- **Модуль:** Typescript. Базовые знания
- **Урок:** Сужение типов (Narrowing)

TypeScript Handbook
from
Иван Петриченко
udemy

МЕХАНИЗМ СУЖЕНИЯ ТИПОВ (NARROWING)

При использовании union типа в аргументах функции может возникнуть ситуация, когда мы хотим выполнить разные операции в зависимости от типа данных. Для этого нам нужно **“сузить типы”**, в простейшем варианте с помощью оператора typeof:

```
function printMsg(msg: string | number): void {
    if (typeof msg === 'string') {
        console.log(msg.toLowerCase());
    } else {
        console.log(msg.toExponential());
    }
    console.log(msg)
}
```

Внутри первой ветки условия msg будет определяться как строка, во второй - как число. А значит им будут доступны разные методы.
Вне условия переменная все так же будет union типом

Операций для сужения типов существуют немало, например:

```
function printMsg(msg: string | number | boolean): void {
    if (typeof msg === 'string' || typeof msg === 'number') {
        console.log(msg.toString());
    } else {
        console.log(msg);
    }
    console.log(msg)
}
```

Комбинация операторов typeof

```ts
function printMsg(msg: string[] | number | boolean): void {
    if (Array.isArray(msg)) {
        msg.forEach((m) => console.log(m));
    } else if (typeof msg === "number") {
        console.log(msg.toFixed());
    } else {
        console.log(msg);
    }
}
```

Метод Array.isArray() позволяет определить массив

```ts
const printReadings = (a: number | string, b: number | boolean) => {
    if (a === b) {
        console.log(a, b);
    }
};
```

Можно применять равенство по значениям и типам

```ts
function checkReadings(readings: { system: number } | { user: number }): void {
    if ("system" in readings) {
        console.log(readings.system);
    } else {
        console.log(readings.user);
    }
}
```

Оператор in позволит определить, существует ли свойство в объекте

```ts
function logValue(x: string | Date) {
    if (x instanceof Date) {
        console.log(x.getDate());
    } else {
        console.log(x.trim());
    }
}
```

Оператор instanceof позволит определить, является ли аргумент экземпляром

```ts
const printReadings2 = (a: number[] | string) => {
    console.log(a.slice(0, 3));
};
```

А иногда нужная операция доступна сразу на разных типах