

Первые шаги в JavaScript

Занятие 2

План занятия

- Операторы
- Операторы сравнения и логические операции
- Условные операторы
- Логические операторы
- Преобразование типов для примитивов

Основные операторы

Операнд

Унарный

Бинарный

Операнд

Унарный

Бинарный

$3 * 5$

Операнд

$3 * 5$

Унарный

```
var x = 2;
```

```
x = -x;  
alert(x);  
// -2
```

Бинарный

Операнд

$3 * 5$

Унарный

```
var x = 2;
```

```
x = -x;  
alert(x);  
// -2
```

Бинарный

```
var x = 1, y = 2;  
alert(y - x);  
// 1
```

Бинарный +

```
var a = 'Hello' + 'GoIt';  
alert(a); // HelloGoIT
```

```
alert('1' + 2); // '12'  
alert(2 + '1'); // '21'
```

```
alert('1' - 2); // '-1'  
alert(6 / '2'); // '3'
```


Унарный +

```
alert(+1); // 1  
alert(+ (1-2)); // -1
```

```
var middle = '2';  
var senior = '3';  
alert(middle + senior); // '23'
```

```
var middle = '2';  
var senior = '3';  
alert(+middle + +senior); // '5'
```

Приоритет

[MDN - Priority](#)

15	Логическое отрицание	справа налево	! ...
	Побитовое отрицание	справа налево	~ ...
	Унарный плюс	справа налево	+ ...
	Унарный минус	справа налево	- ...
	Prefix Increment	справа налево	++ ...
	Prefix Decrement	справа налево	-- ...
	typeof	справа налево	typeof ...
	void	справа налево	void ...
	delete	справа налево	delete ...
14	Умножение	слева направо	... * ...
	Деление	слева направо	... / ...
	Остаток	слева направо	... % ...
13	Сложение	слева направо	... + ...
	Вычитание	слева направо	... - ...
3	Присваивание	справа налево	... = ...
			... += ...
			... -= ...

Присваивание

```
var x = 2 * 2 + 3;  
alert(x); // 7
```

```
var a, b, c;  
a = b = c = 1 + 2;  
alert(a); // 3  
alert(b); // 3  
alert(c); // 3
```

Присваивание

```
var a = 3;
```

```
var b = 2;
```

```
var c = 4 - (a = b + 1);
```

```
alert(a); // 3
```

```
alert(c); // 1
```

Остаток от деления

```
alert(5 % 2); // 1  
alert(8 % 3); // 2  
alert(6 % 3); // 0
```

Инкремент и декремент

```
var i = 2;  
i++;      // i = i + 1.  
alert(i); // 3
```

```
var i = 2;  
i--;      // i = i - 1.  
alert(i); // 1
```

Префикс и постфикс

```
var i = 0;  
alert( ++i ); // 1
```

```
var i = 0;  
alert( i++ ); // 0
```

```
var i = 0;  
i++;  
++i;  
alert( i ); // 2
```

Побитовый операторы

- AND (и) ($\&$)
- OR (или) ($|$)
- XOR (побитовое исключающее или) (\wedge)
- NOT (не) (\sim)
- LEFT SHIFT (левый сдвиг) (\ll)
- RIGHT SHIFT (правый сдвиг) (\gg)
- ZERO-FILL RIGHT SHIFT (правый сдвиг с заполнением нулями) (\ggg)

Сокращенное присваивание

```
var n = 2;  
n = n + 5;  
n = n * 2;
```

```
var n = 2;  
n += 5; // n = 7 (n = n + 5)  
n *= 2; // n = 14 (n = n * 2)  
alert(n); // 14
```

Запятая

```
a = (3, 4);
```

```
alert(a);
```

```
for (a = 1, b = 3, c = a * b; a < 10; a++) {  
    ...  
}
```

Задача

```
var a = 1, b = 1, c, d;
```

```
c = ++a; alert(c);
```

```
d = b++; alert(d);
```

```
c = (2 + ++a); alert(c);
```

```
d = (2 + b++); alert(d);
```

```
alert(a);
```

```
alert(b);
```

Задача

```
var a = 2;
```

```
var x = 1 + (a *= 2);
```

Операторы сравнения и логические значения

Операторы сравнения

- Больше/меньше: $a > b$, $a < b$.
- Больше/меньше или равно: $a \geq b$, $a \leq b$.
- Равно $a == b$.
- «Не равно». В математике он пишется как \neq , в JavaScript — знак равенства с восклицательным знаком перед ним $!=$.

Логические значения

- true
- false

```
alert( 2 > 1 ); // true
alert( 2 == 1 ); // false
alert( 2 != 1 ); // true
```

```
var a = true; // присваивать
var b = 3 > 4; // результат сравнения
alert( b ); // false
alert( a == b ); // (true == false) return false
```

Сравнение строк

```
alert( 'Б' > 'А' ); // true
```


Сравнение строк

```
alert( 'Б' > 'A' ); // true
```

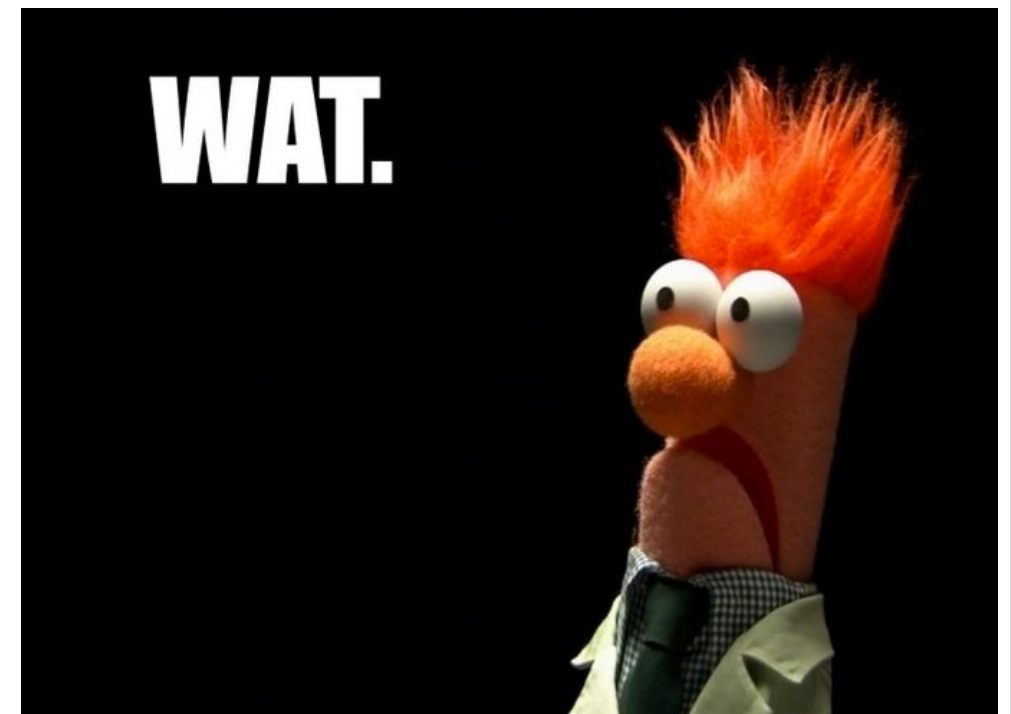


Сравнение строк

```
alert( 'Б' > 'А' ); // true
```



```
alert( 'а' > 'я' ); // true
```



Сравнение строк

```
alert( 'Банан' > 'Аят' ); // true
```

```
alert( 'Вася' > 'Ваня' ); // true
```

```
alert( 'Привет' > 'Прив' ); // true
```

Сравнение строк

```
alert( '2' > '15' ); // true
```

```
alert( +'2' > +'15' ); // false
```

Сравнение странных типов



Сравнение разных типов

```
alert( '2' > 1 ); // true, 2 > 1
alert( '01' == 1 ); // true, 1 == 1
alert( false == 0 ); // true, false -> 0
alert( true == 1 ); // true, true -> 1
```

