

# **Introducción a TS**

Antonio Espín Herranz

# Contenidos

- ¿Qué es TypeScript?
- JavaScript vs TypeScript
- Transpilación
- Sistema de tipos
- Arquitectura de TS
- Enlaces

# ¿Qué es TypeScript?

- **TypeScript** es un lenguaje de programación libre y de código abierto desarrollado y mantenido por **Microsoft**.
- Superconjunto de JavaScript
  - Añade tipos estáticos y objetos basados en clases
- TypeScript es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del **cliente** o de **servidor**,
- Pensado para grandes proyectos
- Lenguaje para frameworks como angular o vue.js
- Soporta ficheros de definición como C/C++

# JavaScript vs TypeScript

- **Tipado en JavaScript**

- Es un lenguaje dinámico, sin tipos definidos.
- Las variables pueden cambiar de tipo en tiempo de ejecución

- **Desarrollo y mantenimiento JS**

- Por su flexibilidad, puede ser más rápido al comenzar proyectos pequeños, pero el mantenimiento puede complicarse a medida que el proyecto crece

- **Tipado en TypeScript**

- Introduce un tipado estático, lo que significa que se pueden especificar los tipos de las variables y funciones, reduciendo errores en el desarrollo.

- **Desarrollo y mantenimiento TS**

- Su tipado estático y características avanzadas como interfaces y generics facilitan el mantenimiento de proyectos grandes y colaborativos

# JavaScript vs TypeScript II

- **JavaScript:**

- No requiere compilación, se ejecuta directamente en navegadores y entornos como **Node.js**.
- Es ampliamente utilizado, con soporte nativo en navegadores y una gran comunidad.

- **TypeScript:**

- Necesita ser compilado a JavaScript antes de ejecutarse, lo que añade un paso al flujo de trabajo.
- Posibilidad de utilizar: **ts-node**
- Aunque está ganando popularidad rápidamente, no tiene soporte nativo en navegadores.
- Cuenta con una fuerte comunidad, especialmente en proyectos grandes y corporativos

# JavaScript vs TypeScript III

- Sintaxis **JS**:

- Simplicidad y flexibilidad; adecuado para principiantes.

