

## Лабораторная работа №2

### Операторы ветвлений и логические условия в JavaScript

#### Цель:

Комментарий: <http://learn.javascript.ru/play> - сервис онлайн для обучения скрипту

#### Оператор if

Оператор `if` («если») получает условие. Он вычисляет его, и если результат — `true`, то выполняет команду.

Если нужно выполнить более одной команды при выполнении условия, то они оформляются блоком кода в фигурных скобках:

```
var year = prompt('В каком году появилась спецификация ECMA-262 5.1?', '');
/* функция prompt создает окно для ввода текста, который передается в
переменную
*/
if (year != 2011)
{
    alert('А вот..');
    alert('..и неправильно!');
}
```

#### Преобразование к логическому типу

Оператор `if` вычисляет и преобразует условие к логическому типу.

**!!! В логическом контексте число 0, пустая строка "", null и undefined, а также NaN являются false, остальные значения — true.**

Например, такое условие никогда не выполнится:

```
if (0) { // 0 преобразуется к false
}
```

А такое — выполнится всегда:

```
if (1) { // 1 преобразуется к true
}
```

Вычисление условия в проверке `if (year != 2011)` может быть вынесено в отдельную переменную:

```
var cond = (year != 2011); // вернет true или false в зависимости от year
if (cond) {
    //какие-то операторы
}
```

#### Пример кода в составе страницы HTML:

```
<!DOCTYPE HTML>
<html>
<head>
<title>Пример 2_1</title>
</head>
<body>
<script>
var year = prompt('В каком году появилась спецификация ECMA-262 5.1?', '');
if (year != 2011)
```

```
{
  alert('А вот..');
  alert('..и неправильно!');
}
</script>
</body>
</html>
```

**Задание:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример2\_1.html и открыть (запустить) его в любом браузере.

### Условие else

Необязательный блок `else` («иначе») выполняется, если условие неверно:

```
var year = prompt('Введите год выхода ECMA-262 5.1', '');
if (year == 2011) {
  alert('Да вы знаток!');
} else {
  alert('А вот и неправильно!'); // любое значение, кроме 2011
}
```

### Несколько условий, else if

Бывает нужно проверить несколько вариантов условия. Для этого используется блок `else if`:

```
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!');
}
```

### Пример кода в составе страницы HTML:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Пример 2_2</title>
  </head>
  <body>
    <script>
      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..');
      }
    </script>
  </body>
</html>
```

```
}  
else {  
    alert('Да, точно в этом году!');  
}  
</script>  
</body>  
</html>
```

**Задание:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример2\_2.html и открыть (запустить) его в любом браузере.

#### Пример кода в составе страницы HTML:

```
<!DOCTYPE HTML>  
<html>  
<head>  
<title>Пример 2_3</title>  
</head>  
<body>  
<script>  
    //объявляем массив как экземпляр класса Array  
    var month_names = new Array("January", "February", "March", "April", "May", "June",  
"July", "August", "September", "October", "November", "December");  
    //создаем экземпляр класса Date  
    var d = new Date();  
    //создаем переменную и получаем день месяца из экземпляра d  
    var current_date = d.getDate();  
    //создаем переменную и получаем номер месяца из экземпляра d  
    //при чем счет месяцев идет с нуля  
    var current_month = d.getMonth();  
    //создаем переменную и получаем год в виде 4х чисел  
    var current_year = d.getFullYear();  
  
    alert("day - "+current_date+" ; month - "+current_month + " ; year - "+current_year);  
  
    if (current_month ==0) {  
        alert('На дворе январь'+ " ; "+ month_names[current_month]);  
    }  
    else if (current_month == 1) {  
        alert(' На дворе февраль'+ " ; "+ month_names[current_month]);  
    }  
    else {  
        alert('Месяц науке не известен!');  
    }  
</script>  
</body>  
</html>
```

**Задание:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример2\_3.html и открыть (запустить) его в любом браузере. Относительно полный список функций работы с датой/временем есть на <http://web-utils.net/javascript/date-time-functions> (в конце статьи в виде таблицы).

#### Индивидуальные задания:

**Вариант 1:** Напишите код, который исправит предыдущий пример и будет отражать по введенной дате соответствующий номер декады месяца.

**Вариант 2:** Напишите код, который исправит предыдущий пример и будет отражать для введенной даты по условиям варианты веков в диапазоне (17 век, 18 век, 19 век, 20 век, 21 век), а для остальных случаев выводила бы сообщение «Век науке не известен».

**Вариант 3:** Напишите код, который получает значение prompt, а затем выводит alert: «маловато», если сумма дня и номера месяца больше 1, «все еще маловато», если сумма дня и номера месяца больше 10, «терпимо», если сумма дня и номера месяца больше 20, «сойдет», во всех остальных случаях.

**Вариант 4:** Напишите код, который исправит предыдущий пример и будет проверять високосный ли год. Если високосный, то выводить «ура, лишний день в году», иначе «все, как обычно».

## Оператор вопросительный знак '?'

Иногда нужно в зависимости от условия присвоить переменную. Например:

