

## src\type\_interface.ts

```
1 // *****
2 // * INTERFACE VS TYPE
3 // *****
4
5 // 1:
6 //? Use custom types when you need unions, intersections, or mapped types.
7 //? Use interfaces when defining object shapes or classes that adhere to a contract.
8
9 // 2:
10 //? Interfaces can extend other interfaces to inherit their members.
11 //? Custom types can use unions and intersections for more complex type compositions.
12
13 // same Interface name treated as one
14
15 type Stud = {
16   name: string;
17   age: number;
18 };
19
20 type StudAddr = {
21   city: string;
22   state: string;
23 };
24
25 type Data = Stud | StudAddr;
26
27 const BioData: Data = {
28   name: "vinod",
29   city: "Pune",
30   state: "MH",
31 };
32
33 console.log(BioData);
34
35 // SUBSCRIBE TO THAPA TECHNICAL
36
37 interface Stud {
38   name: string;
39   age: number;
40 }
41
42 interface StudAddr {
43   city: string;
44   state: string;
45 }
46
47 interface Data extends Stud, StudAddr {}
48
49 class BioData implements Data {
50   constructor(
51     public name: string,
52     public age: number,
```

```
53     public city: string,  
54     public state: string  
55 ) {}  
56 }  
57  
58 const std1 = new BioData("vinod", 29, "pune", "MH");  
59  
60 console.log(BioData);  
61
```