

Объекты в Java Script

Объекты

Объекты же используются для *хранения коллекций* различных значений и более сложных сущностей.

Объект может быть создан с помощью фигурных скобок {...} с необязательным списком свойств. *Свойство* — это пара «ключ: значение», где *ключ* — это строка (также называемая «именем свойства»), а *значение* может быть чем угодно.

Пустой объект («пустой ящик») можно создать, используя один из двух вариантов синтаксиса:

```
1 let user = new Object(); // синтаксис "конструктор объекта"  
2 let user = {}; // синтаксис "литерал объекта"
```

Обычно используют вариант с фигурными скобками {...}. Такое объявление называют *литералом объекта* или *литеральной нотацией*.

Литералы и свойства

При использовании литерального синтаксиса {...} мы сразу можем поместить в объект несколько свойств в виде пар «ключ: значение»:

```
1 let user = {      // объект
2   name: "John",    // под ключом "name" хранится значение "John"
3   age: 30           // под ключом "age" хранится значение 30
4 };
```

У каждого свойства есть ключ (также называемый «имя» или «идентификатор»). После имени свойства следует двоеточие ":", и затем указывается значение свойства. Если в объекте несколько свойств, то они перечисляются через запятую.

Значение в объекте может *быть любого типа*

Для удаления свойства мы можем использовать оператор
delete



Объект, объявленный как константа, может быть
изменён

Имя свойства может состоять из нескольких слов, но тогда оно должно быть заключено в кавычки.

Для свойств, имена которых состоят из нескольких слов, доступ к значению *«через точку» не работает*

Для таких случаев существует альтернативный способ доступа к свойствам *через квадратные скобки.*

```
1 let user = {};  
2  
3 // присваивание значения свойству  
4 user["likes birds"] = true;  
5  
6 // получение значения свойства  
7 alert(user["likes birds"]); // true  
8  
9 // удаление свойства  
10 delete user["likes birds"];
```

Мы можем использовать квадратные скобки в литеральной нотации для создания *вычисляемого свойства*.

```
1 let fruit = prompt("Какой фрукт купить?", "apple");
2
3 let bag = {
4   [fruit]: 5, // имя свойства будет взято из переменной fruit
5 };
6
7 alert( bag.apple ); // 5, если fruit="apple"
```

Существует специальный *оператор "in"* для проверки существования свойства в объекте.

Синтаксис оператора:

```
1 "key" in object
```

Пример:

```
1 let user = { name: "John", age: 30 };  
2  
3 alert( "age" in user ); // true, user.age существует  
4 alert( "blabla" in user ); // false, user.blabla не существует
```