- **Desarrollo:**

- Rápido, simplicidad y flexibilidad

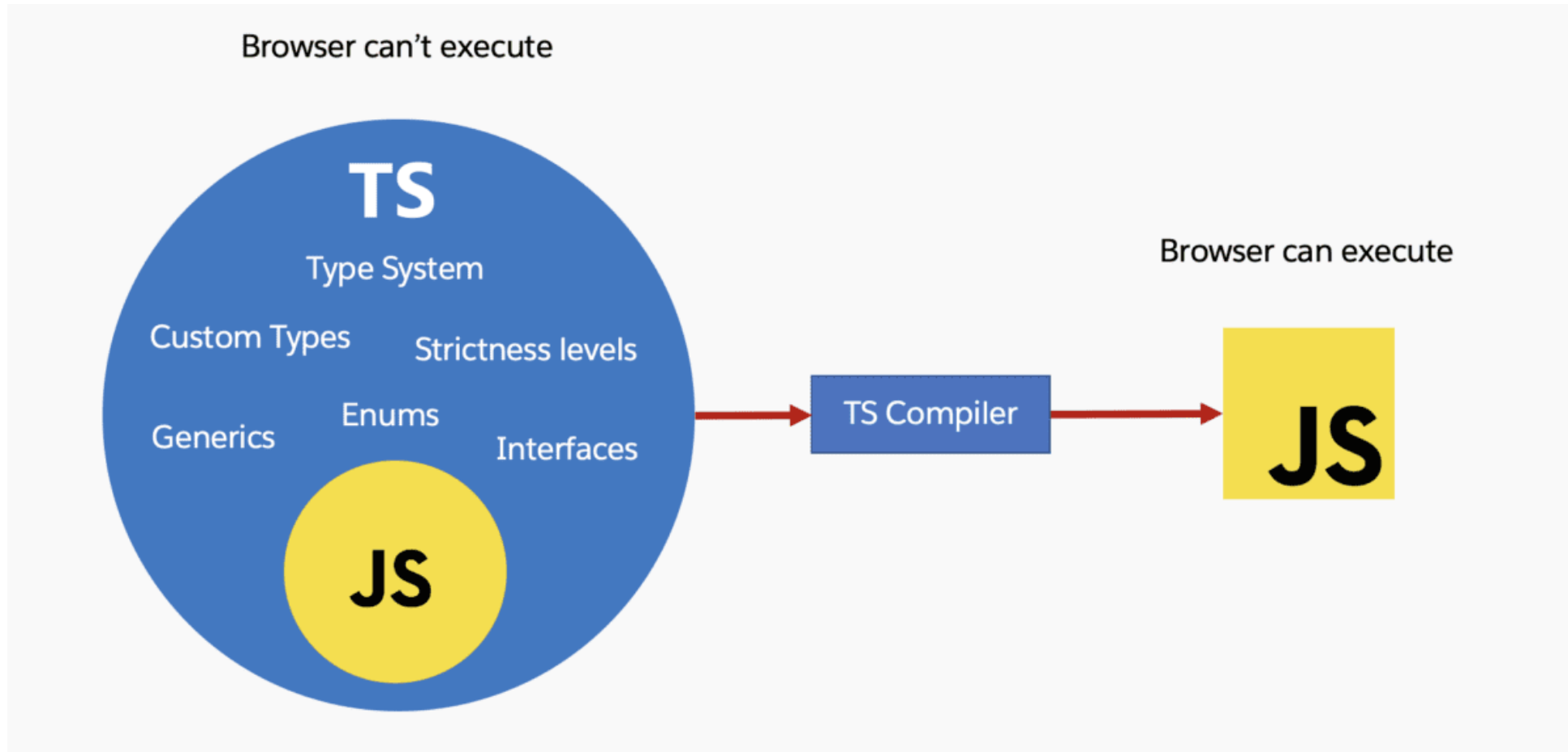
- Sintaxis **TS**:

- Extiende JavaScript, pero introduce características más estrictas y avanzadas, lo cual puede requerir una curva de aprendizaje.

