# Геттеры, сеттеры

Есть два типа свойств объекта. Первый тип это **свойства-данные** (data properties). Второй тип свойств - свойства-аксессоры (accessor properties). По своей сути это функции, которые используются для присвоения и получения значения, но во внешнем коде они выглядят как обычные свойства объекта. Свойства-аксессоры представлены методами: «геттер» – для чтения и «сеттер» – для записи. При литеральном объявлении объекта они обозначаются get и set:

let obj = {

get propName() {

// геттер, срабатывает при чтении obj.propName

},

set propName(value) {

// сеттер, срабатывает при записи obj.propName = value

}

};

Геттер срабатывает, когда obj.propName читается, сеттер – когда значение присваивается.

Снаружи свойство-аксессор выглядит как обычное свойство. В этом и заключается смысл свойств-аксессоров.

let user = {

name: "John",

surname: "Smith",

**get** fullName() {

return `${this.name} ${this.surname}`;

},

**set** fullName(value) {

[this.name, this.surname] = value.split(" ");

}

};

// set fullName запустится с данным значением

**user.fullName** = "Alice Cooper";

alert(user.name); // Alice

alert(user.surname); // Cooper

## Дескрипторы свойств доступа

Дескрипторы свойств-аксессоров отличаются от «обычных» свойств-данных. Свойства-аксессоры не имеют **value** и **writable**, но взамен предлагают функции **get** и **set**.То есть, дескриптор аксессора может иметь:

get – функция без аргументов, которая сработает при чтении свойства,

set – функция, принимающая один аргумент, вызываемая при присвоении свойства,

enumerable – то же самое, что и для свойств-данных,

configurable – то же самое, что и для свойств-данных.

let user = {

name: "John",

surname: "Smith"

};

**Object.defineProperty**(user, 'fullName', {

**get**() {

return `${this.name} ${this.surname}`;

},

**set**(value) {

[this.name, this.surname] = value.split(" ");

}

});

alert(user.fullName); // John Smith

for(let key in user) alert(key); // name, surname