Introducción a la programación

JavaScript

Introducción a la programación

Algoritmo

Un algoritmo es una secuencia ordenada de pasos que sirven para resolver un problema.

- ¿Sabrías explicar el algoritmo de la suma?
- ¿Cómo saldrías del aula?

  ¿Se podría hacer un algoritmo para ello?

Introducción a la programación

Programa

Cuando el problema a resolver lo puede realizar un ordenador entonces el algoritmo “se convierte” en programa.

- Al igual que un algoritmo se descompone en una secuencia de pasos, un programa se compone de una secuencia de instrucciones.

Introducción a la programación

Ciclo de desarrollo de Software

Introducción a la programación

Lenguajes de programación

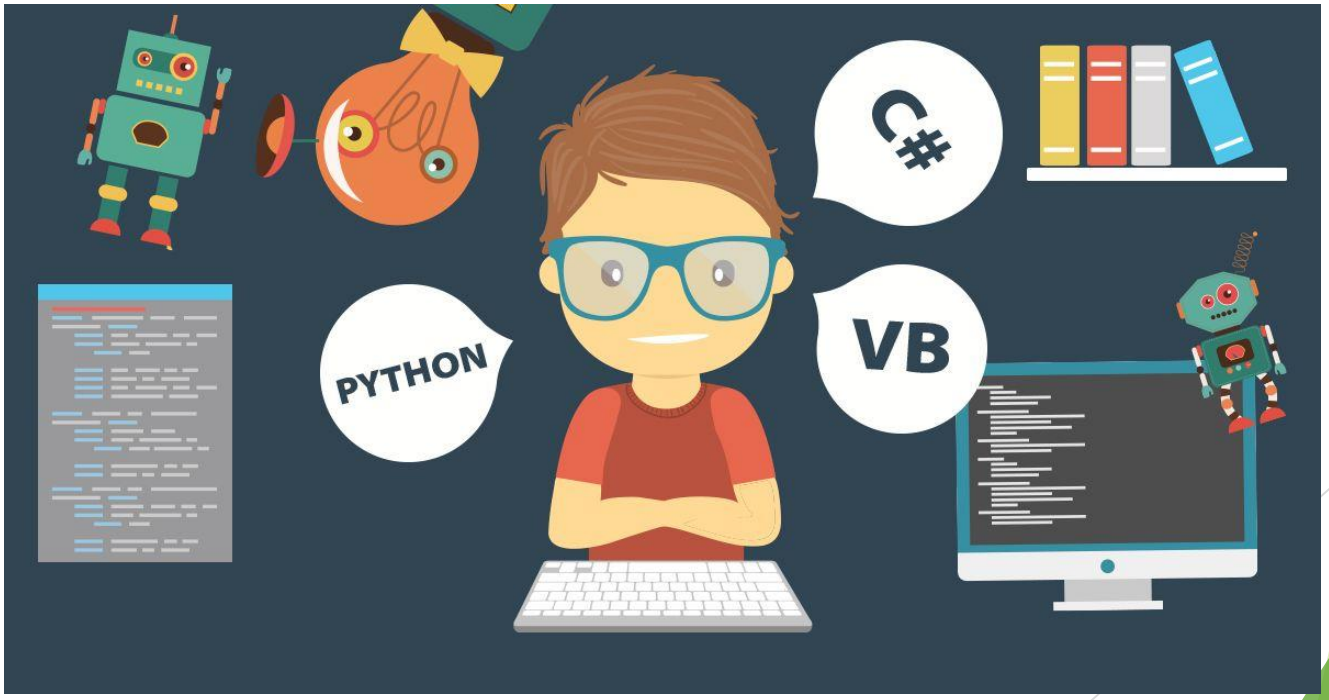
- Lenguaje máquina. El lenguaje máquina o código máquina es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de un ordenador.

```
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
11001010 11001010 11110101 00101011
11001010 11110101 00101011 00101011
11001010 00010111 11110101 00101011
00010111 11110101 00101011 00101011
11001010 11110101 00101011 00101011
```

