

Практические задания

Основные возможности в примерах	Реализация
Создание простейшей программы на JavaScript	
<p>Пример 1. Средствами приложения Блокнот создать HTML-документ и отдельный JS-документ, который в дальнейшем подключается в HTML-документе. При просмотре HTML-документа в браузере должны выводиться окна с сообщениями:</p> <p>1) «Привет! Я alert, который расположен внутри отдельного файла» ;</p> <p>2) «Привет! Я alert, который расположен внутри страницы»;</p> <p>3) «Привет! Я alert, который расположен внутри ссылки».</p> <p>Причем сообщение 3 должно выводиться по щелчке на какой-либо ссылке.</p> <p>Создать кнопку «Ваше имя», после нажатия на которую появляется диалоговое окно для ввода имени. После ввода имени и нажатия ОК должно появиться другое окно с сообщением «Привет, введенное_имя».</p>	<ol style="list-style-type: none"> 1. Создайте рабочую папку workspace, например, на диске D. В ней мы будем сохранять все результаты работы. 2. Создайте внутри папки workspace папку js. В ней будут храниться сценарии JavaScript в отдельных файлах. 3. Откройте приложение Блокнот: Пуск/ Программы/ Стандартные/ Блокнот. 4. Введите следующий JavaScript код: alert('Привет! Я alert, который расположен внутри отдельного файла'); 5. Сохраните документ под именем example1.js в папке workspace\js: Файл / Сохранить как... / в поле Папка, пользуясь кнопкой раскрытия списка, выберите нужный диск / откройте вашу рабочую папку / последовательно откройте папки workspace, js / в поле Имя файла удалите введите example1.js / Сохранить. Не закрывайте приложение Блокнот. 6. В приложение Блокнот выделите весь текст: Правка / Выделить все. Нажмите клавишу Delete для удаления выделенного текста. 7. Введите следующий JavaScript код: <pre> <html> <head> <title>Example 1</title> <script type="text/javascript" language="javascript"> alert('Привет! Я alert, который расположен внутри страницы'); </script> <script type="text/javascript" language="javascript" src="js/example1.js"></script> </head> <body> Нажми на меня

 <input type="button" value="Ваше имя" onclick="alert('Привет, ' + prompt('Андрей') + '!');"> </body> </html> </pre> 8. Сохраните документ под именем example1.htm в папке workspace. 9. Откройте и просмотрите сохраненный документ example1.htm в любом браузере. Например, в браузере Microsoft Internet Explorer. <p>Примечание: onClick – обработчик события click, которое возникает при щелчке мыши на каком-либо элементе. Т.о. при щелчке мыши на ссылке «Нажми на меня» выполняется код, находящийся в кавычка onclick="код". Аналогично для кнопки «Ваше имя».</p>
Типы данных, переменные и преобразование типов	
<p>Пример 2. Создать html-документ, содержащий JavaScript-код, в котором</p>	<ol style="list-style-type: none"> 1. Создайте html-документ example2.htm со следующим содержанием: <pre> <html> <head> </pre>

необходимо:

- 1) Создать переменные разных типов.
- 2) Сложить переменные и вывести результат на экран.
- 3) Переопределить какую-либо переменную, присваивая ей значение другого типа.
- 4) Сложить две однотипные переменные.
- 5) Преобразовать строковую переменную в числовую.
- 6) Вычислить математическое выражение, представленное в виде строки и вывести результат на экран.

```
<title>Example 2</title>
<script type="text/javascript" language="javascript">

</script>
</head>
<body>
```

```
</body>
</html>
```

