



# Ejercicios JavaScript

Objetivos:

Comprobar el entendimiento y aplicación práctica del lenguaje JavaScript; conceptos de Fundamentos de programación y Orientación a Objetos.



# ■ Instrucciones

Crea un documento html (extensión .html) con el código JavaScript embebido por cada uno de los siguientes ejercicios.

Las instrucciones de entrada y salida pueden ser por prompt y alert, o bien por controles input y document.write() excepto por consola.

Nombra a cada documento con el id de cada ejercicio indicado en el título del mismo.

>> Ejemplo

- Ejercicio KC\_EJ01
- Entregar KC\_EJ01.html





# ■ Parte 1

Entrada/salida, variables, tipos de datos y operaciones lógico/aritméticas.



## ■ KC\_EJ01

Crea un programa por el cual se solicite al usuario el radio de un círculo. El programa deberá calcular y mostrar el área.

Ejemplo

>> Entrada: 5

>> Salida: El radio es 78.5



## ■ KC\_EJ02

Crea un programa que solicite dos números y muestre los resultados de todas sus operaciones aritméticas.

Ejemplo

>> Entradas: 30 y 20

>> Salida:

$$30+2 = 32$$

$$30-2 = 28$$

$$30 \times 2 = 60$$

$$30/2 = 15$$

$$30^{**}2 = 900$$

$$30\%2 = 0$$



## ■ KC\_EJ03

Crea un programa que solicite una cantidad en Euros y muestre su equivalente en USD.

Ejemplo

>> Entrada: 10.0

>> Salida: \$10.06 USD



## ■ KC\_EJ04

Crea un programa que solicite tres notas y muestre su media.

Ejemplo

>> Entradas: 8, 9 y 6

>> Salida: La media es 7.6



## ■ KC\_EJ05

Crear un programa que solicite el año y mes de nacimiento de dos personas, y muestre el resultado de compararlas

Ejemplo

>> Entradas:

*mes 1 Enero*

*año 1 1980*

*mes 2 Febrero*

*año 2 1980*

>> Salida: False





## ■ KC\_EJ06

Crea un programa que solicite dos números y muestre los resultados de aplicar comparaciones relacionales.

Ejemplo

>> Entradas 30 y 2

>> Salida:

30 < 2 = False

30 > 2 = True

30 y 2 son iguales = False

30 y 2 son distintos = True





# ■ Parte 2

## Condicionales



## ■ KC\_EJ07

Crea un programa que solicite tres notas y calcule su media.  
Dependiendo del valor de la media, mostrará si el resultado es Apto o No Apto.

Ejemplo

>> Entradas: 10, 9 y 9  
>> Salida: Apto



## ■ KC\_EJ08

Crear un programa que reciba un número entero y muestre si es par o impar, positivo o negativo, o cero.

Ejemplo:

>> Entrada: 23

>> Salida: Positivo impar.



## ■ KC\_EJ09

Crea un programa que reciba una cantidad, y pregunte si desea convertirla de Euros a USD o de USD a Euros y muestre el resultado de la conversión.

### Ejemplo

>> Entradas:

- Cantidad 10.0
- Opción Euros a USD

>> Salida: \$10.06 USD



## ■ KC\_EJ10

Crear un programa que reciba un número del 1 al 7 y muestre el nombre de la película StarWars correspondiente.

Ejemplo

- >> Entrada: 5 (o bien V)
- >> Salida: StarWars V - El imperio contraataca.



## ■ KC\_EJ11

Crear un programa que reciba 3 números e indique cuál es el mayor y el menor.

Ejemplo

>> Entradas: 5, 19 y 2

>> Salida: El mayor es 19, el menor es 2



## ■ KC\_EJ12

Crear un programa que reciba una letra e indique si es vocal o consonante.

Ejemplo

>> Entrada: E

>> Salida: E es letra vocal







# ■ Parte 3

Arreglos y diccionarios



## ■ KC\_EJ13

Crear un programa que contenga una lista de 20 números y muestre un rango de la lista. El inicio y fin del rango serán introducidos por el usuario y el programa deberá validar que sean valores válidos.