- **Desarrollo:**

- Robusto
- Escalabilidad
- Detención temprana de errores

# JavaScript vs TypeScript IV



# Transpilación

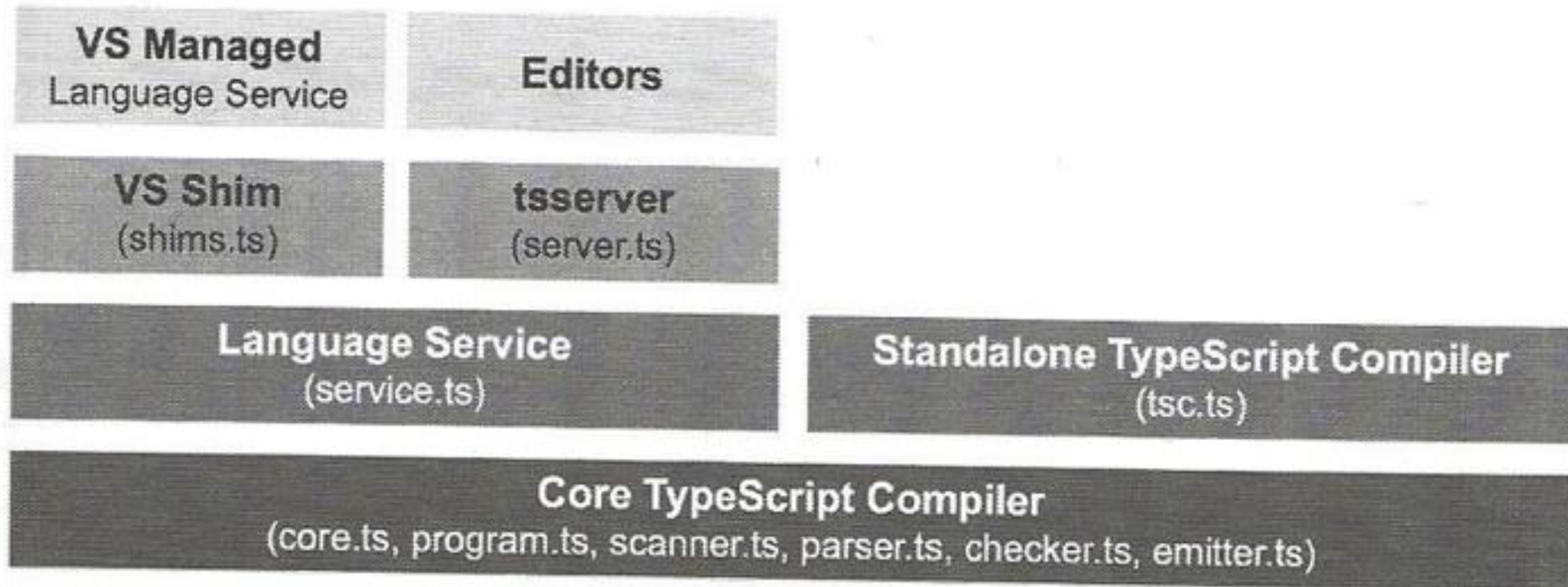
- La **transpilación** es el proceso de convertir el código fuente escrito en un lenguaje de programación a otro lenguaje de programación, manteniendo un nivel similar de abstracción.
- En el caso de TypeScript y JavaScript, la transpilación implica convertir el código TypeScript (con tipado estático y características avanzadas) a JavaScript estándar para que pueda ejecutarse en navegadores o entornos como Node.js.
- En este contexto:
  - Entrada: Código fuente en TypeScript.
  - Proceso: Un transpilador (como el compilador de TypeScript, **tsc**) convierte el código **TypeScript** a **JavaScript**.
  - Salida: Código JavaScript que puede ser entendido y ejecutado por el motor JavaScript.



# Sistema de tipos

- El **sistema de tipos de TS es estructural**.
- Los tipos estructurales se definen por su estructura no por el nombre del tipo.
- Un objeto de la claseA se puede asignar a un objeto de la claseB si A tiene al menos todas las propiedades / métodos de B
  - Ejemplos: TypeScript, Go, Haskell
- Un sistema de tipo nominal. Se identifican con el nombre del tipo.
  - Si A tiene las mismas propiedades que B, no se pueden asignar entre sí dos variables de cada tipo.
    - Ejemplos: C#, Java, PHP y C++

# Arquitectura de TS



# Arquitectura TS

- **Core TypeScript Compiler**
  - Representa el compilador de TS
  - Pieza central de la arquitectura
  - Dos archivos importantes:
    - checker.ts: verifica el código y devuelve errores
    - emitter.js: transpila el código de JS

# Arquitectura TS

- **Standalone TS Compiler**

- Es el comando **tsc**.
- Herramienta en línea para compilar los archivos

- **Language Service**

- Se dedica a integrar TypeScript en entornos de desarrollo.
- Proporciona las funciones de autocompletado, navegación de códigos, etc. para los IDEs

# Opciones de compilación

- Se pueden proporcionar en línea.
- A partir de un fichero de configuración: **tsconfig.json**
  - Principales opciones:
    - **target**: la versión ECMAScript destino
    - **module**: define el estándar del módulo destino
    - **strict**: para el control de tipos

# Enlaces

- Página oficial:
  - <https://www.typescriptlang.org/>
- Documentación:
  - <https://www.typescriptlang.org/docs/>