2. Между тегами <script></script> поместите следующий код:

```
//Определяем строковую переменную str
var str = "2+3=";
//Определяем переменную - число
var intVal = 5;
//Попробуем сложить строку и число и вывести результат в виде сообщения
var sum = str+intVal;
document.write ('Сложили строку "2+3=" и число 5. Результат: ' +
sum + '<br>');
```

```
//Теперь присвоим строковой переменной str численное значение
str = 2;
//Сложим два числа и выведем результат
sum = str+intVal;
document.write('Сложили переменные str=2 и intVal=5. Результат: ' +
sum + '<br>');
```

```
//Присвоим переменной str строку, представляющую собой число
str = "45";
/*
Сложим целочисленную переменную intVal с строкой str,
которую предварительно преобразуем в целое число с помощью функции
parseInt.
Выведем результат на экран.
*/
sum = intVal + parseInt(str);
document.write('Сложили intVal + parseInt(str). Результат: ' + sum +
'<br>');
```

```
//Присвоим переменной str строку, представляющую собой число с
плавающей точкой
str = "34.56";
/*
Сложим целочисленную переменную intVal с строкой str,
которую предварительно преобразуем в число с помощью функции
parseFloat.
Выведем результат на экран.
*/
sum = intVal + parseFloat(str);
document.write('Сложили intVal + parseFloat(str). Результат: ' + sum +
'<br>');
```

```
//Присвоим переменной str строку, представляющую собой математическое
выражение
str = "5+17-3";
//Вычислим значение используя функцию eval и выведем результат на экран
document.write('Вычисляем строку eval("'+str+'"). Результат: ' +
eval(str) + '<br>');
```

3. Откройте и просмотрите сохраненный документ example2.htm в браузере. При правильном написании кода вы увидите строки:

Сложили строку "2+3=" и число 5. Результат: 2+3=5
Сложили переменные str=2 и intVal=5. Результат: 7
Сложили intVal + parseInt(str). Результат: 50
Сложили intVal + parseFloat(str). Результат: 39.56
Вычисляем строку eval("5+17-3"). Результат: 19

	<p>Примечание: Строки, впереди которых находятся символы <code>//</code> - это однострочные комментарии. Они используются для пояснения работы кода. В пункте 2 текст находится на двух строчках из-за недостаточной ширины таблицы. Строки, заключенные между <code>/* */</code> - это многострочный комментарий. Комментарии не воспринимаются как код и не выполняются. Для вывода информации на экран в примере использовался метод document.write. Объект document – это объект в иерархической объектной модели. Он предоставляет всю информацию о html-документе и содержит свойства и методы для работы с ней. Так метод write принимает в качестве значение строку, которую выводит на Web-странице в браузере.</p>
Выражения в JavaScript	
<p>Пример 3. Создать html-документ, содержащий JavaScript-код, в котором необходимо:</p> <ol style="list-style-type: none"> 1) Сложить два числа, не присваивая их переменным – с помощью выражения. Результат вывести на экран. 2) Создать диалоговое окно с каким либо вопросом и двумя кнопками — ОК и Отмена. 3) Обработать результат нажатия кнопок окна с помощью логического выражения. 4) Вывести результат на экран 	<ol style="list-style-type: none"> 1. Создайте html-документ example3.htm со следующим содержанием: <pre> <html> <head> <title>Example 3</title> <script type="text/javascript" language="javascript"> </script> </head> <body> </body> </html> </pre> 2. Между тегами <code><script></script></code> поместите следующий код: <pre> document.write('Сложим два числа 3 и 6 без присваивания их переменным: ' + (3 + 6) + '
'); //Сразу вычисляется выражение в скобках, потом формируется строка //Если не записать (3+6) в скобках, то получится 36 document.write('Без скобок: ' + 3 + 6 + '
'); //Пример на логическое выражение var result = confirm("Вам больше 18 лет?"); //Если нажата ОК, то result=true, отмена - result=false (result) ? document.write('Вам больше 18 лет!
') : document.write('Вам меньше 18 лет!
'); </pre> 3. Откройте и просмотрите сохраненный документ example3.htm в браузере. <p>Примечание: Встроенный метод confirm создает окно с двумя кнопками ОК и Отмена и содержит сообщение, которое передается в качестве параметра. В результате нажатия на какую-либо кнопку метод возвращает результат true или false (ОК=true, Отмена=false).</p>
Операторы присваивания, арифметические, операторы работы со строкой, логические операторы	
<p>Пример 4. Создать html-документ, содержащий JavaScript-код, в котором необходимо:</p> <ol style="list-style-type: none"> 1) Создать выражения с использованием всех операторов присваивания и 	<ol style="list-style-type: none"> 1. Создайте html-документ example4.htm со следующим содержанием: <pre> <html> <head> <title>Example 4</title> <script type="text/javascript" language="javascript"> </script> </head> </pre>