### Ejemplo 1 (entrada no válida)

```
>> Valores: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
>> Entradas: 11 y 23
>> Salida: El rango debe ser de 1 a 20
```

### Ejemplo 2 (entrada válida)

```
>> Valores: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
>> Entrada: 11 y 15
>> Salida: 12, 13, 14, 15.
```



## ■ KC\_EJ14

Crear un programa que reciba un texto y muestre su longitud

Ejemplo

>> Entrada: Hola mundo

>> Salida: Longitud 10



## ■ KC\_EJ15

Crear un programa que reciba los el nombre y las calificaciones de 3 personas. Para cada persona deberá guardar la información en un arreglo. El programa no mostrará resultados de salida.



## ■ KC\_EJ16

Crear un programa que contenga un Arreglo de nombres. Solicitar un índice de la lista y mostrar el valor del índice. El programa deberá validar que el índice es válido.

### Ejemplo

- >> Lista predeterminada: “Pedro Picapiedra”, “Pablo Marmol”, “Bob Esponja”, “Patricio”
- >> Entrada: 2
- >> Salida: Bob Esponja



## ■ KC\_EJ17

Crear un programa que contenga un diccionario con nombres y correos electrónicos. Solicitar el nombre de una persona y mostrar su correo electrónico. Indicar con un mensaje apropiado cuando no se encuentre un resultado válido.

### Ejemplo

```
>> Diccionario predeterminado:  
    Pedro - pedro.picapiedra@gmail.com  
    Pablo - pmarmol123@gmail.com  
    Bob - bob@gmail.com  
  
>> Entrada: Pablo  
>> Salida: pmarmol123@gmail.com
```





# ■ Parte 4

## Bucles



## ■ KC\_EJ18

Crear un programa que muestre los números naturales del 1 al N, donde N será dado por el usuario.

Ejemplo

>> Entrada: 8

>> Salida: 1, 2, 3, 4, 5, 6, 7, 8





## ■ KC\_EJ19

Crear un programa que almacene 10 números dados por el usuario y muestre únicamente los pares.

Ejemplo

>> Entrada: 10, 3, 6, 12, 20, 5, 9, 7, 11, 40

>> Salida: Son pares 10, 6, 12, 20, 40.



## ■ KC\_EJ20

Crear un programa que escriba la suma de los números del 1 a N, donde N será dado por el usuario.

Ejemplo

>> Entrada: 8

>> Salida:  $1+2+3+4+5+6+7+8 = 36$



## ■ KC\_EJ21

Crear un programa que reciba un número natural entero y muestre su tabla de multiplicar del 1 al 10.

Ejemplo

>> Entrada: 5

>> Salida:

5x1=5  
5x2=10  
5x3=15  
5x4=20  
5x5=25  
5x6=30  
5x7=35  
5x8=40  
5x9=45  
5x10=50



## ■ KC\_EJ22

Crear un programa que reciba los nombres y edades de 10 personas. Mostrar únicamente los nombres de las personas que tienen derecho a votar (mayores a 18 años).

### Ejemplo

- >> Entradas: Pedro 23, Bob 15, Patricio 7, Pablo 30, Betty 20, Pebbles 2 ...
- >> Salida: Tienen derecho al voto Pedro, Pablo y Betty.



## ■ KC\_EJ23

Crear un programa que reciba los el nombre y las calificaciones de N personas, mientras que el usuario no escriba “terminar”. Al terminar deberá mostrar la media de calificaciones de cada persona.

### Ejemplo

>> Entradas:

Pedro 6, 8, 7  
Pablo 9, 9, 10  
Bob 10, 10, 10  
**terminar**

>> Salida:

Pedro 7  
Pablo 9  
Bob 10



## ■ KC\_EJ24

Modificar el programa **KC\_EJ23** para que muestre los resultados ordenados por la media, de forma descendente.

Ejemplo

>> Entradas:

Pedro 6, 8, 7

Pablo 9, 9, 10

Bob 10, 10, 10

**terminar**

>> Salida:

Bob 10

Pablo 9

Pedro 7



## ■ KC\_EJ25

Crear un programa que contenga un número aleatorio del 1 al 100, sin mostrarlo, y permitir que el usuario intente adivinarlo. El usuario solamente tendrá 5 oportunidades, en cada oportunidad fallida se le darán pistas para saber si debe intentar con un número mayor o menor.

