HTML - Урок 14

Функции и объекты в JavaScript

Геннадий Караченцев

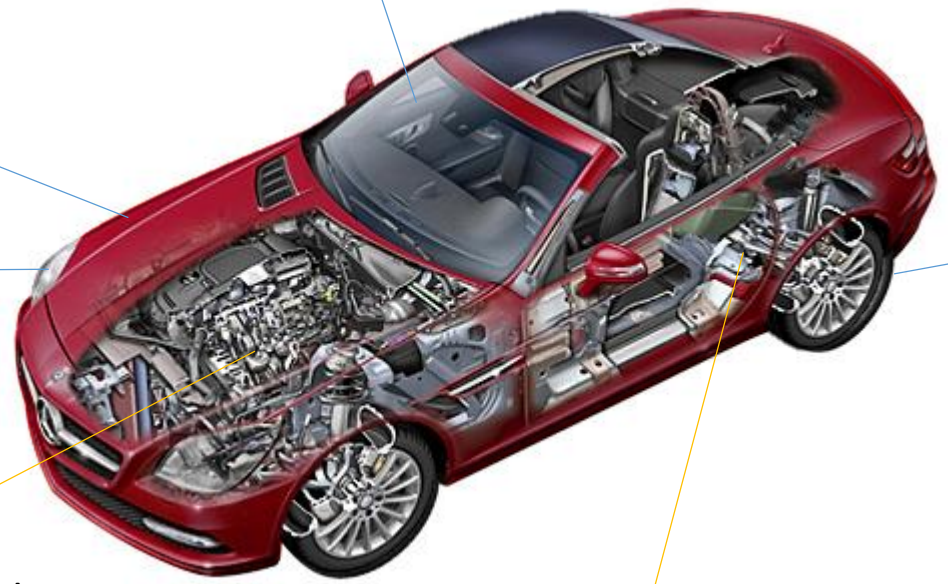
www.weblaboratory.in.ua

Что объекты?

Ассоциативный массив — структура данных, в которой можно хранить любые данные в формате ключ-значение.

По сути, объект это обычный массив, только доступ к элементам объекта, мы можем получить не через интекс(номер) элемента, а по ключу-значению(имени элемента).

Объект простым языком



лобовое стекло : оргстекло

корпус : красный

колеса : резина для шин

цвет шин : черный

фары : ксенон

двигатель : выполняет функцию вращения коленчатого вала

коленчатый вал : выполняет функцию вращения колес

амортизаторы : выполняют функцию смягчения удара

Зачем нужны объекты?

Давайте представим сложную структуру данных, которые относятся к одному типу.

Например кусок программы который содержит переменные, массивы, функции и даже другие объекты.

Все это должно быть в одном месте и простым в доступе.

С этим делом, прекрасно справляются объекты.

Разница структуры данных

Без объекта

```
var itemX = 10;
var itemY = 20;
var string1 = 'Текст 1 строка';
var string2 = 'Текст 2 строка';
var array1 = ["Gena", "Jon", "Stella"];

function getData(x, y){
        return x+y +' '+ array1[0];
}
```

С объектом

```
var objectMain = {
    itemX: 10,
    itemY: 20,
    string1: 'Текст 1 строка',
    string2: 'Текст 2 строка',
    array1: ["Gena", "Jon", "Stella"],
    getData: function(x, y){
        return x+y +' '+ this.array1[0];
    }
}
```

Объекты в JavaScript

Объекты можно создавать самому, добавляя в них ключи дынных.

Также в JavaScript присутствуют уже готовые объекты, которые содержат в себе информацию которую мы можем получить, обратившись к объекту по его имени и через имя обратится к ключу данных, узнав его значение.

