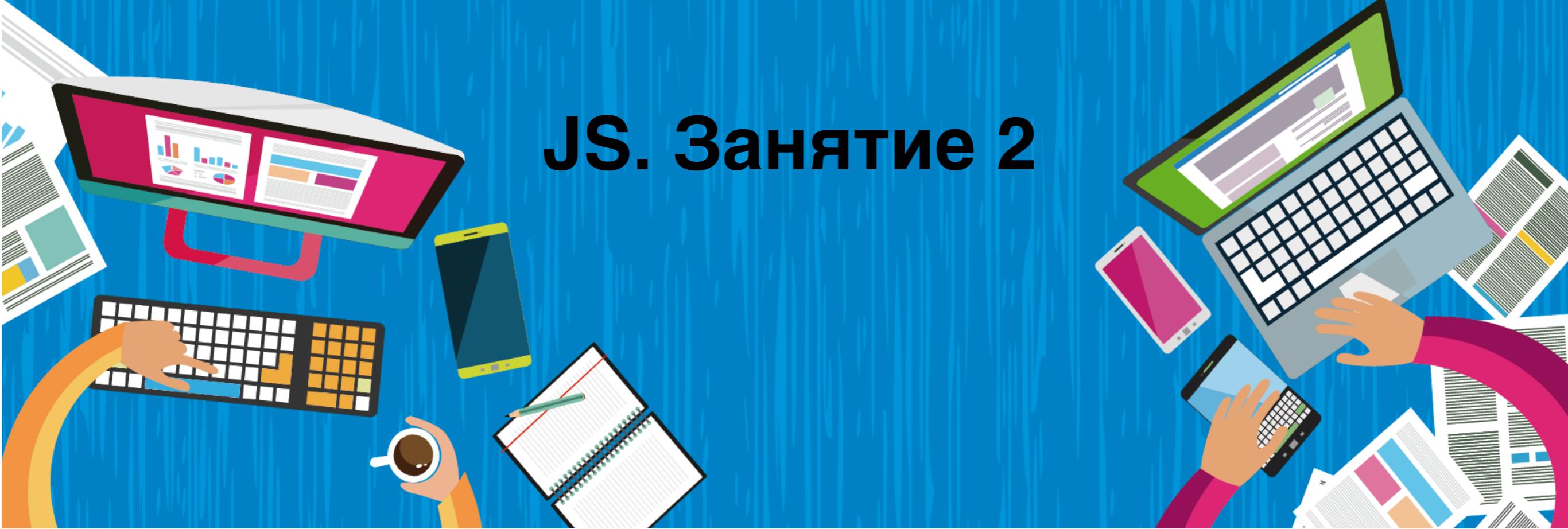


ArtFrontEnd

Операторы, функции и циклы

JS. Занятие 2



План занятия

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

Основные операторы



Оператор

Операнд

$3 * 5$

Операнд – то,
к чему применяется
оператор

Унарный

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

Бинарный

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

Бинарный +

```
var a = 'Welcome to ' + 'JS';  
alert(a); // Welcome to JS
```

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

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

Унарный +

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

14

13

3

Логическое отрицание

Побитовое отрицание

Унарный плюс

Унарный минус

Prefix Increment

Prefix Decrement

typeof

void

delete

Умножение

Деление

Остаток

Сложение

Вычитание

Присваивание

справа налево

! ...

справа налево

~ ...

справа налево

+ ...

справа налево

- ...

справа налево

++ ...

справа налево

-- ...

справа налево

typeof ...

справа налево

void ...

справа налево

delete ...

слева направо

... * ...

слева направо

... / ...

слева направо

... % ...

слева направо

... + ...

слева направо

... - ...

