

## ПРИВАТНЫЕ ПОЛЯ

В **современном** стандарте JS уже существует возможность создавать приватные свойства и методы на уровне языка. Для этого используется символ решетки:



```
1 class Player {
2     #login: string;
3     private _password: string;
4     public server: string;
5     protected consent: boolean;
6 }
7
8 const player = new Player();
9 player.#login // Error
```

Этот функционал останется и **после компиляции**, но в разных вариантах, в зависимости от установленного стандарта. Модификаторы в TS появились намного раньше этой возможности, да и необходимы для совместимости со старыми стандартами

Что использовать: **модификатор private** или **#** - решение за вами, но большинство разработчиков примерно такого мнения:

👉 На **бэкенде** обычно хватает модификатора, так как очень мало кода, который намеренно будет лезть в другие объекты и что-то делать. Более практично контролировать работу самих разработчиков, что и делает TS

👉 На **фронте** есть такие скрипты, которые могут намеренно что-то менять в объектах. Так что имеет смысл самые ценные и уязвимые данные пометить через **#**. Тогда функционал “недоступности” останется и в конечном коде.

**Помните**, что это не скрывает эти данные в прямом смысле этого слова. Они все равно видны в конечном скрипте. Это лишь **ограничивает** функциональный доступ к ним