<p>арифметических операторов.</p> <p>2) Для каждого выражения вывести результат на экран.</p> <p>3) Создать ссылку, при нажатии на которую подсчитывается число нажатий и выводится на экран в виде диалогового окна.</p> <p>4) Создать html-форму, содержащую два текстовых поля для ввода чисел и одно для вывода результата.</p> <p>5) В форме создать четыре кнопки: «Сложить», «Вычесть», «Умножить», «Разделить».</p> <p>6) Каждая кнопка должна совершать действия над введенными в поля числами, соответствующие ее названию, и выводить результат в поле для результатов.</p> <p>7) В этой же форме создать два поля для ввода строк и поле для вывода результата.</p> <p>8) Добавить кнопку «Склеить строки» для сложения строк и их вывода в поле результата.</p> <p>9) В этой же форме создать два выпадающих списка, содержащих по два значения true и false, и текстовое поле для вывода результата.</p> <p>10) Создать 3 кнопки: «И &&», «ИЛИ » и «НЕ !».</p> <p>11) Каждая кнопка должна в соответствии со своим названием совершать действия над значениями, выбранными в выпадающих списках и выводить результат в поле для результатов.</p>	<pre> <body> </body> </html> </pre> <p>2. Между тегами <script></script> поместите следующий код:</p> <pre> var a = 100; var b = 10; var sum = 0; document.write('a=' + a + ', b=' + b + '
'); a += b; document.write('Сложение a+=b. Результат: a=' + a + '
'); document.write('a=' + a + ', b=' + b + '
'); a -= b; document.write('Вычитание a-=b. Результат: a=' + a + '
'); document.write('a=' + a + ', b=' + b + '
'); a *= b; document.write('Умножение a*=b. Результат: a=' + a + '
'); document.write('a=' + a + ', b=' + b + '
'); a /= b; document.write('Деление a/=b. Результат: a=' + a + '
'); document.write('a=' + a + ', b=' + b + '
'); a = 9; b = 7; document.write('Переопределили a=' + a + ', b=' + b + '
'); a %= b; document.write('Остаток от деления a%=b. Результат: a=' + a + '
'); document.write('a=' + a + ', b=' + b + '
'); document.write('Арифметические операторы a * b / a + b - a % b. Результат: a=' + (a * b / a + b - a % b) + '
'); document.write('a=' + a + ', b=' + b + '
'); a++; document.write('Инкремент a++. Результат: a=' + a + '
'); a--; document.write('Декремент a--. Результат: a=' + a + '
'); a = -a; document.write('Унарное вычитание a = -a. Результат: a=' + a + '
'); </pre> <p>3. Сохраните документ, не закрывая Блокнот. Откройте и просмотрите сохраненный документ example4.htm в браузере.</p> <p>4. При правильном написании кода вы увидите строки:</p> <pre> a=100, b=10 Сложение a+=b. Результат: a=110 a=110, b=10 Вычитание a-=b. Результат: a=100 a=100, b=10 Умножение a*=b. Результат: a=1000 a=1000, b=10 Деление a/=b. Результат: a=100 a=100, b=10 Переопределили a=9, b=7 Остаток от деления a%=b. Результат: a=2 a=2, b=7 Арифметические операторы a * b / a + b - a % b. Результат: a=12 a=2, b=7 Инкремент a++. Результат: a=3 Декремент a--. Результат: a=2 Унарное вычитание a = -a. Результат: a=-2 </pre> <p>5. Попробуйте поэкспериментировать со значениями переменных изменяя их.</p> <p>6. Между тегами <body></body> документа добавьте следующий код:</p> <pre> Сколько раз нажата ссылка?

 </pre>
--	---

7. Сохраните документ не закрывая Блокнот, обновите открытую страницу example4.htm в браузере нажатием клавиши **F5**. В результате вы увидите ссылку, после нажатии на которую будет выводиться окно с сообщением о том, сколько раз вы нажали на ссылку.

