Introducción a Javascript

Clase 2

Condicionales

Condicionales

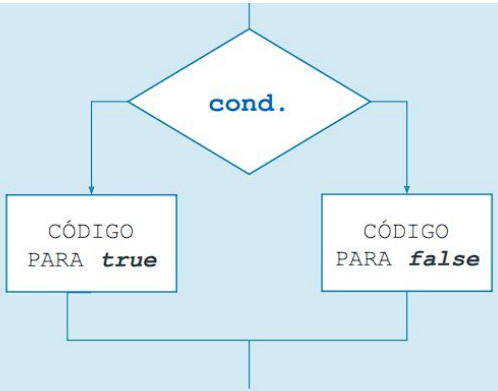
¿Qué son?

- Hasta ahora nuestros algoritmos venían siendo lineales
- A partir de ahora, vamos a tener la posibilidad de comenzar a bifurcar el camino una vez, varias veces, o incluso poder hacer una selección
- En principio los condicionales van a trabajar con valores lógicos, que los obtendremos a través de los operadores de comparación

Condicionales: estructura if [else]

Sintaxis y diagrama de flujo

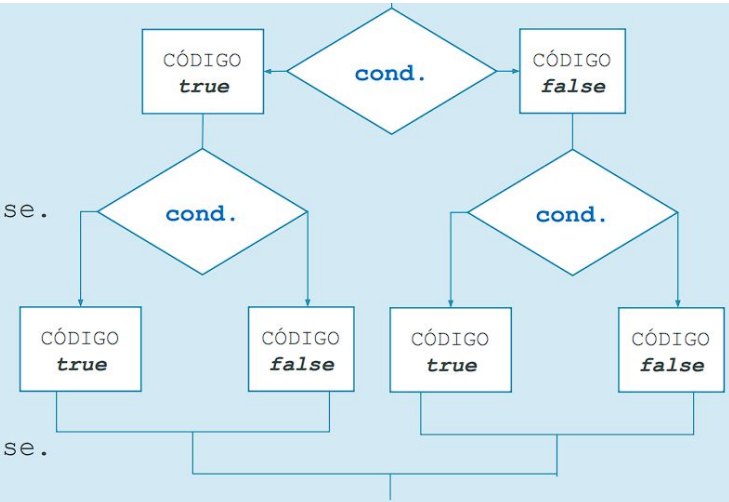
```
if (condición) {
    /*
    El código a ejecutar si la condición se cumple, es
    decir, da true
    */
} else {
    /*
    Optativamente, podemos tener el else, que
    contendrá el código a ejecutar si la condición NO
    se cumple, es decir, da false
    */
}
```



Condicionales: estructura if [else] anidados

Sintaxis y diagrama de flujo

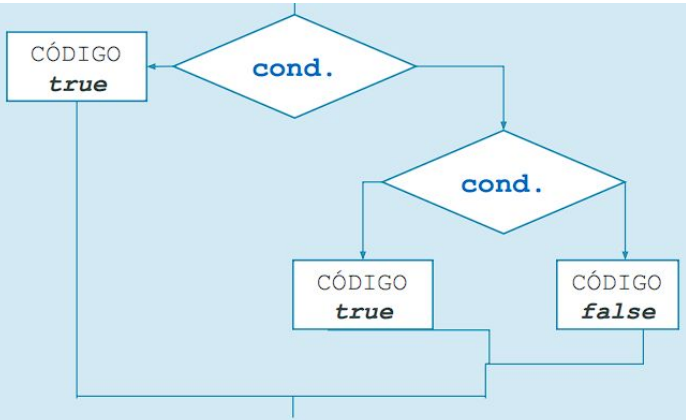
```
if (condición) {
    // Código si es true.
    if (condición anidada) {
        // Código si es true.
    } else {
        // Optativamente, código si es false.
    }
} else {
    // Optativamente, código si es false.
    if (condición anidada) {
        // Código si es true.
    } else {
        // Optativamente, código si es false.
    }
}
```



Condicionales: estructura if [else if] [else]

Sintaxis y diagrama de flujo

```
if (condición 1) {
    // Código si es true.
} else if (condición 2) {
    // Código si es true.
} else if (condición 3) {
    // Código si es true.
} else if (condición n) {
    // Código si es true.
} else {
    // Optativamente, código si es false.
}
```



Condicionales: Operador Ternario

Sintaxis

- (condición) ? /* código true */ : /* código false */;
- Es un operador que evalúa una condición y tendrá preparado las posibles operaciones o valores para los casos true y false de la condición.
- Al igual que un if [else], puede tener anidamiento

Condicionales: estructura switch case

Sintaxis

```
switch (variable) {
    case valor 1:
        // Código si la variable tiene el valor 1.
        break;
    case valor 2:
        // Código si la variable tiene el valor 2.
        break;
    default:
        // Código si la variable no tiene ninguno de los valores comparados.
}
```

Condicionales: estructura switch case

- Es un selector de casos, compara si el valor de la variable es estrictamente igual a alguno de los valores de los case
- Funciona como un if [else if] [else], limitado solamente a comparaciones ===
- Cada case se cierra con un break, que indica que “rompe” el switch si entro al case
- Al igual que los condicionales, tiene un caso por defecto, que será el default

Bucles / Ciclos de repetición

Ciclos de repetición

¿Para qué sirven?