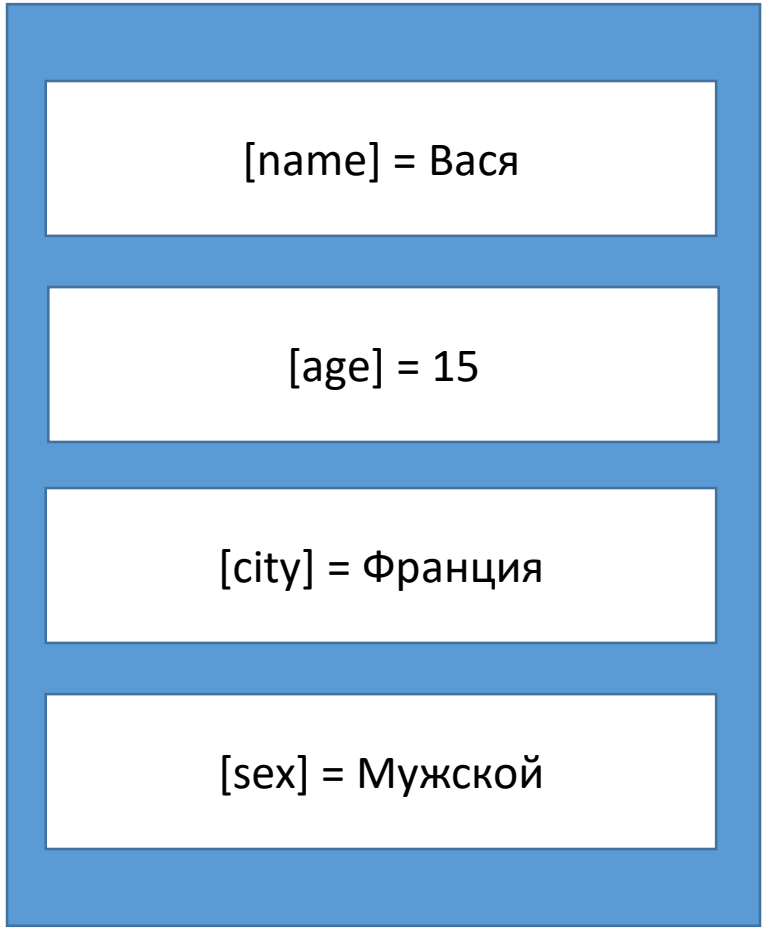
Например в JS есть два основных DOM объекта:

