ESTRUCTURAS DE CONTROL DE FLUJO

Introducción

- Concepto de Estructuras de control de flujo
- ¿Por qué el nombre?
- Tipos de estructuras de control de flujo
  - Condicionales
  - Bucles

Condicional if

Condicionales

¿Tengo esos 816 euros?

SI

Contrato el viaje y me voy a Punta Cana

NO

Me voy de vacaciones al pueblo como todos los años

Nombre y Apellido (*)

E-mail (*)

Género
Hombre Mujer

Intereses
Deporte Política Ciencia

Comentarios

Campos requeridos (*)

Enviar datos Borrar campos

¿Se han rellenado el nombre y el email?

SI

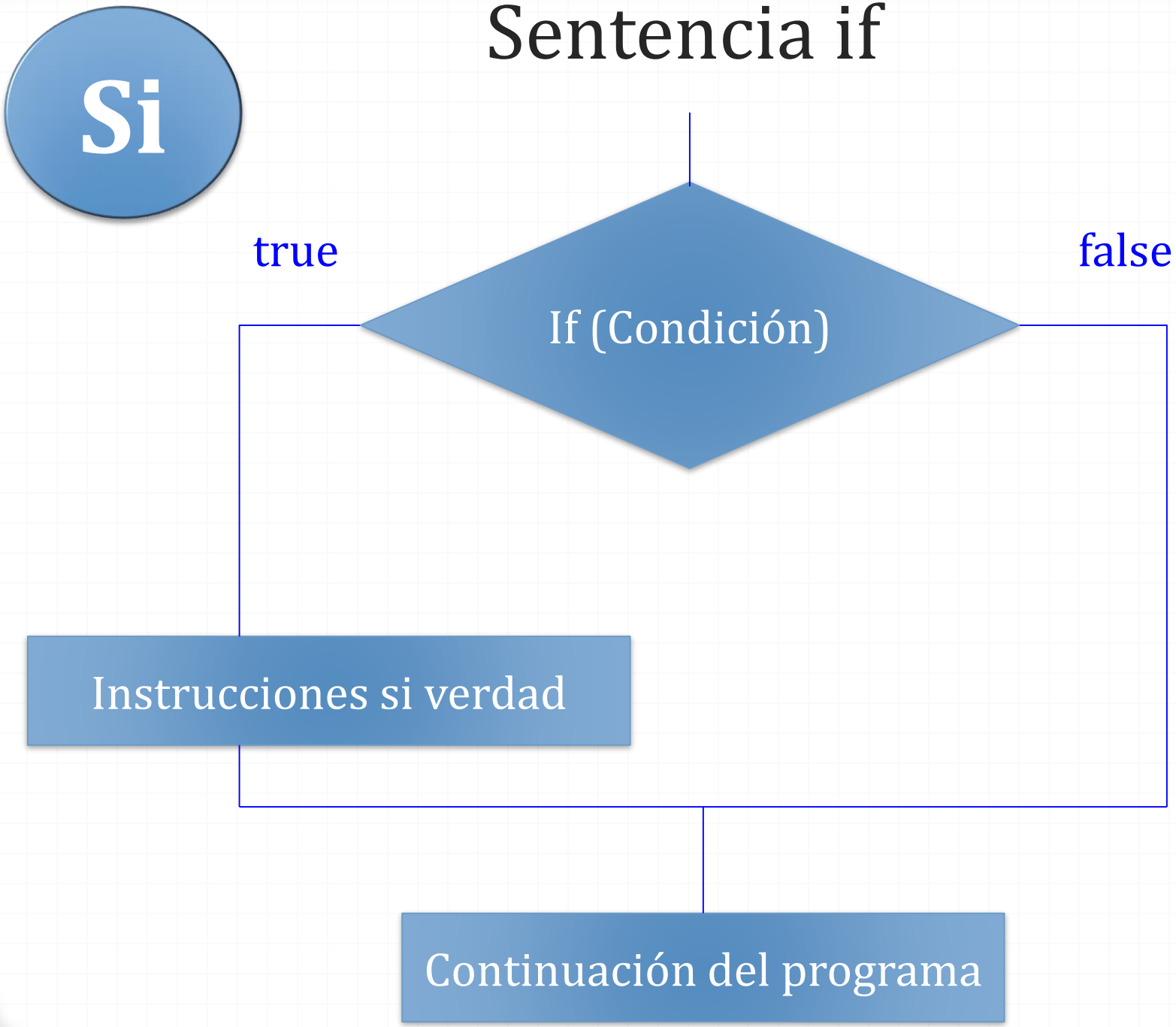
Enviar el formulario

NO

No enviar el formulario

Falta icono mal

Sentencia if



Sintaxis if

```
if (condicion) {
    bloque de instrucciones a ejecutar si la condición es verdad
}
```

- Si la condición se cumple se ejecutan todas las instrucciones que se encuentran dentro de {...}. Si la condición no se cumple no se ejecuta ninguna instrucción contenida en {...} y el programa continúa ejecutando el resto de instrucciones del script.

La palabra if deberá escribirse en minúsculas. Si usáramos letras mayúsculas (como por ejemplo, If o IF) se generará un error de JavaScript.

Operadores de comparación

```
var var1 = 3, var2 = 4;
```

Operador	Descripción	True
== Igualdad	Devuelve true si ambos operandos son iguales.	3 == var1 "3" == var1 3 == '3'
!= Desigualdad	Devuelve true si ambos operandos no son iguales.	var1 != 4 var2 != "3"
=== Estrictamente igual	Devuelve true si los operandos son igual y tienen el mismo tipo.	3 === var1
!== Estrictamente desigual	Devuelve true si los operandos no son igual y/o no tienen el mismo tipo.	var1 !== "3" 3 !== '3'
> Mayor que	Devuelve true si el operando de la izquierda es mas grande que el operando de la derecha.	var2 > var1 "12" > 2
>= Mayor o igual que (>=)	Devuelve true si el operando de la izquierda es mas grande que el operando de la derecha.	var2 >= var1 var1 >= 3
< Menor que	Devuelve true si el operando de la izquierda es mas pequeño que el operando de la derecha.	var1 < var2 "2" < "12"
<= Menor o igual que	Devuelve true si el operando de la izquierda es mas pequeño o igual que el operando de la derecha.	var1 <= var

