- **Модуль:** Typescript. Необходимый уровень
- **Урок:** Оператор Non-Null and Non-Undefined

TypeScript Handbook
from
Иван Петриченко
udemy

ЕСЛИ МЫ ТОЧНО ЗНАЕМ, ЧТО ОНО СУЩЕСТВУЕТ

Для указания того, что сущность точно существует мы используем оператор **Non-Null and Non-Undefined** - "!"
Даже если TS будет предупреждать вас об ошибке, то этот оператор отключит это поведение. Использовать его стоит только тогда, когда вы на все 100% уверены в наличии сущности. Иначе будут ошибки в рантайме

```ts
interface User {
    login: string;
    addr: string | undefined;
    parents?: {
        mother?: string;
        father?: string;
    };
}

function sendUserData(obj: User, db?: string): void {
    console.log(obj.parents!.father?.toLowerCase(), db!.toLowerCase());
}
```

Мы говорим коду, что свойство parents и аргумент db точно будут.
А значит к ним можно применить дальнейшие операции.

Иногда этот оператор позволяет подкорректировать логику TS и сказать, что функция точно будет синхронной. И другие ситуации, когда TS не уверен в существовании чего-либо на момент использования:

```ts
let dbName: string;
sendUserData(user, "ervervefvf");

console.log(dbName!);

function sendUserData(obj: User, db?: string): void {
    dbName = "12345";
    console.log(obj.parents!.father?.toLowerCase(), db!.toLowerCase());
}
```

В 4й строке TS не знает о том, что значение переменной уже назначено,
поэтому показывает ошибку. Мы можем это исправить,
так как знаем, что эта функция работает синхронно

В литературе прием выше называется Definite Assignment Assertion или Утверждение определенного назначения