

Curso React

Nivelación TypeScript

¿Qué es TypeScript?

- Es un superconjunto de JavaScript que añade tipado estático y otras características
- Creado por Microsoft, es un lenguaje de programación que se compila a JavaScript
- Es compatible con JavaScript, lo que significa que cualquier código JavaScript válido es también código TypeScript válido

¿Por qué usar TypeScript?

- Mejora la mantenibilidad del código
- Ayuda a detectar errores en tiempo de compilación
- Facilita la colaboración en equipos grandes

¿Cómo se compila TypeScript a JavaScript?

- Utilizando el compilador `tsc` que convierte archivos `.ts` a `.js`
- También se puede configurar para compilar automáticamente al guardar los archivos
- Se puede integrar con herramientas de construcción como Webpack o Parcel
- Se puede configurar el archivo `tsconfig.json` para personalizar la compilación
- Se puede usar el comando `tsc --watch` para compilar automáticamente al detectar cambios
- Se puede usar el comando `tsc --init` para crear un archivo de configuración por defecto
- Se puede usar el comando `tsc --version` para verificar la versión instalada del compilador
- Se puede usar el comando `tsc --help` para ver todas las opciones disponibles del compilador

Conceptos Básicos de TypeScript

- Tipos primitivos
 - Son los tipos básicos como `number`, `string`, `boolean`, `null`, `undefined`, `void`, y `any`
- Interfaces
 - Son una forma de definir la forma de un objeto, especificando sus propiedades y tipos
- Clases
 - Son plantillas para crear objetos que pueden incluir propiedades y métodos, con soporte para herencia
- Módulos
 - Son archivos que pueden exportar e importar funcionalidades, permitiendo la organización del código en partes reutilizables
- Tipos genéricos
 - Son una forma de crear componentes o funciones que pueden trabajar con diferentes tipos de datos sin perder la información de tipo
- Enums
 - Son una forma de definir un conjunto de constantes con nombre, facilitando la legibilidad
- Tipos de unión

- Son una forma de definir un tipo que puede ser uno de varios tipos diferentes, utilizando el operador `|`
- Tipos de intersección
 - Son una forma de combinar múltiples tipos en uno solo, utilizando el operador `&`
- Tipos literales
 - Son tipos que representan un valor específico, como una cadena o un número concreto
- Tipos condicionales
 - Son una forma de crear tipos que dependen de una condición, utilizando la sintaxis `T extends U ? X : Y`
- Tipos de utilidad
 - Son tipos predefinidos que ayudan a transformar o manipular otros tipos, como `Partial<T>`, `Readonly<T>`, `Record<K, T>`, entre otros