«Talento Tech»

Front End Js

CLASE 10

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase 10: Estructuras de control y bucles en JavaScript

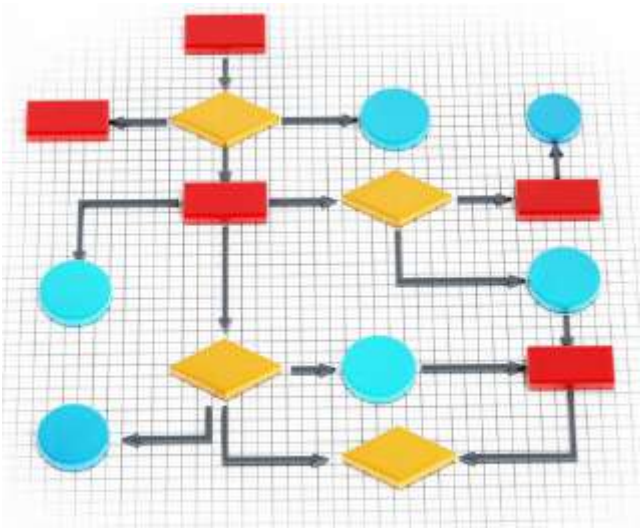
Temario:

1. **Diagrama de flujo**
   - ¿Qué es un diagrama de flujo?
   - Elementos clave de un diagrama de flujo
   - Ejemplo práctico: Preparación de café
   - Utilidad del diagrama de flujo en programación
2. **Condicionales en JavaScript**
   - ¿Qué es un condicional?
   - Tipos de condicionales:
     - `if`
     - `if...else`
     - `else if`
     - Operador ternario
   - Importancia de los condicionales en el desarrollo de aplicaciones
   - Ejemplo práctico: Evaluación de la edad para entrar a un club
3. **Operadores Lógicos y de Comparación**
   - ¿Qué son los operadores lógicos y de comparación?
   - Comparadores:
     - `==` vs `===`
     - `!=` vs `!==`
     - `<`, `>`, `<=`, `>=`
   - Operadores lógicos:
     - `&&` (AND)

        - `||` (OR)
        - `!` (NOT)
    - Combinación de operadores en condiciones complejas
    - Ejemplo práctico: Validación de acceso a un evento según edad y membresía
4. **Bucles en JavaScript**
    - ¿Qué es un bucle?
    - Tipos de bucles:
        - `while`
        - `do...while`
        - `for`
    - Diferencias entre los tipos de bucles
    - Ejemplo práctico: Iterar una lista de productos
5. **Cómo combinar operadores lógicos y bucles**
    - Combinación de operadores con ciclos
    - Casos de uso en la vida real
    - Ejemplo práctico: Filtrado de productos con descuentos en un catálogo

Buenos Aires aprende
Agencia de Habilidades para el Futuro

1. Diagrama de flujo



¿Qué es un diagrama de flujo?

Se trata de un mapa que muestra paso a paso cómo realizar una tarea o resolver un problema: es la representación gráfica del flujo de tu algoritmo. Es una herramienta visual súper útil, sobre todo en programación, porque te permite ver cómo se conectan las decisiones y las acciones de tu programa. Imaginá que querés preparar un café, ¿qué pasos seguirías? Un diagrama de flujo te mostraría, por ejemplo, que primero necesitás hervir el agua, luego poner el café en el filtro, y así sucesivamente.

Elementos claves de un diagrama de flujo

- **Inicio/Fin:** Representan el comienzo o la conclusión del proceso, se muestran en forma de óvalo.
- **Acción/Proceso:** Son las operaciones o tareas que se deben realizar, y se representan por un rectángulo.
- **Decisión:** Muestra una bifurcación en el proceso (una decisión de “sí” o “no”), se representa por un rombo.

- **Flechas:** Indican la dirección en la que fluye el proceso.

Ejemplo práctico: preparar un café

1. Inicio
2. Hervir agua
3. Poner el café en el filtro
4. Verter agua sobre el café
5. Fin

Con este diagrama, antes de escribir una sola línea de código, ya tenés la idea clara de qué debe suceder y en qué orden. Usar diagramas de flujo para planificar tu código ayuda a evitar errores y mejora la organización.

