Environment Setup for Typescript:

- **Step 1: Install Node.js**

  Open your web browser and go to https://nodejs.org/

  On the Node.js website, you'll see a "LTS" (Long Term Support) version and a "Current" version. It's recommended for beginners to choose the LTS version as it's more stable.

  Click on the LTS version download button to start downloading the Node.js installer.

  Once the download is complete, run the installer and follow the on-screen instructions. You can generally use the default settings during installation.

- **Step 2: Install TypeScript**

  After Node.js is installed, open your computer's terminal. On Windows, you can use the Command Prompt or PowerShell.

  To install TypeScript, type the following command and press Enter:

  For Windows users run

  ```
  npm install -g typescript
  ```

  For Mac users run

  ```
  sudo npm install -g typescript
  ```

- **Step 3: Install Visual Studio Code (VSCode)**
  Open your web browser and go to https://code.visualstudio.com/Download.
  Download the appropriate version of Visual Studio Code for your operating system (Windows, macOS, or Linux).
  Once the download is complete, run the installer and follow the on-screen instruction

- **Step 4: Initialize TypeScript Configuratio**n

  Open Visual Studio Code.

In VSCode, open the integrated terminal by clicking on "View" in the top menu and selecting "Terminal" from the dropdown.
In the terminal, navigate to the folder where you want to create your TypeScript project. You can use the cd command to change directories.
Once you're in the desired folder, type the following command to initialize a TypeScript

```
tsc --init
```

- **Step 5: Set Execution Policy (Windows Users Only)**
  If you're a Windows user and encounter an execution policy error, open the Windows PowerShell as an administrator. To do this, find "Windows PowerShell" in the Start menu, right-click it, and select "Run as administrator."
  In the PowerShell window, type the following command and press Enter to set the execution policy
    - `Set-ExecutionPolicy RemoteSigned`

  This allows you to run scripts that you've created locally.

- **Step 6: Make it a Node.js Project**
  In the same terminal where you initialized the TypeScript configuration, type the following command to create a package.json file for your project:
    - `npm init -y`

- **Step 7: Install Types for Node.js**
  In the terminal, type the following command to install TypeScript typings for Node.js
  `npm i @types/node -D`

- **Step 8: Write TypeScript Code**
  Create a new TypeScript file in your project directory. You can use the file extension .ts e.g (filename.ts)

Write your TypeScript code in this file.

```
Let greeting = ‘Hello world’;
console.log(greeting)
```

- **Step 9: Transpile TypeScript to JavaScrip**t
  In the terminal, run the following command to transpile your TypeScript code into

  ```
  tsc
  ```

- **Step 10: Run Transpiled JavaScript**
  After transpilation, you'll find a corresponding JavaScript file in your project directory.
  In the terminal, run the JavaScript file using Node.js with the following command:

  ```
  node filename
  ```

  After running above command in the terminal you’ll get following output:
  Hello world