

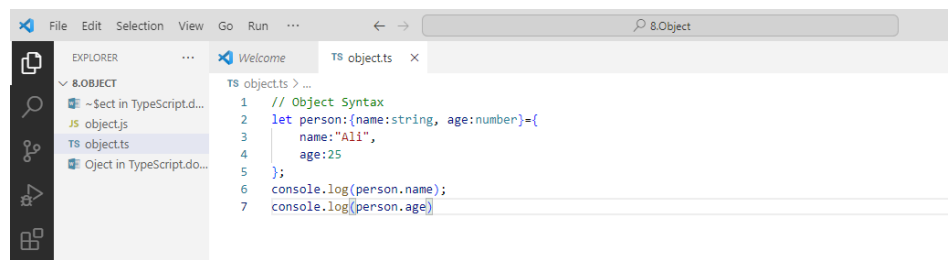
# Object in TypeScript

An object is a fundamental data type that represents a collection of key-value pairs, where keys are strings or symbols, and values can be of any data type, including other objects. Objects in TypeScript are similar to objects in JavaScript, but TypeScript adds static typing to enhance code quality and maintainability.

There are different ways to define objects in TypeScript:

## Object Literal Syntax:

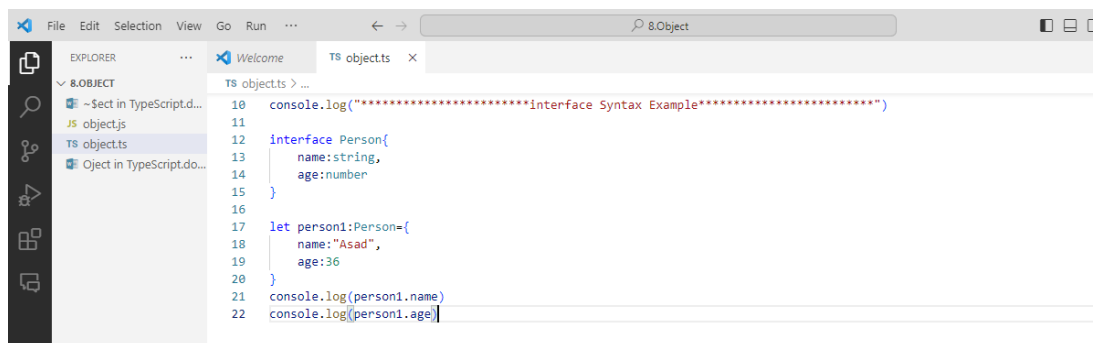
The Object Literal Syntax in TypeScript allows you to define and create objects using a concise and straightforward syntax. It involves specifying key-value pairs within curly braces { }, where keys are typically strings or symbols, and values can be of any data type, including other objects.

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named '8.OBJECT' with files like 'TS objects'. The main editor window is open to 'TS objects.ts' and contains the following TypeScript code:

```
1 // Object Syntax
2 let person:{name:string, age:number}={
3   name:"Ali",
4   age:25
5 };
6 console.log(person.name);
7 console.log(person.age);
```

## Interface:

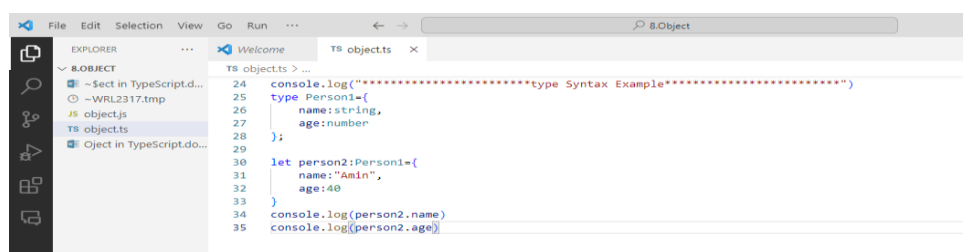
You can use an interface to define the shape of an object and then create objects that adhere to that interface.

A screenshot of the Visual Studio Code editor showing the 'TS objects.ts' file. The code defines an interface and creates an object that conforms to it:

```
10 console.log("*****Interface Syntax Example*****");
11
12 interface Person{
13   name:string,
14   age:number
15 }
16
17 let person1:Person={
18   name:"Asad",
19   age:36
20 }
21 console.log(person1.name)
22 console.log(person1.age);
```

## Type:

Similar to interfaces, you can use the type keyword to define the shape of an object, but assignment operator additionally used before start of curly braces.

A screenshot of the Visual Studio Code editor showing the 'TS objects.ts' file. The code defines a type and creates an object that conforms to it:

```
24 console.log("*****type Syntax Example*****");
25 type Person1={
26   name:string,
27   age:number
28 };
29
30 let person2:Person1={
31   name:"Amin",
32   age:40
33 }
34 console.log(person2.name)
35 console.log(person2.age);
```