8. Добавьте после кода ссылки следующий код, представляющий собой форму для ввода данных:

```
<form name="form1">
```

```
Число1: <input type="text" name="first" value="30" size="4">
```

```
Число2: <input type="text" name="second" value="4" size="4">
```

```
Результат: <input type="text" name="result" value="" size="6"> <br> <br>
```

```
<input type="button" value="Сложить"
onclick="document.form1.result.value=parseInt(document.form1.first.value)
+ parseInt(document.form1.second.value)">
```

```
<input type="button" value="Вычесть"
onclick="document.form1.result.value=parseInt(document.form1.first.value)
- parseInt(document.form1.second.value)">
```

```
<input type="button" value="Умножить"
onclick="document.form1.result.value=parseInt(document.form1.first.value)
* parseInt(document.form1.second.value)">
```

```
<input type="button" value="Разделить"
onclick="document.form1.result.value=parseInt(document.form1.first.value)
/ parseInt(document.form1.second.value)">
```

```
<br> <br>
```

```
</form>
```

9. Сохраните документ не закрывая Блокнот, обновите открытую страницу example4.htm в браузере нажатием клавиши **F5**. В результате вы увидите форму:

Число1: Число2: Результат:

10. Попробуйте поводить различные числа в поля «Число 1» и «Число 2» и посмотрите результат работы скрипта.

11. Между тегами <form></form> после предыдущих элементов введите следующий код:

```
Операции со строками: <br>
```

```
Строка1: <input type="text" name="str1" value="Java">
```

```
Строка2: <input type="text" name="str2" value="Script">
```

```
Результат: <input type="text" name="newStr" value=""> <br>
```

```
<input type="button" value="Склеить строки"
```

```
onclick="document.form1.newStr.value=document.form1.str1.value +
document.form1.str2.value"> <br> <br>
```

12. Сохраните документ не закрывая Блокнот, обновите открытую страницу в браузере. В результате вы увидите форму:


Операции со строками:

Строка1: Строка2: Результат:

13. Попробуйте поводить различные строки в поля «Строка 1» и «Строка 2» и посмотрите результат работы скрипта.

14. Между тегами <form></form> после предыдущих элементов введите следующий код:

```
Логические операторы:
```

	<pre> <select name="bool1"> <option value=1>true</option> <option value=0>false</option> </select> <select name="bool2"> <option value=0>false</option> <option value=1>true</option> </select> Результат:<input type="text" name="logRes" value="">
 <input type="button" value="И &&" onclick="document.form1.logRes.value = ((parseInt(document.form1.bool1.value) && parseInt(document.form1.bool2.value)) == 0) ? false : true;"> <input type="button" value="ИЛИ " onclick="document.form1.logRes.value = ((parseInt(document.form1.bool1.value) parseInt(document.form1.bool2.value)) == 0) ? false : true;"> <input type="button" value="НЕ !" onclick="document.form1.logRes.value = (!parseInt(document.form1.bool1.value)) == 0) ? false : true;"> </pre> <p>15. Сохраните документ и обновите открытую страницу в браузере. В результате вы увидите еще одну форму:</p>  <p>Кнопка «НЕ !» работает только для первого элемента.</p> <p>16. Попробуйте выполнить различные логические операции, нажимая на кнопки для различных комбинаций значения.</p> <p>Примечание:</p> <ul style="list-style-type: none"> • Запись постфикса перед переменной ++sum в свойстве onclick ссылки приводит к увеличению значения переменной sum на 1 перед тем, как она будет выведена с помощью метода alert. Если написать sum++, то при первом вызове alert sum = 0. • Так как объект document предоставляет всю информацию о html-документе, то через него осуществляется доступ к переменным, стоящим ниже его по иерархии. Для текстового поля «Строка 1» элементом-родителем является элемент-форма – form1, а для form1 родитель – document. Поэтому для доступа к текстовому полю «Строка 1» нужно записать такую последовательность: document.form1.str1. Однако str1 – это не значение, а объект. Для получения значения служит свойство value: document.form1.str1.value • При написании кода для изучения работы логических операторов значению true соответствовало значению 1, а false – 0. Это допустимо по отношению к переменным Boolean-типа.
Условный оператор - if...else	
<p>Пример 5. Создать html-документ, содержащий JavaScript-код, в котором необходимо:</p> <ol style="list-style-type: none"> 1) Создать две целочисленные переменные и сравнить их с помощью оператора сравнения. 2) Результат сравнения 	<ol style="list-style-type: none"> 1. Создайте html-документ example5.htm со следующим содержанием: <pre> <html> <head> <title>Example 5</title> </head> <body> <script type="text/javascript" language="javascript"> var a = 3; var b = 8; </pre>