Строгое равенство

```
alert( 0 == false ); // true
```

```
alert( '' == false ); // true
```

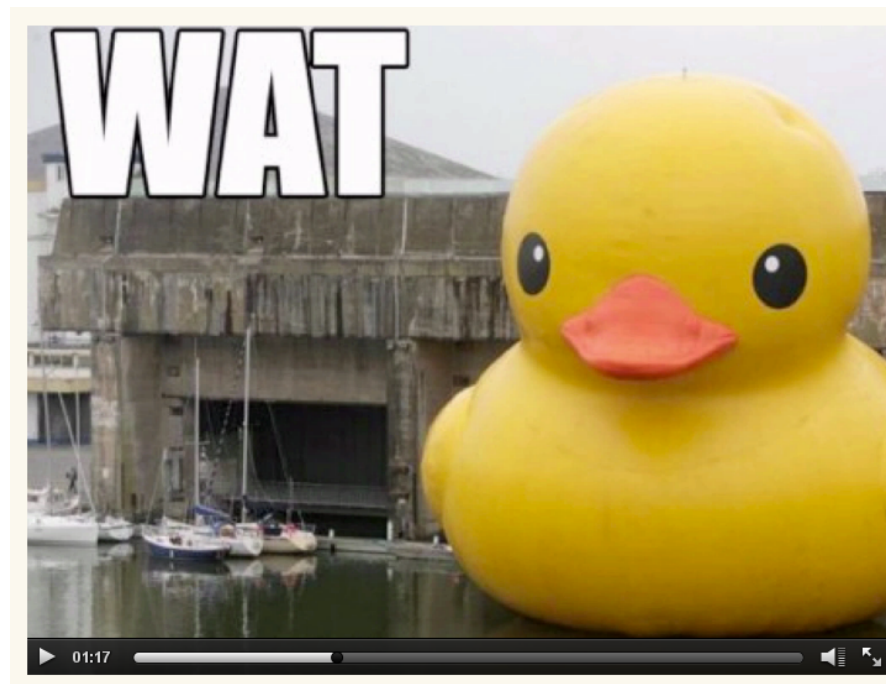
```
alert( 0 === false ); // false
```

Сравнение с null и undefined

```
alert( null > 0 ); // false  
alert( null == 0 ); // false
```


Сравнение с null

```
alert( null > 0 ); // false  
alert( null == 0 ); // false  
  
alert( null >= 0 ); // true
```



Сравнение с undefined

```
alert( undefined > 0 ); // false  
alert( undefined < 0 ); // false  
alert( undefined == 0 ); // false
```

Условные операторы

Оператор if

```
var year = prompt( 'What year today?', '' );
```

```
if (year != 2015) alert( 'Wrooooong!' );
```

```
if (year != 2015) {  
    alert( 'Родителей...' );  
    alert( '...в школу!' );  
}
```

Оператор if

- Число 0, пустая строка "", null и undefined, а также NaN являются false,
- Остальные значения — true.

```
if (0) {  
  
}
```

```
if (1) {  
  
}
```

Оператор if else

```
var year = prompt( 'Введите год появления  
стандарта ECMA-262 5.1', '' );
```

```
if (year == 2011) {  
    alert( 'Да вы знаток!' );  
} else {  
    alert( 'А вот и неправильно!' );  
}
```

Оператор if + multi else

```
var year = prompt( 'В каком году появилась  
спецификация ECMA-262 5.1?', '' );
```

```
if (year < 2011) {  
    alert( 'Это слишком рано..' );  
} else if (year > 2011) {  
    alert( 'Это поздновато..' );  
} else {  
    alert( 'Да, точно в этом году!' );  
}
```

Оператор ?



Оператор ?

условие ? значение1 : значение2

```
var access;  
var age = prompt( 'Сколько вам лет?', '' );  
  
if ( age > 18 ) {  
    access = 'Захады дарагой!';  
} else {  
    access = 'ЭЭЭ! Малэнький ишо!';  
}  
  
alert( access );
```

Оператор ?