Ejemplo

Vamos a crear una página web que me muestre un mensaje indicándome si en agosto me puedo ir de vacaciones a Punta Cana teniendo en cuenta el precio del viaje y el dinero del que yo dispongo.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Ejercicio condicionales</title>


</head>
<body>
    <h1>Viaje a Punta Cana</h1>

    <script>
        var precioViaje=816;
        var dineroDisponible=1000;

        if (dineroDisponible>=precioViaje) {
            alert ("Sácate el billete");
            dineroDisponible=dineroDisponible-precioViaje;
        }

        alert("Te quedan " + dineroDisponible + " euros");

    </script>

</body>
</html>
```

Modificar el script anterior y suponer que el dinero disponible ahora son 600 euros

Modificar el script anterior para que ambos precios nos los pida por pantalla

Operadores lógicos

Supongamos x=5 e y=7

Operador	Significado	True
&& (Y lógico)	Evalúa varias condiciones y devuelve cierto cuando todas ellas son ciertas	(x < 10 && y > 5)
|| (O lógico)	Evalúa varias condiciones y devuelve cierto cuando al menos una de ellas es cierta	(x==10 || y>5)
! (Negación)	Niega la condición a evaluar	!(x>y)

Los dos ejemplos siguientes son iguales:

```
var mostrarMensaje = true;

if(mostrarMensaje == true) {
   alert("Hola Mundo");
 }
```

```
var mostrarMensaje = true;

if(mostrarMensaje) {
   alert("Hola Mundo");
}
```

Recuerda que el operador de comparación es == no =

Operadores lógicos

- Ejemplo de uso del operador OR

```
var mostrado = false;
var usuarioPermiteMensajes = false;
if(!mostrado || usuarioPermiteMensajes) {
    alert("Es la primera vez que se muestra el mensaje");
}
```

mostrado = false => !mostrado = true	true || false = true
usuarioPermiteMensajes = false	

- El único caso en el que este operador devuelve falso es cuando todas las condiciones son falsas.

Operadores lógicos

- Ejemplos de uso del operador de negación:

```
var mostrado = false;
if(!mostrado) {
   alert("Es la primera vez que se muestra el mensaje");
}
```

mostrado = false => !mostrado = true	El mensaje se muestra

```
var mostrado = false;
var usuarioPermiteMensajes = true;
if(!mostrado && usuarioPermiteMensajes) {
    alert("Es la primera vez que se muestra el mensaje");
}
```

mostrado = false => !mostrado = true	true && true = true
usuarioPermiteMensajes = true	

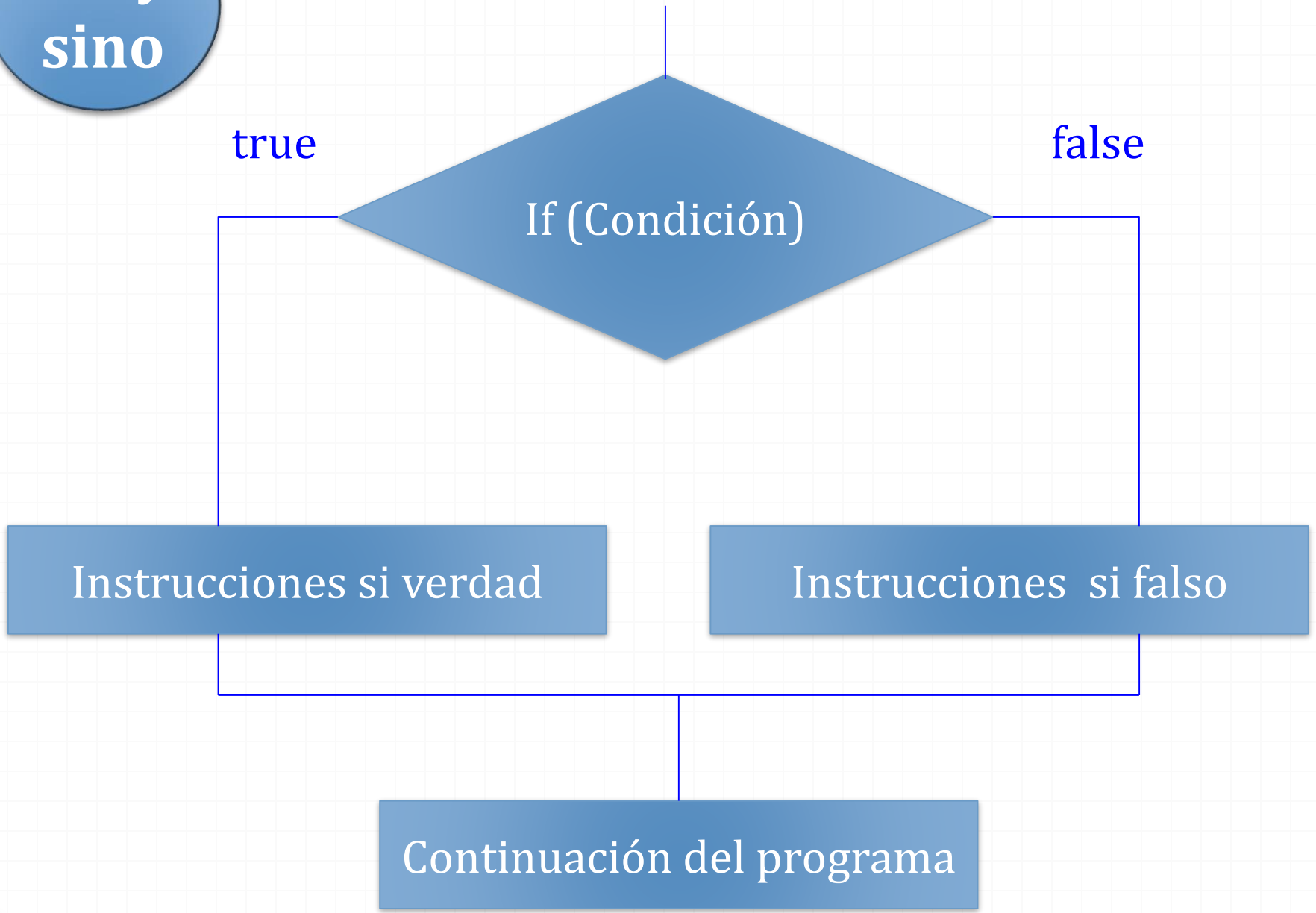
Manos a la obra

Crear un script que pida al usuario un mes del año y si ese mes es agosto le muestre un mensaje que le diga "Tienes ganas de vacaciones ¡¡¡eh pillín!!!

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Ejercicio condicionales</title>


</head>
<body>
    <h1>Meses del año</h1>

    <script>
        var mes= prompt("Dime un mes del año");

        /*Comprobamos si el mes es agosto*/
        if (mes == "agosto") {
            alert ("Tienes ganas de vacaciones, ¡¡¡eh Pillín!!!");
        }

        document.write("El mes que has escogido es "+mes);

    </script>

</body>
</html>
```

