- **Модуль:** Typescript. Базовые знания
- **Урок:** Тип never

TypeScript Handbook
from
Иван Петриченко
udemy

ТИП NEVER

Существуют функции, которые **никогда не заканчиваются возвращаемым значением**: они могут специально вернуть ошибку или “зависнуть”. В таком случае функция возвращает специальный тип `never`

Например:

```ts
const createError = (msg: string) => {
    throw new Error(msg);
};
```

Функция вернет ошибку и никогда не дойдет до return
Но если поместить ошибку в условие, то тип уже будет не never

```ts
const createNever = () => {
    while (true) {

    }
};
```

Функция “зависнет” на бесконечном цикле и ничего не вернет.
Тоже самое и с бесконечной рекурсией.
Использовать нет смысла, так как скрипт просто зависнет, но знать желательно

Этот тип можно использовать для “исчерпывающей проверки”, когда мы хотим сказать коду, что в каком-то случае функция ничего возвращать не будет. Почти всегда это разветвленные условия:

```ts
function logBrtMsg(isBirthday: boolean, userName: string, age: number): string {
    if (isBirthday === true) {
        return `Congrats ${userName.toUpperCase()}, age: ${age + 1}`;
    } else if (isBirthday === false) {
        return "Too bad";
    }
    return createError("Error");
}
```

Последняя ветка в условии else не прописана, а значит там может быть undefined
Но если мы вернем там never, то все будет в порядке, ведь в рантайме появится ошибка
Но обычно условия стараются прописать полностью 🙂