

Язык JavaScript является языком сценариев.

В этом учебнике очень кратко будут рассмотрены основные возможности языка JavaScript, но при этом общее понимание и его возможности будут раскрыты.

Учебник написан в 2014 году для проекта <http://snakeproject.ru/>

Все замечания и отзывы Вы можете оставить на сайте послав мне письмо в разделе контакты: <http://snakeproject.ru/contacts.php> , примеры можно например опробовать в интерактивном интерпретаторе по адресу <http://snakeproject.ru/js.php> (не забудьте в этом случае поменять console.log() например на document.write())

1. Где записывается код?

Сам код можно записать в тегах `<script></script>`, либо в отдельном файле.

Пример в тегах:

```
<html>
<head></head>
<body>
<script> document.write(1); </script>
</body>
</html>
```

Пример в отдельном файле(в этом случае в файле script.js теги `<script></script>` писать не стоит):

```
<html>
<head>
<script src='script.js'></script>
</head>
<body>
</body>
</html>
```

2. Комментарии

Существует два типа комментариев:

```
//Однострочный
/* И
многострочный
комментарий */
```

3. Переменные

Имя переменной не может начинаться с цифры.

Пример объявления переменной:

```
var per;
```

Присваивание значения при объявлении:

```
var per = 2;
```

Узнать тип переменной:

```
typeof per;
```

4. Арифметические операторы

Рассмотрим самые распространенные примеры(в данных примерах вывод идет в консоль. Воспользоваться ею можно например в GoogleChrome с помощью ctrl + shift + j):

```
//Сложение
```

```
console.log(1+1);
```

```
//Вычитание
```

```
console.log(1-1);
```

```
//Умножение
```

```
console.log(2*2);
```

```
//Деление
```

```
console.log(2/2);
```

```
//Скобки дают приоритеты выполнения операторов
```

```
console.log(2 + 2 * 2);
```

```
console.log((2 + 2) * 2);
```

5. Строки

Рассмотрим примеры работы со строками:

```
//Создадим некоторые строки
var str1 = "str1tr";
var str2 = "str2tr";

//Склеим
console.log(str1 + str2);

//Выберем кусок строки по позиции с ограниченным количеством символов
console.log(str1.slice(1, 3));

//Выберем кусок строки по позиции до конца строки
console.log(str1.slice(2));

//Разобьем строку по символам в массив
console.log(str1.split(""));

//Разобьем строку по символам в массив с ограниченным количеством символов
console.log(str1.split(", 2));

//Позиция совпадения символов слева
console.log(str1.indexOf('tr'));

//Позиция совпадения символов справа
console.log(str1.lastIndexOf('tr'));
```

6. Преобразования типов

Рассмотрим примеры преобразования типов переменных:

```
var per = '12.3 km';

//Пример с преобразованием строки в число с десятичной частью.
console.log(parseFloat(per));

//Пример с преобразованием строки в целое число.
console.log(parseInt(per));
```

```
var per = 12.3;

//Преобразование числа с десятичной частью в строку
console.log(per.toString());

//Преобразование числа с десятичной частью в строку в шестнадцатеричном формате и
обратно.
console.log(per.toString(16));

per16 = per.toString(16);
console.log(parseInt(per16, 16));
```

7. Циклы

Цикл while(Делай пока условие верно):

```
var i = 0;
while(i < 3)
{
    console.log(++i);
    //i++;
}
```

Отличается от while только тем, что выполнится вне зависимости от условия хотя бы один раз:

```
var i=0;
do{
    console.log(i);
}while(0 != i);
```

Цикл for(стартовая переменная; условие; действие)

```
for(var i=0;i<3;i++)
{
```

```
    console.log(i);  
};
```

Цикл if будет выполняться по условию(в данном примере добавлены еще несколько условий else if и else, они не обязательны):

```
var i = 123;  
if(i == 1)  
{  
    console.log('if - works!');  
}  
else if(i == false)  
{  
    console.log('else if - works!');  
}  
else  
{  
    console.log('else - works!');  
};
```

Другой синтаксис цикла if(первое значение после ? будет в случае истины, второе в случае лжи):

```
i = 2;  
x = (i == 1) ? 'One' : 'Not One';  
console.log(x);
```

Конструкция switch(выполнится case удовлетворяющий условию, далее для остановки используется оператор break, иначе выполнятся все последующие case):

```
i = 1;  
switch(i){
```

```
    case 0:
        console.log(i);
        break;
    case 1:
        console.log(i);
        break;
}
```

8. Функции

Функции – некоторые куски программного кода, которые выполняются при вызове, умеют принимать значения и возвращать некоторые значения с помощью оператора return.

Рассмотрим пример функции с передачей ей двух аргументов:

```
function func(arg1, arg2)
{
    console.log('Func and arg1: '+arg1+ ' and arg2: '+arg2);
    return arg1 + arg2;
}
console.log(func(1, 2));
```

Рассмотрим пример функции с передачей ей одного аргумента, и последующей передачей второго аргумента, который попадет в анонимную функцию, после чего вернется результат:

```
function func(arg1)
{
    return function(arg2){return arg1 + arg2;
};
};
var r = func(1);
```

```
console.log(r(3));
```

Рассмотри пример с использованием ключевого слова `this`, которое указывает на саму функцию:

```
function func(arg1, arg2)
{
    this.arg1 = arg1;
    this.arg2 = arg2;
    this.get = function()
    {
        return [this.arg1, this.arg2];
    };
}
var f1 = new func(1, 2);
var f2 = f1.get();
document.write(f2);
```

9. Объекты

Объекты – это наборы свойств и методов.

