

# Import Shortcuts

In this lesson, you will discuss about how to use a simplified how importation of module.

## WE'LL COVER THE FOLLOWING ^

- What is an Import Shortcut?
- Configuration
- Web Development
- Unit Test Framework

## What is an Import Shortcut? #

TypeScript's configuration can create shortcuts to specific areas of the code. A shortcut can be useful in a situation where many imports are deep in another branch of the code which requires a lot of navigating; for example

```
../../../../../../../../..
```

## Configuration #

To set up a shortcut, TypeScript's configuration **must** has a `baseUrl` in the `tsconfig.json`. Then, it's possible to add many shortcuts. Each of the shortcuts go in the `paths` list and are relative to the `baseUrl`. The definition of each path contained is a keys-values list, for which the key is the pattern to look for when importing and the value is the location to look for in the actual TypeScript file.

The configuration can look like the following:

```
"baseUrl": "./src",
"paths": {
  "@link/folder1/*": ["folderInSrc/a/b/c/folder1/*"],
  "@link/folder2/*": ["folderInSrc/x/y/z/*"]
}
```



The paths' keys don't need to start with an `@`, but it's a good way to differentiate between shortcuts and actual paths.

## Web Development #

The import shortcut is a mapping pattern that is well accepted among web projects. If you are using TypeScript while developing a web solution, you may need to have to use a plugin with [Webpack](#) for example or similar to another bundler. Without a plugin, you may need to manually have a correspondence mapping under `alias` of your webpack's configuration.

## Unit Test Framework #

Depending on the unit test framework that you are using, you may have to replicate the mapping pattern in the specific configuration file of the framework. For example, Jest requires to have a `moduleNameMapper` with every shortcut re-defined.