

Introducción a la programación en JavaScript

Módulo 2 - Formato

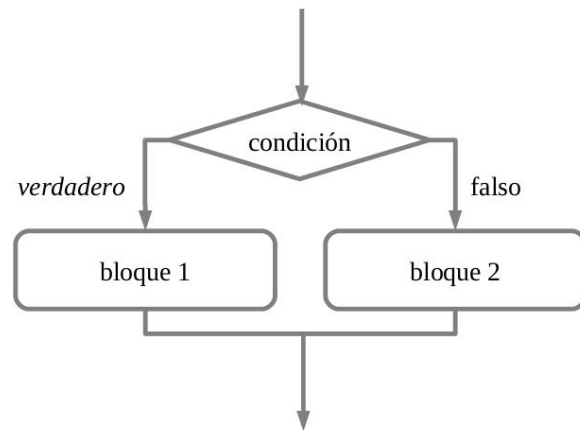
Estructuras de control de flujo

Estructuras de control de flujo

Si solamente trabajamos con operadores y variables, generamos una sucesión de pasos lineales. No podemos decidir si algo se mostrará o no según determinadas circunstancias o en todo caso repetir **una instrucción tantas veces**.

Para realizar estas secuencias más complejas y con múltiples posibilidades se debe trabajar con estructuras de **control de flujo**.

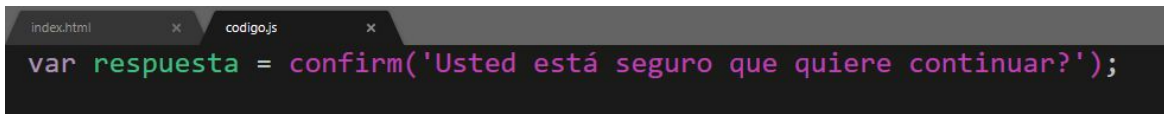
Para poder avanzar, necesitamos recordar algunos tipos de variables y operadores que vamos a utilizar en estas **nuevas estructuras**.



Tipo booleano

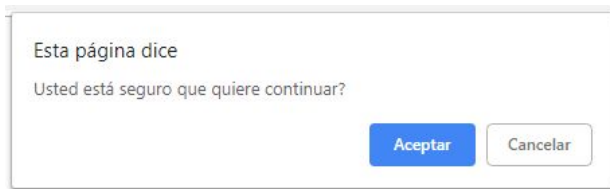
Los tipos de variables de dato booleanos son aquellos conformados únicamente por 2 posibles valores: **true/false**.

Un booleano puede ser definido en forma explícita, por ejemplo, asignando un "true" a una variable. Recordemos que, por ejemplo, si generamos una ventana de confirmación estamos esperando que el resultado según el usuario acepte o no , es **verdadero/falso**.

A screenshot of a code editor with two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active and shows the following JavaScript code:

```
var respuesta = confirm('Usted está seguro que quiere continuar?');
```

El resultado será el siguiente:

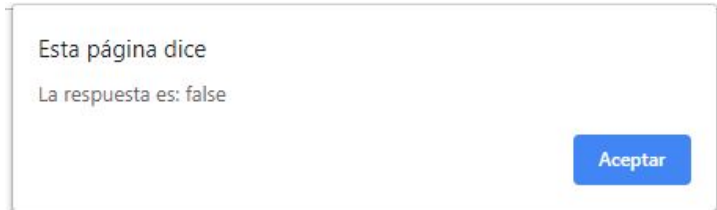


Tipo booleano

Vamos a mostrar el resultado en una alerta:

```
index.html x  código.js x  
var respuesta = confirm('Usted está seguro que quiere continuar?');  
alert('La respuesta es: ' + respuesta);
```