Introducción a la programación

Lenguajes de programación

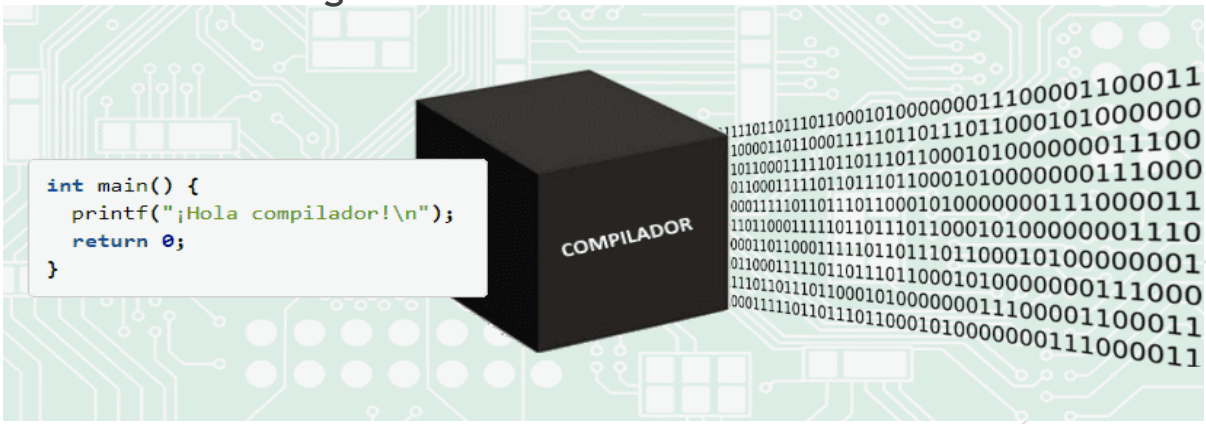
- Lenguaje de alto nivel. Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad con que los ejecutan las máquinas.



Introducción a la programación

Lenguajes de programación

- Compilador. Un compilador es un tipo de traductor que transforma un programa entero de un lenguaje de programación a otro. Usualmente el lenguaje objetivo es código máquina, aunque también puede ser traducido a un código intermedio o a texto.
- Intérprete. Un intérprete es un programa informático capaz de analizar y ejecutar otros programas. Los intérpretes se diferencian de los compiladores o de los ensambladores en que los intérpretes sólo realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.



JavaScript

Lenguajes de programación

- Es un lenguaje interpretado.
- Se utiliza principalmente en su forma del lado del cliente.
- Se utiliza principalmente en páginas web, aunque cada vez más fuera de ellas (documentos PDF, widgets de escritorio, etc.).

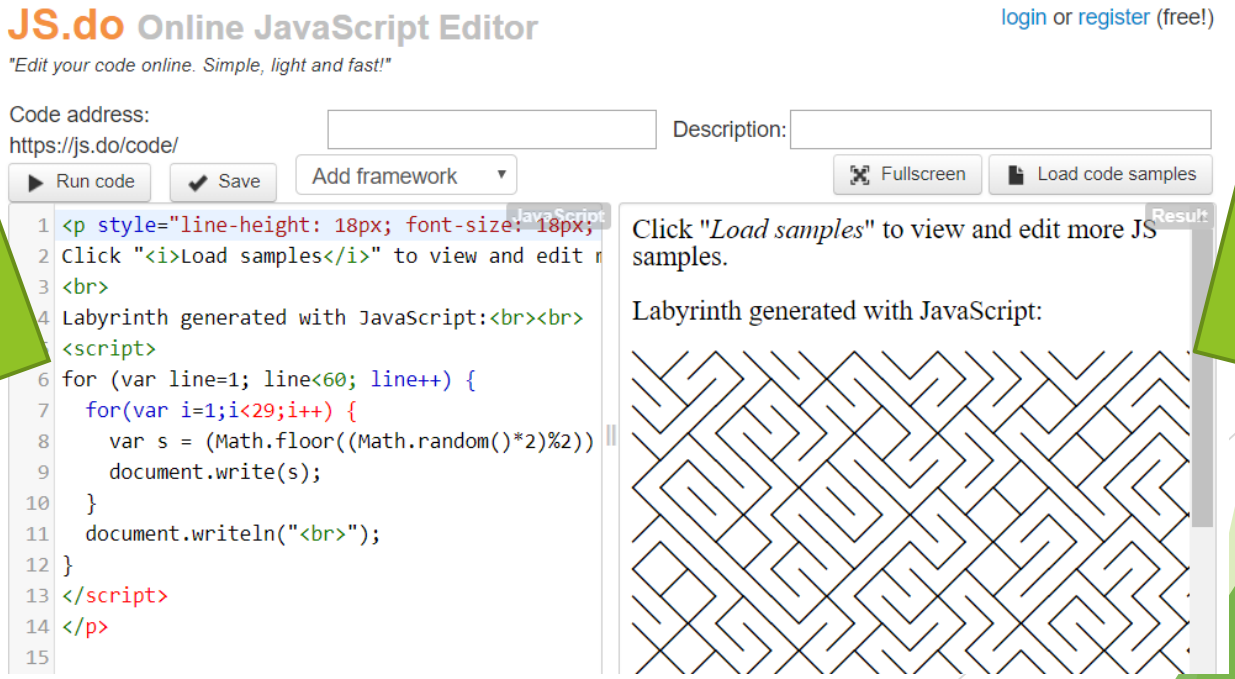

```
<!DOCTYPE html>
<html>
<head>
    <title>Apenas un guión</title>
</head>
<body>
<h1>
    <script language="javascript" type="text/javascript">
        document.write ("¡Hola Mundo!")
    </script>
</h1>
</body>
</html>
```