`window` и `document`

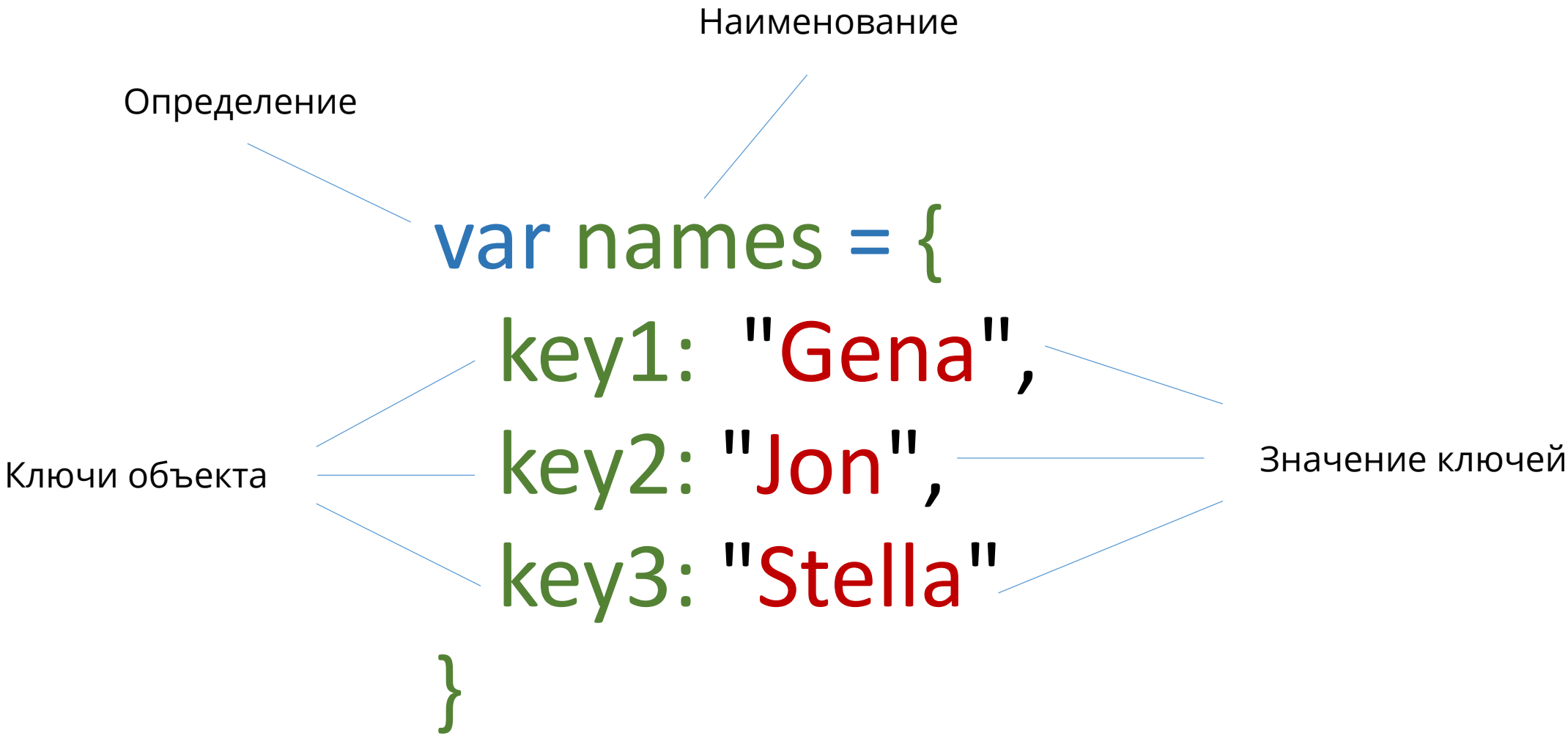
Простым языком

Массив

Объект



Строение объекта



Обращение к данным объекта

```
var names = {
    key1:  "Gena",
    key2: "Jon",
    key3: "Stella"
}
```

Обращение через точку

```
names.key1 // Результат "Gena"
```

Обращение как к элементу массива

```
names['key2'] // Результат "Jon"
```

Создание нового ключа

```
names['newKey'] = 150 // добавить newKey со значением 150
names.newKey    = 150 // добавить newKey со значением 150
```

Удаление ключа

```
delete names['newKey'] // удалит newKey
```

Типы данных внутри объекта

```
var object = {
    name:  "Gena",    // Строка
    number: 1,        // Число
    boolean: false,   // Булева переменная
    function: function(), // Функция
    array: ["Gena", "Jon", "Stella"], // Массив

    // Объект внутри объекта
    object: {
        name:  "Anton",
        number: 2,
        boolean: true
    }
}
```

Перебор объекта

Перебор объекта происходит точно также, как и массива, через цикл for, только немного с другим синтаксисом.

Пример

```
var names = {
    key1:  "Gena",
    key2: "Jon",
    key3: "Stella"
}


for (var key in names){
    console.log(key + ' : ' + names1[key]);
}
```

Функции

Функция — вызов куска программы, который выполняет определённые действия и может возвращать результат выполнения этого действия.

Функция имеет имя, по которому можно ее вызвать.

Также функция может принимать аргументы (данные), которые в последствии могут быть использованы внутри функции.