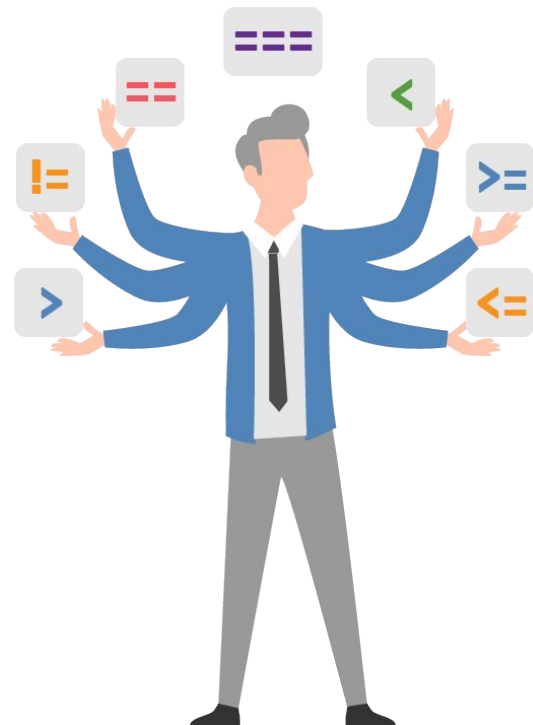
Si el usuario cancela, el resultado será el siguiente:



Operadores relacionales

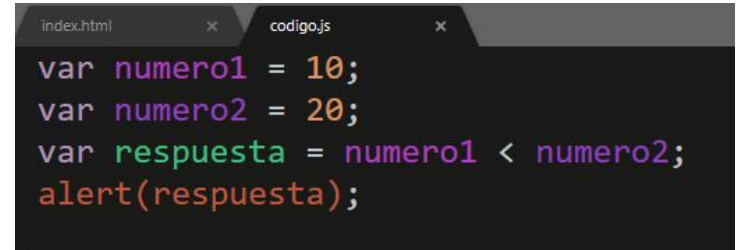
Vamos a recordar qué tenemos la posibilidad de trabajar con operadores relacionales, por ejemplo:

- > Mayor a
- < Menor a
- >= Mayor o igual a
- <= Menor o igual a
- == "Igual a" => Utilizado para verificar si 2 valores coinciden
- != "Distinto de" => Utilizado para verificar si 2 valores NO coinciden
- === "Idéntico a" => Utilizado para verificar si 2 valores coinciden y son del mismo tipo de dato



Operadores relacionales

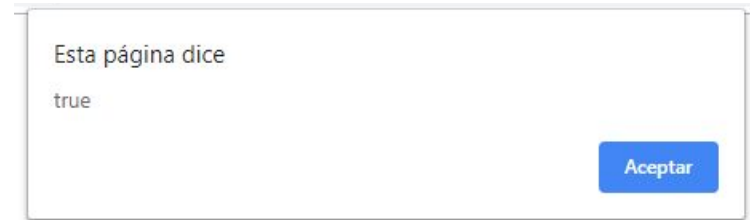
Si utilizamos **operadores relacionales**, podríamos generar el siguiente código, donde nos preguntamos si el **numero1** es menor al **numero2**. Es decir, si 10 es menor a 20...

A screenshot of a code editor with two tabs: 'index.html' and 'codigo.js'. The 'codigo.js' tab is active and contains the following JavaScript code:

```
var numero1 = 10;  
var numero2 = 20;  
var respuesta = numero1 < numero2;  
alert(respuesta);
```

The code is color-coded: 'var' is purple, 'numero1' and 'numero2' are green, '10' and '20' are orange, '<' is blue, and 'respuesta' is green.

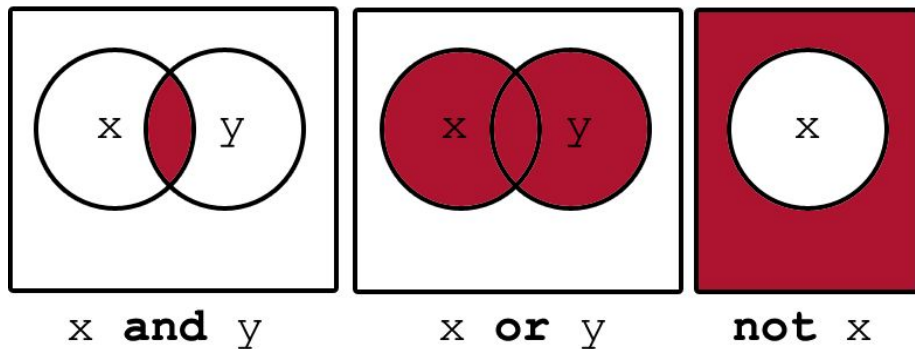
Como el resultado da **true(booleano)**, la respuesta:



Operadores lógicos

También podemos utilizar **operadores lógicos** para generar procesos más complejos donde decimos por ejemplo que un **número es igual o mayor** a otro pero también qué debe cumplir o no con otra condición.