JavaScript

Entorno de programación

JavaScript es un lenguaje interpretado, eso significa que no precisa de un compilador que genere un archivo ejecutable sino que se ejecuta a través una aplicación que hace de intérprete.

Utilizaremos la aplicación web https://js.do/



Editor de código JavaScript

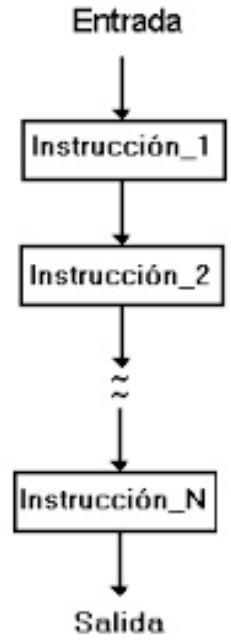
Resultado de la ejecución

JavaScript

Ejecución secuencial

JavaScript es un lenguaje secuencial, es decir, las instrucciones se ejecutan en secuencia, una detrás de otra según el orden en que están escritas.

Existen instrucciones que permiten alterar la secuencia de ejecución que veremos más adelante.



JavaScript

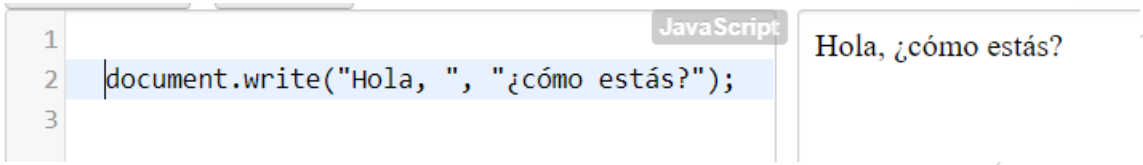
Instrucciones para escribir en pantalla

Queremos que nuestro programa pueda escribir un mensaje o un resultado en la pantalla para que lo pueda ver el usuario que utiliza el programa. Para ello usaremos la siguiente instrucción.

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

Algunos aspectos a tener en cuenta:

- document.write es el nombre de la instrucción y se debe escribir siempre tal cual, sin mayúsculas ni espacios ni ninguna otra modificación.
- El mensaje que quiero escribir se escribe dentro de los paréntesis que siguen a la instrucción.
- Si el mensaje se compone de varias partes o “trozos”, separo cada una de ellas con una coma (,).



JavaScript

Instrucciones para escribir en pantalla

Otras consideraciones:

- El texto que compone un mensaje y que quiero que se escriba de manera literal debe escribirse entre comillas dobles(“) o simples (‘).
- Si una parte del mensaje contiene solo un número, una expresión matemática, una variable o el nombre de una función NO se escribe entre comillas.
- Para escribir un salto de línea debemos utilizar el código HTML correspondiente, que es
.

```
document.write("Tengo ", 18, " años, <br> es decir, ", 18*365," días.")
```

Tengo 18 años,
es decir, 6570 días.

