- **Модуль:** Typescript. Классы
- **Урок:** Статичные свойства и методы

TypeScript Handbook
from
Иван Петриченко
udemy

СТАТИЧНЫЕ СВОЙСТВА И МЕТОДЫ

В **современном** стандарте JS уже существует возможность создавать статичные свойства и методы класса. Для этого используется ключевое слово **static**. Это позволяет создать у класса **сущность, общую для каждого экземпляра.** Общая настройка, общий метод, что-то, что в общем характеризует этот класс.

```
class Player {
    static game: string = "COD";
}
const game = Player.game; // "COD"
```

Для имен статичных методов и свойств **не могут** использоваться встроенные, например name. К ним можно применять модификаторы видимости (private, protected). В таком случае, использовать их можно только внутри самих классов в таком формате:

```
class Player {
    private static game: string = "COD";

    static getGameName() { // Make it static too
        return Player.game;
    }
}

const game = Player.getGameName();
```

Классическим примером статичных методов и свойств является класс Math: **Math.random(), Math.ceil(), Math.PI и тд.**
Но в TS **не существует** такого понятия, как статичные классы. Даже если класс состоит только из static сущностей, то мы не можем создать конструкцию **static class Test {}**

Это не имело бы особого смысла, так как в JS больше вариативности. Вместо static свойства вы можете использовать переменную,а вместо метода - функцию, например. Так что пользуйтесь static с умом, когда действительно сущность привязана к классу.

Для их инициализации статичных свойств существуют **статичные блоки.** Они позволяют установить значение только один раз, при первой инициализации объекта

```
function setName() {
    return 'COD';
}

class Player {
    private static game: string;

    static {
        Player.game = setName();
    }
}
```