- **Модуль:** Typescript. Необходимый уровень
- **Урок:** Модификаторы свойств: optional (Property Modifiers)

TypeScript Handbook
from
Иван Петриченко
udemy

НЕОБЯЗАТЕЛЬНЫЕ СВОЙСТВА И АРГУМЕНТЫ

Для указания того, что свойство в объекте или аргумент в функции может быть необязательным, используется оператор optional - "?"
В объекте ставим его после названия свойства:

```ts
interface User {
    login: string;
    password: string;
    age?: number;
    addr?: string;
}
```

```ts
interface User {
    login: string;
    password: string;
    age: number;
    parents?: {
        mother?: string;
        father?: string;
    };
}
```

Бывают ситуации, когда объекты должны соблюдать строгую форму, даже если значения у свойства нет. Свойство должно быть всегда. В таком случае нам поможет **union тип**, а не optional operator:

```ts
interface User {
    login: string;
    password: string;
    age: number;
    addr: string | undefined;
}
```

В функциях optional operator устанавливается после названия аргумента:

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

Теперь при использовании такого аргумента внутри тела функции используется optional chaining operator (ES2020)

Он позволяет сделать запрос к свойству или методу объекта и если его нет, то просто вернуть undefined. Это позволяет избегать ошибок