**[Изменяем "prototype"](https://learn.javascript.ru/function-prototype" \l "izmenyaem-prototype)**

важность: 5

В коде ниже мы создаём нового кролика new Rabbit, а потом пытаемся изменить его прототип.

Сначала у нас есть такой код:

function Rabbit() {}

Rabbit.prototype = {

eats: true

};

let rabbit = new Rabbit();

alert( rabbit.eats ); // true

1. Добавим одну строчку (выделенную в коде ниже). Что вызов alert покажет нам сейчас?
2. function Rabbit() {}
3. Rabbit.prototype = {
4. eats: true
5. };
6. let rabbit = new Rabbit();
7. Rabbit.prototype = {};

alert( rabbit.eats ); // ?

1. …А если код такой (заменили одну строчку)?
2. function Rabbit() {}
3. Rabbit.prototype = {
4. eats: true
5. };
6. let rabbit = new Rabbit();
7. Rabbit.prototype.eats = false;

alert( rabbit.eats ); // ?

1. Или такой (заменили одну строчку)?
2. function Rabbit() {}
3. Rabbit.prototype = {
4. eats: true
5. };
6. let rabbit = new Rabbit();
7. delete rabbit.eats;

alert( rabbit.eats ); // ?

1. Или, наконец, такой:
2. function Rabbit() {}
3. Rabbit.prototype = {
4. eats: true
5. };
6. let rabbit = new Rabbit();
7. delete Rabbit.prototype.eats;

alert( rabbit.eats ); // ?

Ответ:

1. True
2. False
3. True
4. undefined

### [Создайте новый объект с помощью уже существующего](https://learn.javascript.ru/function-prototype" \l "sozdayte-novyy-obekt-s-pomoschyu-uzhe-suschestvuyuschego)

важность: 5

Представьте, что у нас имеется некий объект obj, созданный функцией-конструктором – мы не знаем какой именно, но хотелось бы создать ещё один объект такого же типа.

Можем ли мы сделать так?

let obj2 = new obj.constructor();

Приведите пример функции-конструктора для объекта obj, с которой такой вызов корректно сработает. И пример функции-конструктора, с которой такой код поведёт себя неправильно.

Ответ: Мы можем использовать такой способ, если мы уверены в том, что свойство "constructor" существующего объекта имеет корректное значение.

Например, если мы не меняли "prototype", используемый по умолчанию, то код ниже, без сомнений, сработает:

function User(name) {

this.name = name;

}

let user = new User('John');

let user2 = new user.constructor('Pete');

alert( user2.name );

Но если кто-то перезапишет User.prototype и забудет заново назначить свойство "constructor", чтобы оно указывало на User, то ничего не выйдет.

Например:

function User(name) {

this.name = name;

}

User.prototype = {}; // (\*)

let user = new User('John');

let user2 = new user.constructor('Pete');

alert( user2.name );