<p>вывести на экран.</p> <p>3) Создать форму для проверки целого числа на четность/нечетность.</p> <p>4) Форма должна представлять собой текстовое поле для ввода числа и кнопку для осуществления проверки.</p> <p>5) Результат об четности/нечетности вывести на экран с помощью диалогового окна.</p>	<pre>document.write('a=' + a + ', b=' + b + '
'); if (a > b) { document.write('a>b'); } else { document.write('a<=b'); } </script> </body> </html></pre> <p>2. Сохраните документ без закрытия Блокнота и откройте созданную html-страницу в браузере. При правильном выполнении примера должны получиться строки:</p> <p>a=3, b=8 a<=b</p> <p>3. Внутри тега <body></body> после закрывающего тега </script> добавьте код формы со следующим содержанием:</p> <pre><form name="form1"> <input type="text" name="n" value="2"> <input type="button" value="Четное или нечетное?" onclick="if(parseInt(document.form1.n.value) % 2 == 0){ alert('Четное'); } else { alert('Нечетное'); }"> </form></pre> <p>4. Сохраните документ и обновите открытую страницу в браузере.</p> <p>Примечание: Оператор % вычисляет остаток от деления. Все четные числа делятся на 2 без остатка. Поэтому выражение в скобках if принимает значение true когда n делится на 2 без остатка.</p>
--	--

Цикл с условием - do...while

<p>Пример 6. Создать html-документ, включающий в себя JavaScript, который с помощью цикла do...while вычисляет факториал числа. Для ввода целочисленного значения создается диалоговое окно. Результат вычислений выводится на экран в виде строки.</p>	<p>1. Создайте html-документ example6.htm со следующим содержанием:</p> <pre><html> <head> <title>Example 6</title> </head> <body> <!-- Вычисление факториала--> <script type="text/javascript"> var n = parseInt(prompt('Введите целое число:', 5)); var factorial = 1; if (n < 0){ n *= -1; } var buf = n; do { factorial *= n--; } while (n > 0); document.write('Факториал ' + buf + '! равен ' + factorial); </script></pre>
--	--

	<pre></body> </html></pre> <p>2. Сохраните документ и откройте его в браузере.</p> <p>Примечание: Цикл do...while выполняется всегда хотя бы один раз независимо от условия цикла.</p>
Цикл с условием - while	
<p>Пример 7. Создать html-документ, включающий в себя JavaScript-код, который с помощью цикла while возводит заданное число в указанную степень. Для ввода числа и степени создаются диалоговые окна. Результат вычислений выводится на экран в виде строки.</p>	<p>1. Создайте html-документ example7.htm со следующим содержанием:</p> <pre><html> <head> <title>Example 7</title> </head> <body> <!-- Вычисление степени числа--> <script type="text/javascript"> var n = parseInt(prompt('Введите целое число:', 2)); var grade = parseInt(prompt('Введите степень(целое, неотрицательное), в которую нужно возвести число ' + n + ':', 3)); var buf = grade; var result = 1; if (n == 0.0) result = 0; else if (n != 0 && grade == 0) result = 1; else { (grade < 0) ? grade *= -1 : grade; buf = grade; while (grade > 0) { result *= n; grade--; } } document.write(n + '<sup>' + buf + '</sup>=' + result); </script> </body> </html></pre> <p>2. Сохраните документ и откройте его в браузере.</p>
Цикл с параметром – for	
<p>Пример 8. Создать html-документ, включающий в себя JavaScript, который с помощью цикла for вычисляет факториал числа. Для ввода целочисленного значения создается диалоговое окно. Результат вычислений выводится на экран в виде строки.</p>	<p>1. Создайте html-документ example8.htm со следующим содержанием:</p> <pre><html> <head> <title>Example 8</title> </head> <body> <!-- Вычисление факториал--> <script type="text/javascript"> var n = parseInt(prompt('Введите целое число:', 5)); var factorial = 1; for (var i = 1; i <= n; i++){ factorial *= i; }</pre>

