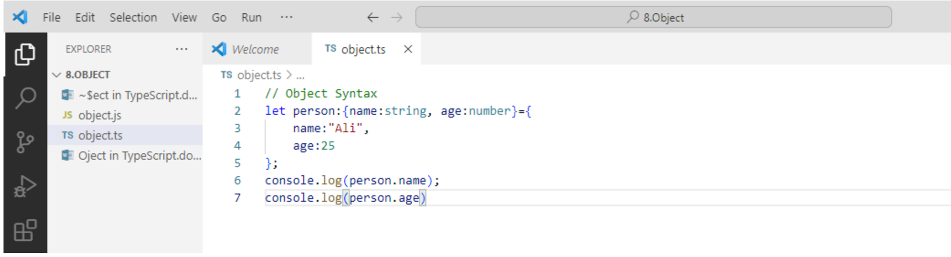
Object in TypeScript

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:

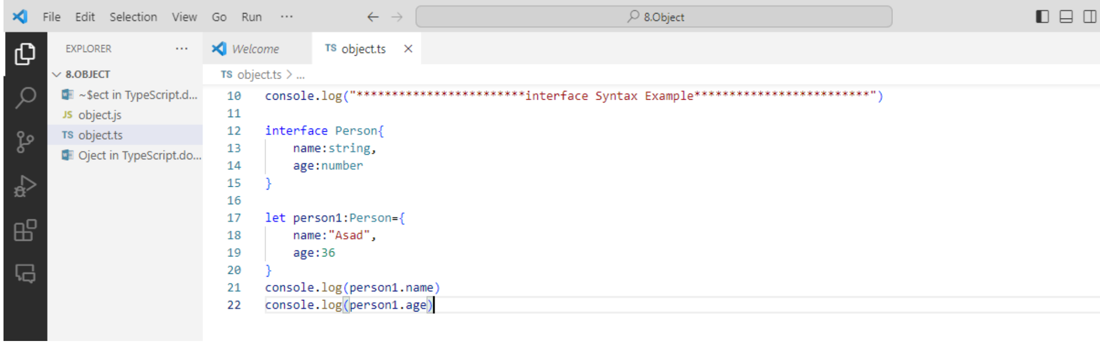
The Object Literal Syntax in TypeScript allows you to define and create objects using a concise and straightforward syntax. It involves specifying key-value pairs within curly braces {}, where keys are typically strings or symbols, and values can be of any data type, including other objects.



```
// Object Syntax
let person:{name:string, age:number}={
    name:"Ali",
    age:25
};
console.log(person.name);
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.



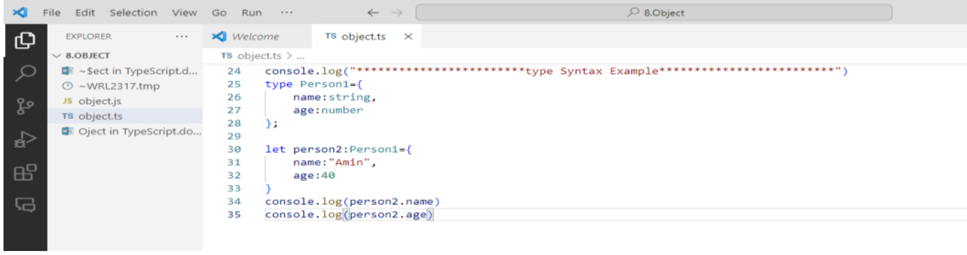
```
console.log("************************interface Syntax Example************************")

interface Person{
    name:string,
    age:number
}

let person1:Person={
    name:"Asad",
    age:36
}
console.log(person1.name)
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.



```
console.log("************************type Syntax Example************************")
type Person1={
    name:string,
    age:number
};

let person2:Person1={
    name:"Amin",
    age:40
}
console.log(person2.name)
console.log(person2.age)
```