Sentencia if … else

Si…y sino



Sentencia if ... else

- Sintaxis

```
if(condicion) {
   ...
}
else {
   ...
}
```

- Ejemplo:

```
var edad = prompt("Dime tu edad");
if(edad >= 18) {
   alert("Eres mayor de edad");
}
else {
   alert("Aún no eres mayor de edad");
}
```

Manos a la obra

- El cálculo de la letra del Documento Nacional de Identidad (DNI) es un proceso matemático sencillo que se basa en obtener el resto de la división entera del número de DNI y el número 23. A partir del resto de la división, se obtiene la letra seleccionándola dentro de un array de letras.
- `var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];`
- Por tanto si el resto de la división es 0, la letra del DNI es la T y si el resto es 3 la letra es la A.
- Con estos datos, elaborar un pequeño script que:
  1. Pida al usuario su DNI y lo almacene en una variable. También deberá pedir al usuario la letra de su DNI y apuntarla en otra variable.
  2. En primer lugar (y en una sola instrucción) se debe comprobar si el número es menor que 0 o mayor que 99999999. Si ese es el caso, se muestra un mensaje al usuario indicando que el número proporcionado no es válido y el programa no muestra más mensajes.
  3. Si el número es válido, se calcula la letra que le corresponde según el método explicado anteriormente.
  4. Una vez calculada la letra, se debe comparar con la letra indicada por el usuario. Si no coinciden, se muestra un mensaje al usuario diciéndole que la letra que ha indicado no es correcta. En otro caso, se muestra un mensaje indicando que el número y la letra de DNI son correctos.

Sentencia else if

- Las sentencias if…else se pueden encadenar para realizar varias comprobaciones seguidas:

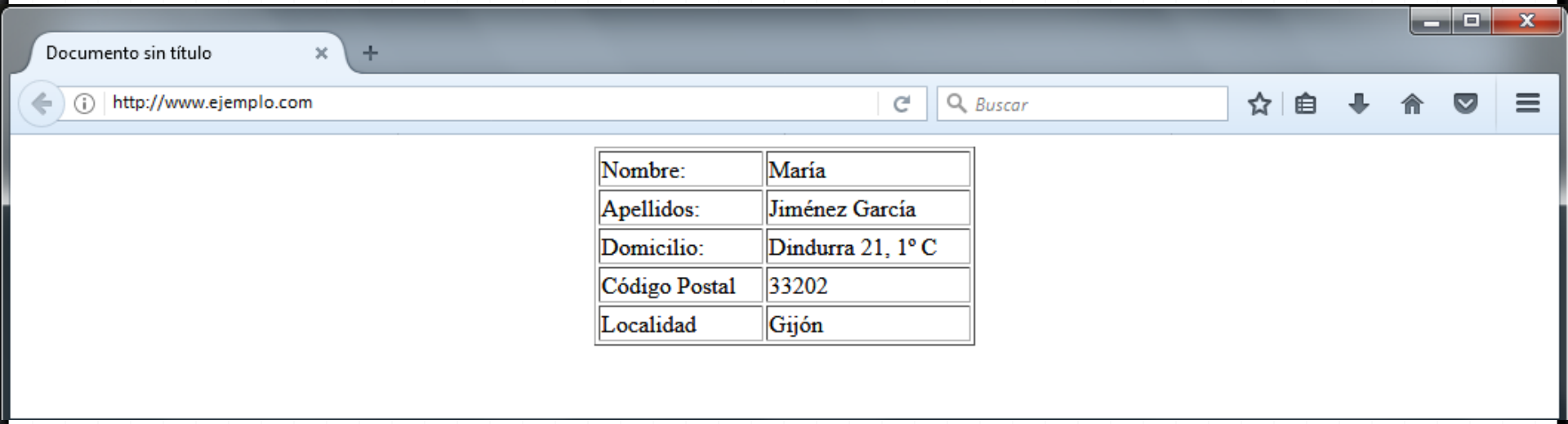
```
if(edad < 12) {
   alert("Todavía eres muy pequeño");
}
else if(edad < 19) {
   alert("Eres un adolescente");
}
else if(edad < 35) {
   alert("Aun sigues siendo joven");
}
else {
   alert("Piensa en cuidarte un poco más");
}
```

Manos a la obra

- Realizar una página web que contenga un script que solicite al usuario el nombre y edad de dos amigos. Luego debe comparar las edades de ambos amigos y mostrar un mensaje en pantalla donde indique si los dos amigos tienen la misma edad o cuál de los dos es mayor.

