



React 1 of 3

Step 3 – The tsc compiler

Let's bootstrap a simple Typescript Node.js application locally on our machines

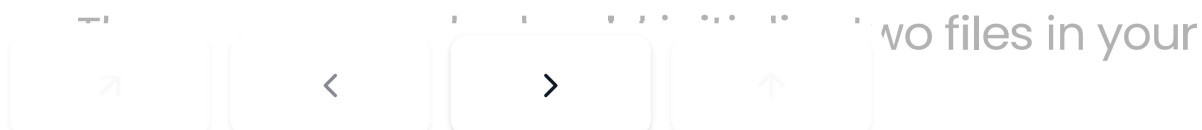
Step 1 – Install tsc/typescript globally

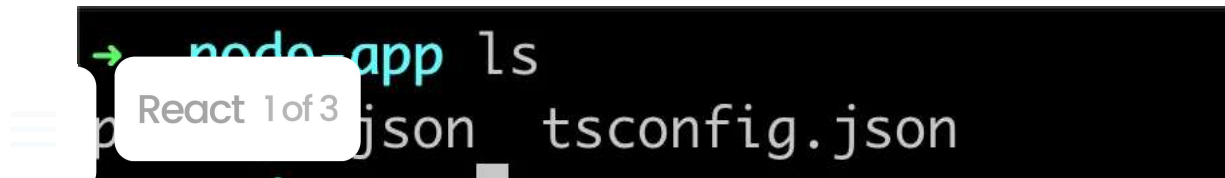
```
npm install -g typescript
```



Step 2 – Initialize an empty Node.js project with typescript

```
mkdir node-app  
cd node-app  
npm init -y  
npx tsc --init
```





Step 3 – Create a a.ts file

```
const x: number = 1;  
console.log(x);
```

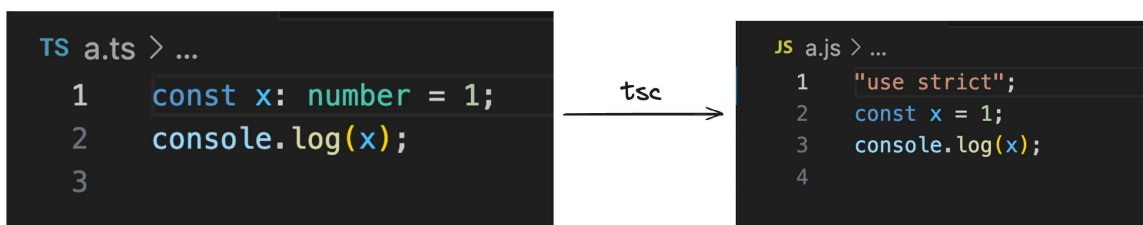


Step 4 – Compile the ts file to js file

```
tsc -b
```



Step 5 – Explore the newly generated index.js file



Notice how there is no typescript code in the
v React 1 of 3 It's a plain old js file with no types

Step 7 – Delete a.js

Step 6 – Try assigning x to a string

Make sure you convert the `const` to `let`

```
let x: number = 1;  
x = "harkirat"  
console.log(x);
```



Step 7 – Try compiling the code again

```
tsc -b
```



Notice all the errors you see in the console. This tells you there are `type` errors in your codebase.

Also notice that no `index.js` is created anymore

```
→ node-app tsc -b  
a.ts:2:1 - error TS2322: Type 'string' is not assignable to type 'number'.  
2 x = "harkirat"  
  ~
```

This is the high level benefit of typescript. It lets you

React 1 of 3 errors at **compile time**

Step 1 – Types of languages

1. Strongly typed vs loosely typed

The terms **strongly typed** and **loosely typed** refer to how programming languages handle types, particularly how strict they are about type conversions and type safety.

Strongly typed languages

1. Examples – Java, C++, C, Rust
2. Benefits –

Loosely typed languages

1. Examples – Python, Javascript, Perl, php
2. Benefits

easy to write code

easy to bootstrap

2. Stricter codebase
React 1 of 3
... catch
errors at compile
time

3. Low learning curve

Code does work



Code doesn't work ❌

```
#include <iostream>

int main() {
  int number = 10;
  number = "text";
  return 0;
}
```



```
function main() {
  let number = 10;
  number = "text";
  return number;
}
```

People realised that javascript is a very power language, but lacks types. **Typescript** was introduced as a new language to add **types** on top of javascript.



Step 2 – What is Typescript


What is typescript?

TypeScript is a programming language developed and maintained by Microsoft.

It is a strict **syntactical superset** of JavaScript and adds optional static typing to the language.

Where/How does typescript code run?

Typescript code never runs in your browser. Your browser runs the **JavaScript** code.

- 
1. Javascript is the runtime language (the thing that **React** runs in your browser/nodejs runtime)
 2. Typescript is something that compiles down to javascript
 3. When typescript is compiled down to javascript, you get **type checking** (similar to C++). If there is an error, the conversion to Javascript fails.

Typescript compiler

tsc is the official typescript compiler that you can use to convert **Typescript** code into **Javascript**

There are many other famous compilers/transpilers for converting Typescript to Javascript. Some famous ones are -

1. esbuild
2. swc



React 1 of 3