JavaScript

Урок №7: Объекты

Автор Бахшиллоев Бехзод
Ментор по Frontend в PRO UNITY

ОБЪЕКТЫ

```
let obj = new Object()
let obj = {}

Свойства объектов:
let obj = {
        name: 'John'
}
obj.name = 'John'

Методы объектов(действия, функции)

let obj = {
        sayName: function(){
                        alert('John')
                    }
}
```

МАССИВЫ

Разновидность объектов с элементами по порядку

```
let arr = ['1', {}, [], 25]
arr[0] == '1'
arr[2] == []
```

Методы массивов:
arr.push('a') - добавляет элемент в конец массива
arr.pop() - удаляет последний элемент из массива и возвращает его
arr.shift() - удаляет из массива первый элемент и возвращает его
arr.unshift('a') - добавляет элемент в начало массива
arr.split(s) - превращает строку в массив, s - разделитель
arr.join(s) - превращает массив в строку, s - разделитель
delete arr[1] - удаляет второй элемент
arr.splice(index, count, elem1...) - удалить count элементов, начиная с index и заменить на элементы elem1...
arr.slice(begin, end) - копирует часть массив с begin до end не включая
arr.sort(fn) - сортировка массива. Если не передать функцию сравнения – сортирует элементы как строки.
arr.reverse() - меняет порядок элементов на обратный.
arr.concat(item1...) - создаёт новый массив, в который копируются элементы из arr, а также item1...

Методы перебора

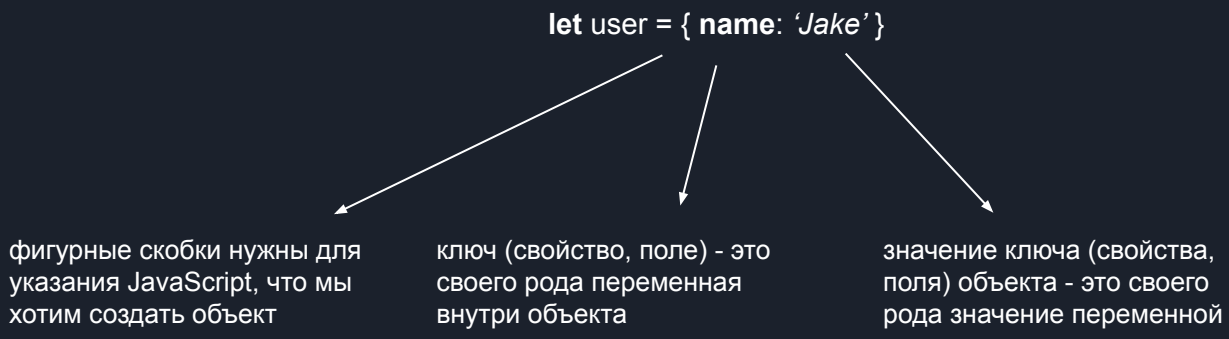
arr.forEach
arr.map
arr.every/some
arr.filter
arr.reduce

Объекты.

Объект - такой тип данных, который может в сочетать в себе несколько значений (примитивных или составных).
Если у нас есть несколько переменных, связанных одним смыслом, мы их помещаем внутрь объекта:

```
let user = {
  name: ‘Vasya’,
  age: 28,
  gender: ‘Male’,
  work: ‘EasyCode’
};
```

Объект - это всегда пара **ключ**: *значение*, помещенные внутрь фигурных скобок {}, несколько пар разделяются запятыми:



Объекты. Чтение. Запись.

let obj = {*name*: ‘Katniss’, *age*: 16}, obj2 = {}, obj3 = {};

Получить (прочитать) свойство объекта можно двумя способами:

1. с помощью точки - объект.имяСвойства: obj.name // ‘Katniss’
2. с помощью квадратных скобок и имени ключа в кавычках: obj[‘name’] // ‘Katniss’

Добавить новое свойство объекту или перезаписать значение существующего поля можно этими же двумя способами:

1. obj3.name = ‘Snow’;
2. obj2[‘name’] = ‘Gale’;

Если мы хотим получить свойство объекта с помощью строковой переменной, мы всегда должны использовать квадратные скобки:

```
let person = {
  name: ‘Nikolay’,
  age: 25
};

let property = ‘name’;
person[property]; // ‘Nikolay’
property = ‘age’;
person[property]; // 25
```

При вычислении переменной property JavaScript подставит
значение этой переменной:
person[‘Nikolay’];

Объекты а внутри другие объекты.

```
let complexObj = {
  name: ‘Peeta’,
  age: 16,
  info: {
    married: false,
    country: ‘District 12’
  }
};

complexObj.info.married // false
complexObj[‘info’][‘married’] // false
complexObj[‘info’].married // false
```

Полезные ссылки: Объекты №1 Объекты №2

Массивы. Тоже объекты.

Массив - это набор данных (переменных), связанных одним смыслом и помещенных внутрь квадратных скобок [ ]. Несколько переменных внутри массива разделяются запятыми.
Внутри массива предпочтительнее использовать данные одного типа (только строки или только числа, или только объекты и т.п.).

let names = [‘Vasya’, ‘Petya’, ‘Slava’];
let products = [{title: ‘Колбаса’, price: 25}, {title: ‘Хлеб’, price: 10}];
let fibonachi = [1, 2, 3, 5, 8, 13, 21];

Для того, чтобы обратиться к определённому элементу массива, нужно рядом с именем массива в квадратных скобках указать номер элемента:

names[0] // ‘Vasya’
names[2] // ‘Slava’

Для того, чтобы получить количество элементов внутри массива (длину или размер массива), нужно воспользоваться свойством массива length:

names.length // 3

Массивы. Примеры.

```
let list = [1,2,3,4];
let collection = [‘one’, ‘two’, ‘three’];
let users = [{name: ‘Smith’, age: 45}, { name: ‘Neo’, age: 25 }];

list.length // 4
collection.length // 3

list[0] // 1
list[1] // 2
list[3] // 4

collection[2] // three

users[0] // {name: ‘Smith’, age: 45}
users[0].name // ‘Smith’
users[1].age // 25
```

Оператор typeof

Оператор **typeof** возвращает строку, указывающую тип операнда, не производя конечных вычислений

typeof ‘string’ || typeof(‘string’) // “string”
typeof 5 || typeof(5) // “number”
typeof false // "boolean"
typeof undefined // "undefined"
typeof null // ooops!.. "object" → an old js mistake
typeof {name: ’Haymitch’} // "object"
typeof function foo(){} // “function”

Задачи:

1. Создать объект с полем **product**, равным *‘iphone’*
2. Добавить в объект поле **price**, равное *1000* и поле **currency**, равное *‘dollar’*
3. Добавить поле **details**, которое будет содержать объект с полями *model* и *color*

Все поля добавлять по очереди, не создавать сразу готовый объект со всеми полями.