Ejemplo

>> Número a adivinar: 32

- Entrada: 10
- Salida: Intenta con un número mayor (te quedan 4 oportunidades)
- Entrada: 40
- Salida: Intenta con un número menor (te quedan 3 oportunidades)
- Entrada: 35
- Salida: Intenta con un número menor (te quedan 2 oportunidades)
- Entrada: 32
- Salida: ¡Bien, has adivinado! :D





# ■ Parte 5

## Funciones





## KC\_EJ26

Crear un programa con una función `pintaFila()` que arme las filas de una tabla html. Completar el programa con la estructura de una tabla e invocando a la función N veces, donde N es un valor introducido por el usuario.

Ejemplo

```
>> Entrada: 4
>> Salida:
    <table>
        #ciclo invocando 4 veces a la función pintarFila()
        <tr><td></td></tr>
        <tr><td></td></tr>
        <tr><td></td></tr>
        <tr><td></td></tr>
        #- - - fin ciclo
    </table>
```



## ■ KC\_EJ27

Crear un programa que contenga una función esPalindromo(texto) y determine si dicho texto es un palíndromo.

Ejemplo

```
>> Entrada: anita lava la tina  
>> Salida: El texto ingresado ES palíndromo
```



## ■ KC\_EJ28

Crear un programa que contenga una función `pintarRectangulo(lado, figura)`. Esta función deberá pintar en la página un cuadrilátero de lado x lado con la figura proporcionada.

Ejemplo:

```
>> Entrada: lado 10, figura +
```

```
>> Salida:
```

```
+++++++  
+       +  
+       +  
+       +  
+       +  
+       +  
+       +  
+       +  
+       +  
+++++++
```



## ■ KC\_EJ29

Modificar el programa **KC\_EJ24** (promedio de alumnos) de forma tal que el cálculo del promedio se realice a través de una función.



## ■ KC\_EJ30

Crear un programa que simule una máquina expendedora de gaseosas con las siguientes características:

- Las bebidas disponibles son: Fanta, Pepsi, 7Up.
- Todas las bebidas tienen un costo de €1,0
- La máquina recibe monedas de 10, 20 y 50 cent, y €1,0

El programa deberá permitir que el usuario seleccione una bebida e ingrese una a una las monedas. El programa deberá detenerse cuando el importe de la gaseosa haya sido completado y, de ser necesario, determinar el sobrante.





# ■ Parte 6

## Classes (prototipos)



# ■ KC\_EJ31

Crear una clase Alumno con los siguientes atributos:

>> numero\_matricula, nombre, apellido, correo\_electronico, estatus\_inscrito.

La matrícula deberá ser numérica, mientras que correo\_electronico, nombre y apellido como textos. El atributo estatus\_inscrito deberá ser un valor booleano.



## ■ KC\_EJ32

Crear una clase Módulo con los siguientes atributos:

>> Listado\_alumnos, fecha\_inicio, fecha\_fin.

El listado de alumnos deberá ser tipo Lista y contener objetos de tipo Alumno creado en el ejercicio *KC\_EJ31*.

En la misma clase Módulo deberá implementar métodos para

>> agregar objetos Alumno a la Lista

>> buscar un alumno

>> mostrar todos los alumnos con estatus\_inscrito == True.







# ■ Parte 7

## Objetos



## ■ KC\_EJ33

Usando las clases de los ejercicios **KC\_EJ31** y **KC\_EJ32**, crear un programa que contenga una instancia de la clase Módulo, con instancias de alumnos predefinidos en init. El programa permitirá al usuario:

- >> Ver todos los alumnos enrolados.
- >> Buscar un alumno por matrícula.





# ■ Parte 8

Herencia (cadenas de prototipos)



## ■ KC\_EJ34

Crear las clases AlumnoRemoto y AlumnoPresencial, ambas subclases de la clase Alumno creada en el ejercicio **KC\_EJ31**.

- >> AlumnoRemoto deberá contar con los atributos numero\_matricula, nombre, apellido, correo\_electronico, estatus\_inscrito, skype, huso\_horario.
- >> Mientras que AlumnoPresencial deberá definir los atributos numero\_matricula, nombre, apellido, correo\_electronico, estatus\_inscrito, numero\_asiento.