- Cuando queramos que un proceso se repita, utilizaremos estas estructuras de control
- Los principales son : while, do-while, for
- Los ciclos avanzados son: for-in, for-of, forEach

Ciclos: estructura while

¿Cómo funciona?

- Es un condicional
- La primera vez que pregunta:
  - si la condición se cumple, entra al ciclo.
  - si la condición no se cumple, no entra al ciclo.

- Una vez cumplido el ciclo, vuelve a preguntar:
  - Mientras la condición sea true, vuelve a entrar.
  - Si la condición deja de cumplirse (false), sale del ciclo.

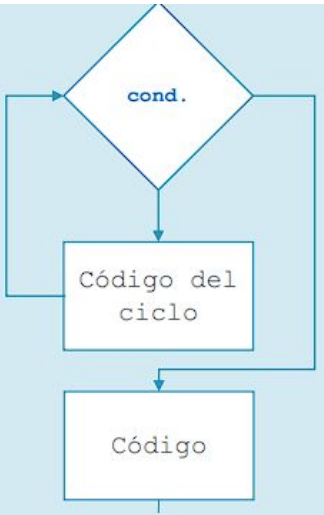
- Atención: si dentro del ciclo no hay algo que modifique la condición, tendremos un siniestro "ciclo infinito", ya que nada lo detendrá (bueno, si, cerrar la pestaña del navegador).

Ciclos: estructura while

Sintaxis y diagrama de flujo

```
while (condición) {
    /*
    El código a ejecutar mientras la condición se
    cumpla, es decir, dé true
    */
}
```

Primero pregunta, por lo que si no se cumple la condición inicial, no entrará nunca al ciclo.



Ciclos: estructura do-while

¿Cómo funciona?

- También es un condicional, pero a diferencia del while, va a entrar una vez al ciclo sí o sí
- Luego de terminar la primer vuelta, pregunta:
  - si la condición se cumple, vuelve a entrar al ciclo
  - si la condición no se cumple, no vuelve a entrar al ciclo

- Si vuelve a entrar, al llegar al final, funciona como el while
  - Mientras la condición sea true, vuelve a entrar.
  - Si la condición deja de cumplirse (false), sale del ciclo.

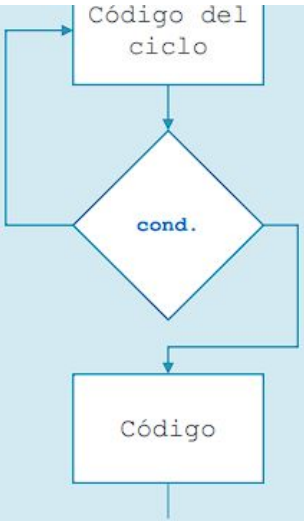
- Atención: si dentro del ciclo no hay algo que modifique la condición, tendremos un siniestro "ciclo infinito", ya que nada lo detendrá (bueno, si, cerrar la pestaña del navegador)

Ciclos: estructura do-while

Sintaxis y diagrama de flujo

```
do {
    /*
    El código a ejecutar mientras la condición se
    cumpla, es decir, dé true
    */
} while (condición)
```

A diferencia delwhile, este ciclo nos asegura al menos la ejecución de una vuelta.



Ciclos: estructura for

¿Cómo funciona?

- Cuando sabemos la cantidad de veces que queremos que se repita un ciclo
- La estructura se divide en tres partes
  - Valor inicial, será desde "donde empiezo a contar"
  - Condición para entrar al ciclo
  - Incremento o decremento

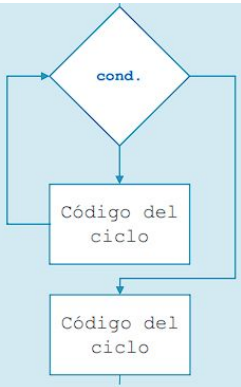
- Utilizaremos una variable que se encargará de contar
  - Generalmente la llamaremos i (de iteración)
  - Puede declararse en la misma estructura o antes de la misma

- Para la condición utilizaremos los operadores > >= <= < (no siempre será así, pero por ahora seamos felices)

Ciclos: estructura for

Sintaxis y diagrama de flujo

```
for (valor inicial; condición; inc/dec) {
    /*
    El código a ejecutar mientras la condición se
    cumpla, es decir, dé true
    */
}
```



Si el valor inicial cumple la condición, entra por primera vez al ciclo (sino, no entrará).
Luego de la primer vuelta en adelante, realizará el incremento o decremento.
Luego del incremento o decremento, vuelve a verificar la condición. Mientras la condición se cumpla, sigue entrando.

Bucles / Usos cotidianos

Ciclos: Uso del do-while

¿Que usos útiles podemos darle al do-while?

- Ingresar valores cuando no esté determinada la cantidad de veces de repetición
- Validaciones
- Cuando utilizamos el do-while para repetir un proceso, conviene no depender de uno de los valores ingresados, sino consultarle al usuario si desea continuar o salir del ciclo

Fin de la clase

Este es el espacio para dudas y/o preguntas existenciales