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
```

Побитовые операторы

- AND (и) (&)
- OR (или) (|)
- XOR (побитовое исключающее или) (^)
- NOT (не) (~)
- LEFT SHIFT (левый сдвиг) (<<)
- RIGHT SHIFT (правый сдвиг) (>>)
- ZERO-FILL RIGHT SHIFT (правый сдвиг с заполнением нулями) (>>>)

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++) {  
    ...  
}
```

Задачка

```
var a = 2;
```

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

Тест на внимательность

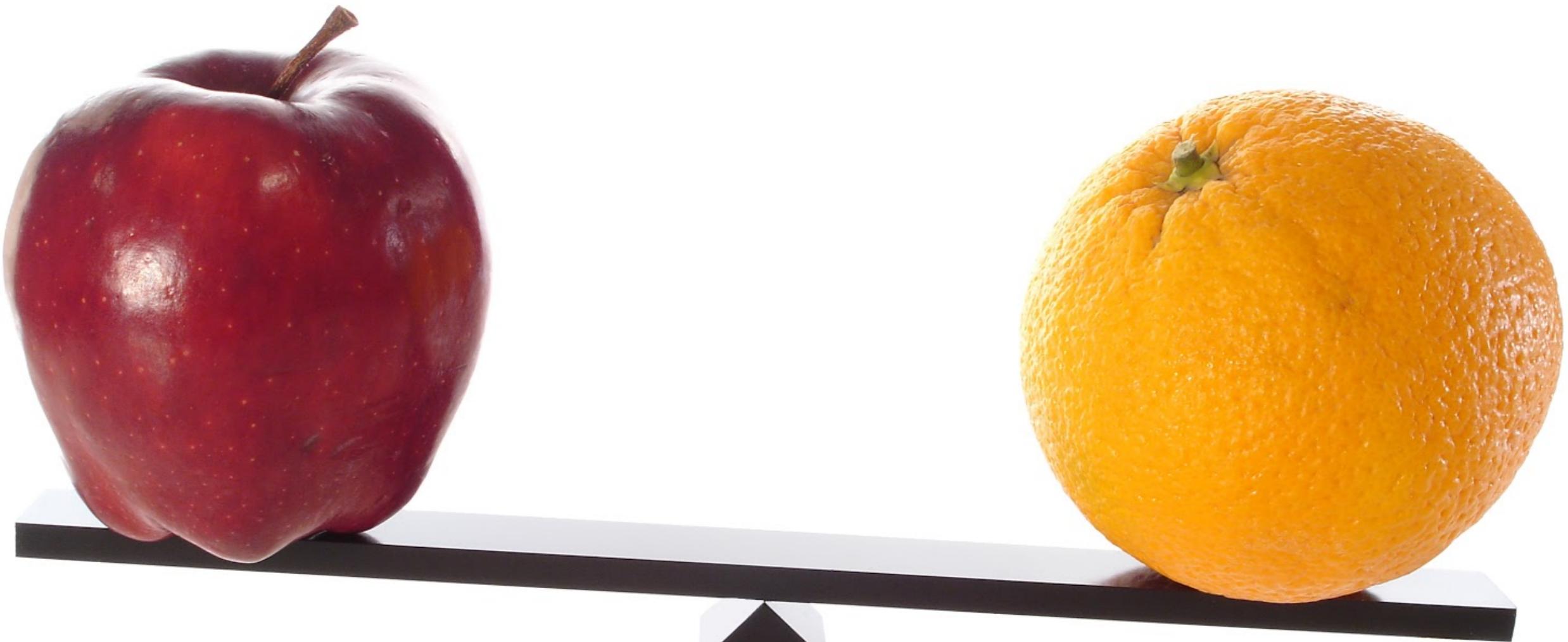
```
var a = 1, b = 1, c, d;
```

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

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

```
alert(a);  
alert(b);
```

