.

1. Confeccionar una clase llamada suma, que contenga dos atributos (valor1, valor2) y tres métodos: cargarvalor1, cargarvalor2 y retornarresultado. Implementar la clase suma. La definición de un objeto de la clase que deben plantear es:

```
let s=new suma();  
s.primerValor(10);  
s.segundoValor(20);  
document.write('La suma de los dos valores es:'+s.retornarResultado());
```

Definición de varias clases.

1. Plantear una clase persona (que permita cargar por teclado su nombre y edad). Por otro lado crear una clase empresa que tenga como atributo la edad tope para que una persona pueda ingresar como trabajador en la misma (60 años). Confeccionar un pequeño programa en JavaScript que defina 3 objetos de la clase persona y 1 de la clase empresa. Mostrar cuántas de esas personas están inhabilitadas para ingresar como trabajadores de acuerdo a la edad tope.

Vectores con componentes de tipo objeto.

1. Confeccionar una clase persona que permita almacenar el nombre y la edad. Luego definir un vector de 4 componentes de tipo persona. Imprimir el nombre de la persona de mayor edad; decir también si hay más de una persona con la edad mayor.

Creación de objetos literales

1. Crear un objeto que represente los datos de un participante de un juego. Definir las propiedades nombre, puntos y los métodos:  
imprimir (mostrar el nombre de jugador y los puntos actuales)  
aumentarPuntos (permite incrementar la cantidad de puntos actuales del jugador)  
verificarSiGano (mostrar un mensaje si los puntos superan a 1000)

Array: Diferentes formas de crearlos

1. Confeccionar un programa que defina dos array, en uno almacenar los días de la semana y en otro los nombres de los meses. Emplear dos sintaxis diferentes para crear e inicializar sus componentes.

Array: densos

1. Crear un vector con 10 valores aleatorios comprendidos entre 1 y 500. Luego crear otros dos vectores y copiar los elementos menores a 250 en el segundo vector y los mayores o iguales a 250 en el tercer vector. Imprimir los tamaños del segundo y tercer vector. Imprimir los tres vectores.



## Programación II Guía de JS

### Array: no densos o dispersos

1. Crear un vector disperso que representen premios de una rifa. Las posiciones indican los números que tienen premio. Almacenar en dichas posiciones los premios. Genera 10 premios con los montos 1000,2000,3000 etc.  
La cantidad de números de la rifa son 1000.  
Generar valores aleatorios para los números con premio.  
Imprimir Los números con premio y los montos de los mismos.

### Array: métodos push y pop

1. Crear un vector con 5 valores aleatorios comprendidos entre 1 y 1000. Luego extraer los dos últimos elementos sumarlos y mostrarlos. Imprimir también el tamaño final del vector.

### Array: métodos sort y reverse

1. Cargar los sueldos de un conjunto de empleados. Finalizar la carga al ingresar cero. Mostrar por pantalla los sueldos ordenados de mayor a menor.

### Array: método splice

1. Crear un vector con una lista de valores. Luego borrar las componentes donde hay un 2 y al mismo tiempo insertar dos elementos con el valor 1.

### Array: método join

1. Crear un vector con los datos de 5 personas. Luego imprimir por pantalla uno abajo de otro utilizando el método join para generar un string y agregarle entre elementos la etiqueta html `<br>`.

### Array: método toString

1. Crear un vector de 10 elementos. Inicializar los valores del vector cuando se crea. Inicializar dos variables que resulten de llamar primero al método `toString()` y la segunda variable inicializarla llamando al método `join()` sin parámetros. Mostrar un mensaje si el resultado de la ejecución de los dos métodos retorna el mismo valor (empleando un `if` para comparar los string)

### Funciones: objeto arguments

1. Confeccionar una función que obtenga y retorne el mayor valor de una lista de enteros que le pasamos como parámetros.

### Funciones: anidadas

1. Confeccionar una función que reciba un parámetro y defina una variable local. Definir dentro de la función que acabamos de crear una función anidada que contenga un parámetro. En la función anidada imprimir el parámetro de la función que la contiene, el parámetro que tiene la función anidada y el valor de la variable local de la función principal.

### Funciones `setInterval`, `clearInterval`, `setTimeout` y `clearTimeout`

1. Confeccionar una página que luego de 10 segundos redirija a otro sitio empleando para ello la función `setTimeout`.





## Programación II

### Guía de JS

Operador condicional ?:

1. Ingresar el sueldo bruto de un trabajador. Si supera a 10000 aplicar un descuento del 10%, en caso contrario el descuento es del 5%.

Expresiones regulares: cuantificadores o repeticiones {x} {x,y} {x,}

1. Validar que se ingrese un número de 3 dígitos enteros, el carácter punto y 2 dígitos decimales.
2. Solicitar el ingreso de la coordenada de un punto. El formato a ingresar por teclado es:  
(999,999)  
Los números pueden tener entre 1 y 3 dígitos.