

# JavaScript/jQuery

## Основни оператори:

- декларация на променлива:

```
var foo = 'hello world';  
var foo = 'hello world';
```

- изрази (скобите определят приоритета):

```
a = 2 * 3 + 5; // 11;  
b = 2 * (3 + 5); // 16;
```

- конкатенация:

```
var foo = 'hello';  
var bar = 'world';  
console.log(foo + ' ' + bar); // 'hello world'
```

- умножение и деление:

```
var a = 2 * 3;  
var b = 2 / 3;  
console.log('a= ' + a + ' b=' + b);
```

- инкрементация и декрементация:

```
var i = 1;  
var j = i++; // increment: j = 2;  
  
var m = 2;  
var n = m--; // decrement: n = 1;
```

## Операции с числа и низове:

- събиране / конкатенация:

```
var a = 1;  
var b = '2';  
console.log(a + b); // 12
```

- низ в число:

```
* функция Number()  
var a = 1;  
var b = '2';  
console.log(a + Number(b)); // 3
```

```
* използвайки (+) оператора  
console.log(a + +b); // 3
```

```
* използвайки (*1)  
console.log(a + b*1); // 3
```

## Логически оператори:

- променливи които връщат TRUE:

```
'any string';  
[]; // an empty array  
{}; // an empty object  
1; // any non-zero number
```

- променливи които връщат FALSE:

```
0;  
" "; // an empty string  
NaN; // JavaScript's "not-a-number"  
variable  
null;
```

- логическо AND и OR:

```
var a = 1;  
var b = 0;  
var c = 2;  
  
a || b; // returns 1, which is true  
b || a; // returns 1, which is true  
a && b; // returns 0, which is false  
a && c; // returns 2, which is true  
b && a; // returns 0, which is false
```

## Оператори за сравнение:

```
var a = 1;  
var b = 0;  
var c = '1';  
var d = 2;
```

```
a == b; // returns false  
a != b; // returns true  
a === c; // returns true; careful!  
a === c; // returns false  
a !== c; // returns true  
a === parseInt(c); // returns true  
a > d; // returns false  
d > c; // returns true  
a <= c; // returns true
```

## Условна конструкция

```
if statement  
if...else statement  
if...else if...else statement  
if (условие1) { конструкция 1;}else if (условие2) { конструкция 2; }else{ конструкция 3;}
```

## Конструкцията switch

```
switch(условие){  
    case резултат_от_условието_1:  
        конструкция1;  
        break;  
    case резултат_от_условието_2:  
        конструкция2;  
        break;  
    default:  
        конструкция3;  
        break;  
}
```

## Цикли

### Цикъл FOR

```
for ([initialisation]; [conditional];  
[iteration])  
[loopBody]
```

### Цикъл WHILE

```
while ([conditional]) [loopBody]
```

### Цикъл DO-WHILE

```
do [loopBody] while ([conditional])
```

### Break и continue

- спиране на цикъла – break:

- пропускане на следващата итерация на цикъла – continue.

## Масиви

- достъпване на елемент от масив чрез index:      - промяна на стойността на елемент на масива:

```
var myArray = [ 'hello', 'world', 'foo',  
'bar' ];  
console.log(myArray[3]); // logs 'bar'
```

```
var myArray = [ 'hello', 'world' ];  
myArray[1] = 'changed';
```

- дължина на масив:

```
var myArray = [ 'hello', 'world' ];  
console.log(myArray.length); // logs 2
```

- добавяне на елемент към масив:

```
var myArray = [ 'hello', 'world' ];  
myArray.push('new'); // ["hello", "world",  
"new"]
```

### Операции с масиви:

```
var myArray = [ 'h', 'e', 'l', 'l', 'o' ];  
var myString = myArray.join(""); // 'hello'  
var mySplit = myString.split(""); // [ 'h', 'e', 'l', 'l', 'o' ]
```

### Функции

- декларация на функция:

```
function myFunction() { /* do something  
*/ }  
  
var myFunction = function() { /* do  
something */ }
```

- връщане на стойност от функция:

```
function greet(person, greeting) {  
    var text = greeting + ', ' + person;  
    return text;  
};  
  
console.log(greet('Rebecca', 'hello'));
```

- извикване на функция:

```
function greet(person, greeting) {  
    var text = greeting + ', ' + person;  
    console.log(text);  
};  
  
greet('Rebecca', 'Hello');
```

## Jquery

### jQuery селектори

- клас / id селектор - \$('feature'); \$('#important');
- таг селектор - \$('p'); \$('div');
- наследствен селектор - \$('li strong');
- мултипъл селектор - \$('p, h1');
- атрибут селектор - \$('a[target="\_blank"]'); \$('[href]'); \$('[href="#"]');
- селектори с филтри - \$('div:has(strong)');

селектор this – показва избрания елемент в този манипулатор

```
$('#div').click(function(event){  
    $(this);  
});
```

### jQuery събития

```
$(селектор).on('събитие', function(event){});
```

- \$('div').click() - събитие при „click“ на селектирания елемент;
- \$('div').change() - събитие при промяна на селектирания елемент;
- \$('div').keypress() - отбелязва натискането на който и да е бутон;
- \$('form').submit() - отбелязва „submit“ събитие на избраните елементи;
- \$('div').toggle() - скрива или показва избраните елементи;
- \$('div').hover() - обединява едно или две събития към избрания елемент, които да се изпълнят, когато курсора е върху елемента или го напуска.

### **jQuery манипулиране на HTML**

- `$('#div').html(content);` - задаване на съдържание;
- `$('#div').append(content);` - добавяне в края;
- `$('#div').prepend(content);` - добавяне в началото;
- `$('#div').after(content);` - след елемента;
- `$('#div').before(content);` - преди елемента;
- `$('#h1').text();` - задаване на текст;
- `$('#p').empty();` - изчистване на съдържанието;

### **jQuery манипулиране на CSS**

`$("#p").css("background-color","yellow");`

```
$("#p").css({  
    "background-color":"yellow",  
    "font-size":"200%"  
});
```

`$("#div1").height("200px");`

`$("#div2").width("300px");`

`$('#div').scrollTop(0);`

### **jQuery HTML методи**

`$('#div').addClass('new-class')` - добавяне на клас;

`$('#div').removeClass('old-class')` - изтриване на клас;

`$('#div').hasClass('some-class')` - проверка за наличие на клас;

`$('#input').attr('name') / $('#input').attr('name', 'some-name')` - вземане/задаване на стойност на атрибут;

`$('#a').removeAttr('href')` - изтриване на елемент/атрибут;

`$('#input').val() / $('#input').val('some-val')` - вземане / задаване на стойност на атрибут value.