Typescript

TypeScript is JavaScript with added syntax for types.

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**.

npx create-react-app myapp typescript

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.

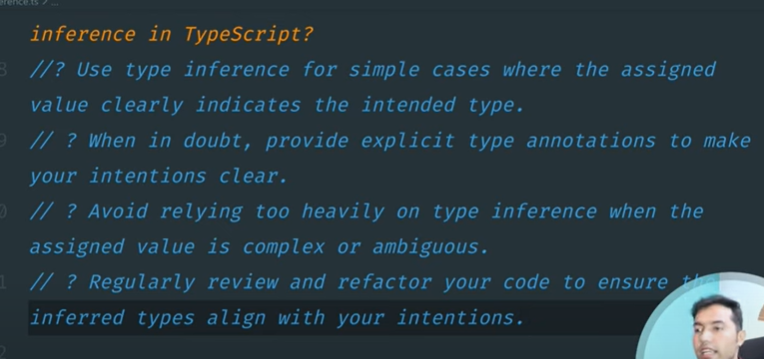
TypeScript uses compile time type checking. Which means it checks if the specified types match **before** running the code, not **while** running the code.

The compiler can be configured using a tsconfig.json file.

npx tsc --init

Type annotations –Explicit – let firstName: string = "Dylan";

Type inference - Implicit- let firstName = "Dylan";

Type inference in typescript refers to the ability of the typescript compiler to automatically determine and assign types to variables, expressions and function return values based on their usage and context in the code 

Special types –

Any - any can be a useful way to get past errors since it disables type checking, but TypeScript will not be able provide type safety, and tools which rely on type data, such as auto completion, will not work. Remember, it should be avoided at "any" cost...

Unknown –

unknown is best used when you don't know the type of data being typed. To add a type later, you'll need to cast it.

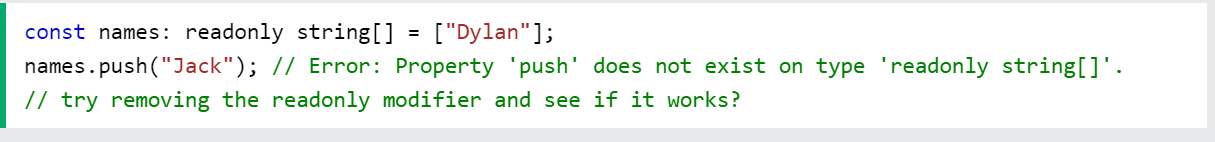
Unknown type is safer alternatives to any because it still enforces type checking and type safety.

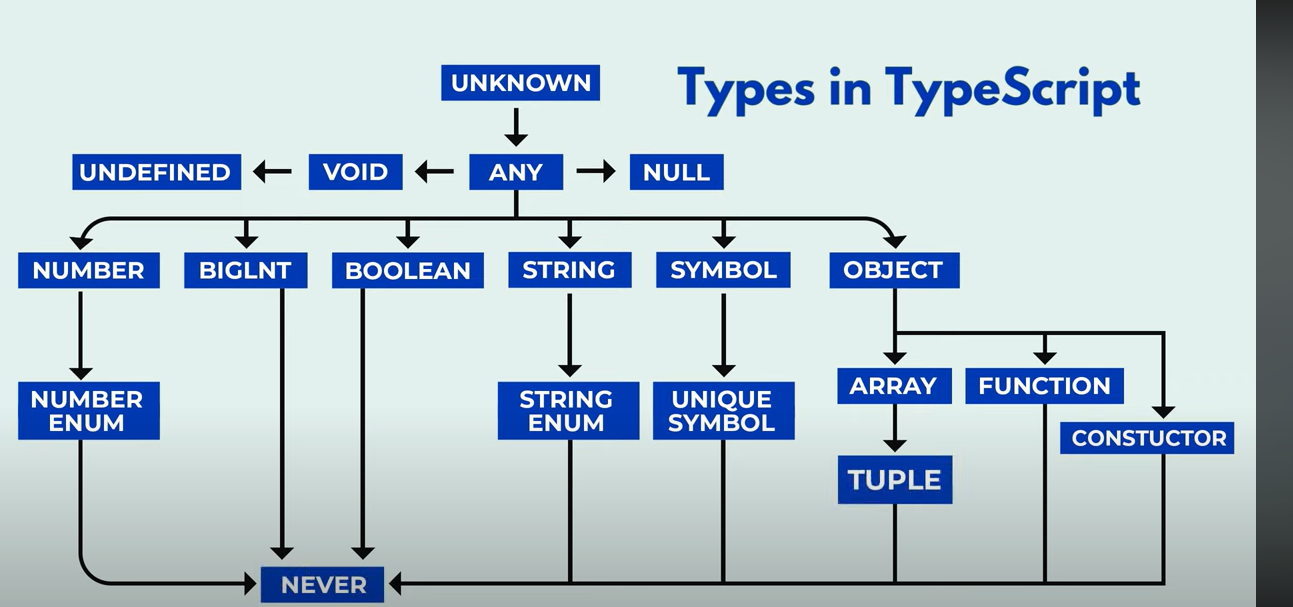
Casting is when we use the "as" keyword to say property or variable is of the casted type.