2. Condicionales en JavaScript

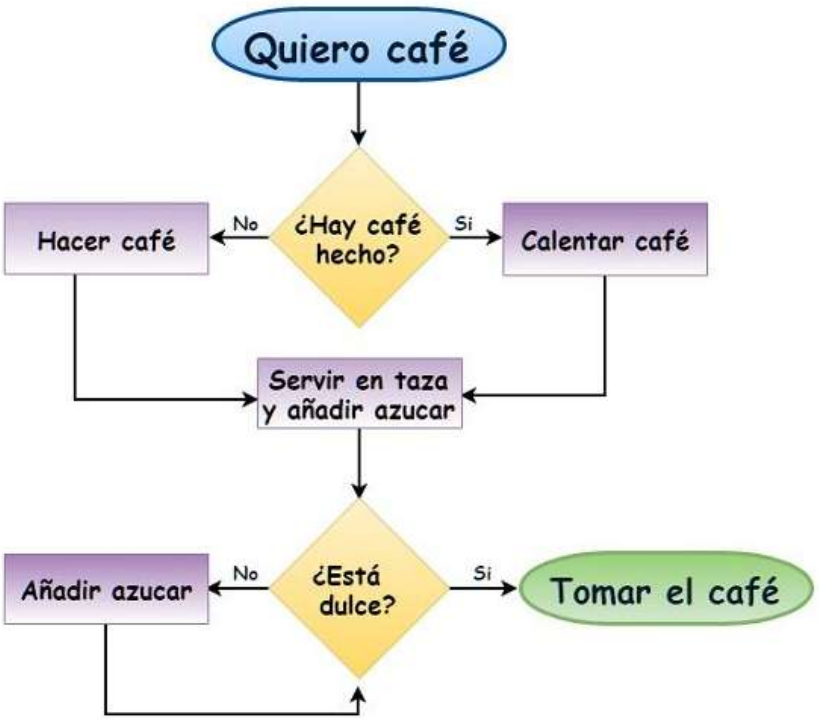
¿Qué es una estructura condicional?

Una estructura condicional permite, en programación, la toma de decisiones. Tendrá esencialmente dos partes: **la condición** (*si…* ocurre X acción) y la acción a llevar a cabo como **consecuencia** (*entonces…* realizar Y acción). Los condicionales sólo pueden dar dos estados de respuesta, ya que la condición sólo puede ser o verdadera o falsa.

Pensémoslo en el mundo real: si salís de casa y ves que está lloviendo, vas a llevar un paraguas. En programación funciona de la misma forma: el programa evalúa una condición y decide qué acción tomar en función del resultado.

```
if (llueve) {
   llevarParaguas();
 } else {
   salirSinParaguas();
 }
```

Tomar un café con condicionales (rombos) sería algo como esto:



Buenos Aires aprende
Agencia de Habilidades para el Futuro

Tipos de Condicionales:

if: Este es el más básico y simplemente ejecuta un bloque de código si la condición es verdadera.

```
if (edad >= 18) {
   console.log("Sos mayor de edad.");
 }
```

if...else: Esto permite ejecutar un bloque de código si la condición es verdadera, y otro en el caso de que sea falsa.

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

else if: Si necesitás manejar múltiples condiciones, `else if` te da esa flexibilidad.

```
if (edad >= 18) {
   console.log("Sos mayor de edad.");
 } else if (edad >= 13) {
   console.log("Sos un adolescente.");
 } else {
```

```
    console.log("Sos un niño/a.");
}
```

Operador ternario: Es una versión compacta de `if...else`. Ideal para condiciones simples.

```
let resultado = edad >= 18 ? "mayor de edad" : "menor de edad";
```

Ejemplo práctico: evaluación de la edad

```
let edad = 20;

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

Con esto, podés determinar si alguien puede entrar a un boliche o no, según su edad.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

3. Operadores lógicos y de comparación

¿Qué son los operadores lógicos y de comparación?

Son herramientas que usás para comparar valores y tomar decisiones basadas en esas comparaciones. Por ejemplo, podés comparar si dos números son iguales, o si un valor es mayor que otro.

Comparadores

- `==` Verifica si dos valores son iguales, pero no necesariamente del mismo tipo de dato.
- `===` Compara tanto el valor como el tipo de dato.
- `!=` Verifica si dos valores son diferentes.
- `!==` Verifica si dos valores son diferentes, incluyendo su tipo de datos.
- `<`, `>`, `<=`, `>=` Comparan valores numéricos.

Operadores lógicos

- **&& (AND):** Se cumplen todas las condiciones.
- **|| (OR):** Se cumple al menos una de las condiciones.
- **! (NOT):** Invierte el valor de la condición.

Ejemplo práctico: acceso a un evento