Manos a la obra

- Realizar una página web que contenga un script que permita al usuario elegir un color de fondo



Calculadora sencilla

- Vamos a empezar creando nuestro código base

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Mi primera calculadora</title>
    <script>


    </script>

</head>
<body>




</body>
</html>
```

- Solicitamos los datos al usuario:

```
var num1=prompt("Introduce el primer número");
var num2=prompt("Introduce el segundo número");
var operacion=prompt("¿Qué operación deseas realizar? (Suma / Resta / Multiplicación / División").toUpperCase();
```

- Definimos el if para la primera operación

```
if (operacion=="SUMA") {
    alert (num1+num2);
}
```

- Definir el código para la resta

```
if (operacion=="SUMA") {
    alert (parseInt(num1)+parseInt(num2));
} else if (operacion=="RESTA") {
    alert (parseInt(num1)-parseInt(num2));
}
```

- Definir el parseInt antes del if
- Definir las otras dos operaciones que nos faltan

```
if (operacion=="SUMA") {
    alert (num1 + num2);
} else if (operacion=="RESTA") {
    alert (num1 - num2);
} else if (operacion=="MULTIPLICACIÓN") {
    alert (num1 * num2);
} else if (operacion=="DIVISIÓN") {
    alert (num1 / num2);
} else  {
    alert ("Lo sentimos. Esa operación no está permitida");
}
```

- Definir la operación de multiplicación y división para que la interprete correctamente tanto si el usuario escribe la palabra acentuada como sin acentuar.

```
if (operacion=="SUMA") {
    alert (num1 + num2);
} else if (operacion=="RESTA") {
    alert (num1 - num2);
} else if (operacion=="MULTIPLICACIÓN" || operacion=="MULTIPLICACION") {
    alert (num1 * num2);
} else if (operacion=="DIVISIÓN" || operacion=="DIVISION") {
    alert (num1 / num2);
} else  {
    alert ("Lo sentimos. Esa operación no está permitida");
}
```

Control de errores

```
if (!isNaN(num1) && !isNaN(num2)){

    var operacion=prompt("¿Qué operación deseas realizar? (Suma / Resta / Multiplicación / División").toUpperCase();

    num1=parseInt(num1);
    num2=parseInt(num2);

    if (operacion=="SUMA") {
        alert (num1 + num2);
    } else if (operacion=="RESTA") {
        alert (num1 - num2);
    } else if (operacion=="MULTIPLICACIÓN" || operacion=="MULTIPLICACION") {
        alert (num1 * num2);
    } else if (operacion=="DIVISIÓN" || operacion=="DIVISION") {
        alert (num1 / num2);
    } else {
        alert ("Esa operación no está contemplada");
    }
} else {
    alert ("No se puede realizar la operación");
}
```

Calculadora 06

```
if (!isNaN(num1) && !isNaN(num2)){

    var operacion=prompt("¿Qué operación deseas realizar? (Suma / Resta / Multiplicación / División").toUpperCase();

    num1=parseInt(num1);
    num2=parseInt(num2);

    if (operacion=="SUMA") {
        alert (num1 + num2);
    } else if (operacion=="RESTA") {
        alert (num1 - num2);
    } else if (operacion=="MULTIPLICACIÓN" || operacion=="MULTIPLICACION") {
        alert (num1 * num2);
    } else if (operacion=="DIVISIÓN" || operacion=="DIVISION") {
        alert (num1 / num2);
    } else {
        alert ("Esa operación no está contemplada");
    }
} else {

    if (typeof(num1)=="string" || typeof(num2)=="string") {
        alert ("Uno o ambos valores de los que has introducido no son numéricos");
    } else {
        alert ("Ha ocurrido un error no identificado");
    }

}
```

Estructura switch

Introducción

- La estructura `if...else` se puede utilizar para realizar comprobaciones múltiples y tomar decisiones complejas. Sin embargo, si todas las condiciones dependen siempre de la misma variable, el código JavaScript resultante es demasiado redundante:

```
if(numero == 5) {
...
}
else if(numero == 8) {
...
}
else if(numero == 20) {
  ...
} else {
...
}
```

Definición

- En estos casos, la estructura switch es la más eficiente, ya que está especialmente diseñada para manejar de forma sencilla múltiples condiciones sobre la misma variable. Su definición formal puede parecer compleja, aunque su uso es muy sencillo:

