

INDEXED ACCESS TYPES

Для получения **типа значения** в определенном свойстве используется прием **Indexed Access Types** (дословно: получение типа по индексу):

```
1 interface ICompany {
2     name: string;
3     debts: number;
4 }
5
6 type CompanyDebtsType = ICompany["debts"]; // number
```

При использовании будем получать подсказки о доступных свойствах. Можно применять на объектных типах, в том числе и для получения значений **вложенных свойств**:

```
1 interface ICompany {
2     name: string;
3     debts: number;
4     management: {
5         owner: string;
6     };
7 }
8
9 type CompanyOwnerType = ICompany["management"]["owner"]; // string
```

Получение **одного** конкретного типа из массива типов:

```
1 interface ICompany {
2     name: string;
3     debts: number;
4     departments: Department[];
5 }
6
7 type CompanyDepartmentsType = ICompany["departments"][number]; // Department
```

Для использования переменной при получении типа используется оператор **typeof**. Алгоритм такой же: получаем **тип** из переменной и применяем его:

```
1 interface ICompany {
2     name: string;
3     debts: number;
4 }
5
6 const debts = "debts";
7 type CompanyDebtsType = ICompany[typeof debts]; // number
```

Если переменная объявлена как **let**, то её сначала нужно привести к литеральному типу:

```
1 let debts: "debts" = "debts";
2 // Или
3 let debts = "debts" as "debts";
```