

## МАССИВЫ В TYPESCRIPT

Массивы в TS работают точно так же, как и в классическом JS и цель их использования такая же. Мы лишь можем указывать, какие **типы** будут в нем и какая **структура** массива у нас должна быть

```
1 const departments: string[] = ["dev", "design", "marketing"];  
2 const nums: number[] = [4, 5, 6]
```

Запись `string[]` , `number[]`, `{}[]` и тп. - это указание того, что массив состоит из типов, указанных перед `[]`

А если указаны двойные квадратные скобки `[] []` - это массив массивов с определенными данными:

```
1 const nums: number[][] = [  
2     [3, 5, 6],  
3     [3, 5, 6],  
4 ];
```

Для создания массивов с разными типами данных внутри используются кортежи (tuples), union type или аннотация `any[]` (плохой вариант)

За счет указания типов внутри массива, TS “знает” чем является каждый его член и дает нам подсказки про недопустимые операции как с самим массивом, так и с его членами



```
1 const departments: string[] = ["dev", "design", "marketing"];
2 departments.push(5);
```

Ошибка! Нельзя “пушить” число в массив, состоящий из строк



```
1 const departments: string[] = ["dev", "design", "marketing"];
2 const department = departments[0];
3 department.parseInt();
```

Ошибка! Первый элемент - это строка, у неё не существует такого метода

Во время использования методов массива TS определяет **тип аргумента внутри callback-функций**. Это позволяет внутри тела функции применять только допустимые операции.

**Tip!** Лучше указывать аннотации аргументов самостоятельно, это поможет поддерживать код при будущих изменениях без ошибок



```
1 const departments: string[] = ["dev", "design", "marketing"];
2 const report = departments
3   .filter((d: string) => d !== "dev")
4   .map((d: string) => `${d} - done`);
```

Аргумент d автоматически определен как строка, но мы дополнительно указали тип вручную



```
1 const departments: string[] = ["dev", "design", "marketing"];
2 const report = departments
3   .filter((d) => d !== "dev")
4   .map((d) => {
5     return 4;
6   })
7   .map((d) => `${d + 10} - done`);
```

Без указания типов, на 7й строке TS определит d как число, ведь в 6й мы вернули массив чисел. И операция может выполняться не так, как нужно вам. Подсказки от TS не будет в данном случае!