**Typescript**

**What is TypeScript?**

TypeScript is a syntactic superset of JavaScript which adds static typing.

This basically means that TypeScript adds syntax on top of JavaScript, allowing developers to add types.

**Why should I use TypeScript?**

JavaScript is a loosely typed language. It can be difficult to understand what types of data are being passed around in JavaScript.

In JavaScript, function parameters and variables don't have any information! So developers need to look at documentation, or guess based on the implementation.

TypeScript allows specifying the types of data being passed around within the code, and has the ability to report errors when the types don't match.

For example, TypeScript will report an error when passing a string into a function that expects a number. JavaScript will not.

**Is TypeScript better than JavaScript?**

It depends on your needs! Here’s a quick comparison:

✅ **Why TypeScript is better than JavaScript**

1. Static Typing – Helps catch errors early and makes code more predictable.
2. Better IDE Support – Autocompletion, refactoring, and navigation are much smoother.
3. Improved Maintainability – Large projects benefit from strict type checking.
4. Modern Features – Supports the latest JavaScript features while ensuring backward compatibility.
5. Object-Oriented Programming (OOP) Support – More structured code with interfaces, generics, and Enums.

❌ **When JavaScript Might Be Better**

1. Simpler & More Flexible – No need for compilation, so it's faster to set up.
2. Smaller Projects – If you're working on a quick script, TypeScript may be overkill.
3. More Libraries & Frameworks – Some JavaScript libraries don't have great TypeScript support.
4. Learning Curve – TypeScript requires learning types, interfaces, and configurations.

🎯 **Final Verdict**

* Use TypeScript for large-scale applications, enterprise projects, or when you need maintainability.
* Use JavaScript for small projects, quick prototypes, or if you prefer a more flexible coding style.

**TypeScript Data Types**

1. [Number](https://www.tutorialsteacher.com/typescript/typescript-number)
2. [String](https://www.tutorialsteacher.com/typescript/typescript-string)
3. [Boolean](https://www.tutorialsteacher.com/typescript/typescript-boolean)
4. [Array](https://www.tutorialsteacher.com/typescript/typescript-array)
5. [Tuple](https://www.tutorialsteacher.com/typescript/typescript-tuple)
6. [Enum](https://www.tutorialsteacher.com/typescript/typescript-enum)
7. [Union](https://www.tutorialsteacher.com/typescript/typescript-union)
8. [Any](https://www.tutorialsteacher.com/typescript/typescript-any)
9. [Void](https://www.tutorialsteacher.com/typescript/typescript-void)
10. [Never](https://www.tutorialsteacher.com/typescript/typescript-never)

To run typescript programs, we need to install Node.js and Visual Studio Code.

The command to create type script project

npm create vite@latest my-app -- --template react-ts

cd my-app

npm install

npm run dev

<http://localhost:5173/>

Types of Props

1. String of Literals
2. Children
3. Optional Props

The react **intrinsicAttributes** play a vital role in type checking by ensuring that the built-in attributes of JSX elements are correctly typed. This helps in preventing errors and ensuring that components behave as expected.

Chapter 1

* Alternative to JavaScript (Superset)
* Allows us to use strict types
* Supports modern features (arrow, functions, let, constant)
* Extra features (generics,interfaces,tuples)

**To understand typescript, one has to know following things**

1. JavaScript
2. Arrow Functions
3. DOM
4. Classes