

## **2. Fundamentos de la Programación**

# ¿Qué es programar?

Una forma de comunicación

- Con el ordenador y con otros programadores
- Extremadamente precisa y clara
- Basada en ideas muy simples
  - El ordenador es muy, muy tonto
  - Pero muy, muy rápido
- Que se combinan para crear ideas más complejas

# ¿Qué es programar?

Programar es algo fácil de aprender...

- Ideas sencillas
- Con significados muy claros

...pero difícil de dominar

- Los elementos se pueden combinar de muchas formas
- Hasta el nivel de complejidad que queramos

# **A tener en mente**

Si algo no funciona como esperabas:

- ¡No te frustres!
- NO es culpa del ordenador
- NO es un bug del lenguaje
- ¡El problema está en tu código!
- Respira hondo y repásalo una vez más

# **Valores y variables**

# Valores

Los “sustantivos” que entiende el Javascript están divididos en categorías:

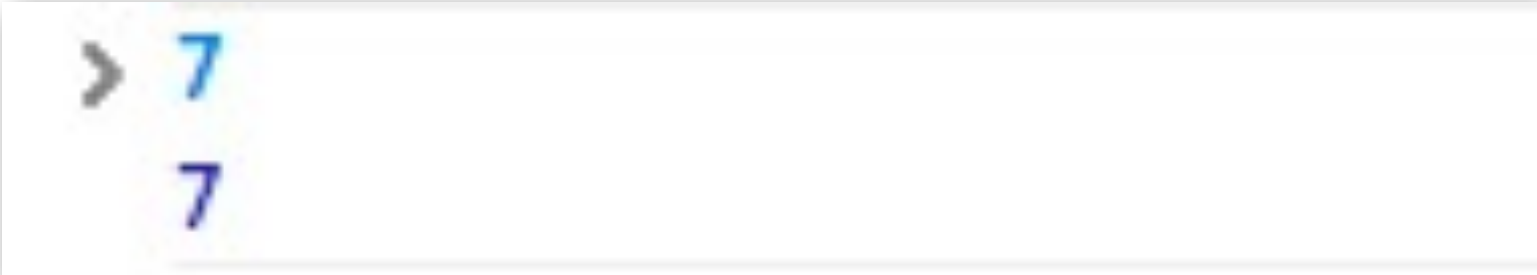
- Números
- Booleanos
- Cadenas de texto
- Funciones
- Objetos
- **undefined**

# Valores

Podemos referirnos a un valor simplemente tecleándolo en la consola o en el código

Por ejemplo, para referirnos al número 7

- Abre una consola
- Teclea el número 7 y Enter



```
> 7  
7
```

# Valores

Para referirnos a un valor booleano (verdadero o falso), tecleamos **true** o **false**

```
> true  
true  
> false  
false  
> |
```



# Valores

Para referirnos a una cadena de caracteres, tecleamos el texto entre comillas (dobles o simples)

```
> "¡Soy una cadena de caracteres!"  
  "¡Soy una cadena de caracteres!"  
> |
```

```
> 'Yo también...'  
  "Yo también..."  
> |
```

# Valores

Nos saltamos los objetos y las funciones hasta más adelante...

# Valores

**undefined** significa “¡No lo se (todavía)!”

- Se representa escribiendo **undefined**

```
> undefined  
undefined  
> |
```

# Valores

También podemos expresar un valor como el resultado de una operación con otros valores

Otra manera de referirse al número 7

- Abre una consola
- Teclea: **5 + 2**

```
> 5 + 2  
7  
>
```

# Valores

¿Cómo puedo referirme al número 7 utilizando solo el dígito **1** y la operación **+**?

# Valores

¿Cómo puedo referirme al número 7 utilizando solo el dígito **1** y la operación **+**?

```
> 1 + 1 + 1 + 1 + 1 + 1 + 1  
  7  
>
```

# Valores

¿Cómo puedo referirme al número 7 sin utilizar el dígito **7** ni la operación **+**?

