



TypeScript (parte3 tsconfig.ts)

LABORATORIO DE COMPUTACIÓN III

UTNFRA – TÉCNICO SUPERIOR EN PROGRAMACIÓN --

tsconfig.json

La presencia del archivo `tsconfig.json` en un directorio indica que es un directorio raíz de un proyecto TypeScript.

En este archivo se puede configurar entre otras cosas las opciones del compilador.

Para generar el archivo `tsconfig.json` debemos escribir en una terminal el siguiente comando: `tsc --init`

Ejemplo de archivo tsconfig.json

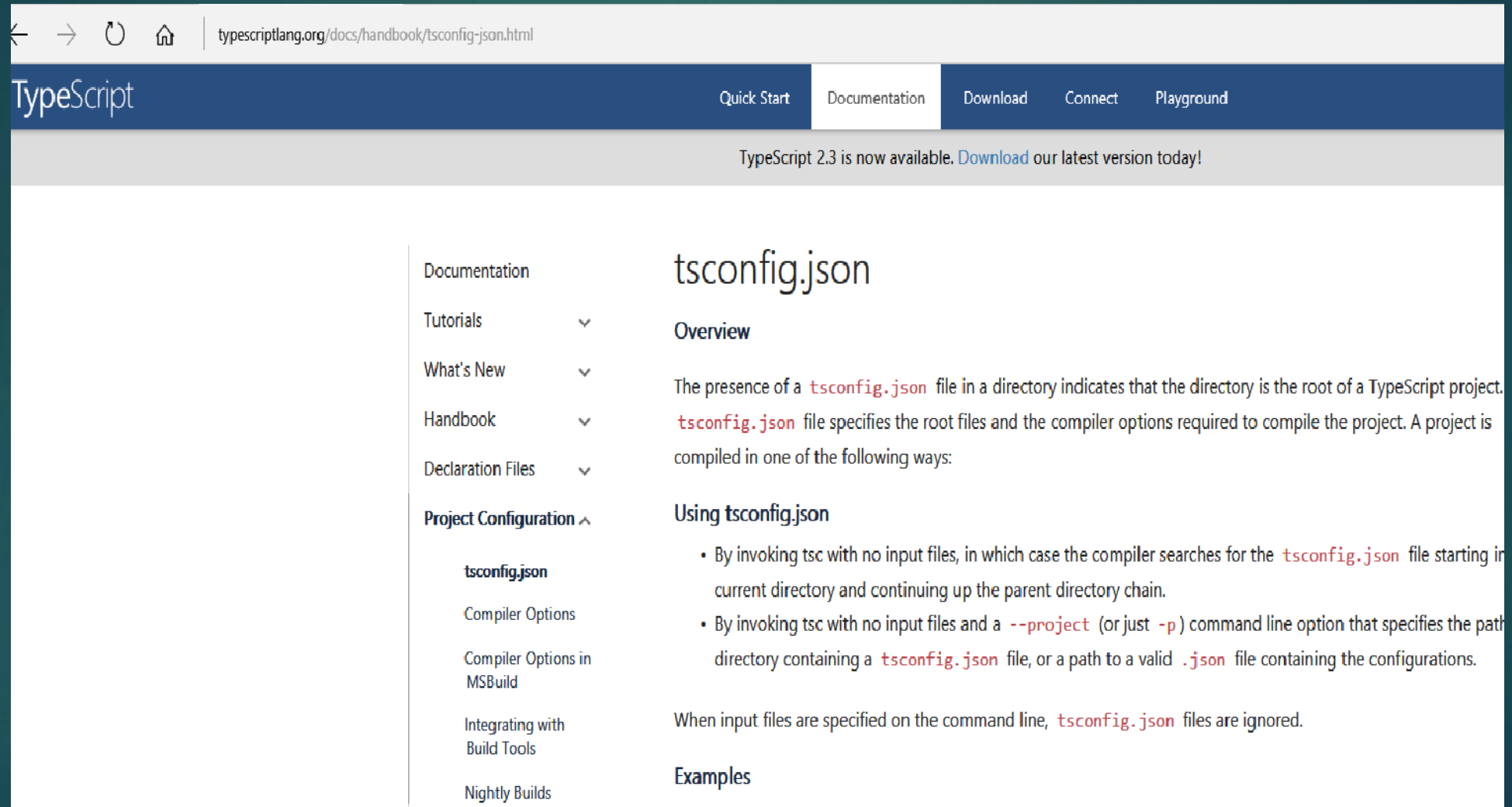
```
{
  "compilerOptions": {
    "module": "commonjs",
    "noImplicitAny": true,
    "removeComments": true,
    "preserveConstEnums": true,
    "sourceMap": true
  },
  "files": [
    "core.ts", "
    sys.ts",
    "types.ts",
    "scanner.ts",
    "parser.ts",
    "utilities.ts",
    "binder.ts",
    "checker.ts",
    "emitter.ts",
    "program.ts",
    "commandLineParser.ts",
    "tsc.ts",
    "diagnosticInformationMap.generated.ts"
  ]
}
```

sourceMap

sourceMap nos permite depurar el código
typeScript directamente en la consola del
navegador

