

Front-end

JavaScript

JS. Object

- Объекты

JS. Object

Объект — это один из типов данных в JavaScript, который предназначен для хранения коллекции различных значений и более сложных сущностей.

Всего в JavaScript существует 8 типов данных: Number, BigInt, String, Boolean, null, undefined, Symbol и Object.

Но все кроме объекта являются примитивными. При присвоении переменной значения примитивного типа, оно хранится в ней непосредственно

JS. Object

Когда мы хотим одной переменной присвоить значение другой, содержащей примитивный тип, копирование осуществляется по значению (по английски – copy by value)

В JavaScript объект – это **набор свойств «имя: значение»**. При этом имена ещё очень часто называют ключами. При этом значение свойства может содержать что угодно. Если в качестве значения используется функция, то такое свойство называют методом.

Объекты – это ссылочный тип. Поэтому, когда вы присваиваете объект переменной, вы на самом деле присваиваете ей не сам этот объект, а ссылку на него, которая указывает на то место в памяти компьютера, где он находится.

JS. Object

Литеральный синтаксис

Литеральный синтаксис объявления объекта – это один из самых простых способов его создания. Обычно он используется, когда необходимо объявить только один объект.

Литерал объекта записывается с помощью фигурных скобок {...}, внутрь которых помещают свойства, отделенные друг от друга посредством запятой

```
const person = {  
  firstName: 'Иван',  
  lastName: 'Иванович',  
  age: 28,  
  speak: function () {  
    console.log('person can speak');  
  }  
};
```


JS. Object

В этом примере объект содержит четыре свойства: `firstName`, `lastName`, `age` и `speak`. При этом свойство `speak` является методом, так как его значение – это функция.

Новый формат записи методов выполняется без использования ключевого слова `function` и двоеточия:

```
const person = {  
  firstName: 'Иван',  
  lastName: 'Иванович',  
  age: 28,  
  speak () {  
    console.log('person can speak');  
  }  
};
```


JS. Object

Как можно создать объект

`let cap = new Object({name: 'david', lastName: 'Duchovny'});` // => конструктор объекта

`let cap1 = {};` // => литерал объекта

`let cap2 = Object.create({});`

`cap2.name = 'david';`

`cap2.lastName = 'Duchovny'`

JS. Object

Объекты в JavaScript, как и в других языках программирования, легче понять, если провести аналогию с объектами «реального мира». При этом любой объект имеет характеристики (свойства) и может выполнять некоторые задачи (методы).

Например, объект `car` имеет свойства: `color`, `engine` и `model`. Свойства – это характеристики объекта, они обычно описываются с помощью существительных и прилагательных. Также объект имеет собственные методы: `drive`, `park`, `start` и `stop`. Методы – это поведение объекта, его функции, они обычно обозначаются посредством глаголов.

JS. Object

```
const car = {  
  color: 'orange',  
  engine: '3.6L V6',  
  model: 'AL3',  
  drive() {  
    console.log('driving');  
  },  
  park() {  
    console.log('parking');  
  },  
  start() {  
    console.log('starting');  
  },  
  stop() {  
    console.log('stopping');  
  }  
}
```


JS. Object

Свойства

Свойства состоят из ключа и значения. Отделяется ключ от значения посредством двоеточия. Свойства подобны переменным, но в составе объекта.

Обращение к свойствам выполняется через точку или квадратные скобки:

Через точку

```
const firstName = person.firstName;
```

```
const lastName = person.lastName;
```

Через квадратные скобки, ключ в этом случае нужно указывать в виде строки

```
const age = person['age'];
```

```
const getFullName = person['getFullName'];
```


JS. Object

Значения, связанные с соответствующими ключами можно не только получить, но и присвоить им новые значения:

Изменим значения свойств firstName и lastName

```
person.firstName = 'Иван';
```

```
person.lastName = 'Михайлов';
```

А также добавить новые свойства объекту:

Добавим к объекту person свойство middleName со значением 'Сергеевич'

```
person.middleName = 'Сергеевич';
```


JS. Object

Удаление свойств из объекта осуществляется с помощью оператора delete:

Удалим свойство middleName из объекта person

```
delete person.middleName;
```