# Valores

¿Cómo puedo referirme al número 7 sin utilizar el dígito **7** ni la operación **+**?

```
> 10 - 3  
  7  
  
>
```



# Valores

¿Cómo puedo referirme al número 7 sin utilizar el dígito **7** ni la operación **+**?

```
> 63 / 9
```

```
7
```

```
>
```

# Valores

¿Cómo puedo referirme al número 7 sin utilizar el dígito **7** ni la operación **+**?

```
> 14 * 0.5
```

```
7
```

```
>
```

# Valores

¿Cómo puedo referirme al número 7 sin utilizar el dígito **7** ni la operación **+**?

```
> 25 % 9
```

```
7
```

```
>
```

# Valores

Las operaciones que se pueden aplicar:

- ¡Dependen del tipo del valor!
- Por ejemplo, con números se puede hacer aritmética

```
> 4 + 4 - 5  
3  
> |
```

- Pero con cadenas no tiene sentido!

```
> "John Paul Jones" + "Robert Plant" + "Jimmy Page" - "John Bonham"  
NaN  
> |
```

# Valores

## Operaciones con números

### Aritmética

Suma: +

Resta: -

Multiplicación: \*

División: /

Módulo (resto): %

### Comparaciones

Mayor-menor: <, >, <=, >=

Equivalencia: ==, !=

# Valores

¿Es 10 mayor que 1?

```
> 10 > 1  
true
```

¿Es 10 menor o igual que 1?

```
> 10 <= 1  
false
```

¿Es 10 igual que 10?

```
> 10 == 10  
true
```

¿Es 10 diferente de 10?

```
> 10 != 10  
false
```

# Valores

## Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo?  
¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

# Valores

Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo?  
¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

$$17 \% 4$$



# Valores

Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo?  
¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

$$17 - (17 \% 4) / 4$$

# Valores

Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo? ¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

¿Por qué no funciona?

# Valores

## Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo? ¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

$$17 - ((17 \% 4) / 4) != (17 - (17 \% 4)) / 4$$

# Valores

## Agrupar operadores

- Tengo 17 bombones en una caja a repartir entre 4 amigos. ¿A cuántos bombones toca cada amigo?  
¿Cuántos me sobran? (un bombón no se puede partir en trozos más pequeños?)

$$(17 - (17 \% 4)) / 4$$

$$(17 - 1) / 4$$

$$16 / 4$$

$$4$$

# Valores

Haz esto en la consola, en una sola línea

- Piensa en un número
- Añádele 7
- Sustrae 2
- Sustrae tu número original
- Multiplica el resultado por 4
- Sustrae 2
- El resultado es... ¡18!

# Valores

Comprueba si es cierto que

La mitad de la cuarta parte de la edad de tu compañero multiplicada por el año en el que estamos es menor o igual que mil veces el resto de dividir tu edad entre el doble de los alumnos de la clase

# Valores

Operaciones con cadenas de caracteres

Concatenar: +

Equivalencia: ==, !=

Comparación: <, >, <=, >=

# Valores

Concatenar cadenas

```
> "Jimmy" + " " + "Page"  
"Jimmy Page"
```

Equivalencia

```
> "Led Zeppelin" == "Deep Purple"  
false  
> "asdf" == "ASDF"  
false
```

Comparación

```
> "Ala" < "Baraja"  
true  
> "Zapato" > "Castillo"  
true
```



# Valores

¿Qué pasa si hacemos esto?

```
"Jackson " + 5
```

# Valores

Operaciones con booleanos

Combinación: **&&**, **||**

Negación: **!**

# Valores

Negación: lo opuesto

```
> !false  
true  
> !true  
false
```

```
> !(4 > 2)  
false
```

# Valores

¿Verdadero o falso?

- `!( 1 == 1)`
- `!("Hola" <= "Adios")`
- `!false == true`
- `!( !false == true )`
- `!!true == !!!false`

# Valores

Conjunción: ... y ...

```
> true && true  
true  
> true && false  
false  
> false && false  
false
```

Si ambos lados son verdaderos, el total es verdadero

# Valores

Disyunción: ... o ...

```
> true || true  
true  
> true || false  
true  
> false || false  
false
```

Si alguno de los lados es verdadero, el total es verdadero

# Valores

¿Verdadero o falso?

- `true && (true && true)`
- `true && (true || false)`
- `(true && false) || (false && true)`
- `(8 != 7) || (1 > 2) || !(true && true)`
- `("Morrison" == "Jim") || ("The Doors" > "Cream")`

# Variables

¿Qué era un valor?