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

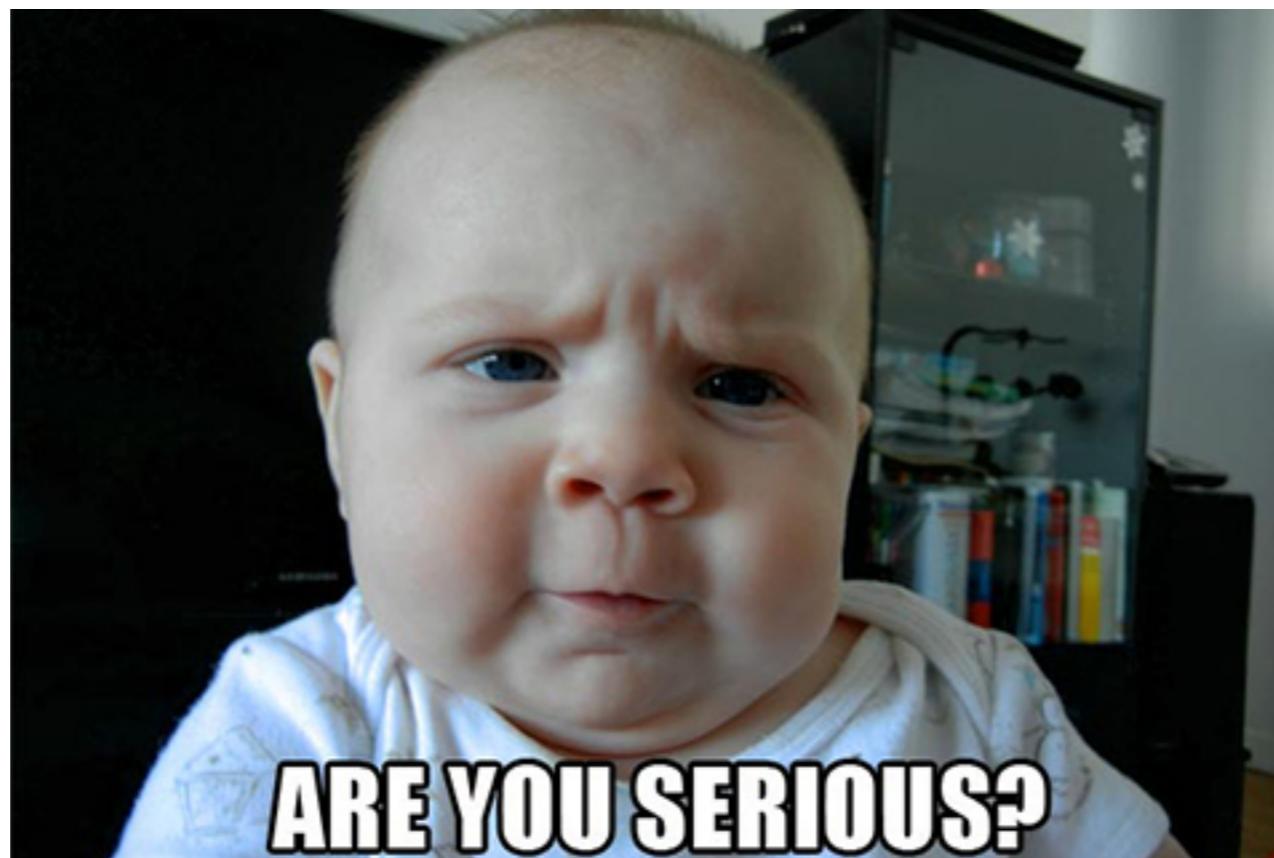
```
alert( 2 > 1 ); // true
alert( 2 == 1 ); // false
alert( 2 != 1 ); // true
```

```
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
```

```
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 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

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



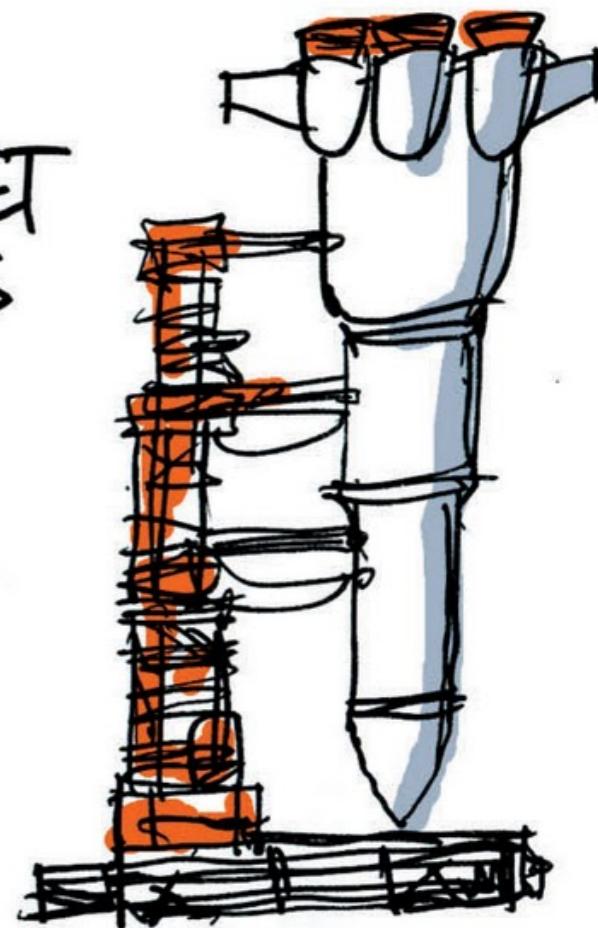
Сравнение с undefined

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



IF... BANKERS
WERE
FIREMEN

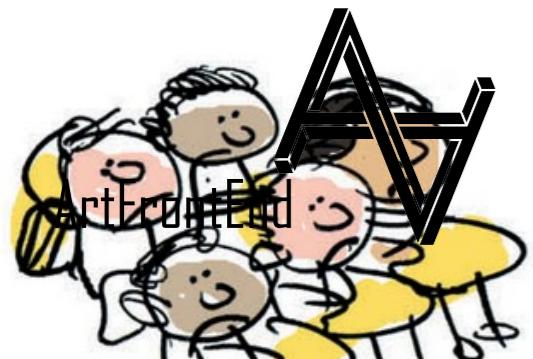
IF... WALL STREET
BANKERS
WERE
FIREMEN



Условные операторы



IF... FINANCIERS
WERE
SCHOOL
TEACHERS...



IF...
FIREMEN,
ROCKET SCIENTISTS
AND TEACHERS
GOT MILLION
DOLLAR
BONUSES...



THE SALT LAKE TRIBUNE
PASLEY 09

WE
WOULDN'T
BE IN
THIS
MESS.
artapp.pro

Оператор if