	<pre>document.write('Факториал ' + n + '! равен ' + factorial); </script> </body> </html></pre> <p>2. Сохраните документ и откройте его в браузере.</p>
Оператор множественного выбора – switch	
<p>Пример 9. Создать html-документ, включающий в себя JavaScript-код, позволяющий по номеру месяца определить его название и вывести на экран в виде строки. Номер месяца вводится через диалоговое окно.</p>	<p>1. Создайте html-документ example9.htm со следующим содержанием:</p> <pre><html> <head> <title>Example 9</title> </head> <body> <!-- Определение названия месяца по его номеру--> <script type="text/javascript"> var n = parseInt(prompt('Введите номер месяца от 1 до 12:', 12)); switch (n) { case 1: document.write('Январь'); break; case 2: document.write('Февраль'); break; case 3: document.write('Март'); break; case 4: document.write('Апрель'); break; case 5: document.write('Май'); break; case 6: document.write('Июнь'); break; case 7: document.write('Июль'); break; case 8: document.write('Август'); break; case 9: document.write('Сентябрь'); break; case 10: document.write('Октябрь'); break; case 11: document.write('Ноябрь'); break; case 12: document.write('Декабрь'); break; default : document.write('Месяц не определен'); } </script> </body> </html></pre>

2. Сохраните документ и откройте его в браузере.	
События	
<p>Пример 10. Создать html-документ, включающий в себя JavaScript-код, в котором реализуется обработка различных событий, таких как click, load, change, focus и т.д.</p>	<ol style="list-style-type: none"> Создайте html-документ example10.htm со следующим содержанием: <pre> <html> <head> <title>Example 10</title> </head> <body onload="alert('Добро пожаловать на эту замечательную страницу')"> Ссылка

 </body> </html> </pre> Сохраните документ без закрытия Блокнота и откройте созданную html-страницу в браузере. Нажмите на ссылку «Ссылка». Добавьте между тэгами <body></body> после ссылки следующую форму: <pre> <form action="" method="post" onsubmit="alert('Нажата кнопка submit. Передача данных формы'); return false;" name="form1"> Поле1: <input type="text" name="field1" onfocus="alert('Выбрано первое поле');" onkeyup="document.form1.field2.value=this.value">
 Поле2: <input type="text" name="field2" onfocus="alert('Выбрано второе поле');">
 Список: <input type="submit" value="Кнопка submit"> </form>

 </pre> Сохраните документ без закрытия Блокнота и обновите открытую страницу в браузере. Попробуйте понажимать кнопки формы, поводить различные данные в форме. Добавьте после закрывающего тега </form> следующий код: <pre> Подведите сюда курсор </pre> Сохраните документ и обновите открытую страницу в браузере. Наведите курсор на созданную только что ссылку.
Функции	
<p>Пример 11. Создать функцию для вычисления площади прямоугольника. Для ввода данных о сторонах прямоугольника создать соответствующую html-форму.</p>	<ol style="list-style-type: none"> Создайте html-документ example11.htm со следующим содержанием: <pre> <html> <head> <title>Example 11</title> <script type="text/javascript"> function square(a, b) { var s = parseFloat(a) * parseFloat(b); return s; } </script> </head> <body> </body> </html> </pre>

	<p>2. Между тэгами <body></body> создадим форму: Вычисляем площадь прямоугольника <form name="form1"> Сторона a:<input type="text" name="a" value="1">
 Сторона b:<input type="text" name="b" value="1">
 Площадь: <input type="text" name="s" value="">
 <input type="button" value="Вычислить" onclick="document.form1.s.value = square(document.form1.a.value, document.form1.b.value);"> </form></p> <p>2. Сохраните документ и откройте его в браузере 3. Введите численные данные для сторон и нажмите «Вычислить»</p>
Проверка на заполнение полей формы	
<p>Пример 12. Создать html-форму для передачи каких-либо данных и функцию, проверяющую форму на заполнение.</p>	<p>1. Создайте html-документ example12.htm, содержащий форму для ввода имени, email-адреса, URL сайта и сообщения:</p> <pre> <html> <head> <title>Example 12</title> <script type="text/javascript"> </script> </head> <body> <form action="" name="mail" onsubmit="return checkFields(this);" method="post"> Имя:<input type="text" name="name" value="">
 E-mail:<input type="text" name="email" value="">
 URL сайта: <input type="text" name="url" value="">
 Сообщение:
 <textarea rows="6" cols="20" name="message"></textarea>
 <input type="submit" value="Отправить"> <input type="reset" value="Очистить"> </form> </body> </html> </pre> <p>2. Сохраните документ без закрытия Блокнота и откройте его в браузере. Убедитесь, что получилась форма:</p> <div data-bbox="758 1411 1204 1836" data-label="Form"> <p>Имя: <input type="text"/></p> <p>E-mail: <input type="text"/></p> <p>URL сайта: <input type="text"/></p> <p>Сообщение: <input type="text"/></p> <p><input type="button" value="Отправить"/> <input type="button" value="Очистить"/></p> </div> <p>3. Создайте функцию для проверки на пустоту полей формы и добавьте ее между тэгами <script></script>:</p> <pre> function checkFields(form) { if (form.name.value == "") { alert('Не заполнено имя!'); form.name.focus(); return false; } } </pre>

	<pre> } if (form.email.value == "") { alert('Не заполнен email!'); form.email.focus(); return false; } if (form.url.value == "") { alert('Введите URL сайта!'); form.url.focus(); return false; } if (form.message.value == "") { alert('Введите сообщение!'); form.message.focus(); return false; } alert('Форма заполнена!!! Все ОК!'); //Прерываем отправку данных формы return false; } </pre> <p>4. Сохраните документ и обновите страницу, содержащую его в браузере.</p> <p>5. Попробуйте отправить форму без заполнения полей либо не со всеми заполненными полями.</p> <p>Примечание: При создании формы мы задали параметр onsubmit="return checkFields(this);". Это значит, что в зависимости от того, что вернет функция – true или false, форма будет отправлена (true) или не будет (false). Когда одно из полей не заполнено, выполняется блок if. в котором есть оператор return false; Он не допускает отправки формы. Для удобства в конце функции находится также return false. В реальных условиях должно быть return true; В качестве параметра функции передается this. this – это указатель на объект формы, который заменяет document.mail и освобождает программиста от ввода длинной информации.</p>
Создание собственных объектов	
<p>Пример 13. Создать объект, который носит информацию об автомобиле и создать несколько экземпляров объекта и поиграться со свойствами и методами объектов.</p>	<p>1. Создайте js-документ example13.js в папке js:</p> <pre> //Объект Car function Car(model, color, year) { //Свойства объекта this.model = model "неизвестно"; this.color = color "черный"; this.year = year 1900; //методы объекта this.toString = function() { return this.model + ' ' + this.color + ' ' + this.year; } this.go = function(speed) { return 'Автомобиль ' + this.toString() + ' движется со скоростью ' + speed + ' км/ч'; } this.stop = function () { return 'Автомобиль ' + this.toString() + ' остановился'; } this.signal = function () { return 'Автомобиль ' + this.toString() + ' подал сигнал'; } } </pre>

