

Конструкторы. Прототипное наследование

1. Конструкторы, создание объектов через "new"
2. Прототипное наследование
3. F.prototype
4. Встроенные прототипы
5. Методы прототипов, объекты без свойства `__proto__`

Задачи

Кофемашина

базовый

Разработайте функцию-конструктор `CoffeeMachine(name, power)`, которая создает объект для кофеварки. В момент создания объекту задаются свойства-данные, значения которых могут быть переданы как параметры.

- **name** содержит строку с именем машины
- **power** содержит числовое значение. По умолчанию 50.

Стандартные методы для машин содержатся в свойстве `prototype` конструктора:

- **on()** выводит в консоль строку '`<name> power on`'. **on()** можно вызвать один раз в начале выполнения скрипта или после вызова **off()** (также единожды). Некорректные вызовы не должны производить каких-либо действий.
- **off()** выводит в консоль строку '`<name> power off`'. Этот и все последующие методы работают только после вызова **on()** и до вызова **off()**.
- **setPower(power)** устанавливает числовое значение `power` в соответствующее свойство объекта (допустимые значения - целые числа от 0 до 100).
- **makeCoffee()** выводит в консоль строку '`Cup of coffee strength in <power>` (при котором кофе был приготовлен)'. Метод выполняется 2 секунды.

```
const myCoffeeMachine = new CoffeeMachine('Ingrid',
20);

console.log(myCoffeeMachine.name); // Ingrid
console.log(myCoffeeMachine.power); // 20

myCoffeeMachine.on(); // Ingrid power on
myCoffeeMachine.power.setPower(70);

myCoffeeMachine.makeCoffee();
/*(after 2 seconds) Cup of coffee strength in 70*/

myCoffeeMachine.off(); // Ingrid power off
```

Примечание 1. Вместо двух отдельных реализации свойства-данных *power* и метода *setPower()* можно использовать одно **свойство-аксессор** (рекомендовано автором).

Специальная кофемашина

базовый

Разработайте новую функцию-конструктор `specialCoffeeMachine(name, coffeeName, defaultPower)`, которая создает отдельную кофеварку для `coffeeName`. Значение `coffeeName` может быть строкой с латинскими буквами длиной от 3 до 10. Новая машина может делать только один вид кофе. Потому метод `setPower()` больше для неё не доступен, а метод `makeCoffee` выводит в консоль «Cup of <coffeeName>». Во всем остальном она работает так же как и обычная. Для наследования свойств и методов `CoffeeMachine` используйте вложенный вызов конструктора и прототипы.

```
const espressoMachine = new CoffeeMachine('Emma',
    'espresso', 35);

espressoMachine.on(); // Emma power on

espressoMachine.makeCoffee();
/*(after 2 seconds) Cup of <coffeeName>. */

myCoffeeMachine.off(); // Ingrid power off
```

Примечание 2. На самом деле, в предыдущем задании мы создали класс *CoffeeMachine* в функциональном стиле, а в этом расширим его наследующим классом *specialCoffeeMachine*. Обратите внимание на эту терминологию. Классы в синтаксисе ES6 устроены так же, правда, с некоторыми различиями.