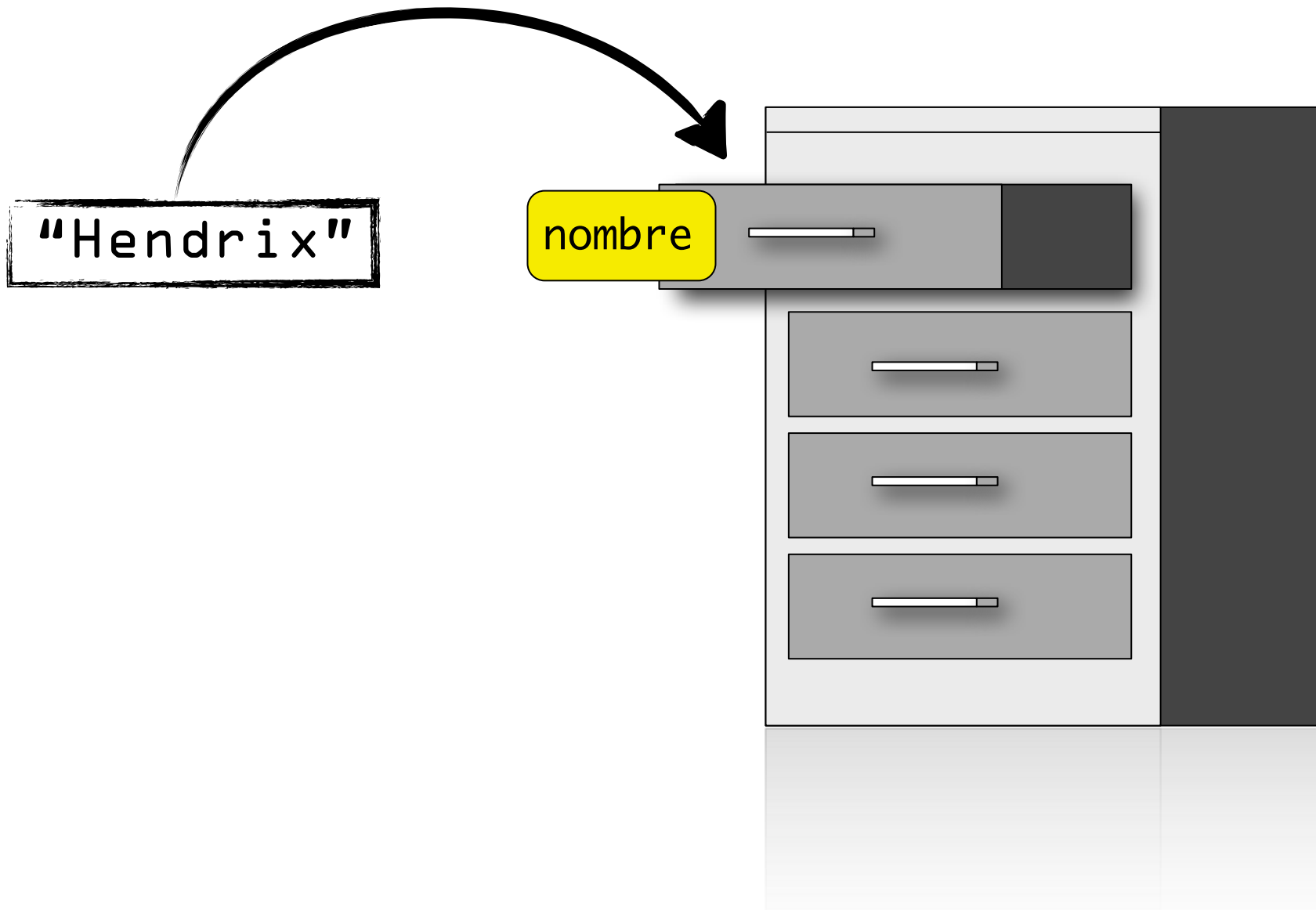
# Variables

Una variable es un “cajón” con una etiqueta

- Tiene nombre
- Puede alojar un valor “dentro”
- Podemos referirnos al valor que “guarda” usando el nombre de su etiqueta

```
var nombre = "Hendrix";
```

# Variables



# Variables

- Se crean con la palabra clave **var**
- Una vez creadas:
  - Accedemos al valor que guardan tecleando su nombre
  - Asignamos un nuevo valor con =

```
> var nombre = "Hendrix"  
undefined  
> nombre  
"Hendrix"  
> nombre = "Jimmy"  
"Jimmy"
```

