

Type Intersection

In TypeScript

Student of GIAIC : Hareem Jaweid (Roll No:00423585)
Onsite Student : Wednesday (7 -10) pm

In TypeScript, an intersection type is used to combine multiple types into one. This means that the resulting type has all the properties and methods of the intersected types. Intersection types are useful when you need a type that combines several other types into a single type. The syntax for creating an intersection type uses the `&` (ampersand) operator.

Student of GIAIC : Hareem Jaweid (Roll No:00423585)

Onsite Student : Wednesday (7 -10) pm

Basic Syntax

The intersection type is created using the & operator.

```
1 type TypeA = {  
2     propertyA: string;  
3 };  
4  
5 type TypeB = {  
6     propertyB: number;  
7 };  
8  
9 type IntersectionType = TypeA & TypeB;
```

In this example,
IntersectionType will
have both propertyA
from TypeA and
propertyB from TypeB.

Student of GIAIC: Hareem Jaweid (Roll No:00423585)

Onsite Student : Wednesday (7 -10) pm

Basic Example

```
1 type Person = {
2     name: string;
3     age: number;
4 };
5
6 type Employee = {
7     employeeId: number;
8     department: string;
9 };
10
11 type EmployeeDetails = Person & Employee;
12
13 const employee: EmployeeDetails = {
14     name: "Hareem",
15     age: 30,
16     employeeId: 6789,
17     department: "HR"
18 };
19
20 console.log(employee);
21
22 //Display at Terminal :
23 // { name: 'Hareem', age: 30, employeeId: 6789, department: 'HR' }
```

Remaining Basic Example:

In this example, `EmployeeDetails` combines the properties of `Person` and `Employee`, so an object of type `EmployeeDetails` must have all four properties: `name`, `age`, `employeeId`, and `department`.

Student of GIAIC : Hareem Jaweid (Roll No:00423585)

Onsite Student : Wednesday (7 -10) pm

Working with Interfaces:

Intersection types can also intersect interfaces:

Here, FlyingCar implements both Drivable and Flyable through the Vehicle type, which is an intersection of these two interfaces.

Student of GIAIC: Hareem Jaweid (Roll No:00423585)

Onsite Student: Wednesday (7 -10) pm

```
1 interface Drivable {  
2     drive(): void;  
3 }  
4 interface Flyable {  
5     fly(): void;  
6 }  
7 type Vehicle = Drivable & Flyable;  
8  
9 class FlyingCar implements Vehicle {  
10     drive() {  
11         console.log("Driving");  
12     }  
13  
14     fly() {  
15         console.log("Flying");  
16     }  
17 }  
18 const myFlyingCar = new FlyingCar();  
19 myFlyingCar.drive();  
20 // Display at Terminal: Driving  
21  
22 myFlyingCar.fly();  
23 // Display at Terminal: Flying  
24  
25 // Display at Terminal: Flying
```

Key Points to Remember:

- **Combining Properties:** The resulting type of an intersection will have all properties from all intersected types.
- **Compatibility:** An object of the intersection type must be compatible with all the intersected types.
- **Structural Typing:** TypeScript uses structural typing, so the resulting type can be any object that satisfies the requirements of all intersected types.
- **Function Types:** Intersection types can combine multiple function types, but the resulting function must be compatible with all the intersected function types.

SUMMARY:

Intersection types are a powerful feature in TypeScript that help in creating flexible and reusable code by combining multiple types into a single, comprehensive type.

Student of GIAIC : Hareem Jaweid (Roll No:00423585)

Onsite Student : Wednesday (7 -10) pm

THANK YOU

Student of GIAIC : Hareem Jaweid (Roll No:00423585)

Onsite Student : Wednesday (7 -10) pm