```
var access;  
var age = prompt('Сколько вам лет?', '');  
if (age > 14) {  
    access = true;  
} else {  
    access = false;  
}  
alert(access);
```

Оператор вопросительный знак '?' позволяет делать это короче и проще.

Он состоит из трех частей:

условие ? значение1 : значение2

Проверяется условие, затем если оно верно — возвращается значение1, если неверно — значение2, например:

```
access = (age > 14) ? true : false;
```

Оператор '?' выполняется позже большинства других, в частности — позже сравнений, поэтому скобки можно не ставить:

```
access = age > 14 ? true : false;
```

#### Пример кода в составе страницы HTML:

```
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Пример 2_4</title>
```

```

</head>
<body>
  <script>
    var access;
    var access2;
    var age = prompt('Сколько вам лет?', '');
    if (age > 14) {
      access = true;
    }
    else
    {
      access = false;
    }
    access2 = (age > 14) ? true : false;
    alert("access = "+access+" ; access2 = "+ access2);

  </script>
</body>
</html>

```

**Задание:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример2\_4.html и открыть (запустить) его в любом браузере.

**Индивидуальное задание:**

**Вариант 1:** Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?:':

```

if (sqrt(a*b) < 4) {
  result = 'Мало';
} else {
  result = 'Много';
}

```

**Вариант 2:** Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?:':

```

if (a==b) {
  result = 'значения равны';
} else {
  result = 'значения не равны';
}

```

**Вариант 3:** Напишите код, который получает значение трех переменных a, b и c через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?:':

```

if ((a*a+b*b)=c*c) {
  result = 'Ура: угадали длины сторон треугольника';
} else {
  result = 'Надо учить теорему';
}

```

**Вариант 4:** Напишите код, который получает значение двух переменных a и b через prompt, а затем выводит alert, затем перепишите следующий if с использованием оператора '?':

```
if ((a+a)==b) {  
    result = 'значения равны';  
} else {  
    result = 'значения не равны';  
}
```

## Несколько операторов '?'

Несколько операторов if..else можно заменить последовательностью операторов '?'.  
Например:

```
var a = prompt('a?', 1);  
  
var res = (a == 1) ? 'значение1' :  
    (a == 2) ? 'значение2' :  
    (a > 2) ? 'значение3' :  
    'значение4';  
  
alert(res);
```

Вопросительный знак проверяет сначала a == 1, если верно — возвращает значение1, если нет — идет проверять a == 2. Если это верно — возвращает значение2, иначе проверка a > 2 и значение3.. Наконец, если ничего не верно, то значение4.

Альтернативный вариант с if..else:

```
var res;  
  
if (a == 1) {  
    res = 'значение1';  
} else if (a == 2) {  
    res = 'значение2';  
} else if (a > 2) {  
    res = 'значение3';  
} else {  
    res = 'значение4';  
}
```

**Задание:** Перепишите if..else с использованием нескольких операторов '?'. Для читаемости — оформляйте код в несколько строк.

```
var message;  
if (login == 'Вася') {  
    message = 'Привет';  
} else if (login == 'Директор') {  
    message = 'Здравствуйте';  
} else if (login == "") {  
    message = 'Нет логина';  
} else {  
    message = "";  
}
```

## Логические операторы

В JavaScript поддерживаются операторы `||` (ИЛИ), `&&` (И) и `!` (НЕ).

Они называются «логическими», но в JavaScript могут применяться к значениям любого типа и возвращают также значения любого типа.

### `||` (ИЛИ)

Оператор ИЛИ выглядит как двойной символ вертикальной черты:

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

**Логическое ИЛИ в классическом программировании работает следующим образом: «если хотя бы один из аргументов true, то возвращает true, иначе — false».**

Получается следующая таблица результатов:

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

При вычислении ИЛИ в JavaScript можно использовать любые значения. В этом случае они будут интерпретироваться как логические.

Например, число 1 будет воспринято как true, а 0 — как false:

```
if ( 1 || 0 ) { // сработает как if( true || false )
  alert('верно');
}
```

Обычно оператор ИЛИ используется в `if`, чтобы проверить, выполняется ли хотя бы одно из условий, например:

```
var hour = 9;

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

Можно передать и больше условий:

```
var hour = 12, isWeekend = true;

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

### Пример кода в составе страницы HTML:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Пример 2_5</title>
  </head>
  <body>
    <script>
      var hour = prompt('Введите интересующий час работы');
      var isWeekend = prompt('Сегодня выходной? (да – true; нет - false)');

      if (hour < 10 || hour > 18 || isWeekend) {
        alert('Офис до 10 или после 18 или в выходной закрыт');
      }
    </script>
  </body>
```

```
</html>
```

**Задание:** Создать приведенный пример документа в любом редакторе. Удобнее использовать редактор Visual Studio или любой другой, поддерживающий языки разметки. Сохранить файл с названием Пример2\_5.html и открыть (запустить) его в любом браузере.