```
switch(variable) {
   case valor_1:
      ...
      break;
   case valor_2:
      ...
      break;
   ...
   case valor_n:
      ...
      break;
   default:
      ...
      break;
}
```

- Dentro del switch se definen todas las comparaciones que se quieren realizar sobre el valor de la variable. Cada comparación se indica mediante la palabra reservada `case` seguida del valor con el que se realiza la comparación. Si el valor de la variable utilizada por `switch` coincide con el valor indicado por case, se ejecutan las instrucciones definidas dentro de ese case.
- Normalmente, después de las instrucciones de cada case se incluye la sentencia `break` para terminar la ejecución del `switch`, aunque no es obligatorio.
  - Si se nos olvida un break, el script se ejecutará desde donde se umple la condición y seguirá ejecutándose independientemente de si se cumple o no la condición
- Las comparaciones se realizan por orden, desde el primer case hasta el último, por lo que es muy importante el orden en el que se definen los case.
- ¿Qué sucede si ningún valor de la variable del switch coincide con los valores definidos en los case? En este caso, se utiliza el valor default para indicar las instrucciones que se ejecutan en el caso en el que ningún case se cumpla para la variable indicada.

Manos a la obra

- Crear un script que pida al usuario que elija entre cuatro frutas: Naranja, Kiwi, Plátano o Manzana.
- Según la fruta que el usuario haya elegido, deberá mostrarse un mensaje diferente:
  - Naranja: La naranja tiene mucha vitamina C
  - Kiwi: El kiwi tiene muchas vitaminas
  - Plátano: El plátano tiene mucho potasio
  - Manzana: La manzana es buena para bajar el colesterol
- Si el usuario teclea cualquier otro nombre de fruta deberá mostrarse un mensaje que diga: Lo siento, esa fruta no lo conozco

ESTRUCTURAS DE CONTROL DE FLUJO

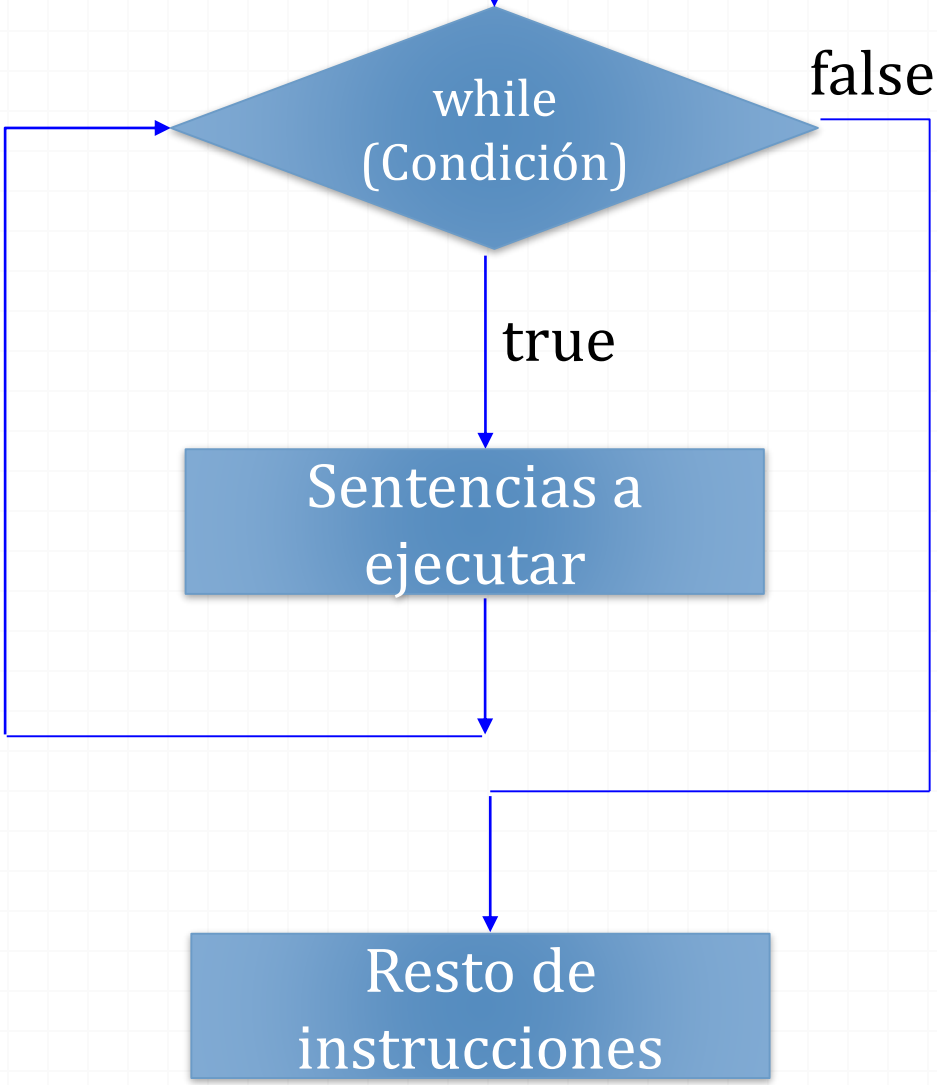
Bucles

Tipos de bucles

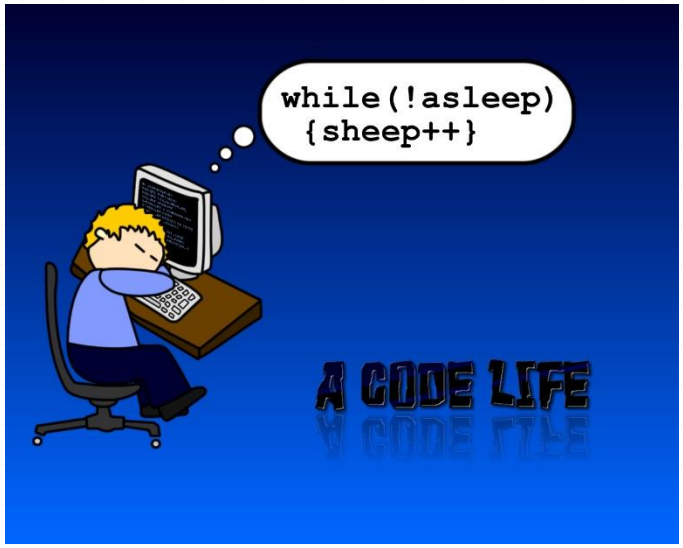
- Bucles
  - Determinados
    - for
  - Indeterminados
    - while
    - do while

while

Mientras



Sintaxis while




```
while (condicion) {
    bloque de instrucciones a repetir mientras la condición del
    bucle sea cierta
}
//Continuación del programa
```

Es necesario que dentro del bloque de instrucciones a repetir haya alguna sentencia que modifique los valores de las variables que intervienen en la condición si no se produciría un bucle infinito que haría fallar el navegador.

Ejemplo

- Crear un script que imprima el contenido de un array

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>while</title>

</head>
<body>
  <h1>Coches del concesionario</h1>

  <script>
    /*Definimos tres variables de tipo array llamada coches*/
    var coches=["BMV", "Seat", "Renault", "Mercedes", "Peugeot", "Citroen"];

    /*Imprimimos el contenido del array completo*/
    document.write("<p>Los coches que hay en el concesionario son: <br/>" + coches +"</p>");

  </script>
</body>
</html>
```

- Vamos a hacer lo mismo del ejercicio anterior pero utilizando el bucle while

```
<script>

  var coches=["BMV", "Seat", "Renault", "Mercedes", "Peugeot", "Citroen"];

  var indice=0;

  while (indice<coches.length) {
      document.write(coches[indice]);
      indice++;
  }

  document.write ("<p> No hay más coches para mostrar </p>");

</script>
```

```
<script>

  var coches=["BMV", "Seat", "Renault", "Mercedes", "Peugeot", "Citroen"];

  var indice=0;

  while (indice<coches.length) {
      document.write(coches[indice] + "<br/>");
      indice++;
  }

  document.write ("<p> No hay más coches para mostrar </p>");

</script>
```

Manos a la obra

Crear un script que pida al usuario por pantalla el nombre de una animal. Mientras el usuario escriba el nombre de un animal deberá pedirle otro. Cuando el usuario teclee la palabra fin, entonces deberá terminar y mostrar un mensaje que diga Has terminado el juego

```
</h1>

    <script>

        nomMascota="";

        while (nomMascota!="Fin"){
            nomMascota=prompt("Introduce el nombre de un animal. Fin para terminar");
        }

        document.write ("Has terminado el juego");

    </script>

</body>
```