Siempre es interesante **graficar estas situaciones** para entender el proceso de descarte, donde algo termina o no dando **true o false**:



Estructuras condicionales

Condicionales

Son aquellas que nos permiten tomar decisiones acerca de qué **sentencias (órdenes)** debe ejecutar el programa, en base a una condición / expresión booleana, es decir si algo es **true/false**.

Se puede decir que a diferencia de los **programas que veníamos generando**, en este caso según el dato de entrada es que se tomará una serie de pasos u otra, por lo tanto el resultado o dato de salida dependerá del dato de entrada.



Condicionales: if

El **condicional if** nos permite tomar una decisión en base a una condición.

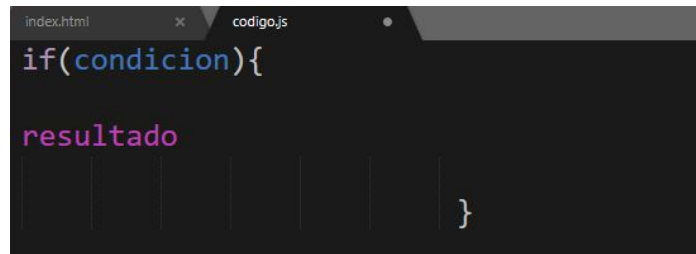
Cuando la condición sea evaluada como verdadera, se ejecutará un bloque con un conjunto de sentencias asociado al **if**.

Siempre, para empezar a trabajar, lo haremos en **pseudocódigo**, esto es esencial para no cometer errores y saber siempre hacia donde apunta **nuestro programa**.

```
Inicio  
  
Entero: Edad  
  
Tomar a través de ventana de usuario la Edad  
  
Si (Edad >= 20) entonces  
  
    Mostrar en alerta "Tu tienes 20 o más"  
  
Fin Si  
  
Fin Algoritmo
```

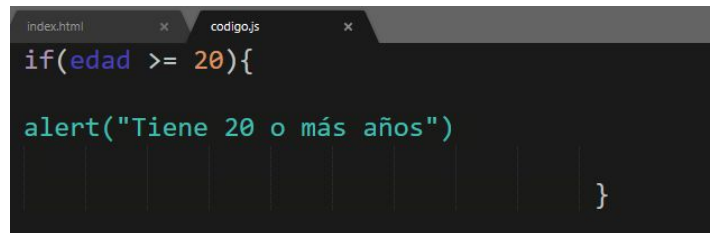
Condicionales: if

A partir del **pseudocódigo** anterior, la sintaxis será similar a la siguiente:



```
index.html x  codigo.js
if(condicion){
    resultado
}
```

Generemos **algo real para ir probandolo en nuestro código**:



```
index.html x  codigo.js x
if(edad >= 20){
    alert("Tiene 20 o más años")
}
```

Condicionales: if

Dado que hemos decidido que el dato de entrada sea a través de una **ventana de usuario**, lo haremos de la siguiente forma:

```
index.html x código.js
var edad = prompt("Cuando años tiene el usuario?")

if(edad >= 20){

  alert("Tiene 20 o más años")

}
```