# Variables

Un ejercicio:

- Guarda tu nombre en la variable **quien**
- Guarda tu cantante favorito en la variable **idolo**
- Ejecuta:

```
quien + " dice que " + idolo + " es un genio!"
```

# Variables

Un ejercicio:

- Ahora cambia **quien** por el nombre de tu compañero
- Guarda su cantante favorito en **idolo**
- Vuelve a ejecutar:

```
quien + " dice que " + idolo + " es un genio!"
```

# Variables

Una variable puede contener cualquier tipo de valor

```
> var variable = 10
undefined
> variable
10
> variable = "Creedence"
"Creedence"
> variable
"Creedence"
> variable = variable == "Creedence"
true
```

# Variables

Empecemos con

```
var contador = 0
```

¿De qué maneras puedo incrementar el valor de **contador**?

# Variables

```
contador = 1
```

```
contador = contador + 1
```

```
contador += 1
```

```
contador++
```

```
++contador
```



# Variables

¿De qué maneras puedo decrementar **contador**?

# Variables

¿De qué maneras puedo decrementar **contador**?

¿De qué maneras puedo decrementar **contador** de dos en dos?

# Variables

Si tenemos que

```
var cosa = 15
```

¿Qué valor tiene que tener **otra** para que sea cierto lo siguiente?

```
cosa == otra
```

# Variables

¿Cuál es el valor de **c** al final? ¿Cuál es el valor de **a**?  
¿Por qué?

```
var a = "Er";  
a += "ic";  
var b = a > "John Lennon" || a + " Clapton";  
a = b;  
var c = !(a > b || a != b) && a != "Eric";
```

¿Dónde está el truco?

# Variables

Cuál es el valor de...

- `true || false`
- `false || true`
- `true || "Layla"`
- `false || "Layla"`
- `"Layla" || "Crossroads"`

# Variables

Moralejas:

- Para Javascript:
  - `0`, `""`, `undefined`, `null` y `false` son falsedades
  - Cualquier otra cosa es verdadero
  - Ejercicio: demuestra que esto es cierto
- `||` y `&&` no solo devuelven `true` o `false`
  - ¿Qué valores devuelve cada uno?

# Variables

Prueba con:

- `true || "Maggie"`
- `"Maggie" || true`
- `false || "Maggie"`
- `false || "Maggie" || "Rod Stewart"`

Prueba cambiando los `||` por `&&`

# Intermedio: Mostrar cosas por pantalla

Las dos formas más sencillas de mostrar mensajes:

- Por consola:

```
console.log("tu mensaje aquí");
```

- En un popup:

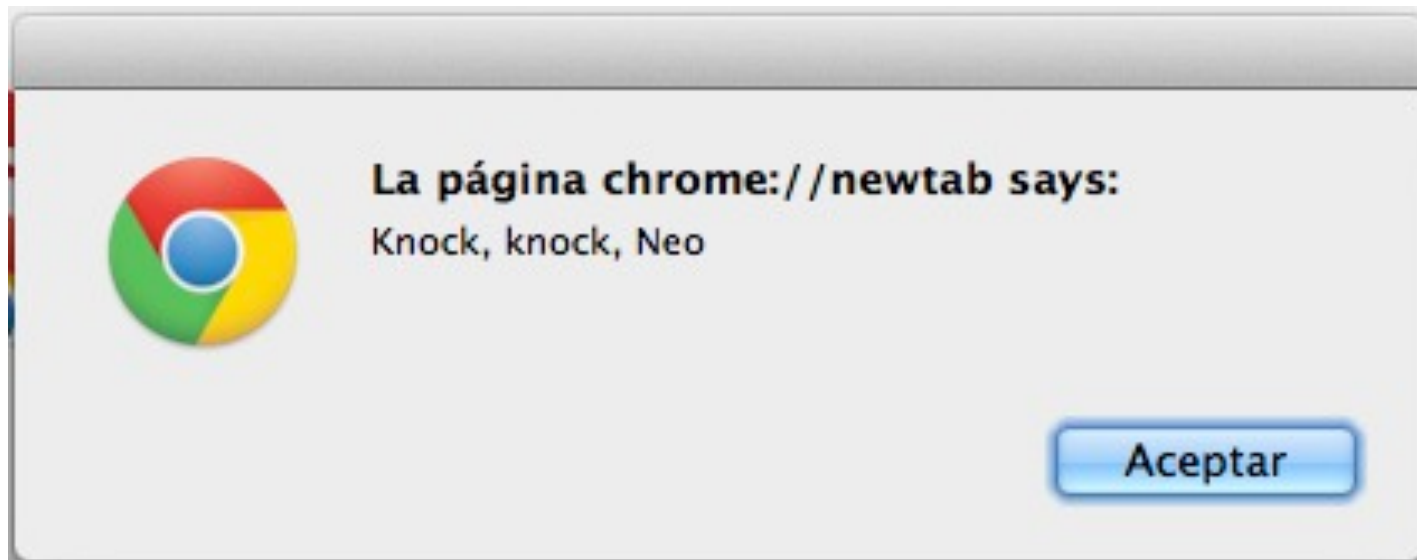
```
alert("Una molesta ventana de alerta");
```



