- **Модуль:** Typescript. Базовые знания
- **Урок:** Union (Объединение)

TypeScript Handbook
from
Иван Петриченко
udemy

МЕХАНИЗМ ОБЪЕДИНЕНИЯ ТИПОВ

Union тип - это механизм, который позволяет объединить несколько типов, чтобы дословно сказать: эта сущность может быть или этим типом, или этим. Для этого используется специальный оператор:

```
1  const message: string | number = 5;
2  const messages: string[] | number[] = ["a", "b"];
```

Обычно мы стараемся сделать так, чтобы в массивах был только один тип данных. Но при помощи union типа мы можем объявить, что в нем есть разные типы:

```
1  const data: (string | number)[] = ['name', 25];
```

Такой синтаксис и говорит о том, что в массиве разные типы данных. Но старайтесь избегать таких ситуаций. При переборе и выполнении определенных действий с элементами разных типов можно получить ошибки. Лучше, чтобы массив был однородным

Одно из самых частых применений - это аннотация аргументов функций:

```
1  function printMsg(msg: string | number): void {
2      console.log(msg);
3  }
4  printMsg(4);
5  printMsg("string");
```

Но нужно быть аккуратными при манипуляциях с этими аргументами внутри тела функции. Вы можете использовать только операции, доступные со всеми типами в union типе. Иначе будет ошибка:

```
1  function printMsg(msg: string | number): void {
2      console.log(msg.toLowerCase());
3  }
```

Метод **toLowerCase()** существует только на строке. Если в функцию попадет число, то, будет ошибка, о чем вас предупредит TS. Для работы с такими ситуациями необходимо использовать механизм **сужения типов** (narrowing, следующий урок в курсе)