JavaScript

Instrucciones para escribir en pantalla

Realiza los siguientes programas JavaScript a modo de ejemplo:

1. Escribe un programa que muestre el siguiente mensaje: *Estoy aprendiendo JavaScript y estoy muy emocionado.*
2. Modifica el programa anterior para escribir el mismo mensaje pero ahora en tres “trozos”.
3. Escribe un programa que muestre el mensaje: *El día tiene 24 horas.* Hazlo de forma que el número 24 sea un “trozo” por sí solo del mensaje.
4. Modifica el programa anterior para que además el mensaje indique cuantos segundos tiene un día.
   NOTA: No calcules previamente cuántos segundos tiene un día, para multiplicar valores en JavaScript usamos el asterisco (*).
5. Escribe un programa que muestre el nombre de los miembros de tu familia o de un grupo de amigos. Cada nombre debe aparecer en una línea diferente.

JavaScript

Variables

En los programas utilizamos frecuentemente valores que, a priori, desconocemos o que podrían cambiar durante la ejecución.

Para almacenarlos y utilizarlos usamos variables. Las variables son un espacio de memoria RAM que utilizamos para nuestro programa.

Una variable se identifica por su nombre que debe comenzar por una letra y contener letras, número y el carácter guion bajo (_).

JavaScript

Variables

En primer lugar debemos declarar la variable, es decir, indicar al programa que vamos a utilizar un determinado espacio de memoria y qué nombre le vamos a dar.

```
var mi_variable
```

```
var nombre, edad, direccion
```

NOTA: Aunque JavaScript lo acepta, no es recomendable utilizar caracteres no ANSI, es decir, Ñ o tildes, en los nombres de las variables.

JavaScript

Variables

Antes de poder utilizar una variable se le debe asignar un valor. En nuestro ejemplo sería como introducir algo en la caja.

Para asignar un valor a una variable se utiliza el signo igual (=).

```
mi_variable = 35
```

```
var nombre = “Josema”, edad
```

NOTA: Cuando asignamos un texto a una variable debemos encerrarlo entre comillas dobles (“) o simples(‘).

JavaScript

Variables

Para utilizar el valor de una variable basta con escribir su nombre en el programa. En nuestro ejemplo sería como mirar qué hay en la caja.

```
document.write (“Me llamo ”, nombre)
```

JavaScript

Obtener información del usuario

En muchas ocasiones vamos a necesitar, como programadores, interaccionar con el usuario de nuestro programa.

Imagina que queremos hacer un programa que salude a un usuario por su nombre. Al programarlo es imposible que yo sepa como se va a llamar la persona que lo ejecute. Además, dependiendo de quién lo haga tendrá un nombre u otro.

La solución es preguntar al usuario su nombre y guardarlo en una variable, para ello utilizamos la instrucción prompt.

```
nombre = prompt ("¿Cómo te llamas?")
```

Variable para guardar el valor introducido

Instrucción para preguntar al usuario

Mensaje que se muestra al usuario

JavaScript

Obtener información del usuario

Cuando ejecutamos un programa JavaScript en una página web, se interpreta que la información introducida por el usuario mediante la instrucción prompt es un texto.

Cuando le pedimos a un usuario un número y queremos realizar alguna operación numérica o de comparación con otro número, en ocasiones, esta interpretación que hace prompt nos lleva a un error en el programa.

Necesitamos indicar explícitamente que queremos que ese valor se trate en el programa como un número y para ello usamos la función:

```
numero = parseInt(prompt("Introduce un número:")
```

Variable para guardar el valor introducido

Instrucción para "convertir" el valor a un número

Instrucción para preguntar al usuario

Mensaje que se muestra al usuario

JavaScript

Variables e introducción de datos de usuario

Realiza los siguientes programas JavaScript a modo de ejemplo:

6. Escribe un programa que pregunte su nombre al usuario, lo guarde en una variable y muestre un mensaje de saludo personalizado.
   Hola Juan! Encantado de conocerte.
7. Modifica el programa anterior para que el usuario además introduzca su edad y se incluya en el mensaje de saludo.
   Hola Juan! Así que tienes 17 años, encantado de conocerte.
8. Escribe un programa que pregunte su nombre a un usuario y lo guarde en una variable, luego le pregunte a qué se quiere dedicar y lo guarde en otra variable, y por último, muestre un mensaje como el siguiente:
   Juan, electricista es una profesión con mucho futuro.
9. Escribe un programa que pida al usuario que introduzca dos números y muestre un mensaje como el siguiente:
   La suma de 4 y 8 es 12.

