

Объекты

Что такое массивы?

Индекс	Значение
0	Иван
1	Петр
2	Николай
3	Александр
4	Дмитрий
5	Сергей
...	

Что такое объекты?

Объект – это набор пар «ключ» (название поля) – «значение»

Название поля	Значение
boss	Иван
topManager	Петр
janitor	Николай
economist	Александр
courier	Дмитрий
developer	Сергей
...	

Создание объекта

```
var obj1 = new Object();
```

```
var obj2 = {};
```

}
—
} Равнозначные способы
 создания объектов без
 свойств

Инициализация объекта значениями

```
var obj3 = {field1: "value1", field2: "value2"};
```

```
var cat = {legs: 4,  
          tails: 1,  
          name: "Барсик",  
          age: 3};
```

Обращение к полям объекта (ЧТЕНИЕ)

```
var cat = {legs: 4,  
  tails: 1,  
  name: "Барсик",  
  age: 3};  
  
alert( cat.legs ); // 4  
alert( cat.name ); // Барсик  
alert( cat.age ); // 3  
alert( cat.surname ); // undefined  
alert( cat.weight ); // undefined
```

Обращение к полям объекта (запись)

```
var cat = {legs: 4,  
  tails: 1,  
  name: "Барсик",  
  age: 3};  
alert( cat.legs ); // 4  
cat.legs = 3;  
alert( cat.legs ); // 3  
cat.name = "Машка";  
alert( cat.name ); // Машка  
  
alert( cat.weight ); // undefined  
cat.weight = 2.6;  
alert( cat.weight ); // 2.6
```

Обращение к полям объекта (альтернативный синтаксис)

```
var cat = {legs: 4,  
  tails: 1,  
  name: "Барсик",  
  age: 3};  
  
alert( cat["name"] ); // Барсик  
alert( cat["tails"] ); // 1  
  
cat["age"] = 4;  
alert( cat["age"] ); // 4  
alert( cat.age ); // 4  
alert( cat["age"] === cat.age ); // true
```

Обращение к полям объекта

```
var cat = {legs: 4,  
          tails: 1,  
          name: "Барсик",  
          age: 3};
```

```
var name = "age";
```

```
alert( cat.name ); // Барсик
```

```
alert( cat[name] ); // 3
```

```
alert( cat["name"] ); // Барсик
```

Обход элементов массива

```
var arr = [1, 3, 5, 7, 9, 122, 2424];  
//привычный способ  
for(var i = 0; i < arr.length; i++)  
{  
    alert(arr[i]);  
}  
//новый способ  
for(var i in arr)  
{  
    alert(arr[i]);  
}
```

Обход свойств и значений объекта

```
var cat = {legs: 4,  
  tails: 1,  
  name: "Барсик",  
  age: 3};
```

```
for(var field in cat)  
{  
  alert(field + " : " + cat[field]);  
}  
  
// legs : 4  
// tails : 1  
// name: Барсик  
// age : 3
```

Функции

Объявление функции

```
function имя_функции(аргументы)
{
    тело функции
}
```

Пример

```
function sayHello()
{
    alert("Hello!");
}
function sum(a, b)
{
    return a + b;
}
```

Вызов объявленной функции

```
function sayHello()  
{  
    alert("Hello!");  
}  
function sum(a, b)  
{  
    return a + b;  
}  
  
var a = sum(5, 3);  
alert(a); // 8  
sayHello(); // Hello!
```

Вызов функции с другим количеством аргументов

```
function alertArguments(a,b,c)
{
    alert(a + ", " + b + ", " + c);
}

alertArguments(1,2,3); // 1,2,3
alertArguments(1,2); // 1,2,undefined
alertArguments(1); // 1, undefined,undefined
alertArguments(); // undefined,undefined,undefined

alertArguments(1,2,3,4,5,6,7); //1,2,3
```

Неопределенное количество аргументов

Перед входом в тело функции автоматически создается объект `arguments`, который содержит все аргументы, переданные в функцию

```
function alertAllArguments ()
{
    for(var i = 0; i < arguments.length; i++)
    {
        alert(i + " : " + arguments[i]);
    }
}
alertAllArguments(1, 3, "Hello", true);
// 0 : 1
// 1 : 3
// 2 : Hello
// 3 : true
```

Что такое функция?

- Функции в JavaScript – экземпляры встроенного класса Function
- Следовательно, функции могут передаваться в качестве аргументов других функций, записываться в переменные и т.д. (аналогично любым другим типам данных)

Хранение функции в переменной

```
function alertArguments(a,b,c)
{
    alert(a + ", " + b + ", " + c);
}
```

```
var f = alertArguments;
f(1,2,3); // 1,2,3
alertArguments(1,2,3); // 1,2,3
```

Передача функции в качестве аргумента другой функции

```
function sayHello()
{
    alert("Hello!");
}
function sayGoodbye()
{
    alert("Goodbye!");
}
function callFunction(f)
{
    f();
}
callFunction(sayHello); // Hello!
callFunction(sayGoodbye); // Goodbye!
```

«Анонимные» функции

```
var x = function() {  
    alert("Вызов анонимной функции!");  
};
```

```
x(); // Вызов анонимной функции!
```

```
function callFunction(f)  
{  
    f();  
}
```

```
callFunction(function() {  
    alert( "Вызов анонимной функции, переданной в " +  
    "качестве аргумента" );  
});
```

```
//Вызов анонимной функции, переданной в качестве аргумента
```

Особенности именованных и анонимных функций

```
sayGoodbye(); //ОШИБКА! sayGoodbye - не функция
var sayGoodbye = function()
{
    alert("Goodbye!");
};
```

```
sayHello(); // Hello!
function sayHello()
{
    alert("Hello!");
}
```