

Condicionales

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Condicionales

Estas nos van a ayudar a decir qué acción vamos a ejecutar. La evaluación de condiciones, sólo puede arrojar 1 de 2 resultados: verdadero o falso (True o False).

Como reparar CUALQUIER COSA



Condicionales en la vida cotidiana

En la vida diaria, actuamos de acuerdo a la evaluación de condiciones, de manera mucho más frecuente de lo que en realidad creemos:

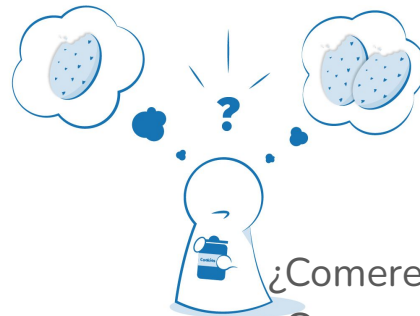
Si el semaforo esta en amarillo && rojo , no cruzar

Si el semaforo esta en verde || no hay semaforo, cruzar con cuidado.



Si el semáforo
está en verde,
cruzar la calle.

Si no, esperar a
que el
semáforo se
ponga en
verde.



¿Comere una galleta? o
¿Comere 2 galletas?

Operadores relacionales

DEV.F
DESARROLLAMOS(PERSONAS);

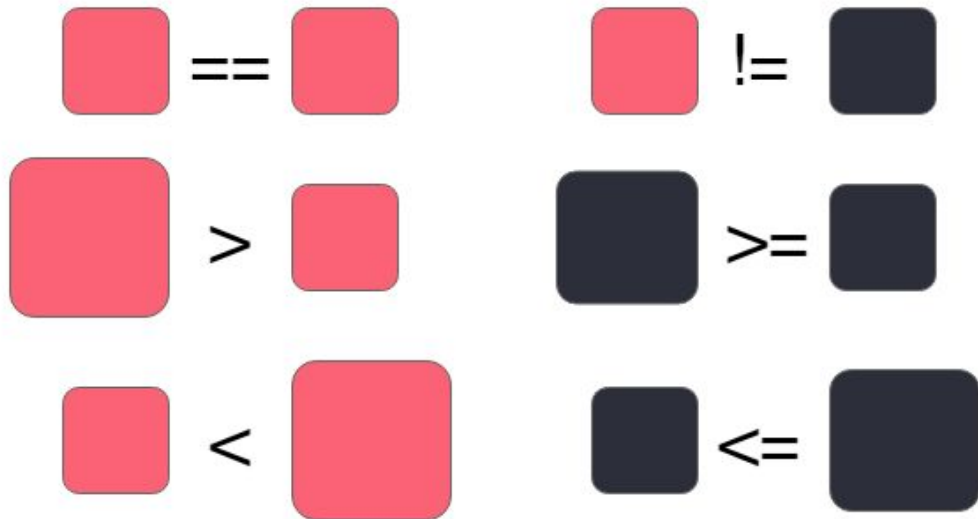
dev

Operadores relacionales

Los operadores relacionales son símbolos que se usan para comparar dos valores.

Si el resultado de la comparación es correcto la expresión considerada es verdadera, en caso contrario es falsa.

“Verde” != “Verde” => false



<

>

<=

>=

==

===

!=

Tablas de Operadores de Comparación Comunes

Simbolo	Significado	Ejemplo	Resultado
==	Igual que	true == true	true
!=	Distinto que	10 != 10	false
<	Menor que	10 < 4	false

Simbolo	Significado	Ejemplo	Resultado
>	Mayor que	3 > 1	true
<=	Menor o igual que	10 <= 9	false
>=	Mayor o igual que	4 >= 10	false

Tipos de Datos Undefined y Null

DEV.F
DESARROLLAMOS(PERSONAS);

dev

Undefined

Una **variable** a la que **no se le ha asignado valor**, o no se ha declarado en absoluto (no se declara, no existe) son de tipo **undefined**.

```
> var saludo;  
< undefined  
  
> saludo;  
< undefined
```

No defino un valor a **saludo**,
por lo tanto es undefined

```
> var saludo = "Hola";  
< undefined  
  
> saludo;  
< "Hola"
```

Aquí si defino un valor a **saludo**,
por lo tanto me devuelve dicho valor: "Hola".

