**The tsconfig.json File**

**Let’s explore what the tsconfig.json file is for.**

TypeScript features are extremely useful: for one, it allows us to add types to regular JavaScript code. It also checks for syntax errors even before run time. It even provides tooltips that show you why some code might throw an error. There are so many great features that come by default. Even with all these great features, TypeScript recognizes the need for flexibility.

Sometimes, you don’t want all the [default rules](https://www.typescriptlang.org/docs/handbook/compiler-options.html) that TypeScript is trying to enforce — and that’s fine. That’s one reason why providing a **tsconfig.json** file is useful. Additionally, you get perks like telling the TypeScript compiler what files to run on and more! So, let’s explore what this file looks like and how it helps.

**Sample tsconfig.json and Breakdown**

The **tsconfig.json** file is always placed in the root of your project and you can customize what rules you want the TypeScript compiler to enforce. Here’s the **tsconfig.json** file we provide in every single exercise and project in our Learn TypeScript course:

{  
  "compilerOptions": {  
    "target": "es2017",  
    "module": "commonjs",  
    "strictNullChecks": true  
  },  
  "include": ["\*\*/\*.ts"]  
}

In the JSON, there are several properties:

* **"compilerOptions"**, which is a nested object that contains the rules for the TypeScript compiler to enforce.
  + **"target"**, the value **"es2017"** means the project will be using the 2017 version of EcmaScript standards for JavaScript.
  + **"module"**, this project will be using **"commonjs"** syntax to import and export modules.
  + **"strictNullChecks"**, variables can only have **null** or **undefined** values if they are explicitly assigned those values.
* **"include"** that determines what files the compiler applies the rules to. In this case **["\*\*/\*.ts"]** means the compiler should check every single file that has a **.ts** extension.

We chose these rules because we wanted to give you an idea of what rules to enforce and still allow your creativity to shine. For your own projects, you can set these rules to your own preferences! The **tsconfig.json** file is great for both individual work and team projects because it allows everyone to be on the same page about how to write their code.

**Usage**

Another neat addition is that by including a **tsconfig.json** file, you can now use the command **tsc** without any arguments in your terminal! The compiler will automatically recognize from your **tsconfig.json** file, what specific files to run on. You can still provide specific files like **tsc fileName.ts** if that’s the only file you want the compiler to check.

**Wrap up**

Check out TypeScript’s [compiler option documentation](https://www.typescriptlang.org/docs/handbook/compiler-options.html) for even more information.

Happy Coding! And remember, when in doubt, *type* it out!