

## 面试题

### 1、类型推论 & 可赋值性

a. 什么是类型推论？

b. 以下代码ts推论出的类型是什么？

```
let a = 1024;
let b = '1024';
const c = 'apple';
let d = [true, false, true];
let e = { name: 'apple' }
let f = null;
```

c. 是否可互相赋值？

```
// 1
let aa = 1024;
let aa1:number = 1024;
aa = aa1; // ??
aa1 = aa; // ??
// 2
let bb:string = 'bb';
let bb1:string | number = 'bb1';
bb = bb1; // ??
bb1 = bb; // ??
bb1 = 2; // ??
bb = bb1; // ??
// 3
let i: 3 = 3;
i = 4; // ??
// 4
let j = [1,2,3];
j.push(4);
j.push('5');
```

d.ts的严格模式和非严格模式下，以下代码的表现差异？

```
var aaa = undefined;
var bbb = null;
let ggg: number = 12;
ggg = aaa;
ggg = bbb;
```

- 2、什么是类型断言？
- 3、type 和 interface的异同
- 4、下面代码块哪些会报错？为什么？

```
// 1
type Options= {
  baseUrl: string
  cacheSize?: number
  env?: 'prod' | 'dev'
}
// 2
class API {
  constructor(options: Options){}
}
// 3
new API({
  baseUrl: 'http://myapi.site.com',
  env: 'prod'
})
// 4
new API({
  baseUrl: 'http://myapi.site.com',
  badEnv: 'prod'
})
// 5
new API({
  baseUrl: 'http://myapi.site.com',
  badEnv: 'prod'
} as Options)
// 6
let badOptions = {
  baseUrl: 'http://myapi.site.com',
  badEnv: 'prod'
}
new API({badOptions})
// 7
let options: Options = {
  baseUrl: 'http://myapi.site.com',
  badEnv: 'prod'
}
new API({badOptions})
```

- 5、有哪些类型装饰器？装饰器作用？执行的顺序是怎样的？
- 6、接口类型有哪些种类