JavaScript

Operadores matemáticos y lógicos

En JavaScript se utilizan los siguientes operadores:

- De asignación: =
- De concatenación: +
- Aritméticos:
  - Suma: +
  - Resta: -
  - Multiplicación: *
  - División: /
  - Cociente de la división entera: \
  - Resto de la división entera: %
- De comparación:
  - Igual: ==
  - Distinto: !=
  - Menor: <
  - Mayor: >
  - Menor o igual: <=
  - Mayor o igual: >=
- Lógicos:
  - Y (and): &&
  - O (or): ||
  - No (not): !
- Incremento y decremento: ++, --

JavaScript

Comentarios

Los programas tienen muchas líneas de código, con multitud de instrucciones, variables, funciones y otros elementos. Mientras se escribe un programa todo se ve muy claro, pero si miramos el código fuente de un programa que escribimos hace semanas, meses, o incluso años, es muy probable que no recordemos la intención con la que escribimos el código.

Esta situación se agrava si miramos el código escrito por otra persona.

Para aclarar la intención de lo que escribimos usamos comentarios.

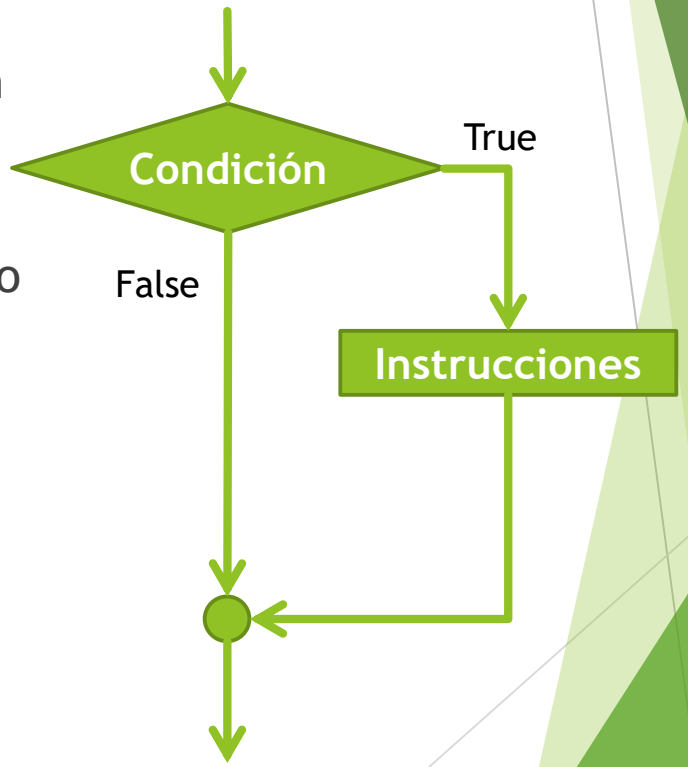
```
// Esto es un comentario de una línea en JavaScript

/* Esto es un comentario de
varias líneas
en JavaScript */
```

JavaScript

Estructuras de control

Estructura condicional

- Se emplea para tomar decisiones en función de una condición.
- Si la condición se cumple se ejecutarán unas instrucciones y si no se cumple se “saltan” esas instrucciones.




```
if (condición)
{
    instrucciones;
}
```

