

Typescript

Resumen

Javascript es un lenguaje potente que tiene una gran nivel de flexibilidad, esta es un de las razones principales que hace que sea un lenguaje no tipado. Programar con en un lenguaje dinámico como Javascript puede que te haga sentir que avanzas rápido y de forma productiva de primeras.

No obstante, esta ventaja se puede convertir en un problema cuando empiezan a explotarte en la cara fallos en tiempo de ejecución. En aplicaciones grandes, en las que el código lo compartes con otros desarrolladores es crucial poder contestar a preguntas del tipo "¿ Qué tipo de argumentos acepta esta función?", "Puedo pasar un null o undefined?", "¿ Qué tipo de valor devuelve?". Si no tenemos un sistema de tipado implantado en nuestro código, la respuestas a estas preguntas son suposiciones.

Construir proyectos de front-end grandes es un desafío, en el que trabajar de forma efectiva y eficiente es crucial para llegar a entregar en fecha y coste. Tener un equipo de desarrollo 100% productivo es crítico, e invertir tiempo en crear un código tipado, puede llegar a ahorrar horas de depuración y minimizar fallos inesperados en producción, aquí es donde Typescript se convierte en una pieza fundamental.

Typescript le añade tipado estático a Javascript, haciendo que el código de tu aplicación sea robusto y sólido.

En esta acción formativa cubrimos el lenguaje Typescript desde conceptos basicos hasta avanzados.

Tabla de contenido

- Introducción al lenguaje Typescript.
- Conceptos básicos
 - Type annotation.
 - Tipos basicos.
 - Boolean.
 - Number.
 - String.
 - Array.
 - Tuple.
 - Enum.
 - Void.
 - Null & Undefined.
 - Never.
 - Any.
 - Type Assertion.
- Interfaces.
 - Syntax.
 - Propiedades opcionales.
 - Propiedades de sólo lectura.
 - Extendiendo interfaces.

- Duck Typing.
- Classes.
 - Sintaxis.
 - Access operators: public, private and protected.
 - Propiedades estáticas.
 - Clases abstractas.
 - Constructor shorthands.
- Funciones.
 - Sintaxis.
 - Argumentos opcionales.
 - Argumentos por defecto.
 - Extrayendo tipos de función a Alias.
 - Funciones como argumentos.
 - Sobrecarga de Funciones.
 - Tipando funciones a través de interfaces.
- Genéricos.
 - Sintaxis.
 - Funciones genéricas.
 - Interfaces genéricos.
 - Clases genéricas.
- Tipos avanzados.
 - Intersection.
 - Union.
 - Guardas, Disambiguation Guards.
 - Guardas definidas por el usuario: duck typing y predicados de tipo.
 - Guardas Typeof.
 - Guardas Instanceof.
 - Alias.
 - String literals.
 - Numeric literals.
 - Keyof.
 - Tipos condicionales.
- Tipos de utilidad.
 - Partial.
 - Required.
 - Readonly.
 - Record.
 - Pick.
 - Omit.
 - Return type.

Ficha del curso

- Duración: 6 Horas
- Rango de estudiantes recomendado: 6 - 12.
- Disponible en formato online (clases en vivo).