Undefined and null-

Never-

Array –





Npx create-react-app myapp typescript

File structure –

Index.js === index.tsx

Bydefault type is ‘any’

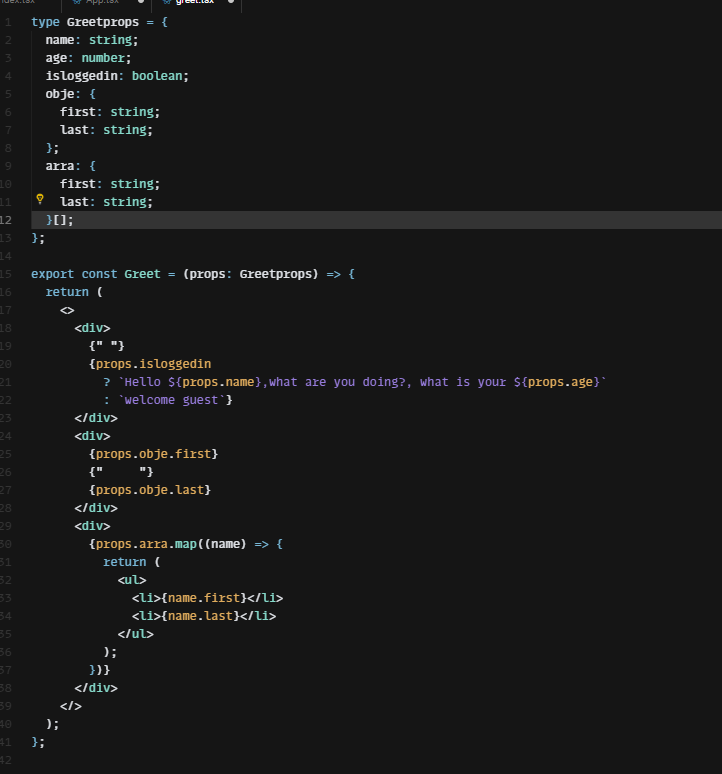
To assign type, there are two ways –

1)implicit – internally

2)explicit -externally like type and inference

Type- when building application

Inference – when building library

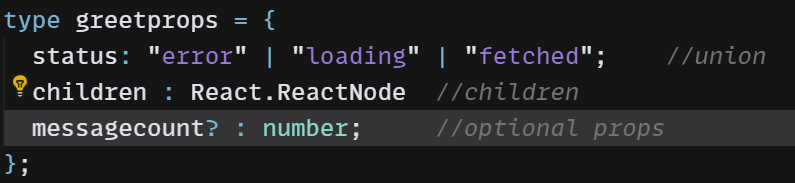


Advanced props –

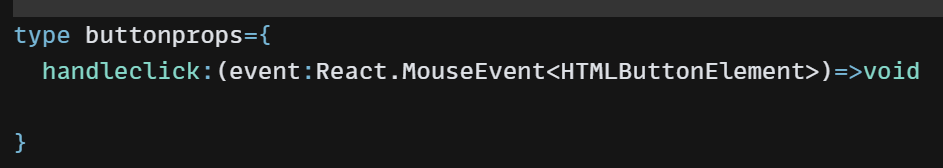
Union of string literal –

Children props –

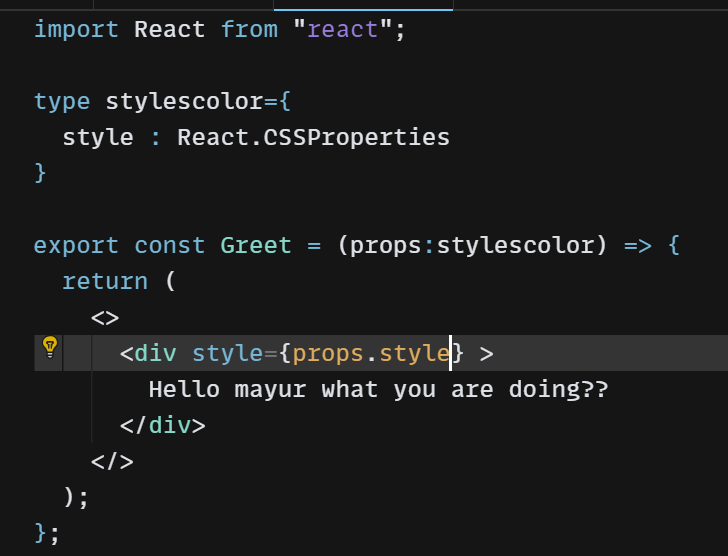
Optional props –



Event props ---

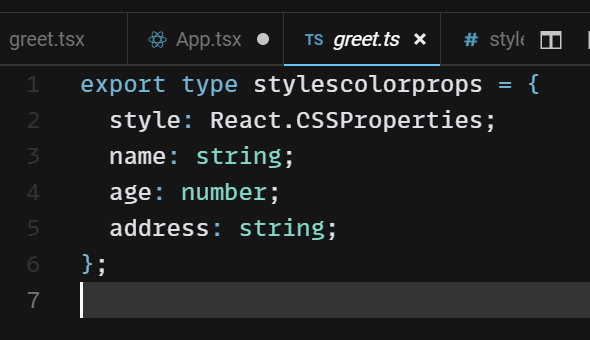
Function - 

Style as props –



Props types and tips –

Index.ts -- file format for types



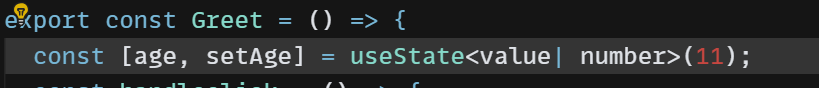
Props destructring –

A screenshot of a computer

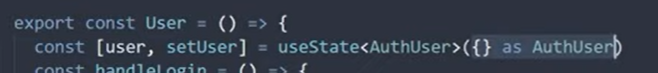
Description automatically generated

Usestate hooks –

If required give types like this --



Usestate type assertion –



Usereducer hook --