- **Модуль:** Typescript. Базовые знания
- **Урок:** Объектные литералы и аннотации функций

TypeScript Handbook
from
Иван Петриченко
udemy

ОБЪЕКТНЫЕ ЛИТЕРАЛЫ

Объектные литералы почти не используются в реальной практике, но в теории их знать полезно. Так мы будем чуть глубже понимать TS и почему иногда возникают ошибки

Если в коде есть объект, который используется в дальнейшем, то он может не подходить по условиям, которые программисту кажутся очевидными на первый взгляд:

```ts
const serverConfig = {
    protocol: "https",
    port: 3001,
};

const startServer = (
    protocol: "http" | "https",
    port: 3000 | 3001
): "Server started" => {
    console.log(`Server started on ${protocol}://server:${port}`);

    return "Server started";
};

startServer(serverConfig.protocol, serverConfig.port);
```

TS покажет ошибку, ведь в **serverConfig.protocol** или **serverConfig.port** может лежать все, что угодно. А функция принимает только определенные значения на вход

Для указания того, какой вид должен быть у объекта и из чего он может состоять, используются объектные литералы:

```ts
const serverConfig: { protocol: "http" | "https"; port: 3000 | 3001 } = {
    protocol: "https",
    port: 3001,
};
```

Ошибка уходит, но использовать такой синтаксис не очень приятно. Для удобства существуют специальные конструкции type и interface из следующих уроков

ОБЪЕКТНЫЕ ЛИТЕРАЛЫ ФУНКЦИЙ

Подобные литералы можно применять в качестве аннотаций и в функциях. (Функция - это объект в JS). Таким образом мы опишем какие аргументы принимает функция, в каком виде и что она может вернуть:

```typescript
const startServer: (protocol: "http" | "https", port: 3000 | 3001) => string = (
    protocol: "http" | "https",
    port: 3000 | 3001
): "Server started" => {
    console.log(`Server started on ${protocol}://server:${port}`);

    return "Server started";
};
```

Самое главное - это отличить аннотацию от объявления самой функции, что при таком синтаксисе не всегда легко. Заметьте, в аннотации данного примера мы возвращаем **string**, а у самой функции указано, что она возвращает определенную строку **"Server started"**

Такой код непросто читать, аннотацию невозможно переиспользовать, так что и применение у него нечастое. Но возможность всегда остается