# Información general de TypeScript

5 minutos

JavaScript se considera uno de los lenguajes de programación más usados del mundo y se ha convertido en el oficial de la Web. Los desarrolladores lo usan para escribir aplicaciones multiplataforma que se pueden ejecutar en cualquier plataforma y explorador.

Aunque JavaScript se usa para crear aplicaciones multiplataforma, no está concebido para aplicaciones grandes que impliquen miles o incluso millones de líneas de código. JavaScript carece de algunas de las características de los lenguajes más maduros que se emplean en las sofisticadas aplicaciones de hoy en día. Para los editores de desarrollo integrado (IDE), puede ser todo un desafío administrar JavaScript y mantener estas grandes bases de código.

TypeScript aborda las limitaciones de JavaScript sin poner en peligro la propuesta de valor clave de JavaScript: la capacidad de ejecutar el código en cualquier lugar y en cualquier plataforma, explorador o host.

## ¿Qué es TypeScript?

TypeScript es un lenguaje de código abierto desarrollado por Microsoft. Se trata de un supraconjunto de JavaScript, lo que significa que puede usar sus conocimientos actuales sobre JavaScript que ya ha desarrollado junto con determinadas características que antes no estaban disponibles.

### Sugerencias de escritura

La característica principal de TypeScript es su sistema de tipos. En TypeScript, puede identificar el tipo de datos de una variable o un parámetro mediante una *sugerencia de tipo*. Con las sugerencias de tipo, se describe la forma de un objeto, lo que proporciona una documentación mejor y permite a TypeScript validar que el código funciona correctamente.

Mediante la comprobación de tipos estáticos, TypeScript al principio del desarrollo detecta problemas de código que JavaScript normalmente no puede detectar hasta que el código se ejecuta en el explorador. Los tipos también permiten describir lo que debe hacer el código. Si

forma parte de un equipo, un compañero que trabaje en el código después de usted podrá entenderlo fácilmente.

Los tipos también potencian las ventajas de inteligencia y productividad de las herramientas de desarrollo, como IntelliSense, la navegación basada en símbolos, la opción Ir a definición, la búsqueda de todas las referencias, la finalización de instrucciones y la refactorización del código.

Escribir tipos puede ser opcional en TypeScript, porque la *inferencia de tipos* permite obtener gran parte de esta potencia sin escribir código adicional. Si TypeScript puede determinar el tipo de datos implícitamente (por ejemplo, cuando se asigna un valor a una variable mediante `let age = 42`), el tipo de datos se deduce automáticamente.

## ¡Pruébelo! Más información sobre los tipos

Echemos un vistazo a un ejemplo para entender el uso de los tipos.

1. Abra el sitio de prueba de TypeScript. Aprenderá más sobre este elemento más adelante en el módulo.

2. Copie y pegue el siguiente ejemplo de código JavaScript en el panel de TypeScript (a la izquierda):

   JavaScript

   ```
   function addNumbers(x, y) {
     return x + y;
   }

   console.log(addNumbers(3, 6));
   ```

   Observe que aparece el mismo código en el panel **.JS** (a la derecha). En este panel se muestra el código JavaScript que TypeScript generará después de su compilación.

3. Seleccione **Ejecutar** para ejecutar el código JavaScript. Luego, seleccione la pestaña **Registros** y observe que el valor `9` se ha registrado en la consola. JavaScript ha asignado el tipo `number` a los parámetros `x` y `y`, y la función ha devuelto un número.

4. Reemplace `3` por `"three"` (incluidas las comillas) en el código TypeScript y, después, ejecútelo. JavaScript ahora asigna el tipo `string` al parámetro `x` y devuelve "three6" (un

tipo de cadena) a la consola. Probablemente ya se haya encontrado con esta situación en otras ocasiones y, como sabe, puede provocar algunos resultados inesperados.

En el panel de TypeScript, observe las líneas rojas onduladas que aparecen bajo los nombres de parámetro de la función `addNumbers`. Las líneas indican que el comprobador de tipos identificó los errores. Mantenga el puntero sobre uno de los parámetros y lea la descripción del error. TypeScript ha asignado implícitamente un tipo de `any`, que es el tipo más amplio porque, básicamente, puede contener cualquier cosa.

5. Actualice el código TypeScript para especificar un tipo para cada parámetro. Reemplace `x` por `x: number` y `y` por `y: number`.

   Observará que ahora los errores han desaparecido de los parámetros, pero que hay uno nuevo en el primer argumento de la llamada a función: "Argument of type 'string' is not assignable to parameter of type 'number'".

6. Reemplace `"three"` por un número para corregir el error. Puede pasar un valor literal, una variable o cualquier otro dato. Como TypeScript entiende la forma del objeto, le notificará el conflicto entre los tipos en tiempo de desarrollo.

7. Revise el código JavaScript y observe que no se ha producido ningún cambio en él. TypeScript podía proporcionar comprobación de tipos durante el desarrollo, pero no ha tenido ningún efecto en el código JavaScript resultante porque no admite tipos.