```
let edad = 22;

let miembroVIP = true;


if (edad >= 18 && miembroVIP) {
 console.log("Acceso concedido al área VIP.");
} else {
 console.log("Acceso denegado.");
}
```

En este ejemplo, el acceso depende tanto de la edad como de ser miembro VIP.

4. Bucles en JavaScript

¿Qué es un bucle?

Un bucle es una estructura que repite acciones en tu programa hasta que una condición deje de cumplirse. Imaginate que tenés que iterar sobre una lista de productos: con un bucle, podés hacerlo fácilmente.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Tipos de bucles

while: El ciclo se ejecuta mientras la condición sea verdadera.

```
let i = 0;

while (i < 5) {

 console.log(i);

 i++;

}
```

do...while: Similar a `while`, pero siempre ejecuta el código al menos una vez.

```
let i = 0;

do {

 console.log(i);

 i++;

} while (i < 5);
```

for: Es más compacto y se usa cuando conocés de antemano cuántas veces se repetirá el bucle.

```
for (let i = 0; i < 5; i++) {

   console.log(i);

 }
```

Ejemplo práctico: iterar productos

```
let productos = ['Laptop', 'Celular', 'Tablet'];

for (let i = 0; i < productos.length; i++) {

 console.log(productos[i]);

}
```

Este ejemplo te muestra cómo recorrer una lista de productos y mostrarlos en la consola.

5. Cómo combinar operadores lógicos y bucles

En muchos casos, vas a necesitar usar operadores lógicos dentro de bucles. Por ejemplo, podés querer filtrar una lista de productos para mostrar solo los que están en descuento.

Ejemplo práctico: filtrar productos en descuento

```
let productos = [

   { nombre: 'Laptop', descuento: true },

   { nombre: 'Celular', descuento: false },

   { nombre: 'Tablet', descuento: true }

 ];

  for (let i = 0; i < productos.length; i++) {

   if (productos[i].descuento) {

     console.log(productos[i].nombre + " tiene descuento.");}}
```

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Con esta clase, ya dominás el uso de condicionales y bucles en JavaScript, herramientas fundamentales para cualquier aplicación dinámica. A medida que vayas practicando, estos conceptos se volverán cada vez más naturales y fáciles de implementar. ¡Adelante y a seguir experimentando!

Ejercicio práctico #1:

Evaluación de condicionales y operadores lógicos

Enunciado: Crear un programa que reciba la edad de una persona y si está en la lista VIP. El programa deberá verificar lo siguiente:

1. Si la persona tiene 18 años o más, permitirle el acceso al evento.
2. Si además de tener 18 años o más, es miembro VIP, darle acceso al área exclusiva.
3. Si la persona tiene menos de 18 años, denegar el acceso.

Tips:

- **Validación de entradas**: Asegurate de que se ingrese una edad válida. Podés usar `isNaN()` para verificar que el valor ingresado sea un número.
- **Uso de operadores lógicos**: Combiná el operador `&&` para verificar que se cumplan ambas condiciones (edad >= 18 y ser miembro VIP).
- **Operador ternario**: Considerá usar un operador ternario si la lógica es sencilla.
- **Consola del navegador**: Mostrá los resultados usando `console.log()` para verificar si se cumplen las condiciones correctamente.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Ejercicio práctico #2:

Iterar sobre una lista de productos con bucles y condicionales

Enunciado: crear un programa que reciba una lista de productos, cada uno con un precio y un indicador de descuento (true o false). El programa debe iterar sobre la lista y:

1. Mostrar todos los productos con descuento.
2. Mostrar el total de productos sin descuento.
3. Al final, mostrar cuántos productos tienen descuento y cuántos no.

Tips:

- **Uso de bucles**: Utilizá un `for` para recorrer la lista de productos.
- **Condicionales**: Dentro del bucle, utilizá `if` para verificar si el producto tiene descuento (`producto.descuento === true`).
- **Validación**: Podés agregar validaciones para asegurarte de que los datos de los productos sean correctos (nombres no vacíos, precios válidos).
- **Consola del navegador**: Mostrá los resultados de los productos con descuento y el total de productos en la consola usando `console.log()`.

Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad