

— JAVASCRIPT —

Выражения. Операторы. Циклы



— ВЫРАЖЕНИЯ —

фраза языка, которая может быть вычислена интерпретатором для получения значения

ОПЕРАТОР:

УНАРНЫЙ

БИНАРНЫЙ

ТЕРНАРНЫЙ

ВЫРАЖЕНИЯ

фраза языка, которая может быть вычислена интерпретатором для получения значения

ОПЕРАТОР:

УНАРНЫЙ

БИНАРНЫЙ

ТЕРНАРНЫЙ

`var a = 3 + 2; // => 5: сложение`

`a = 3 - 2; // => 1: вычитание`

`a = 3 * 2; // => 6: умножение`

`a = 3 / 2; // => 1.5: деление`

`var b = 3 + "2"; // => "32": складывает числа, объединяет строки`

`var x = 1;`

`x = -x; // -1`

`x = +x; // -1`

`var two = "2", three = "3";`

`two + three; // "23"`

`+two + +three; // 5`

приоритет?

ВЫРАЖЕНИЯ

фраза языка, которая может быть вычислена интерпретатором для получения значения

ОПЕРАТОР:

УНАРНЫЙ

БИНАРНЫЙ

ТЕРНАРНЫЙ

`var a = 3 + 2; // => 5: сложение`

`a = 3 - 2; // => 1: вычитание`

`a = 3 * 2; // => 6: умножение`

`a = 3 / 2; // => 1.5: деление`

`var b = 3 + "2"; // => "32": складывает числа, объединяет строки`

`var x = 1;`

`var two = "2", three = "3";`

`x = -x; // -1`

`two + three; // "23"`

`x = +x; // -1`

`+two + +three; // 5`

`var i = 1;`

`var c = ++i; // 2`

`var i = 1;`

`var d = i++;`

`// d = 1, i = 2`

СТРОКОВОЕ:

`var s1 = String(123); // "123"`

`var s2 = "123" + undefined; // "123undefined"`

`var s3 = true + "test"; // "truetest"`

ЧИСЛЕННОЕ:

`var n1 = Number("123"); // 123`

`var n2 = +"123"; // 123`

`var n3 = +true; // 1`

— ЛОГИЧЕСКИЕ ОПЕРАТОРЫ —

|| (ИЛИ)

если хотя бы один из аргументов
true

```
var a = false || true; // true
```

```
a = false || false; // false
```

```
a = 0 || 1; // 1
```

```
a = true || "неважно что"; // true
```

```
a = undefined || " " || false || 0;  
// 0
```

— ЛОГИЧЕСКИЕ ОПЕРАТОРЫ —

|| (ИЛИ)

если хотя бы один из аргументов
true

```
var a = false || true; // true
```

```
a = false || false; // false
```

```
a = 0 || 1; // 1
```

```
a = true || "неважно что"; // true
```

```
a = undefined || " " || false || 0;  
// 0
```

&& (И)

если оба аргумента истинны

```
var b = true && true; // true
```

```
b = false && true; // false
```

```
b = 0 && 1; // 0
```

```
b = 5 && 0; // 0
```

```
b = 1 && 2 && null && 3; // null
```

```
b = 1 && 2 && 3; // 3
```

— ЛОГИЧЕСКИЕ ОПЕРАТОРЫ —

|| (ИЛИ)

если хотя бы один из аргументов
true

```
var a = false || true; // true
```

```
a = false || false; // false
```

```
a = 0 || 1; // 1
```

```
a = true || "неважно что"; // true
```

```
a = undefined || " " || false || 0;  
// 0
```

&& (И)

если оба аргумента истинны

```
var b = true && true; // true
```

```
b = false && true; // false
```

```
b = 0 && 1; // 0
```

```
b = 5 && 0; // 0
```

```
b = 1 && 2 && null && 3; // null
```

```
b = 1 && 2 && 3; // 3
```

! (НЕ)

Сначала приводит к
true/false,
затем возвращает
противоположное
значение

```
var c = !true; // false
```

```
c = !0; // true
```

— ЛОГИЧЕСКИЕ ОПЕРАТОРЫ —

|| (ИЛИ)

если хотя бы один из аргументов
true

```
var a = false || true; // true
```

```
a = false || false; // false
```

```
a = 0 || 1; // 1
```

```
a = true || "неважно что"; // true
```

```
a = undefined || " " || false || 0;  
// 0
```

&& (И)

если оба аргумента истинны

```
var b = true && true; // true
```

```
b = false && true; // false
```

```
b = 0 && 1; // 0
```

```
b = 5 && 0; // 0
```

```
b = 1 && 2 && null && 3; // null
```

```
b = 1 && 2 && 3; // 3
```

! (НЕ)

Сначала приводит к
true/false,
затем возвращает
противоположное
значение

```
var c = !true; // false
```

```
c = !0; // true
```

ЛОГИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ (!!):

```
var result = !!"Строка"; // true
```

```
var result2 = Boolean("Строка"); // true
```


СРАВНЕНИЕ

> < >= <= == != === !==

true/false

СРАВНЕНИЕ

> < >= <= == != === !==

true/false

СРАВНЕНИЕ РАЗНЫХ ТИПОВ

используется числовое преобразование

```
'2' > 1;  
    // true, сравнивается как 2 > 1
```

```
'01' == 1;  
    // true, сравнивается как 1 == 1
```

```
false == 0;  
    // true, false становится числом 0
```

```
true == 1;  
    // true, так как true становится  
    // числом 1.
```

СРАВНЕНИЕ

> < >= <= == != === !==

true/false

СРАВНЕНИЕ РАЗНЫХ ТИПОВ

используется числовое преобразование

```
'2' > 1;  
    // true, сравнивается как 2 > 1
```

```
'01' == 1;  
    // true, сравнивается как 1 == 1
```

```
false == 0;  
    // true, false становится числом 0
```

```
true == 1;  
    // true, так как true становится  
    // числом 1.
```

! Значения `null` и `undefined` равны `==` друг другу и ничему другому.

СТРОГОЕ РАВЕНСТВО

для проверки равенства без преобразования типов используются операторы строгого равенства `===` (тройное равно) и `!==`

```
0 === false; // false, т.к. типы различны
```

СРАВНЕНИЕ

> < >= <= == != === !==

true/false

СРАВНЕНИЕ РАЗНЫХ ТИПОВ

используется числовое преобразование

```
'2' > 1;  
// true, сравнивается как 2 > 1
```

```
'01' == 1;  
// true, сравнивается как 1 == 1
```

```
false == 0;  
// true, false становится числом 0
```

```
true == 1;  
// true, так как true становится  
//числом 1.
```

! Значения null и undefined равны == друг другу и ничему другому.

СТРОГОЕ РАВЕНСТВО

для проверки равенства без преобразования типов используются операторы строгого равенства === (тройное равно) и !==

```
0 === false; // false, т.к. типы различны
```

УСЛОВНЫЕ ОПЕРАТОРЫ

If, else, if else

?

```
1 if (age < 3) {  
2   message = 'Здравствуй, малыш!';  
3 } else if (age < 18) {  
4   message = 'Привет!';  
5 } else if (age < 100) {  
6   message = 'Здравствуйте!';  
7 } else {  
8   message = 'Какой необычный возраст!';  
9 }
```

```
1 условие ? значение1 : значение2
```

```
var access;  
var age = prompt('Сколько вам лет?', '');  
access = age > 14 ? true : false;
```

ЦИКЛЫ

WHILE

```
1 while (условие) {  
2   // код, тело цикла  
3 }
```

```
1 var i = 0;  
2 while (i < 3) {  
3   alert( i );  
4   i++;  
5 }
```

ЦИКЛЫ

WHILE

```
1 while (условие) {  
2   // код, тело цикла  
3 }
```

```
1 var i = 0;  
2 while (i < 3) {  
3   alert( i );  
4   i++;  
5 }
```

DO...WHILE

```
1 do {  
2   // тело цикла  
3 } while (условие);
```

```
1 var i = 0;  
2 do {  
3   alert( i );  
4   i++;  
5 } while (i < 3);
```

ЦИКЛЫ

WHILE

```
1 while (условие) {  
2   // код, тело цикла  
3 }
```

```
1 var i = 0;  
2 while (i < 3) {  
3   alert( i );  
4   i++;  
5 }
```

FOR

```
1 for (начало; условие; шаг) {  
2   // ... тело цикла ...  
3 }
```

```
1 var i;  
2  
3 for (i = 0; i < 3; i++) {  
4   alert( i );  
5 }
```

DO...WHILE

```
1 do {  
2   // тело цикла  
3 } while (условие);
```

```
1 var i = 0;  
2 do {  
3   alert( i );  
4   i++;  
5 } while (i < 3);
```

break;

continue;

— ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ —

alert()

модальное окно с сообщением и приостанавливает выполнение скрипта, пока пользователь не нажмёт «ОК».

```
alert( "Привет" );
```

prompt()

модальное окно с заголовком *title*, полем для ввода текста, заполненным строкой по умолчанию *default*.

```
var result = prompt(title, default);  
var years = prompt('Сколько вам лет?', 100);
```

confirm()

окно с вопросом *question* с двумя кнопками: ОК и CANCEL.

```
var result = confirm( "Ты здесь?" );
```

console.log()

выводит в консоли значение переменной(х), выражения.

```
console.log( "Привет" );
```


— JAVASCRIPT —

Выражения. Операторы. Циклы