Manos a la obra

Modificar el script anterior, para que los animales que el usuario introduce se vayan almacenando en un array y que al final se le muestre al usuario el listado de animales que ha introducido

```
<script>

    var animales=[];
    var i=0;
    nomMascota="";

    while (nomMascota!="Fin"){
        nomMascota=prompt("Introduce el nombre de un animal. Fin para terminar");
        animales[i]=nomMascota;
        i++;
    }

    document.write ("Has añadido los siguientes animales : <br/>");

    indice = 0;
    while (indice<animales.length) {
        document.write(animales[indice] + "<br/>");
        indice++;
    }

    document.write ("<p> Se ha terminado el juego </p>");

</script>
```

Paso 1

Evitar que la palabra fin se añada al array

```
while (nomMascota!="fin"){
    nomMascota=prompt("Introduce el nombre de un animal. Fin para terminar");
    nomMascota=nomMascota.toLowerCase();
    if (nomMascota!="fin") {
        animales[i]=nomMascota;
        i++;
  }
}
```

Paso 2

Mostrar el nombre de los animales en minúscula y con la primera letra en mayúscula

```
while (nomMascota!="fin"){
    nomMascota=prompt("Introduce el nombre de un animal. Fin para terminar");
    nomMascota=nomMascota.toLowerCase();
    if (nomMascota!="fin") {
        nomMascota=nomMascota.charAt(0).toUpperCase() + nomMascota.slice(1);
        animales[i]=nomMascota;
        i++;
  }
}
```

Comentarios finales

Evitar valores vacíos

```
while (nomMascota!="fin"){
    nomMascota=prompt("Introduce el nombre de un animal. Fin para terminar");
    nomMascota=nomMascota.toLowerCase();
    if (nomMascota!="fin" && nomMascota!="") {
        nomMascota=nomMascota.charAt(0).toUpperCase() + nomMascota.slice(1);
        animales[i]=nomMascota;
        i++;
  }
}
```

Comentarios finales

- Check Evitar que esta página cree diálogos adicionales en Firefox y Chrome
- Permitir contenido bloqueado en Internet Explorer
- Mensaje undefined en Internet Explorer

Manos a la obra

Crear un script que pida al usuario por pantalla su nombre, apellidos, dni y edad.
El nombre y los apellidos deben ser datos de tipo texto y no quedar vacíos. Mientras el usuario no introduzca un dato que cumpla estos criterios, se deberá seguir pidiendo que introduzca un nombre o apellidos válido.
El dni tiene que ser un número y no quedar vacío. Mientras el dato introducido no cumpla estas dos condiciones se debe seguir solicitando al usuario que introduzca su dni.
La edad tiene que ser un número y estar entre 18 y 100. Mientras no se cumpla este criterio, deberá seguir solicitando al usuario que introduzca su edad.

Comentarios finales

- Modificar el código para que no haga falta poner un script después de cada campo que muestre el valor introducido. (datos_2)
- Cambiar el último script añadido y ponerlo en el script del head justo después del código que ya contiene (datos_3)
  - ¿Qué ocurre?

ESTRUCTURAS DE CONTROL DE FLUJO

Bucles

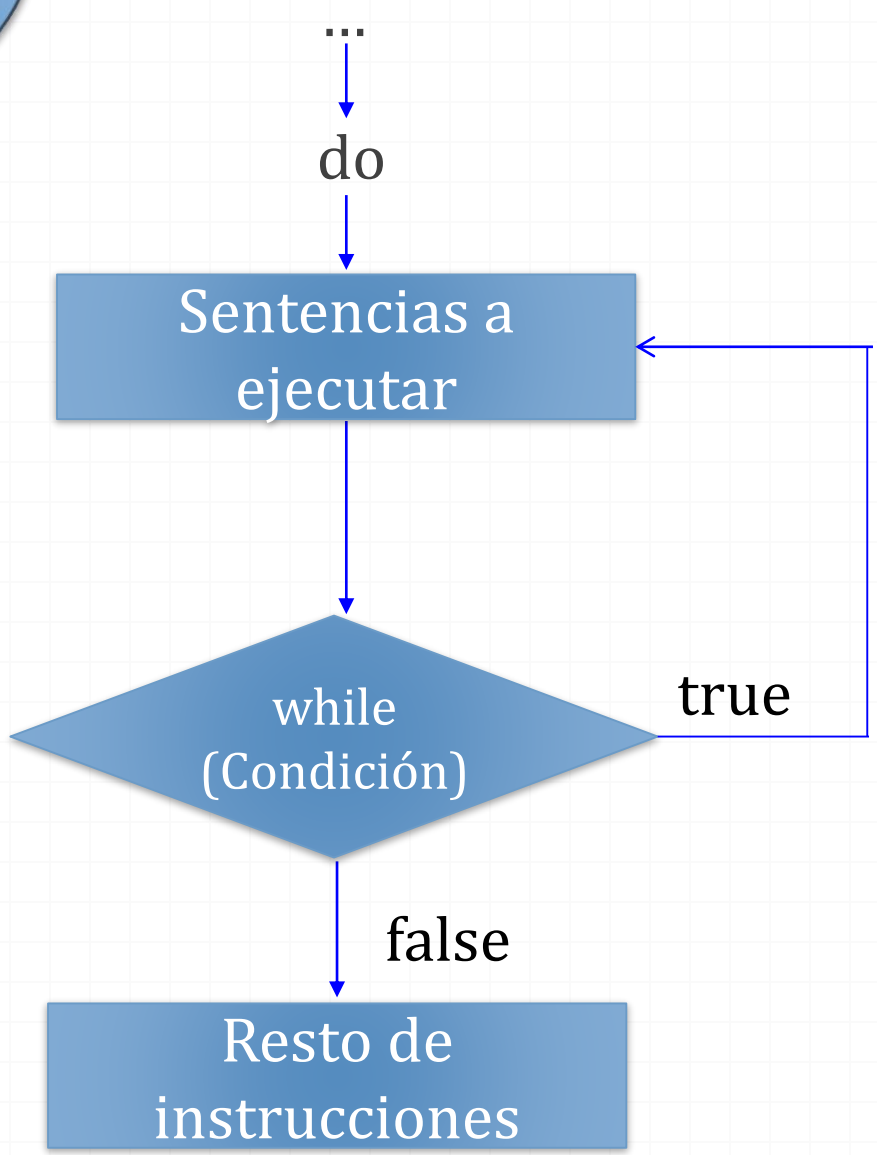
Tipos de bucles

- Bucles
  - Determinados
    - for
  - Indeterminados
    - while
    - do while