### Короткий цикл вычислений

JavaScript вычисляет несколько ИЛИ слева направо. При этом, чтобы экономить ресурсы, используется так называемый «короткий цикл вычисления».

Допустим, вычисляются несколько ИЛИ подряд: `a || b || c || ...`. Если первый аргумент — `true`, то результат заведомо будет `true` (хотя бы одно из значений — `true`), и остальные значения игнорируются.

Это особенно заметно, когда выражение, переданное в качестве второго аргумента, имеет *сторонний эффект* — например, присваивает переменную.

При запуске примера ниже присвоение `x` не произойдет:

```
var x;  
  
true || (x = 1); // просто вычислим ИЛИ, без if  
  
alert(x); // undefined, x не присвоен
```

...А в примере ниже первый аргумент — `false`, так что ИЛИ попытается вычислить второй, запустив тем самым присваивание:

```
var x;  
  
false || (x = 1);  
alert(x); // 1
```

### && (И)

Оператор И пишется как два амперсанда `&&`:

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

**В классическом программировании И возвращает `true`, если оба аргумента истинны, а иначе — `false`**

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

Пример:

```
var hour = 12, minute = 30;  
  
if (hour == 12 && minute == 30) {  
    alert('Время 12:30');  
}
```

К И применим тот же принцип «короткого цикла вычислений», но немного по-другому, чем к ИЛИ.

**Если левый аргумент — `false`, оператор И возвращает его и заканчивает вычисления. Иначе — вычисляет и возвращает правый аргумент.**

```
// 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 && "не важно"); // 0
```

**!!! Приоритет оператора И && больше, чем ИЛИ ||, т.е. он выполняется раньше.**

Поэтому в следующем коде сначала будет вычислено правое И:  $1 \ \&\& \ 0 = 0$ , а уже потом — ИЛИ.

### **! (НЕ)**

Оператор НЕ — самый простой. Он получает один аргумент. Синтаксис:

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

Действия !:

1. Сначала приводит аргумент к логическому типу `true/false`.
2. Затем возвращает противоположное значение.

Например:

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

**В частности, двойное НЕ используются для преобразования значений к логическому типу:**

```
alert( !"строка" ) // true  
alert( !null ) // false
```

### **Индивидуальное задание:**

**Вариант 1:** Напишите код, который получает значение возраста и пола (с проверкой условия, что вводится только «м» или «ж»). Напишите условие `if` для проверки того факта, что переменная `s` возрастом находится между 14 и 25 включительно и если пол мужской, то выводить фразу «Добрый день, молодой человек», если тот же возраст, но пол женский, то выводить фразу «Приветствуем юную даму». Во всех остальных случаях выводить «Привет!».

**Вариант 2:** Напишите код, который получает должность (с проверкой, что должность введена «начальник», «зам.начальника» или «секретарь») и ФИО. Напишите условие `if`, если должность не «секретарь», то выводить «Добрый день, господин начальник», иначе если должность «секретарь», то выводить «Добрый день, ФИО».

**Вариант 3:** Напишите код, который получает значения для трех переменных: цвет, форма, размер. При этом предлагается пользователю в сообщении вводить для цвета - «синий» или «зеленый» или «красный», для формы - «круглый», «квадратный», для размера — «большой», «маленький». Напишите условие, если цвет не синий, а форма круглая и размер маленький, то выводить фразу «Наверное вы подумали про яблоко», если форма квадратная, а цвет любой, то выводить фразу «Наверное вы подумали про нечто квадратное», иначе «Понятия не имею, о чем вы думали».

**Вариант 4:** Напишите код, который получает значения для трех переменных: цвет, форма, размер. При этом предлагается пользователю в сообщении вводить для цвета - «синий» или «зеленый» или «красный», для формы - «круглый», «квадратный», для размера — «большой», «маленький». Напишите условие, если цвет не красный и не зеленый, а форма круглая и размер большой, то выводить фразу «Что-то синее и большое», если форма квадратная, а цвет не синий, то выводить фразу «Наверное вы подумали про яблоко, но почему квадратное?», иначе выводить значения переменных в одном сообщении.

## **Отчет по лабораторной работе**

В соответствии со структурой заготовки отчета и примером оформления оформить в отчете все задания, выполняемые в ходе лабораторной работы, а также индивидуальные задания по вариантам. Файл с отчетом называть по шаблону: **Фамилия\_лаб\_раб\_номер**.

Отчет предоставляется в электронном виде либо лично преподавателю, либо на электронную почту для проверки. Также по результатам лабораторной работы на следующем за ней занятии проводится выборочный опрос по командам языка.