Рассмотрим несколько примеров работы с объектами:

```
//Создадим объект с парами ключ:значение
```

```
var x = { name:'Petr', age:31 };
```

```
//Получим значение ключа name
```

```
console.log('name' in x);
```

```
//Переберем значения с помощью for
```

```
for(var per in x){console.log(per + ' ' + x[per])};
```

```
//Удаление возможно с помощью delete
```

```
delete x.age
```

```
//Присваивание делается с помощью =
```

```
x.lastname = 'Liv';
```

```
//Пример с занесенной функцией
```

```
var y = { name: 'Lev', getName: function(){console.log(this.name);}};
```

```
y.getName();
```

```
console.log(y.name);
```

```
//Пример создания пользовательского объекта
```

```
function func(arg)
```

```
{
```

```
  this.arg = arg;
```

```
  this.get = function()
```

```
    {
```

```
      document.write(this.arg);
```

```
    };
```

```
}
```

```
var nFunc = new func('AnyArgument');
```

```
nFunc.get();
```

10. Массивы

```
//Объявление массивов
```

```
arr=[name='Petr', age=31, lastname='Liv', friend='Lev'];
```

```
//Обращение к массивам
```

```
console.log(arr);
```



```
console.log(arr[0]);  
  
//Длина массива  
console.log(arr.length);  
  
//Обращение по индексу  
arr[0] = 'Nikon';  
console.log(arr[0]);  
  
//перебор значений  
for(var per in arr){ console.log(per + ' ' + arr[per]);}  
  
//Получим значения по позициям  
console.log(arr.slice(1, 3));  
  
//Обратное отображение  
console.log(arr.reverse());  
  
//Сортировка  
console.log(arr.sort());  
  
//Удаление справа  
console.log(arr.pop());  
  
//Добавление справа  
console.log(arr.push('new1', 'new2'));  
  
//Удаление слева  
console.log(arr.shift());  
  
//Добавление слева  
console.log(arr.unshift('new3', 'new4'));  
  
//Удалим по позициям  
console.log(arr.splice(2, 2));  
  
//Вставим по позиции  
console.log(arr.splice(2, 0, 'new5'));
```

11. Регулярные выражения

Сами выражения записываются между символами / и /

Рассмотрим несколько примеров

```
//Найдем позицию с совпадением одного из слов
```

```
var str = 'user@local';  
var reg = /local|mail|test/;  
console.log(str.search(reg));
```

```
//Найдем позицию с совпадением экранированного символа
```

```
var str = 'user/local';  
var reg = /\//;  
console.log(str.search(reg));
```

```
//Найдем позицию с совпадением табуляции
```

```
var str = 'user local';  
var reg = /\t/;  
console.log(str.search(reg));
```

```
//Найдем позицию с совпадением с одним из перечисленных символов
```

```
var str = 'user@local';  
var reg = /[sro]/;  
console.log(str.search(reg));
```

```
//Найдем позицию с совпадением из диапазона символов
```

```
var str = 'user@local';  
var reg = /[e-l]/;  
console.log(str.search(reg));
```

```
//Найдем позицию с совпадением одного любого символа
```

```
var str = 'user@local';  
var reg = /.;/;  
console.log(str.search(reg));
```

```
//Найдем позицию с несовпадением ни с одним из перечисленных символов
```

```
var str = 'user@local';
```

```
var reg = /^[^use]/;

console.log(str.search(reg));

//Вернет позицию 8

var str = 'user@loclocal';

var reg = /^[^*e][o].a/;

console.log(str.search(reg));
```

12. Метод замены текста для регулярных выражений

```
var str = 'user@local';

var reg = /local|mail|test/;

document.write(str.replace(reg, 'newDomen'));
```

13. Дата и время

Список чисел в создании даты:

- год – число из 4х или 2х цифр
- номер месяца (нумерация с 0 до 11)
- дата (от 1 до 31)
- часы (от 0 до 23)
- минуты (от 0 до 59)
- секунды (от 0 до 59)
- миллисекунды (от 0 до 999)

Несколько примеров создания:

```
//Самы простой
```

```
var d = new Date();
```

```
document.write(d);
```

```
//Числовой
```

```
var d = new Date(2014, 1, 23, 01, 02, 03);
```

```
document.write(d);
```

```
//Как формат строки Sun Feb 23 2014 01:02:03 GMT+0400 (Московское время (зима))
```

```
var d = new Date('Feb/23/2014');
```

```
document.write(d);
```

```
var d = new Date('Feb/23/2014/01:02:03');  
  
document.write(d);  
  
//А вот примеры методов для работы с датой  
  
var d = new Date();  
d.setFullYear(2014);  
d.setMonth(1);  
d.setDate(23);  
d.setHours(01);  
d.setMinutes(02);  
d.setSeconds(03);  
d.setMilliseconds(123);  
  
document.write(d.getFullYear());  
document.write(d.getMonth());  
document.write(d.getDate());  
document.write(d.getHours());  
document.write(d.getMinutes());  
document.write(d.getSeconds());  
document.write(d.getMilliseconds());  
document.write(d.getDay());
```

14. Выполнение переданного кода

```
eval(document.write('c'))
```