do...while

Bucle do ... while

Hacer mientras



Sintaxis do ... while

```
do{
    cerveza++;
}while(sobrio);
```

```
do {
    ...
} while(condicion);

//Continuación del programa
```

Es importante no olvidar que después del while() se debe añadir el carácter ; (al contrario de lo que sucede con el bucle while simple).

Ejemplo

- Vamos a modificar el código del último ejercicio del apartado anterior.

```
nombre=prompt ("Introduce tu nombre por favor:");
while (!isNaN(nombre) || nombre==null) {
    nombre=prompt ("Debes introducir un nombre válido:");
}
```

```
do {
    nombre=prompt ("Introduce tu nombre por favor:");
} while (!isNaN(nombre) || nombre==null);

do {
    apellidos=prompt("Introduce tus apellidos por favor:");
} while (!isNaN(apellidos) || apellidos==null);

do {
    dni=prompt("Introduce tu dni por favor:");
} while (isNaN(dni) || dni=="");

do {
    edad=prompt("Introduce tu edad por favor:");
} while (edad<=18 || edad>100 || isNaN(edad));
```

Recordar que por temas de compatibilidad con IE es aconsejable en el prompt añadir una coma y un mensaje o “”

Manos a la obra

Crear un script que genere un número aleatorio entre 0 y 10. Posteriormente, deberá solicitar al usuario un número. Si el número introducido por el usuario no es el mismo que el que ha calculado el sistema deberá seguir pidiéndole un número.

```
<script>
    var aleatorio=Math.round(Math.random()*10);

    var numero;

    numero=0;

    do {
        numero=prompt("Introducir un valor entre 0 y 10","");
    } while (aleatorio!=numero);


</script>
```

Manos a la obra

Modificar el script del ejercicio anterior para que según el número introducido por el usuario, el sistema emita una ayuda indicándolo si el número aleatorio es mayor o menor que el número introducido. Cuando el usuario acierte el número, deberá emitirse un mensaje de enhorabuena.

```
do {
    numero=prompt("Introducir un valor entre 0 y 10","");

    if (aleatorio>numero) {
        alert ("Mi número es más alto. Inténtalo de nuevo");
    }

    if (aleatorio < numero) {
        alert ("Mi número es más bajo. Inténtalo de nuevo");
    }

} while (aleatorio!=numero);

alert("¡¡¡Genial has acertado!!!. Eres todo un vidente");
```

Manos a la obra

Modificar el script del ejercicio anterior para que además del mensaje de enhorabuena nos diga en cuántos intentos hemos conseguido acertar el número.

```
<script>
    //Generamos un número aleatorio
    var aleatorio=Math.round(Math.random()*10);

    var numero, intentos;

    numero=0;
    intentos=0;

    do {
        numero=prompt("Introducir un valor entre 0 y 10","");
        if (aleatorio>numero) {
            alert ("Mi número es más alto. Inténtalo de nuevo");
        }
        if (aleatorio < numero) {
            alert ("Mi número es más bajo. Inténtalo de nuevo");
        }
        intentos++;
    } while (aleatorio!=numero);

    alert("Has acertado y sólo te ha llevado " + intentos + " intentos. ¡¡¡Eres todo un vidente!!!");

</script>
```

¿while o do while?

- Podríamos resolver el ejercicio anterior con while en lugar de do while?

```
var aleatorio=Math.round(Math.random()*10);

var numero, intentos;

numero=0;
intentos=0;

while (aleatorio!=numero) {

    numero=prompt("Introducir un valor entre 0 y 10","");

    if (aleatorio>numero) {
        alert ("Mi número es más alto. Inténtalo de nuevo");
    }

    if (aleatorio < numero) {
        alert ("Mi número es más bajo. Inténtalo de nuevo");
    }

    intentos++;
}

alert("Has acertado y sólo te ha llevado " + intentos + " inte
```