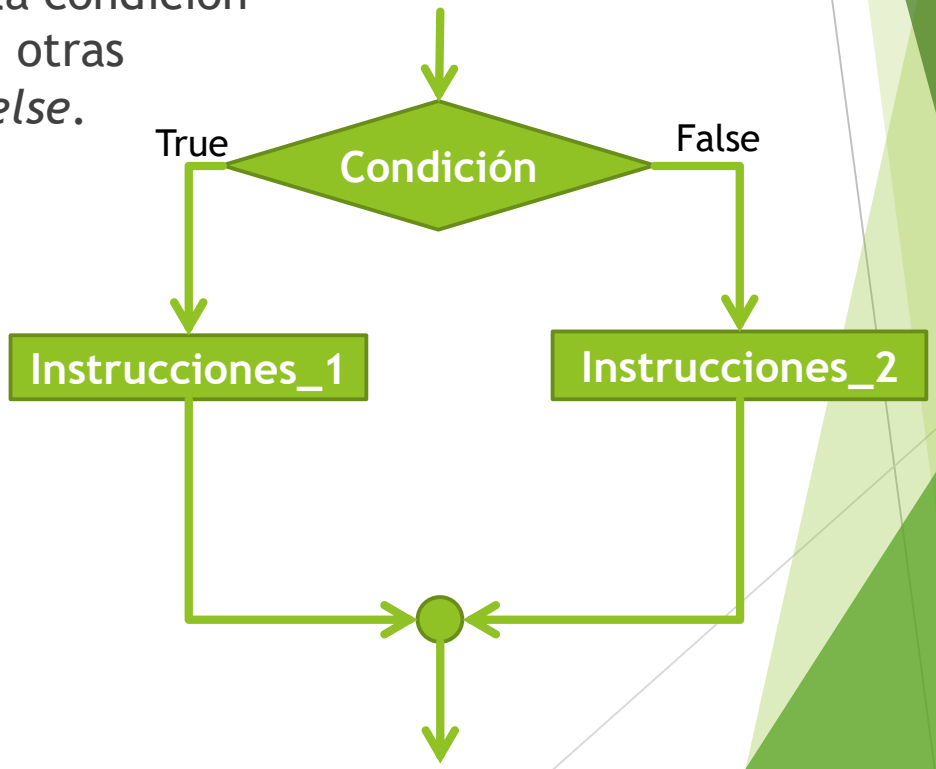
JavaScript

Estructuras de control

Estructura condicional

- Si queremos que cuando la condición no se cumpla se ejecuten otras instrucciones utilizamos *else*.

```
if (condición)
{
    instrucciones_1;
}
else
{
    instrucciones_2;
}
```

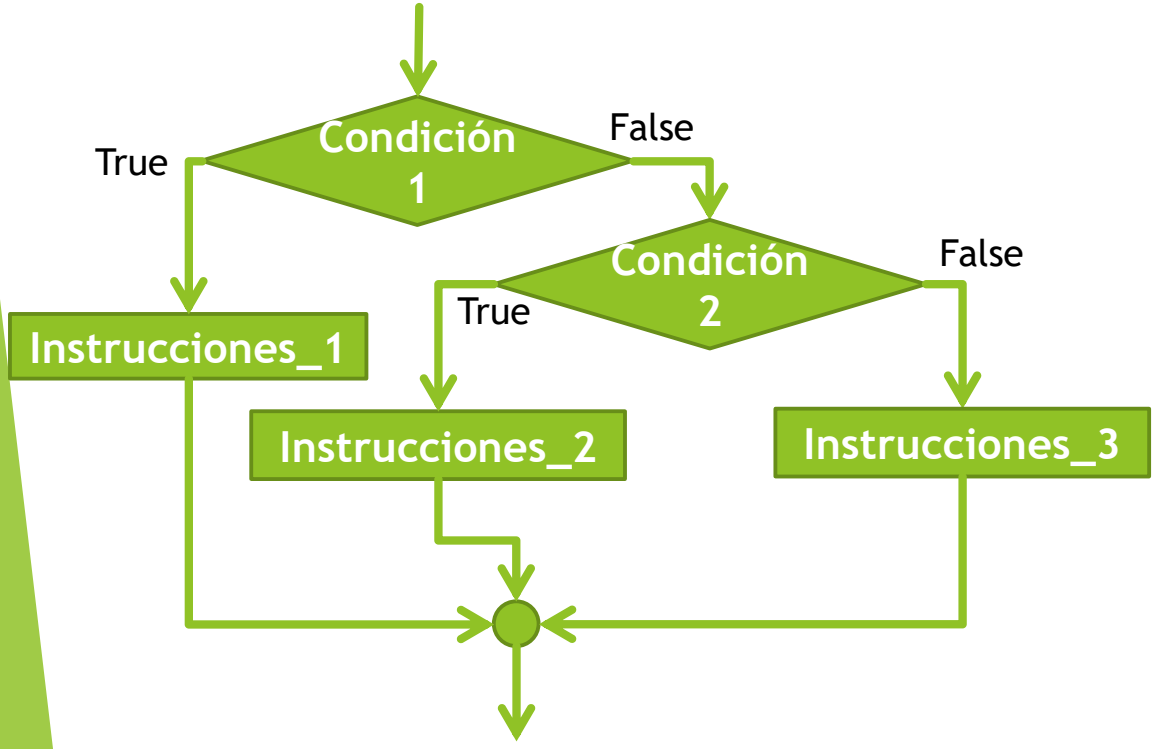


JavaScript

Estructuras de control

Estructura condicional

- Si necesitamos incluir más de dos posibilidades usamos *else if*.




```
if (condición_1)
{
    instrucciones_1;
}
else if (condición_2)
{
    instrucciones_2;
}
else
{
    instrucciones_3;
}
```