Null

El valor **null** es un literal de Javascript que representa intencionalmente un valor nulo o "vacío". Solemos indicarlo cuando una operación no devuelve resultados relevantes.



typeof

El operador **typeof** devuelve una cadena (string) que nos indica el tipo del dato sin evaluarlo.

```
> var saludo = "Hola";
```

```
< undefined
```

```
> typeof saludo;
```

```
< "string"
```

```
> typeof 3;
```

```
< "number"
```

```
> typeof "3";
```

```
< "string"
```

```
> var verdadero = true;
```

```
< undefined
```

```
> typeof verdadero;
```

```
< "boolean"
```

Diferencias entre Null y Undefined

```
> undefined === null  
< false
```

null

```
> typeof null  
< "object"
```

```
> var a = null;  
< undefined  
  
> var b = null;  
< undefined  
  
> a+b;  
< 0
```

undefined

```
> typeof undefined  
< "undefined"
```

```
> var c;  
< undefined  
  
> var d;  
< undefined  
  
> c + d;  
< NaN
```

Non-zero value



null



0



undefined



Operadores lógicos

DEV.F
DESARROLLAMOS(PERSONAS);

dev

And (&&)

Con el operador **and** (&&), **solo si ambas condiciones se cumplen** (son verdaderas), el resultado **es verdadero**.

22<67 && -5<0		Resultado final
true	true	true
true	false	false
false	true	false
false	false	false

Or (||)

Con el operador **or(||)**, si una sola condición se cumple (es verdaderas), el resultado es **verdadero**.

Or		
true	true	true
true	false	true
false	true	true
false	false	false

Hoy es jueves ó está lloviendo.

Falsy

Los siguientes valores se evalúan como falso (también conocido como valores Falsy)

- `false`
- `undefined` > `if(undefined) console.log(true); else console.log(false); false`
- `null` > `if(null) console.log(true); else console.log(false); false`
- `0` > `if(0) console.log(true); else console.log(false); false`
- `NaN` `if(NaN) console.log(true); else console.log(false); false`
- the empty string (`""`) `if("") console.log(true); else console.log(false); false`

Truthy

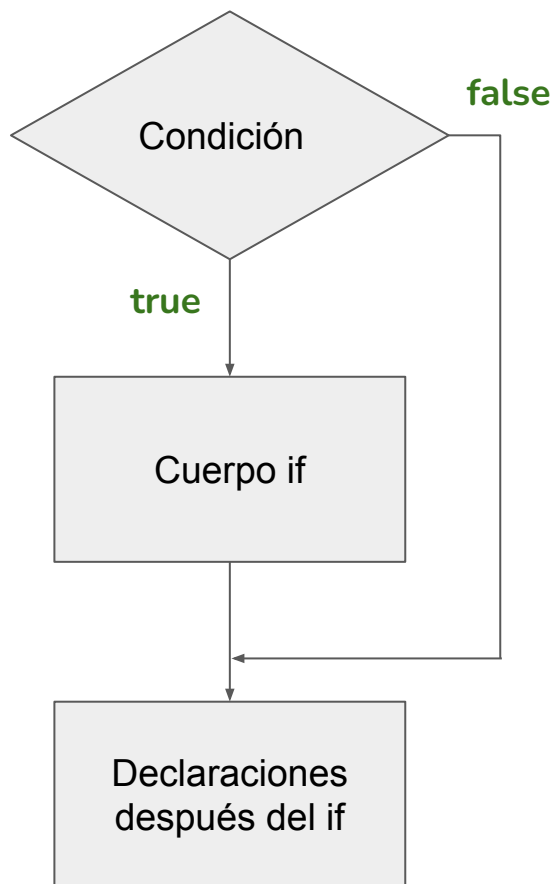
Los siguientes valores se evalúan como verdadero (también conocido como valores Truthy)

- `true`
- `{}` `if({}) console.log(true); else console.log(false); true`
- `[]` `if([]) console.log(true); else console.log(false); true`
- `42` `if(42) console.log(true); else console.log(false); true`
- `"0"` `if("0") console.log(true); else console.log(false); true`
- `"false"` `if("false") console.log(true); else console.log(false); true`

Tomando decisiones: Declaraciones If / else

DEV.F
DESARROLLAMOS(PERSONAS);

dev



Declaraciones If / else

if ejecuta una sentencia si una condición especificada es evaluada como verdadera. Si la condición es evaluada como falsa, otra sentencia puede ser ejecutada (**if else** o **else**).