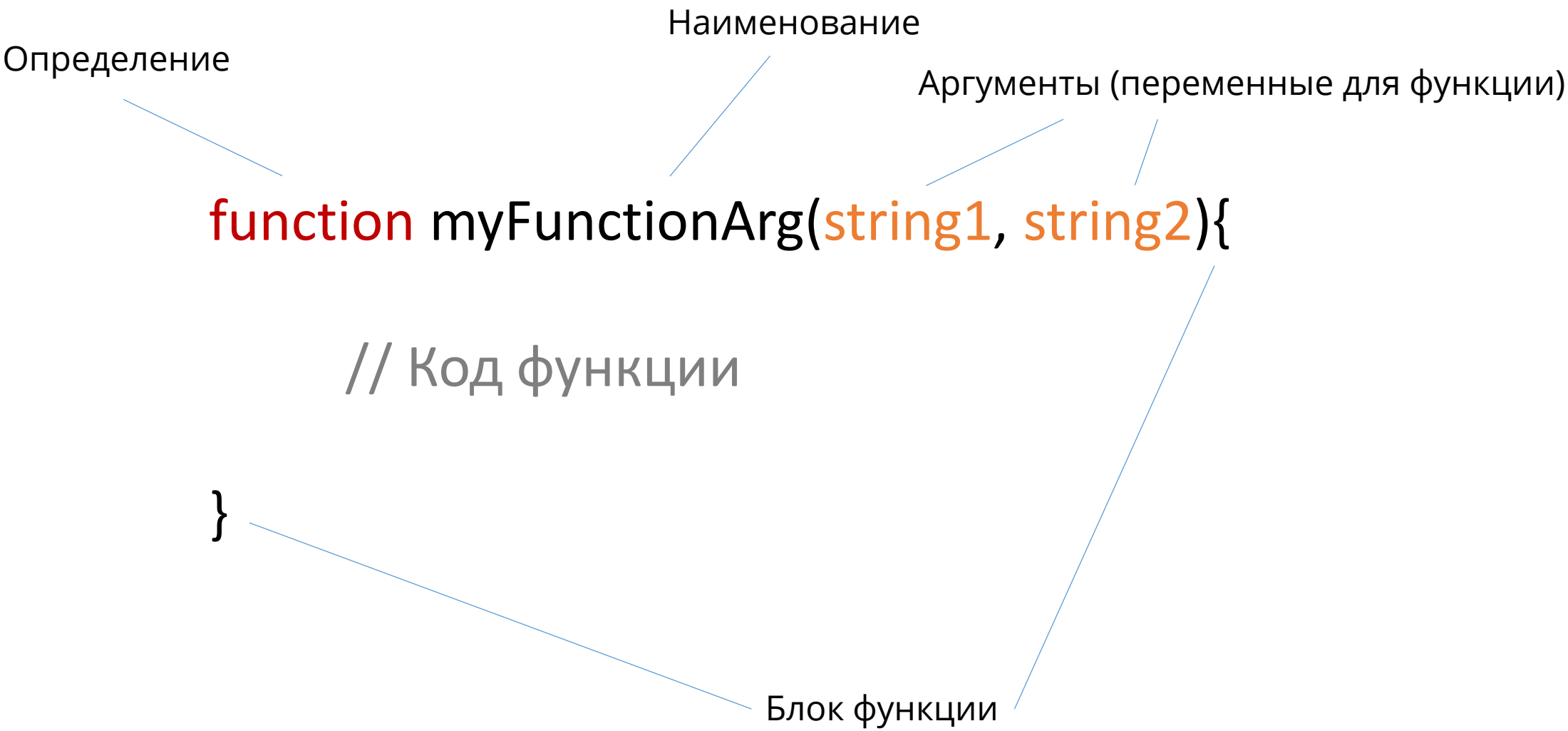
Простая функция

```
function myFirstFunction(){
    console.log('Hello world! I’m function');
}
myFirstFunction();
```

Функция с аргументом

```
function myFunctionArg(string1, string2){
    console.log('Hello' + string1 + '. I’m' + string2 + '!');
}
myFunctionArg('Геннадий', 'JavaScript');
```

Строение функции



Возврат значения

Функция может возвращать значение, после выполнения своего тела функции.
Для этого достаточно использовать команду return

```
function myFunctionArg(x, y){
    return x + y;
}
var y = myFunctionArg(10, 5); // y будет равен 15

function myFunctionCalc(x, y){
   var y = x + y;
   if (y > 10){
        return true;
   }
   return false;
}
```

При выполнении команды return функция заканчивает свою работу.

Функции внутри объекта

Пример

```
var names = {
    text1:  "Gena",
    fn1: function(){
        return 'Привет мир!' ;
    },
    fn2: function(x, y){
        return x + y;
    }
}
names.fn1(); // результатом будет строка 'Привет мир'
names.fn2(5, 10); // результатом будет число 15
```

Переменная this

this переводится с английского как «этот», что полностью характеризует его значение.

Эта переменяет свое значение в зависимости от того, где она была взывана.

Пример

```
var thisVar = this; // вернет объект window


var names = {
    text1:  "Gena",
    key2: "Jon",
    key3: "Stella",
    fn1: function(){
        return this; // вернет объект names
    }
}
```