JavaScript

Estructuras condicional

Ejemplo:

```
var nota
nota = parseInt(prompt("¿Qué nota has sacado?"))
if (nota >= 5)
{
   document.write("¡Has aprobado!")
}
else
{
   document.write("¡Has suspendido!")
}
```

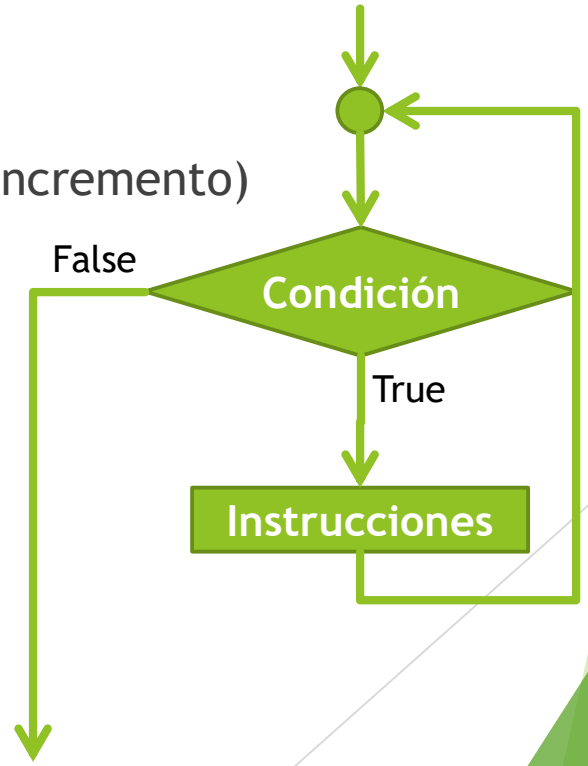
JavaScript

Estructuras repetitivas

Estructura *for*

- Se utiliza para ejecutar un bloque de instrucciones un determinado número de veces.

```
for (var_control=inicio; condición; incremento)
{
    instrucciones;
}
```



JavaScript

Estructura de repetición *for*

Ejemplo:

```
// Vamos a contar del 1 al 10

for (i=1;i<=10;i++)
{
      document.write(i, “<br>”)
}
```

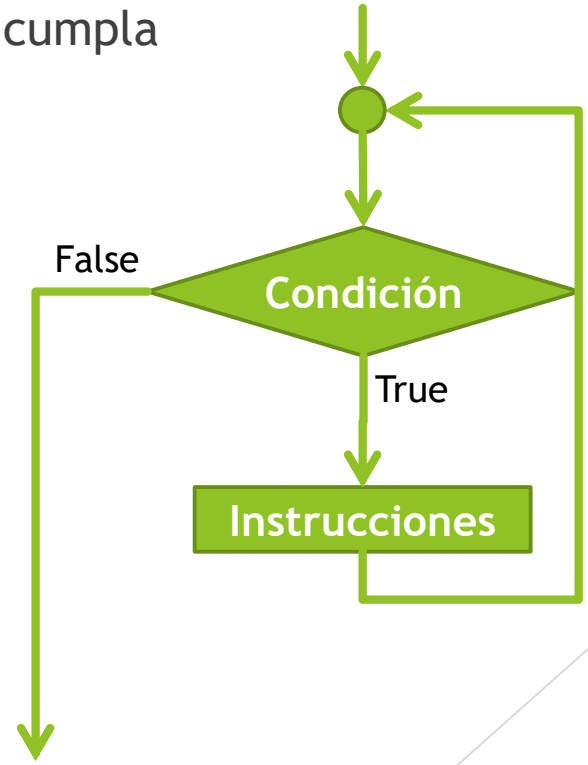
JavaScript

Estructuras repetitivas

Estructura *while*

- Se emplea para ejecutar un bloque de instrucciones mientras se cumpla una determinada condición.

```
while (condición)
{
    instrucciones;
}
```



JavaScript

Estructura de repetición *while*

Ejemplo:

```
var num
num = parseInt(prompt(“Introduce un número entero del 1 al 3”))

while (num<1 || num>3)
{
   alert(“¡He dicho un número entero del 1 al 3!”)
   num = prompt(“Introduce un número entero del 1 al 3”)
}

document.write(“Gracias!!”)
```