```
var year = prompt('What year today?' , '') ;  
  
if (year != 2016) alert( 'Wroooong!' ) ;  
  
if (year != 2016) {  
    alert( 'Посмотри...' ) ;  
    alert( '...в свой телефон!' ) ; }
```

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 ЕСМА-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 ЕСМА-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 = 'Заходи дарагой!'; }
```

```
else {
```

```
    access = 'ЭЭЭ! Маленький ишо!'; }
```

```
alert(access);
```

Оператор ?

условие ? значение1 : значение2

```
var access;
```

```
var age = prompt('Сколько вам лет?', ''');
```

```
access = (age > 18) ? 'Захады дарагой!' :  
'ЭЭЭ! Малэнъкий 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 );
```

Задачка

```
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" ); // 'truetest'
```

```
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

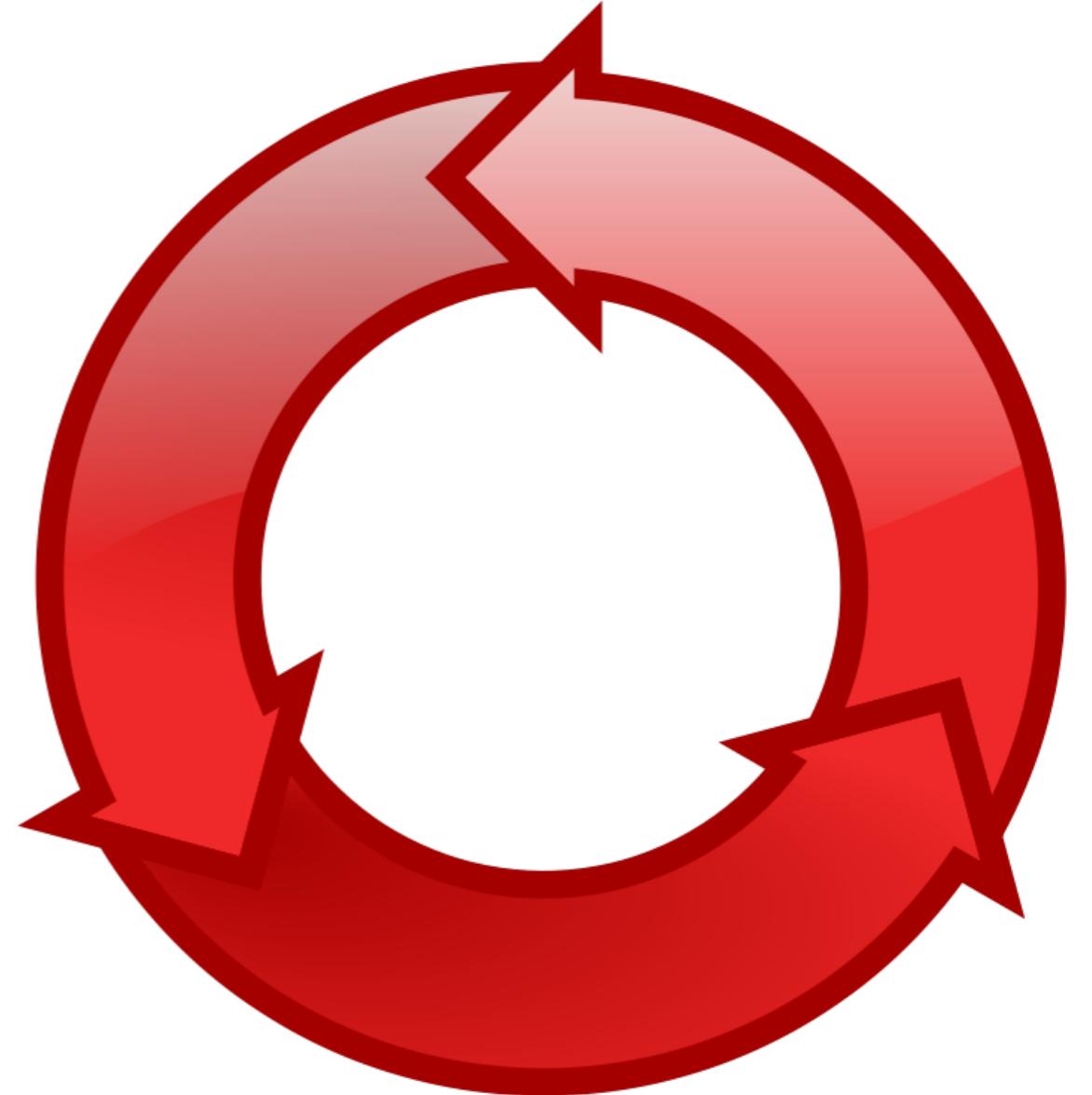
Всегда true

Логическое преобразование

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

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

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



Циклы

Множественный выбор

Switch

```
switch(x) {  
    case 'value1': // if (x === 'value1')  
    ...  
    [break]  
  
    case 'value2': // if (x === 'value2')  
    ...  
    [break]  
  
    default:  
    ...  
    [break]  
}
```

Switch

```
var a = 2 + 2;

switch (a) {
    case 3:
        alert( 'Маловато' );
        break;
    case 4:
        alert( 'В точку!' );
        break;
    case 5:
        alert( 'Перебор' );
        break;
    default:
        alert( 'Я таких значений не знаю' );
}
```

Switch

```
var a = 2 + 2;  
  
switch (a) {  
    case 3:  
        alert('Маловато');  
    case 4:  
        alert('В точку!');  
    case 5:  
        alert('Перебор');  
    default:  
        alert('Я таких значений не знаю');  
}  
  
