step00_helloworld

```
let message = "Hello World"; // Infering Types,
// take your cursor on the variable name
console.log(message);

let num = 5; // TypeScript infers 'number' for 'num'
let message = "hello"; // TypeScript infers 'string' for 'message'
```

Get Node.js v22.4.11

TypeScript can be installed through three installation routes depending on how you intend to use it: an npm module, a NuGet package or a Visual Studio Extension

TypeScript in Your Project via npm

manager like npm, yarn or pnpm to download TypeScript into your project.

```
npm install typescript --save-dev
```

TypeScript in Your Project with Visual Studio

The Nuget Package Manager Console (found in Tools > NuGet Package Manager > Package Manager Console) and then running:

```
Install-Package Microsoft.TypeScript.MSBuild
```

Generate a tsconfig.json file

Open your terminal in your project's root directory.

Install typescript globally with

```
npm install typescript@latest -g
```

Generate a tsconfig file by running: tsc --init.

Make it a Node.js project by giving the following command:

```
npm init -y
```

Install types for Node.js

```
npm i @types/node -D
```

Create .gitignore file

Create file app.ts

Transpile JavaScript by running: **tsc**

Run transpiled javascript by running the following command: **node app.js**

JSON

1. JSON stands for Javascript Object Notation.
2. JSON is a text-based data format that is used to store and transfer data.
3. // JSON syntax

   ```
   {
    "name": "Vipin",

    "age": 21,

    "gender": "male",
   }
   ```

But wait, Is JSON is similar to javaScript objects?

The Answer is No.

1. JavaScript objects can contain functions but JSON not.
2. JavaScript objects can only be used in JavaScript but JSON can be created and used by other programming languages.

JSON Data

1. JSON data consists of key/value pairs similar to JavaScript object properties.
2. The key and values are written in double quotes separated by a :.
3. Example :

// JSON data

```
 "name": "Vipin"
```

4. JSON data requires double quotes for the key.

JSON Object

1. The JSON object is written inside curly braces { }.
2. JSON objects can contain multiple key/value pairs.
3. Example :

   ```
   // JSON object

   { "name": "Vipin", "age": 21 }
   ```

JSON Array

1. JSON array is written inside square brackets [ ].

   Example :

   ```
   // JSON array

   [ "Vipin", "Ankit", "Raj"]
   ```

Accessing JSON Data

1. We can access JSON data using the dot notation.
2. Example :

```
// JSON object

    const detail = {"name": "Vipin", "age": 21}

  // accessing JSON object

  console.log (detail. name); // Vipin
```

We can also use square bracket syntax [] to access JSON data.

Example :

```
 // JSON object

 const detail = {

     "name": "Vipin",

     "age": 21
 }
// accessing JSON object
console.log(detail["age"]); // Vipin
```

What is difference between JSON and JSON Object.

Understanding JSON vs. JSON Objects in TypeScript 📝

JSON (JavaScript Object Notation) and JSON objects are fundamental concepts in modern web development, especially when working with TypeScript or JavaScript. Here’s a quick breakdown to clarify their differences:

JSON: A text-based data format used for data interchange, always represented as a string.

Example:

```
{ "key": "value",  "numberKey": 123,  "arrayKey": [1, 2, 3], "objectKey": {  "nestedKey": "nestedValue" }}
```

JSON Object in TypeScript: A JavaScript object resulting from parsing valid JSON data, which can be dynamically accessed and manipulated in TypeScript, often with the help of defined types or interfaces for clarity and type safety.

Example:

```
const jsonString = '{"name":"John","age":30,"city":"New York"}';
const jsonObject = JSON.parse(jsonString);
console.log(jsonObject.name); // Outputs: John
console.log(jsonObject.age); // Outputs: 30
console.log(jsonObject.city); // Outputs: New York
```

Syntax Error

```
lett message = "Hello World";//syntax error (MISTAKE IN KEYWORD)

console.log(message);
```

The Output:

```
app.ts:1:1 - error TS1435: Unknown keyword or identifier. Did you mean 'let'?
```

Type Error

```
let message = "Hello World";

console.loger(message);  /// TYPE MISTAKE IN PROPERTY
```

The Output:

```
app.ts:2:9 - error TS2551: Property 'loger' does not exist on type 'Console'. Did you
mean 'log'?
```

Assignability_Error

```
let message = "Hello World"; // type assign error

message = 6;

console.log(message);
```

The Output:

```
app.ts:2:1 - error TS2322: Type 'number' is not assignable to type 'string'.
```

```
let message = 6;

message = “hello World”;

console.log(message);
```

The Output:

```
app.ts:2:1 - error TS2322: Type 'string' is not assignable to type 'number'.
```