ESTRUCTURAS DE CONTROL DE FLUJO

Bucles

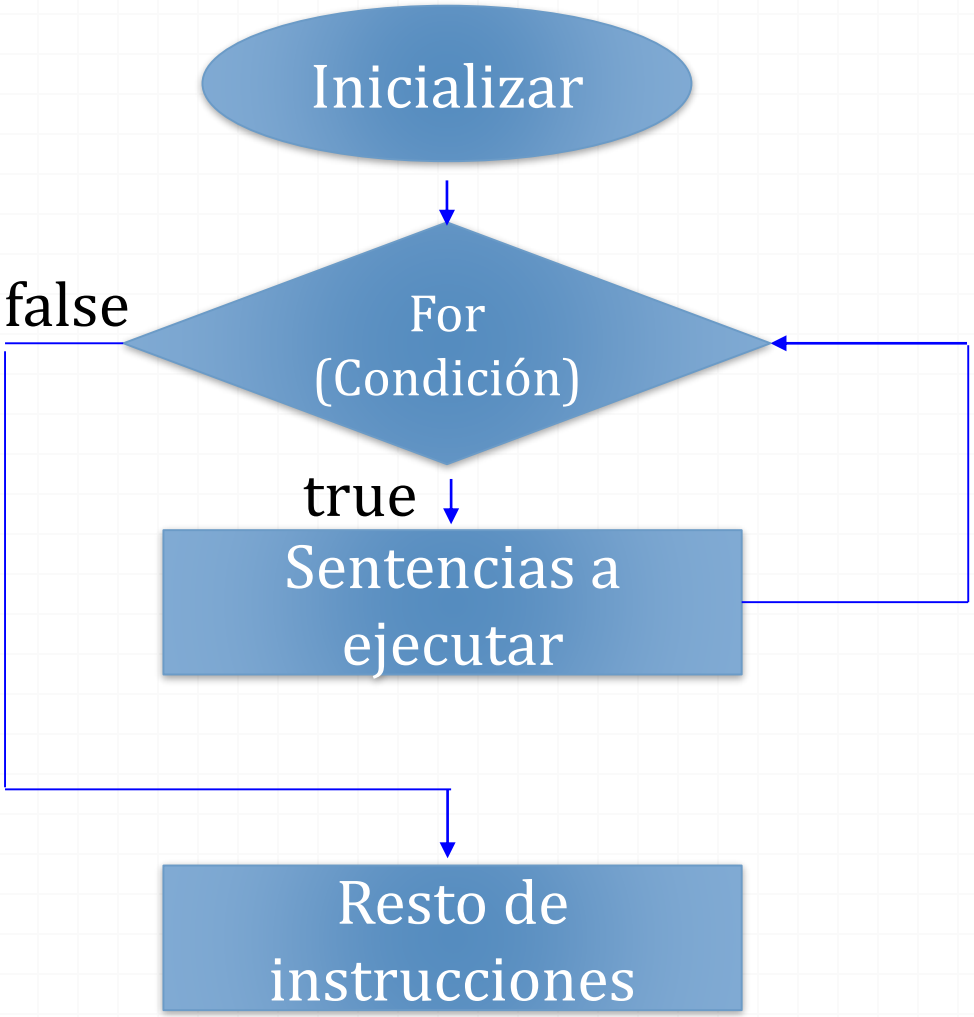
Tipos de bucles

- Bucles
  - Determinados
    - for
  - Indeterminados
    - while
    - do while

for

Bucle for

Para...



Sintaxis for

```
for(inicializacion; condicion; actualizacion) {
   ...
}
//Continuación del programa
```

Manos a la obra

```
for(var i = 0; i < 5; i++) {
   document.write("Hola mundo"+"<br/>");
}
```

Inicialización. Se crea la variable i y se le asigna el valor 0
La inicialización sólo se tiene en cuenta justo antes de empezar a ejecutar el bucle. Las siguientes repeticiones no se tiene en cuenta.

Condición. El bucle se ejecutará mientras se cumplan las condiciones y se dejan de ejecutar justo después de comprobar que la condición no se cumple

Actualización. Como la variable i se ha inicializado a un valor de 0 y la condición para salir del bucle es que i sea menor que 5, si no se modifica el valor de i de alguna forma, el bucle se repetiría indefinidamente.
Por ese motivo, es imprescindible indicar la zona de actualización, en la que se modifica el valor de las variables que controlan el bucle:

i++

En este caso, el valor de la variable i se incrementa en una unidad después de cada repetición

Manos a la obra

Crear un script que escriba los números impares que hay entre el 1 y el 20.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Bucle for</title>
</head>

<body>

    <h1>Primeros impares</h1>

    <script>
        for (var i=1; i<20;i+=2) {
            document.write(i + "<br/>");
        }

    </script>
</body>
</html>
```

Manos a la obra

El siguiente script muestra los días de la semana contenidos en un array.

```
<script>
   var dias = [“Lunes”, “Martes”, “Miércoles”, “Jueves”, “Viernes”,
   “Sábado”, “Domingo”];
   document.write(dias[0]+"<br>");
   document.write(dias[1]+"<br>");
   document.write(dias[2]+"<br>");
   document.write(dias[3]+"<br>");
   document.write(dias[4]+"<br>");
   document.write(dias[5]+"<br>");
   document.write(dias[6]);
</script>
```

Reescribir el código anterior definiendo un bucle con el menor número de sentencias posibles

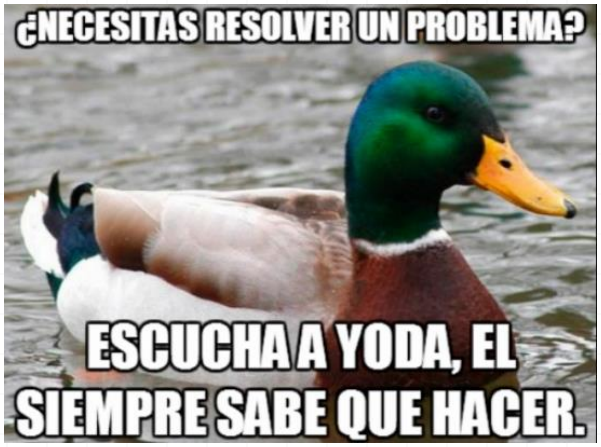
Manos a la obra

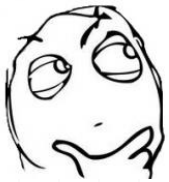
```
<script>
    var dias = ["Lunes", "Martes", "Miércoles", "Jueves",
    "Viernes", "Sábado", "Domingo"];

    for(var i=0; i<7; i++) {
       document.write(dias[i]+"<br>");
    }
</script>
```

Manos a la obra

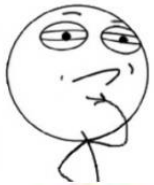
- Crear un script que primeramente muestre un mensaje al usuario pidiéndole diferentes modelos de coche y cuando el usuario teclee una palabra (por ejemplo, fin) termine de pedirle coches y entonces muestre los coches que el usuario ha añadido.
  - Este script es muy parecido al de los animales que hicimos en while solo que utilizaremos el bucle for para recorrer el array.

Bucle infinito

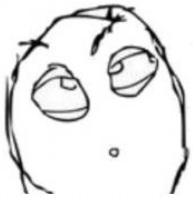




Un momento, si le hago caso a Yoda le debo hacer caso al pato.



Y si le hago caso al pato, debo seguir el consejo de Yoda.



Pero si escucho a Yoda debo escuchar al pato, para seguir el consejo de Yoda que le haga caso al pato.

```
for(;;) {
  alert("Enhorabuena, has ganado un coche");
}
```