# Intermedio: Mostrar cosas por pantalla

```
> console.log("Despierta, Neo");  
Despierta, Neo  
← undefined
```

```
> alert("Knock, knock, Neo")
```



# Intermedio: Interactuar con el usuario

Un par de formas rudimentarias de interacción:

- Confirm: ¿Si o no?

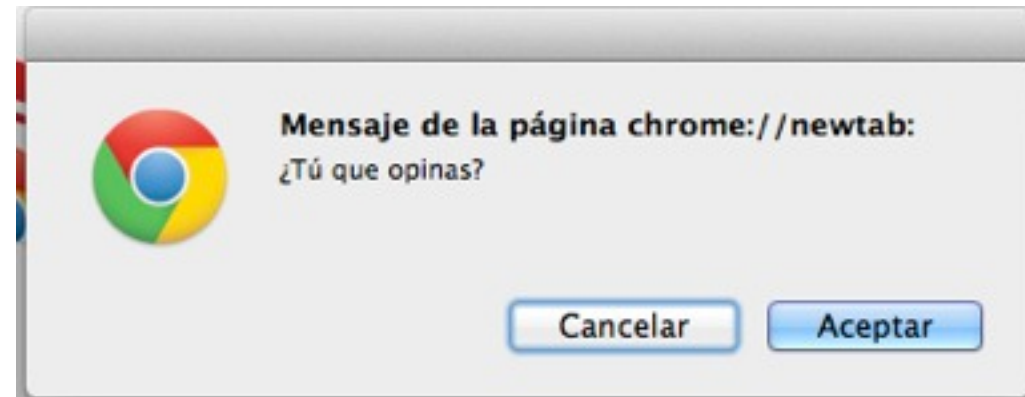
```
confirm("¿Quieres la pastilla roja?");
```

- Prompt: Dime algo

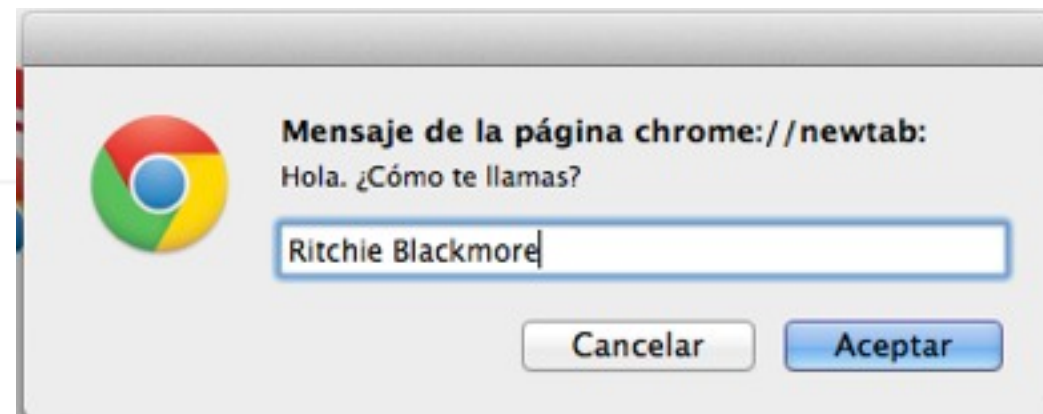
```
prompt("Dime una canción de UFO");
```

# Intermedio: Interactuar con el usuario

```
> confirm("¿Tú que opinas?")  
false
```



```
> prompt("Hola. ¿Cómo te llamas?")  
"Ritchie Blackmore"
```



# Intermedio: Interactuar con el usuario

Ejercicio:

- Crea una nueva página y un nuevo fichero js vacío
- Escribe un programa que:
  1. Pregunte al usuario su nombre
  2. Guarde el nombre en una variable
  3. Pregunte al usuario otra cosa (a tu elección) y guarde la respuesta
  4. Muestre un saludo personalizado utilizando sus respuestas

# Condicionales

- Tomar una decisión
- Ejecutar una parte del código u otra según una condición
- Las condiciones han de ser valores booleanos, verdadero o falso
- Las condiciones se construyen operando sobre valores y variables

# Condicionales

```
if (expresión) {  
    // Este código solo corre  
    // si expresión es verdadero  
    // ...  
}
```

# Condicionales

Ejercicio:

Preguntar al usuario si quiere escuchar un consejo

Si el usuario responde que sí: muéstrale un consejo

Si responde que no, no le muestres nada

# Condicionales

## Ejercicio

- Pídele al usuario que puntúe del 1 al 10 qué opina de los Rolling Stones
- Si su respuesta es mayor o igual que 7, dile “Hot Stuff!”
- Si su respuesta es menor que 7, dile “It’s Only Rock’n’Roll (But I Like It)”

¡Cuidado! ¡Tiene truco!



# El truco: cadenas a números

El valor “7” NO es un número!

- Es del tipo cadena de caracteres
- Pero podemos convertirlo en el número que representa
- Con la utilidad **parseInt(cadena)**

```
> parseInt("7")  
7
```

# Condicionales

```
if (expresión) {  
    // Este código solo corre  
    // si expresión es verdadero  
    // ...  
} else {  
    // Este código solo corre  
    // si expresión es falso  
    // ...  
}
```

# Condicionales

Reescribe el ejercicio anterior usando **else**

# Condicionales

```
if (expresión) {  
    // Este código solo corre  
    // si expresión es verdadero  
    // ...  
} else if (expresión_2){  
    // Este código solo corre  
    // si expresión es falso  
    // y expresión_2 es verdadero  
    // ...  
} else {  
    // Este código solo corre  
    // si expresión es falso  
    // y expresión_2 es falso  
}
```

# Condicionales

- Pregúntale dos cosas al usuario con **confirm**
- Muéstrale un mensaje u otro según sus respuestas sean verdaderas o falsas

# Bucles

- Ejecutar un bloque de código varias veces
- Dos tipos:
  - Ejecutar el bloque mientras una condición sea válida
  - Ejecutar el bloque un número definido de veces

# Bucles

```
while (expresion) {  
    // ...  
}
```

1. Si **expresion** es verdadero
2. Ejecuta el bloque de código
3. Cuando acabe el bloque, ve al paso 1

# Bucles

```
while (true) {  
    console.log("Socorrooo!!");  
}
```



# Bucles

```
while (true) {  
    console.log("Socorrooo!!");  
}
```

¡Cuidado: Bucle Infinito!

# Bucles

¿Cómo podríamos hacer un bucle que no fuera infinito?

# Bucles

```
var respuesta;  
var nombre = prompt("Hola. ¿Cómo te llamas?");  
  
while (respuesta != "déjame en paz!") {  
    respuesta = prompt("¡Qué rollo! Cuéntame algo, " + nombre);  
}
```

# Bucles

Canción para niños “10 botellas verdes”

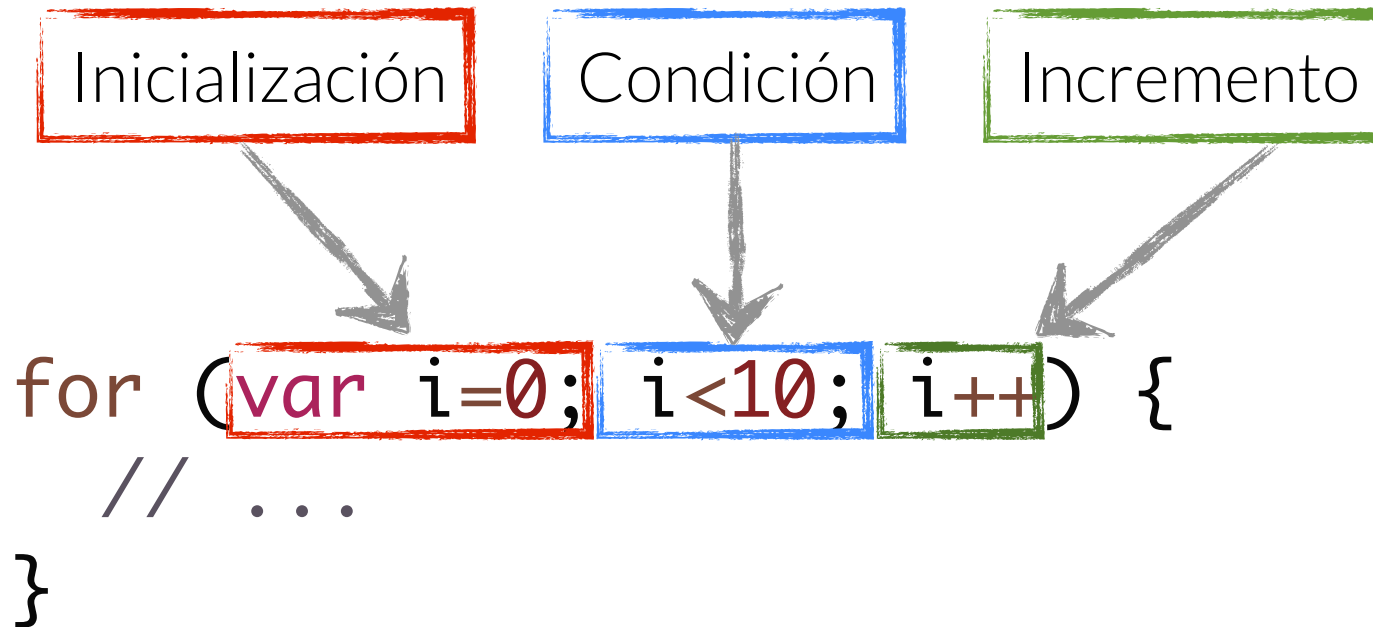
- Si no conocéis la canción: <http://www.youtube.com/watch?v=Xkwx0g1-yPE>
- Haz un bucle que muestre en la consola el texto de esta canción, desde 10 botellas hasta 1 botella

# Bucles

```
for (var i=0; i<10; i++) {  
    // ...  
}
```

1. Ejecuta la inicialización
2. Comprueba que la condición es cierta
3. Ejecuta el bloque
4. Ejecuta el incremento
5. Vuelve al paso 2

# Bucles



# Bucles

Ejercicio:

Haz un programa que cuente del 1 al 10 en la consola

# Bucles

Ejercicio:

Haz un programa que cuente de 10 a 1 en la consola



# Bucles

Ejercicio:

Haz un programa que muestre por consola todos los múltiplos de 7 menores que 500

# Bucles

## Ejercicio: fizz buzz

- Recorre los números del 1 al 100
- Muestra el número por la consola...
- Pero si el número es divisible entre 3, en vez del número muestra “fizz”
- O si el número es divisible entre 5, muestra “buzz”
- O si el número es divisible entre 3 y 5, muestra “fizz buzz”

# Bucles

Ejercicio: fizz buzz

```
1  
2  
fizz  
4  
buzz  
fizz  
7  
8  
fizz  
...
```

# Bucles

Ejercicio: Adivina qué número estoy pensando

- Al empezar a jugar, el ordenador piensa un número (1-100)
- El usuario lo intenta adivinar
- Si el número introducido es menor, se informa al usuario y se le pide que lo vuelva a intentar
- Si el número es mayor, lo mismo
- Cuando el usuario acierta, se le felicita, se le comunica cuántos turnos ha tardado en acertar y se le pregunta si quiere volver a jugar

# Bucles

Lo que necesitas saber:

`Math.random()`; número aleatorio entre 0 y 1

```
> Math.random()  
0.08254580339416862
```

`Math.floor()`; descarta la parte decimal

```
> Math.floor(14.234)  
14
```

# Tiempo de vida

Un programa Javascript...

- Se ejecuta en el navegador del usuario
- “Dentro” de la página web
- Tiene acceso a casi todos los elementos de la página
- Cuando el usuario sale de la página, se termina
- Si el usuario refresca la página, vuelve a empezar
- Dos usuarios viendo la misma página ejecutan dos copias independientes del programa. No se afectan entre sí.