

[illegible]

```
type PartialUser = Partial<{ id: string; name: string;>;

// Default {}

type Response<Data extends object = { content: string }> = {
  data: Data
  isErr: boolean
}

type BlogResponse = Response<{ title: string, text: string }>

const res: Response = { content: "2233" };

```

types to functions

```
const objectKeys = <T extends object>(obj: T) => {  
  return Object.keys(obj) as Array<keyof T>;  
}
```

```
interface Bar<T> {  
  <T> (args: T): T; // function  
  someProp: T; // normal key  
}
```

```
class GenericNumber<NumType> {  
  zeroValue: NumType;  
  add: (x: NumType, y: NumType) => NumType;  
}  
  
&nbsp;  
let myGenericNumber = new GenericNumber<number>();  
myGenericNumber.zeroValue = 0;  
myGenericNumber.add = function (x, y) {  
  return x + y;  
};
```

[Newer](#)

[Older](#)

2024-02-26

[TypeScript] 03 Skills

2024-02-26

[TypeScript] 01 Type vs Interface

zliangfeng © 2022-2025

Made with [Montaigne](#) and by [anton](#) 