* **Fundamentales:** JavaScript, CSS, HTML5 y DOM
* **Web UI framework:** Angular y NPM
* **Integración:** Integración Angular Application con REST API

**Introducción a JavaScript**

**JavaScript**  es un lenguaje de programación dinámico, versátil y de alto nivel. Ampliamente utilizado en el desarrollo web, ha evolucionado para convertirse en un lenguaje de propósito general utilizado tanto del lado del cliente como del servidor.

**Características principales**

* **Single Thread:** Opera en un solo hilo de ejecución, lo que significa que maneja una secuencia de operaciones. Esto ayuda a evitar problemas de concurrencia, aunque se debe de tener cuidado al gestionar tareas que podrían bloquear el hilo principal.
* **Lenguaje Interpretado:** Es interpretado en lugar de compilado, lo que significa que el código se ejecuta directamente por el navegador o el entorno de ejecución como Node.js. Esto permite un desarrollo más ágil, ya que se pueden ver los resultados inmediatamente sin necesidad de compilar el código.
* **Basado en eventos:** Es altamente orientado a eventos, lo que significa que puede escuchar y responder a eventos específicos, como clics del ratón, pulsaciones de teclas o la carga de una página. Esto es crucial para la creación de interfaces de usuario interactivas.
* **Multiparadigma:** Admite múltiples paradigmas de programación, lo que permite a los desarrolladores elegir el enfoque que mejor se adapte a sus necesidades. Se puede programar de manera orientada a objetos, funcional y procedimental, lo que ofrece flexibilidad en la forma de estructurar el código.
* **Lenguaje Asíncrono:** Gracias a su modelo de ejecución basado en eventos y el uso de promesas, async/await, y callbacks, JavaScript maneja operaciones asincrónicas de manera efectiva. Esto permite que los desarrolladores realicen tareas como la carga de datos desde un servidor sin bloquear la interfaz de usuario.
* **Entornos:** Aunque JavaScript fue diseñado para ejecutarse en navegadores web, con la llegada de Node.js, también puede ejecutarse en servidores. Esto lo convierte en un lenguaje de programación tanto del lado del cliente como del lado del servidor. Además, JavaScript puede utilizarse en entornos como React Native para desarrollar aplicaciones móviles, en Electron para aplicaciones de escritorio, y en herramientas como MongoDB para bases de datos.

**Operadores en JavaScript**

**Tabla

Descripción generada automáticamente**

**El spread operator genera una copia, no trabaja con referencia.**

**Estructuras de Control en JavaScript**

**Condicionales:** Los condicionales permiten ejecutar bloques de código dependiendo de si se cumple o no una condición específica.

* **If/else:** Se utiliza para tomar decisiones basadas en condiciones booleanas. Si la condición es verdadera, se ejecuta el primer bloque; si es falsa, se ejecuta el bloque del else.

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

* **else if:** Permite comprobar múltiples condiciones de manera secuencial.

**Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente**

* **switch:** Se utiliza para ejecutar diferentes bloques de código basados en el valor de una expresión. Es útil cuando hay muchas condiciones basadas en un solo valor.

**Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

**Repetición:** Las estructuras de repetición permiten ejecutar un bloque de código varias veces, basándose en condiciones.

* **for:** Se utiliza cuando se conoce el número exacto de iteraciones.

**Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

* **for:** Se utiliza cuando no se conoce el número de iteraciones y se desea repetir el bloque mientras se cumpla una condición.

**Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

* **do...while:** Similar a while, pero garantiza que el bloque se ejecute al menos una vez.

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente** **Imagen que contiene Texto

Descripción generada automáticamente**

* **forEach:** Se utiliza para iterar sobre elementos de un array sin necesidad de usar índices.

**Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

* **for...of:** Se utiliza para iterar sobre elementos de objetos iterables, como arrays y strings, sin incluir índices.

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

* **for...in:** Se utiliza para iterar sobre las propiedades enumerables de un objeto.

**Texto

Descripción generada automáticamente** **Texto

Descripción generada automáticamente**

* **break:** Se utiliza para salir de un bucle o un switch prematuramente.
* **continue:** Se utiliza para saltar a la siguiente iteración del bucle.

**UseStrict en JavaScript**

Es una declaración en JavaScript que se utiliza para habilitar el **modo estricto** y hace que JavaScript funcione de una manera más segura y estricta, ayudando a evitar errores comunes y malas prácticas en la escritura de código.

**¿Para qué sirve?**

* **Evitar errores silenciosos:** Ciertos errores que JS normalmente ignoraría o trataría como advertencias, se convierten en errores que detienen la ejecución del código. Por ejemplo, asignar un valor a una variable no declarada genera un error en modo estricto.
* **Eliminar características no seguras:** Deshabilita algunas características de JS que se consideran problemáticas o inseguras. Por ejemplo, el modo estricto no permite el uso de watt y restringe el uso de eval.
* **Restringir el uso de palabras reservadas:** En modo estricto, no puedes utilizar nombres de variables que coincidan con palabras reservadas para futuras versiones de ECMAScript, lo que ayuda a evitar conflictos con nuevas características del lenguaje.

**¿Cómo se utiliza?** Para activar el modo estricto en JavaScript, simplemente incluyes la cadena **"use strict";** al comienzo de un script o de una función.