Sintaxis if básica

Puedo ejecutar un código en el caso de que se cumpla una condición, solamente utilizando if:

```
if (condición) sentencia1
```

If cuando ejecuta una sola línea de código (sentencia)

```
if (condición) {  
    sentencia1;  
    sentencia2;  
    sentencia3;  
}
```

Si el if ejecuta múltiples sentencias, debe llevar {}.

Sintaxis if-else básica

Con la estructura if else, puedo ejecutar un código si se cumple la condición o en caso contrario ejecutar otro en su lugar:

```
if (condición) {  
    //código a ejecutar si la condición es verdadera  
} else {  
    //ejecuta este otro código si la condición es falsa  
}
```

Sintaxis if-else básica: Ejemplo

Con la estructura if else, puedo ejecutar un código si se cumple la condición o en caso contrario ejecutar otro en su lugar:

```
> var edad = 20;
```

```
< undefined
```

```
> if (edad >= 18) console.log("Eres mayor de edad")  
   else console.log("Eres menor de edad");
```

```
Eres mayor de edad
```

Sintaxis if - else if

Con la estructura **if - else if**, puedo ejecutar un código si se cumple con la condición especificada en cada caso:

```
if (condición1) {  
    //código a ejecutar si la condición1 es verdadera  
} else if (condición2) {  
    //ejecuta este otro código si la condición2 es verdadera  
} else if (condición3) {  
    //ejecuta este otro código si la condición3 es verdadera  
} else if (condición4) {  
    //ejecuta este otro código si la condición4 es verdadera  
}
```

Sintaxis if - else if: Ejemplo

Con la estructura **if - else if**, puedo ejecutar un código si se cumple con la condición especificada en cada caso:

```
> var semaforo = "amarillo";  
  if (semaforo == "verde") {  
    console.log("Avanza")  
  } else if (semaforo == "amarillo") {  
    console.log("Comienza a frenar")  
  } else if (semaforo == "rojo") {  
    console.log("Detente")  
  }  
}
```

Comienza a frenar

Sintaxis if - else if - else

Con la estructura **if - else if**, puedo ejecutar un código si se cumple con la condición especificada en cada caso y al final con **else** ejecutar un código si no se cumplio ninguna condición anterior.

```
if (condición1) {  
    //código a ejecutar si la condición1 es verdadera  
} else if (condición2) {  
    //ejecuta este otro código si la condición2 es verdadera  
} else if (condición3) {  
    //ejecuta este otro código si la condición3 es verdadera  
} else {  
    //ejecuta este otro código si ninguna condición se cumple  
}
```


Sintaxis if - else if - else: Ejemplo

Con la estructura **if - else if**, puedo ejecutar un código si se cumple con la condición especificada en cada caso y al final con **else** ejecutar un código si no se cumplió ninguna condición anterior.

```
> var semaforo = "morado";  
  if (semaforo == "verde") {  
    console.log("Avanza")  
  } else if (semaforo == "amarillo") {  
    console.log("Comienza a frenar")  
  } else if (semaforo == "rojo") {  
    console.log("Detente")  
  } else {  
    console.log("Por favor introduce un color de semáforo válido");  
  }
```

Por favor introduce un color de semáforo válido

Demostración: ¿Puedo vacunarme?



**GOBIERNO DE
MÉXICO**

Pertenezco a uno de los siguientes grupos y me quiero vacunar :

- Mayores de 18 años que residen en un municipio fronterizo del Norte del país.
- Embarazadas mayores de 18 años con más de 9 semanas de gestación.
- Personas que cumplen 30 años o más en este año.