условие ? значение1 : значение2

```
var access;  
var age = prompt( 'Сколько вам лет?', '' );  
  
access = ( age > 18 ) ? 'Захады дарагой!' :  
            'ЭЭЭ! Малэнький ишо!';  
  
alert( access );
```

Оператор ?

```
var age = prompt( 'возраст?', 18 );
```

```
var message = ( age < 3 ) ? 'Здравствуй, малыш!' :  
    ( age < 18 ) ? 'Привет!' :  
        ( age < 100 ) ? 'Здравствуйте!' :  
            'Какой необычный возраст!';
```

```
alert( message );
```

Задача

```
if ( '0' ) {  
    alert( 'Привет' );  
}
```

Логические операторы

|| (ИЛИ)

```
var result = a || b;
```

```
alert( true || true ); // true  
alert( false || true ); // true  
alert( true || false ); // true  
alert( false || false ); // false
```

```
var hour = 9;
```

```
if ( hour < 10 || hour > 18 ) {  
    alert( 'Офис до 10 или после 18 закрыт' );  
}
```

|| (ИЛИ)

```
var x;
```

```
true || (x = 1);
```

```
alert(x); // undefined
```

Значение ИЛИ

```
alert( 1 || 0 ); // 1
```

```
alert( true || 'неважно что' ); // true
```

```
alert( null || 1 ); // 1
```

```
alert( undefined || 0 ); // 0
```

```
alert( undefined || '' || false || 0 ); // 0
```


&& (И)

```
var result = a && b;
```

```
alert( true && true ); // true  
alert( false && true ); // false  
alert( true && false ); // false  
alert( false && false ); // false
```

```
if ( 1 && 0 ) { // true && false  
    alert( 'noup!' );  
}
```

&& (И)

*// Первый аргумент – true,
// Поэтому возвращается второй аргумент*

```
alert( 1 && 0 ); // 0
```

```
alert( 1 && 5 ); // 5
```

*// Первый аргумент – false,
// Он и возвращается, а второй аргумент
игнорируется*

```
alert( null && 5 ); // null
```

```
alert( 0 && 'aaaa' ); // 0
```

&& круче ||

```
alert( 5 || 1 && 0 ); // 5
```

! (НЕ)

- Сначала приводит аргумент к логическому типу true/false.
- Затем возвращает противоположное значение.

```
var result = !value;
```

```
alert( !true ); // false
```

```
alert( !0 ); // true
```

```
alert( !!"строка" ); // true
```

```
alert( !!null ); // false
```

Преобразование типов для примитивов

Преобразование типов для примитивов

String

Number

Boolean

Строковое преобразование

```
var a = true;
```

```
alert( a ); // 'true'
```

```
alert( true + "test" ); // 'true test'
```

```
alert( "123" + undefined ); // '123undefined'
```

Численное преобразование

```
var a = +'123'; // 123
```

```
var b = Number('123'); // 123
```

undefined

NaN

null

0

true / false

1 / 0

Строка

Пробельные символы по краям обрезаются.
Далее, если остаётся пустая строка, то 0,
иначе из непустой строки «считывается»
число, при ошибке результат NaN.

Численное преобразование

// после обрезания пробельных символов останется "123"

```
alert( +"  \n 123  \n  \n" ); // 123
```

```
alert( +true ); // 1
```

```
alert( +false ); // 0
```

```
alert( "\n0 " == 0 ); // true
```

```
alert( "\n" == false );
```

```
alert( "1" == true );
```

Логическое преобразование

undefined, null	false
Числа	Все true, кроме 0, NaN — false.
Строки	Все true, кроме пустой строки "" — false
Объекты	Всегда true

Логическое преобразование

```
alert(!!'0'); // true
```

```
alert(!!' '); // любые непустые строки, даже из  
пробелов – true!
```

```
alert( 0 == "\n0\n" ); // true
```

План занятия

- Операторы
- Операторы сравнения и логические операции
- Условные операторы
- Логические операторы
- Преобразование типов для примитивов



2015