	<pre> } 2. Создайте html-документ example13.htm в папке workspace. В документе следует подключить скрипт js/example13.js: <html> <head> <title>Example 13</title> <script type="text/javascript" src="js/example13.js"></script> </head> <body> </body> </html> </pre> <p>3. Сохраните документ без закрытия Блокнота и откройте страницу в браузере.</p> <p>4. Создайте несколько экземпляров объекта и попробуйте изменять их свойства, вызывать их методы. Для этого поместите следующий код между тэгами <body></body>:</p> <pre> <script type="text/javascript"> var car1 = new Car("BA3 2109", "красный", 1989); var car2 = new Car("Audi 100", "коричневый", 1992); var car3 = new Car("Запорожец", "розовый", 1980); document.write(car1.toString()+'
'); document.write(car2.toString()+'
'); document.write(car3.toString()+'
'); document.write(car2.go(87)+'
'); document.write(car3.go(180)+'
'); document.write(car1.go(142)+'
'); document.write(car1.stop()+'
'); document.write(car3.signal()+'
'); </script> </pre> <p>3. Сохраните документ и обновите страницу в браузере. В результате работы скрипта должны получиться строки:</p> <pre> BA3 2109 красный 1989 Audi 100 коричневый 1992 Запорожец розовый 1980 Автомобиль Audi 100 коричневый 1992 движется со скоростью 87 км/ч Автомобиль Запорожец розовый 1980 движется со скоростью 180 км/ч Автомобиль BA3 2109 красный 1989 движется со скоростью 142 км/ч Автомобиль BA3 2109 красный 1989 остановился Автомобиль Запорожец розовый 1980 подал сигнал </pre> <p>4. Попробуйте создать еще несколько объектов и поиграться с ними.</p>
Использование встроенных объектов и создание массивов	
<p>Пример 14. Создать функцию, выводящую текущую дату в виде строки используя объект Date. С помощью объекта Array создать массивы, хранящие названия месяцев и дней недели. Создать текстовое поле в котором непрерывно будет выводиться текущая дата.</p>	<p>1. Создайте js-документ example14.js в папке js. В нем создайте массивы, хранящие названия месяцев и дней недели:</p> <pre> //Массив, содержащий названия месяцев var months = new Array("Янв", "Фев", "Мар", "Апр", "Май", "Июн", "Июл", "Авг", "Сен", "Окт", "Ноя", "Дек"); //Массив, содержащий названия дней недели. var dayNames = new Array("Понедельник", "Вторник", "Среда", "Четверг", "Пятница", "Суббота", "Воскресенье"); </pre> <p>2. Добавим в этот же файл функцию, выводящую полную текущую дату:</p> <pre> function getCurrentDate() { //Создаем объект Date для работы с датой </pre>

```

var date = new Date();

var dateStr = 'Сегодня ';
dateStr += dayNames[date.getDay()] + ', '
dateStr += date.getDate() + ' ';
dateStr += months[date.getMonth()] + ' ';
dateStr += date.getFullYear() + ' ';
dateStr += date.getHours() + ' ч. ';
dateStr += date.getMinutes() + ' мин. ';
dateStr += date.getSeconds() + ' сек.';
return dateStr;
}

```

3. Сохраните этот файл и создайте новый html-файл в папке workspace. В html-документе подключите созданный ранее файл example14.js. Страница должна содержать текстовое поле для вывода даты:

```

<html>
<head>
  <title>Example 14</title>
  <script type="text/javascript" src="js/example14.js"></script>
</head>
<body>

  <form name="dateForm">
    <input type="text" name="time" value="" size="100">
  </form>

</body>
</html>

```

4. Сохраните документ без закрытия Блокнота и откройте страницу в браузере.

5. Добавьте между тэгами <body></body> скрипт, содержащий функцию printDate, который получает текущую дату вызовом функции getCurrentDate и выводит ее в текстовом поле, созданном выше:

```

<script type="text/javascript">
  function printDate() {
    document.dateForm.time.value = getCurrentDate();
  }
</script>

```

6. Для непрерывного вывода даты и времени после функции printDate добавьте строку:

```

window.setInterval("printDate()", 1000);

```

Метод setInterval встроенного объекта window непрерывно вызывает функцию printDate через интервал 1000 мс, т.е. через каждую секунду.

7. Сохраните документ и обновите страницу в браузере. Вы получите текстовое поле, в котором непрерывно будет выводиться текущая дата и время.

Примечание:

Встроенный объект **Date** служит для работы с датой и временем и содержит множество методов. Метод getDay выводит номер дня недели. Поэтому мы создавали массив с названиями дней недели. Номер дня недели совпадает с индексом элемента массива. getDate выводит число. getMonth выводит номер месяца. По аналогии с днями недели создан массив названий месяцев. getFullYear выводит год, например 2009. getHours выводит час, getMinutes – минуты, getSeconds – секунды.