// 'В точку!'  
// 'Перебор'  
// 'Я таких значений не знаю'
```

Switch

```
var a = 1;
```

```
var b = 0;
```

```
switch (a) {  
    case b + 1:  
        alert( 1 );  
        break;
```

```
default:
```

```
    alert('нет-нет, выполнится вариант выше')
```

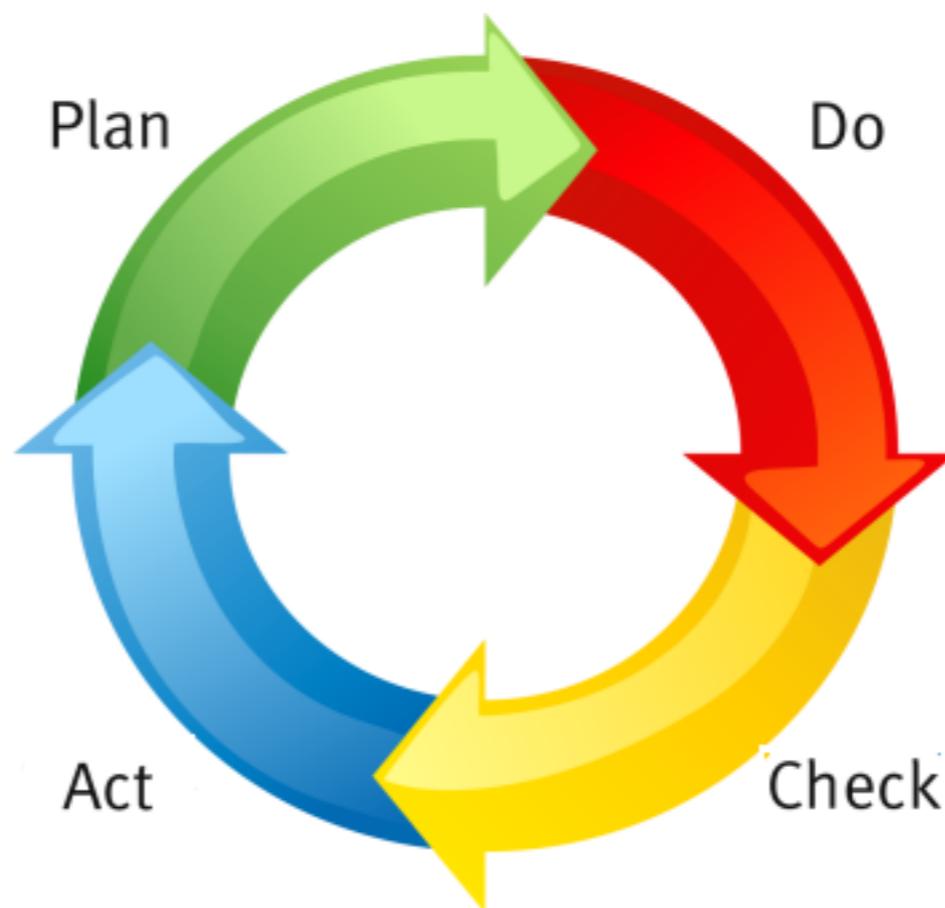
```
}
```

Группировка case

```
var a = 2 + 2;  
  
switch (a) {  
    case 4:  
        alert('Верно!');  
        break;  
  
    case 3:  
    case 5:  
        alert('Неверно!');  
        alert('Немного ошиблись, бывает.');  
        break;  
  
    default:  
        alert('Странный результат, очень странный');  
}
```

Что будет при `arg = 0, 1, 2, 3, 4 ?`

```
var arg = prompt('Введите arg?', '');
switch(arg) {
    case '0':
    case '1':
        alert( 'Один или ноль' );
    case '2':
        alert( 'два' );
        break;
    case 3:
        alert( 'Никогда не выполнится' );
    default:
        alert( 'Неизвестное значение: ' + arg );
}
```



While

Четные числа от 0 до 12

```
console.log(0);
console.log(2);
console.log(4);
console.log(6);
console.log(8);
console.log(10);
console.log(12);
```

Цикл



while

```
var number = 0;  
while (number <= 12) {  
    console.log(number);  
    number = number + 2;  
}
```

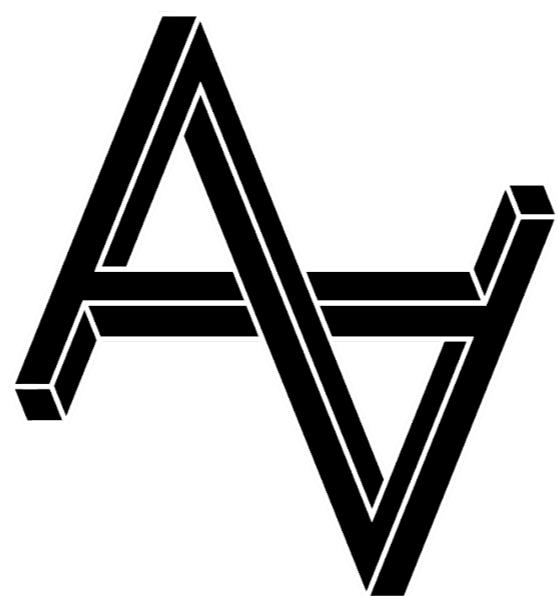
```
// → 0  
// → 2  
// ... 12
```

while

```
var result = 1;
var counter = 0;
while (counter < 10) {
    result = result * 2;
    counter = counter + 1;
}
console.log(result);
// → 1024
```

Бесконечный цикл

```
while (true) {  
    // ...  
}
```



2016