## Otras características de código TypeScript

TypeScript tiene características de creación de código adicionales que no se encuentran en JavaScript:

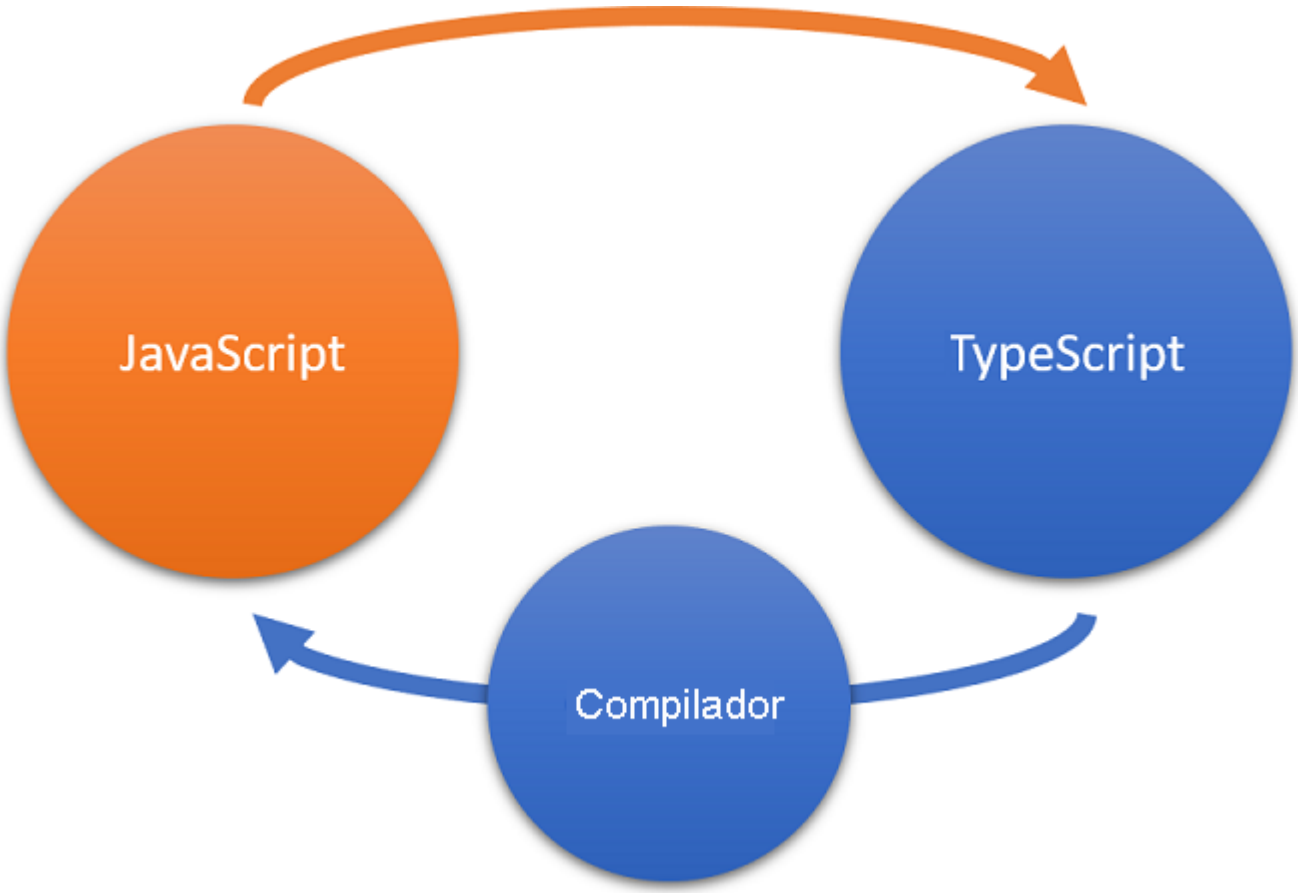
- Interfaces
- Espacios de nombres
- Genéricos
- Clases abstractas
- Modificadores de datos
- Valores opcionales
- Sobrecarga de funciones
- Elementos Decorator
- Tipos de utilidad
- Palabra clave readonly

Obtendrá más información sobre algunas de estas características en módulos posteriores.

# Compatibilidad de TypeScript con JavaScript

TypeScript es un supraconjunto estricto de ECMAScript 2015 (ECMAScript 6 o ES6). Esta relación significa que todo el código JavaScript es también código TypeScript, por lo que un programa escrito en TypeScript puede consumir JavaScript sin problemas.

Los exploradores solo entienden JavaScript. Para que la aplicación funcione, cuando la escriba en TypeScript, deberá compilar el código y convertirlo a JavaScript. El código TypeScript se transforma en código JavaScript mediante el uso del compilador de TypeScript o un transpilador compatible con TypeScript. El código JavaScript resultante es limpio y simple, y se ejecuta en cualquier lugar en el que se ejecuta JavaScript: en un explorador, en Node.js o en las aplicaciones.




**Importante**

Cuando trabaje con TypeScript, recuerde que, en casi todas las situaciones, el código TypeScript se compilará (o se transpilará) en JavaScript y que el código JavaScript se

ejecutará mediante el entorno de ejecución. Puede usar TypeScript en *cualquier* proyecto en el que use JavaScript.

# Migración de JavaScript a TypeScript

La adopción de TypeScript no es una elección binaria, por lo que puede migrar el código base gradualmente. Puede empezar por anotar el código JavaScript existente con JSDoc y, después, cambiar algunos archivos para que los compruebe TypeScript. Puede preparar el código base con el tiempo para que se convierta por completo.

Para obtener más información sobre este proceso, consulte Tutoriales de TypeScript: Migración desde JavaScript .

## Siguiente unidad: Ejercicio: Selección de un editor de TypeScript

Continuar >

¿Qué tal lo estamos haciendo? ☆ ☆ ☆ ☆ ☆