“sourceMap” = true

Toda la información acerca de tsconfig.json



The screenshot shows the TypeScript website's documentation page for `tsconfig.json`. The browser's address bar displays `typescriptlang.org/docs/handbook/tsconfig-json.html`. The website's navigation bar includes the TypeScript logo, a 'Quick Start' link, and a 'Documentation' link which is currently selected. Other links in the bar are 'Download', 'Connect', and 'Playground'. A banner below the navigation bar states: 'TypeScript 2.3 is now available. [Download](#) our latest version today!'. On the left side, a sidebar menu lists various documentation sections: 'Documentation', 'Tutorials', 'What's New', 'Handbook', 'Declaration Files', 'Project Configuration' (which is expanded), 'tsconfig.json' (the current page), 'Compiler Options', 'Compiler Options in MSBuild', 'Integrating with Build Tools', and 'Nightly Builds'. The main content area is titled 'tsconfig.json' and contains an 'Overview' section. The overview text explains that the presence of a `tsconfig.json` file in a directory indicates it is the root of a TypeScript project, and that this file specifies the root files and compiler options. It also mentions that a project can be compiled in one of two ways. Below this is a section titled 'Using tsconfig.json' which lists two methods: 1) invoking `tsc` with no input files, where the compiler searches for `tsconfig.json` starting in the current directory and moving up the parent directory chain; 2) invoking `tsc` with no input files and a `--project` (or just `-p`) command line option that specifies the path to a directory containing a `tsconfig.json` file, or a path to a valid `.json` file containing the configurations. A note states that when input files are specified on the command line, `tsconfig.json` files are ignored. The page concludes with an 'Examples' section.

typescriptlang.org/docs/handbook/tsconfig-json.html

TypeScript

Quick Start Documentation Download Connect Playground

TypeScript 2.3 is now available. [Download](#) our latest version today!

Documentation

Tutorials

What's New

Handbook

Declaration Files

Project Configuration

tsconfig.json

Compiler Options

Compiler Options in MSBuild

Integrating with Build Tools

Nightly Builds

tsconfig.json

Overview

The presence of a `tsconfig.json` file in a directory indicates that the directory is the root of a TypeScript project. The `tsconfig.json` file specifies the root files and the compiler options required to compile the project. A project is compiled in one of the following ways:

Using tsconfig.json

- By invoking `tsc` with no input files, in which case the compiler searches for the `tsconfig.json` file starting in the current directory and continuing up the parent directory chain.
- By invoking `tsc` with no input files and a `--project` (or just `-p`) command line option that specifies the path to a directory containing a `tsconfig.json` file, or a path to a valid `.json` file containing the configurations.

When input files are specified on the command line, `tsconfig.json` files are ignored.

Examples

Remover comentarios

Si no queremos que alguien pueda leer los comentarios que tenemos en el código TypeScript podemos removerlos fácilmente agregando en el archivo tsconfig.json la siguiente línea:

```
"removeComments" : true
```

Si queremos que algún comentario quede en el .js debemos escribirlo de la siguiente manera: `/*! */`



Incluir y excluir archivos

Si quisiéramos que algunos archivos no se incluyan en la transpilación y otros si se deben incluir. Debemos agregar en el archivo `tsconfig.json` las directivas necesarias para ello.

No son opciones del compilador así que las agregamos después.

“include” / “exclude”

```
{
  "compilerOptions": {
    "module": "commonjs",
    "target": "es5",
    "noImplicitAny": false,
    "sourceMap": false,
    "removeComments": true
  },
  "include": [
    "app/**/*"
  ],
  "exclude": [
    "carpeta/",
    "src/"
  ]
}
```


Archivo de Salida - outFile -

Podemos transpilar varios archivos en un único archivo de salida.
Para ello en la consola debemos colocar la siguiente directiva:

```
tsc --outFile build/app.js ap2 ap1 ap3
```

Cambiar versión de JavaScript

Si queremos cambiar la versión de ECMAScript (JavaScript) a la cual queremos transpilar debemos indicarlo en el archivo `tsconfig.json` modificando la opción del compilador `target`.

Por Ejemplo: `"target" : "es6"`

O por consola: `tsc app.ts --target es6`

Modo “Observador”

Podemos hacer que TypeScript identifique cambios en los archivos y haga la transpilación de manera automática. Para ello debemos agregar un watch por consola mediante el siguiente comando:

```
tsc -w ó tsc -watch
```

Para dejar de “observar” se debe teclear en la consola ctrl + c