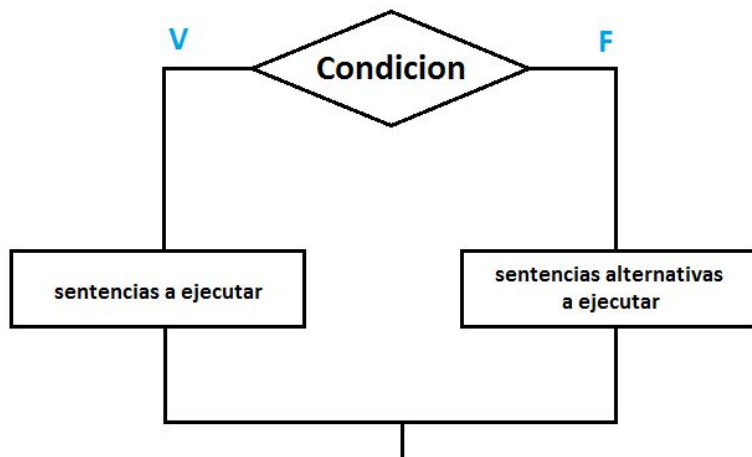
Como el usuario, **efectivamente tiene 20 o más**, el resultado en el navegador será el siguiente:

Esta página dice
Tiene 20 o más años

Aceptar

Condicionales: if, else

En este caso fijamos una alternativa, de esta **forma un algoritmo y un pseudocódigo** que representen esta situación sería:



Inicio

Entero: Edad

Tomar a través de ventana de usuario la Edad

Si(Edad >=20) entonces

Mostrar en alerta "Tu tienes 20 o más"

Sino

Mostrar en alert "Usted tiene menos de 20 años"

Fin Si

Fin Algoritmo

Condicionales: if, else

Vamos a generar el código del caso anterior en nuestro **codigo.js**

```
index.html x  codigo.js x
var edad = prompt("Cuando años tiene el usuario?")
if(edad >= 20)
    {alert("Tiene 20 o más años")}
else
    { alert("Usted tiene menos de 20 años")}
```

El resultado de no cumplirse la condición es:

Esta página dice

Usted tiene menos de 20 años

Aceptar

Condicionales: if, else, elseif

La estructura **elseif**, nos brinda una posibilidad de que se ejecute **una serie de pasos si una opción (no la primera) es verdadera**.

Por ejemplo, nuestro **pseudocódigo** podría ser:

```
Inicio
Entero: Edad
Tomar a través de ventana de usuario la Edad
Si(Edad >20) entonces
    Mostrar en alerta "Usted tiene más de 20 años"
Sino
    Si(Edad=20 o Edad > 15)
        Mostrar en una alerta "Usted tiene entre 20 y 15 años"
Sino
    Mostrar en una alerta "Usted tiene menos de 15 años, adios!"
FinSi
Fin Algoritmo
```

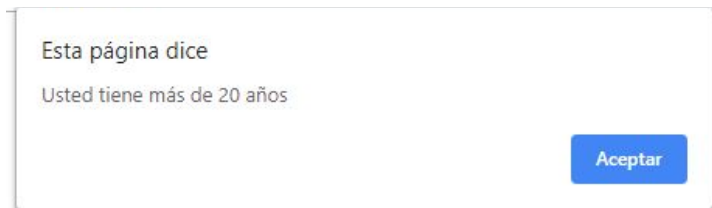

Condicionales: if, else, elseif

Nuestro código podría verse de la siguiente forma:

```
codigo.js x index.html x
var edad = prompt("La edad del usuario es?")

if (edad > 20) {
  alert("Usted tiene más de 20 años")
} else if (edad == 20 || edad > 15) {
  alert("Usted tiene entre 20 y 15 años")
} else {
  alert("Usted tiene 15 años o menos, adios!")
}
```

El resultado si el **usuario ingresa 25** será el siguiente:



Condicionales: if, else, elseif

En el caso de que el **usuario ingrese 15 por ejemplo**, el resultado en el navegador a través de una **ventana de alerta** será el siguiente:



Probemos también hacerlo con un **document.write()** para alternar entre las diferentes formas vistas de output en Javascript.

Revisión

- Repase el concepto de **estructura** de **control de flujo**.
- Genere diferentes opciones de **pseudocódigo** para trabajar con **condicionales**.
- Implemente **if, else, elseif**.
- Combine los **diferentes tipos de output vistos**.
- Implemente los conocimientos en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente.



¡Muchas gracias!

¡Sigamos trabajando!