

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

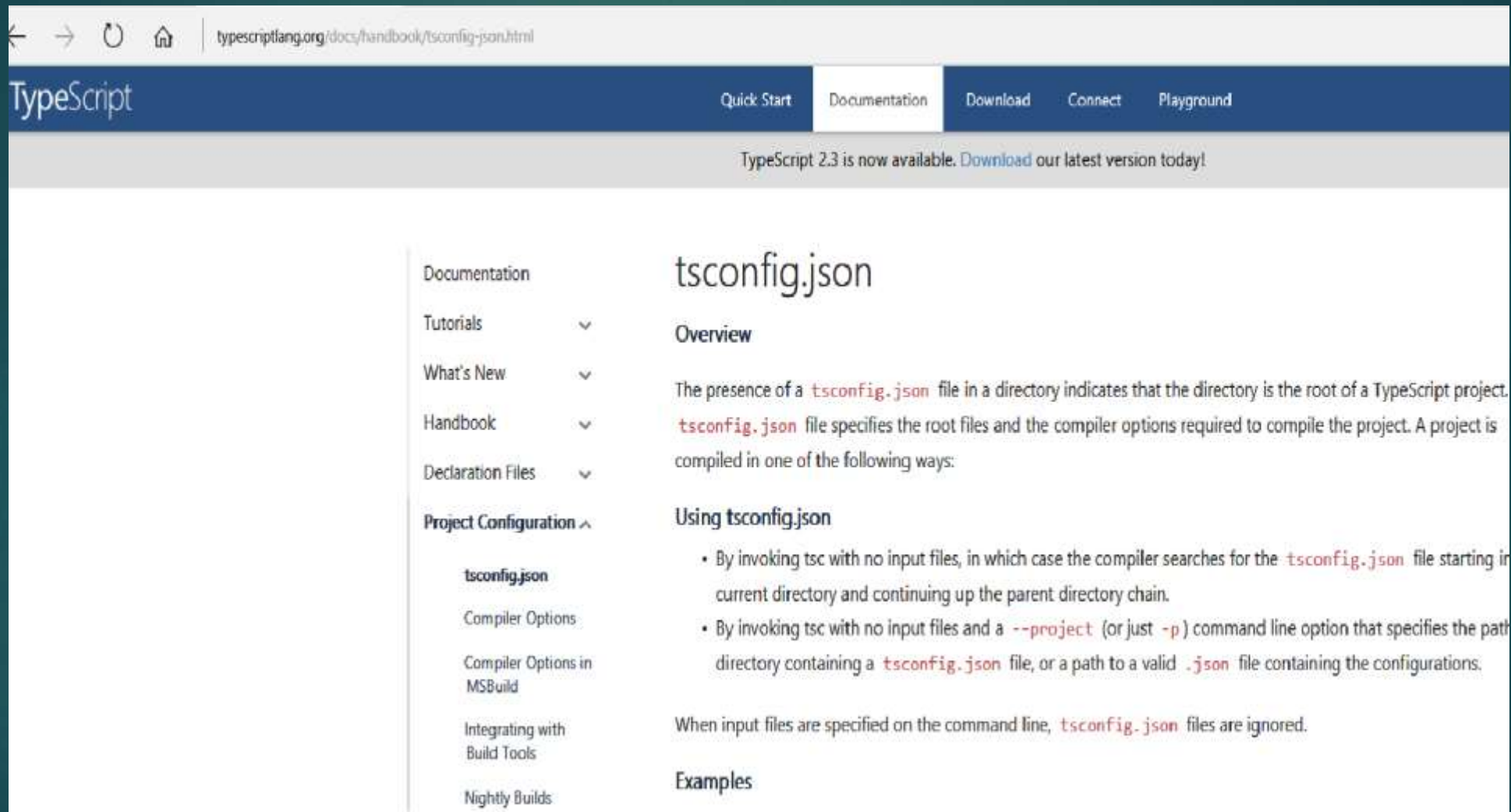
```
{
  "compilerOptions": {
    "module": "commonjs",
    "noImplicitAny": true,
    "removeComments": true,
    "preserveConstEnums": true,
    "sourceMap": true
  },
  "files": [
    "core.ts", "
    sys.ts",
    "types.ts",
    "scanner.ts",
    "parser.ts",
    "utilities.ts",
    "binder.ts",
    "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 documentation page for `tsconfig.json`. The browser address bar displays `typescriptlang.org/docs/handbook/tsconfig-json.html`. The page features a dark blue header with the TypeScript logo and navigation links: Quick Start, Documentation (active), Download, Connect, and Playground. A banner below the header states: "TypeScript 2.3 is now available. [Download](#) our latest version today!".

On the left, a sidebar lists the documentation structure:

- Documentation
- Tutorials
- What's New
- Handbook
- Declaration Files
- Project Configuration (expanded)
 - tsconfig.json**
 - Compiler Options
 - Compiler Options in MSBuild
 - Integrating with Build Tools
 - Nightly Builds

The main content area is titled `tsconfig.json` and includes the following sections:

Overview

The presence of a `tsconfig.json` file in a directory indicates that the directory is the root of a TypeScript project. `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