

¿Qué voy a aprender?

JavaScript es el lenguaje de programación más usado en la actualidad, y TypeScript hace de él un lenguaje más potente y más escalable.

En este curso aprenderás a dominar TypeScript, una tecnología que te va a permitir utilizar los últimos estándares ECMAScript y las metodologías de los grandes lenguajes de programación a la hora de escribir aplicaciones web o Node.js. En este curso aprenderás cómo separar tu código fuente en módulos que se comunican entre sí, cómo gestionar y organizar las dependencias con Webpack, cómo utilizar clases, interfaces y decoradores, y cómo optimizar el código que le enviamos al navegador.

En el curso vas a ver cómo se aplican mediante TypeScript los patrones más consolidados en los grandes lenguajes de Programación Orientada a Objetos, y mediante el sistema de tipado conseguirás un código mucho más fiable y con menos errores. Aprenderás a realizar aplicaciones mucho más escalables y más fáciles de mantener.

TypeScript es conocido por ser el lenguaje más adecuado para programar aplicaciones con Angular, pero cuando finalices el curso lo utilizarás para cualquier aplicación basada en JavaScript, ya sea de frontend (Angular, React, Vue, Ionic, KnockOut o sin framework) o de backend (Node.js).

Plazas Limitadas

Duración: 5 semanas (40 horas aprox. Incluyendo lecciones en video, apuntes, realización de ejercicios y práctica final.)

Online: Accede cuando y desde donde quieras.

Tutelados: Seguimiento continuo. Foro de dudas y consultas con respuesta en 24 horas laborables. Corrección de práctica.

Bonificable por Fundae.

Precio: 300€ + Iva*

* Consultar promociones en www.trainingit.es

¿Qué conocimientos son necesarios?

- Experiencia programando con JavaScript, preferiblemente con ECMAScript 2015 (ES6).
- Experiencia en programación orientada a objetos.
- Es recomendable, aunque no imprescindible, tener cierta experiencia trabajando con el marcado HTML y el modelo DOM.

Material necesario

- Editor de código, preferiblemente Visual Studio Code.

Profesor

Mario González

Desarrollador web y formador en tecnologías web y en herramientas de ayuda al desarrollo. Admirador confeso de JavaScript, TypeScript, Angular, Node.js y Git, paso la mitad de mis días aplicando estas tecnologías y la otra mitad ayudando a otros desarrolladores a adentrarse en ellas por primera vez o a profundizar en sus metodologías y técnicas recomendadas.



Acerca de TrainingIT

TrainingIT es una iniciativa en colaboración con **HiringIT** para dar formación bonificable de altísima calidad en temas relacionados con IT: Desarrollo back y front, sistemas, bases de datos, data science, inteligencia artificial, big data...

Nos diferenciamos sobre todo por la calidad de los cursos, que nacen con la mentalidad de un curso presencial: profesor dedicado, ejercicios, tests, prueba final, tutorías, diploma de certificación...

Contáctanos en info@trainingit.es o en el 656.27.65.04.



Temario del curso

1. Fundamentos de TypeScript.

1. Historia.
2. Límites de JavaScript.
3. Transpilación.
4. Complemento a JS.
5. Integración con el IDE
6. Errores en tiempo de compilación

2. Entorno de desarrollo.

1. Visual Studio Code. Instalación, configuración y extensiones útiles.
2. NodeJS y npm.
3. Gestión de dependencias mediante npm y package.json.
4. Instalación de paquetes npm. Versiones semánticas.
5. TypeScript como paquete npm. El compilador tsc.
6. Scripts en package.json.

3. JavaScript moderno.

1. ECMAScript6. Novedades importantes.
2. Módulos. Integración con librerías de terceros y código JS externo.
3. Herramientas para poder programar con ES6: transpiladores y module bundlers.
4. Webpack. Uso y configuración.

4. TypeScript.

1. Configuración y uso del compilador.
2. Los tipos. El tipado y el compilador.
 1. Tipos básicos.
 2. Enums.
 3. Union types.
 4. Genéricos y type assertions.
3. Funciones.
 1. Parámetros.
 2. Sobrecarga.
 3. Function types.
4. Módulos.
 1. Sintaxis.
 2. Encapsulación.
 3. Importación y resolución de nombres y rutas.
 4. Exportación. Definiendo la API de nuestros módulos.
5. Clases.
 1. Sintaxis. Propiedades y métodos. Visibilidad de los miembros.
 2. Getters y setters.
 3. Miembros estáticos.
 4. Miembros abstractos.
 5. Herencia.
 6. Interfaces. La importancia de las interfaces en el sistema de tipado.
6. Decoradores.
 1. Metadatos.
 2. Tipos.
 3. Uso de decoradores.
 4. Creación de decoradores.

