

ЛАБОРАТОРНАЯ РАБОТА № 1

ТЕМА: ВВЕДЕНИЕ В JavaScript

Web-документ, отображаемый браузером, – это результат исполнения программ, созданных на разных языках. Для описания структуры используется язык разметки (xhtml), для описания внешнего вида – язык стилей (css). Для описания поведения документа, его реакции на действия пользователя используется язык сценариев (javascript).

Подключение к странице

Исполняет JavaScript-код браузер. В него встроен интерпретатор JavaScript. Следовательно, выполнение программы зависит от того, когда этот интерпретатор получает управление. Опишем несколько способов размещения кода JavaScript на странице:

1. В теговом контейнере `<BODY>...</BODY>`.

```
<body>
...
<script > команды скрипта</script>
...
</body>
```

2. В теговом контейнере `<HEAD>...</HEAD>` - если код скрипта представляет собой функцию, которая вызывается в ответ на какое-либо событие.

```
<head>
...
<script type="text/javascript"> Здесь находятся команды сценария </script>
...
</head>
```

3. Во внешнем файле. По аналогии с тем, как стили подключаются к странице с помощью элемента `link`, сценарии подключаются с помощью элемента `script`, только файл имеет расширение не `.css`, а `.js`.

```
<head>
...
<script type="text/javascript" src="my.js" > </script>
...
</head>
```

4. Обработчик события указывается прямо в теге, без заключения в теги `<script > </script>`

```
<input type="button" value="Нажать" onClick="window.alert('Нажмите еще раз')">
```

Выполнение операторов сценария

Существует несколько способов определения момента запуска сценария. Вот некоторые из них:

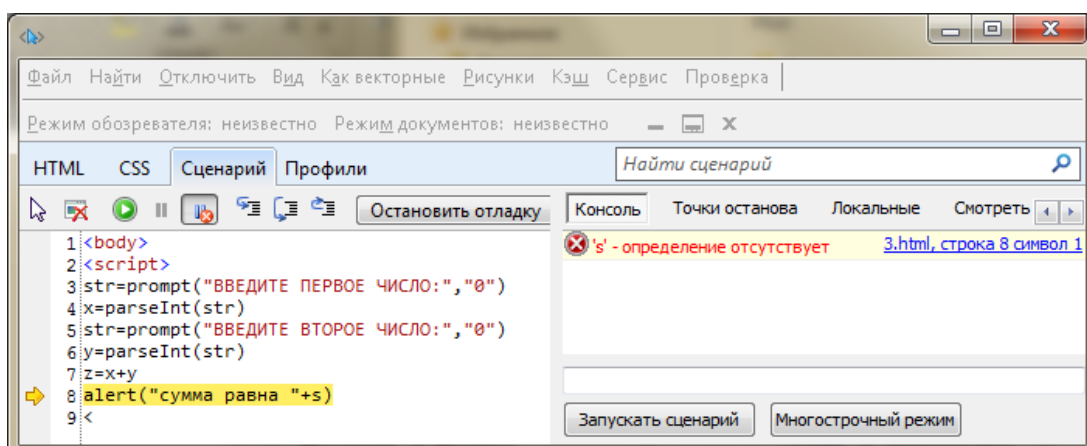
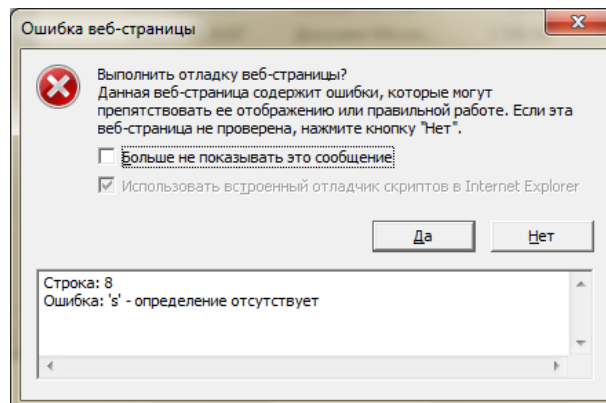
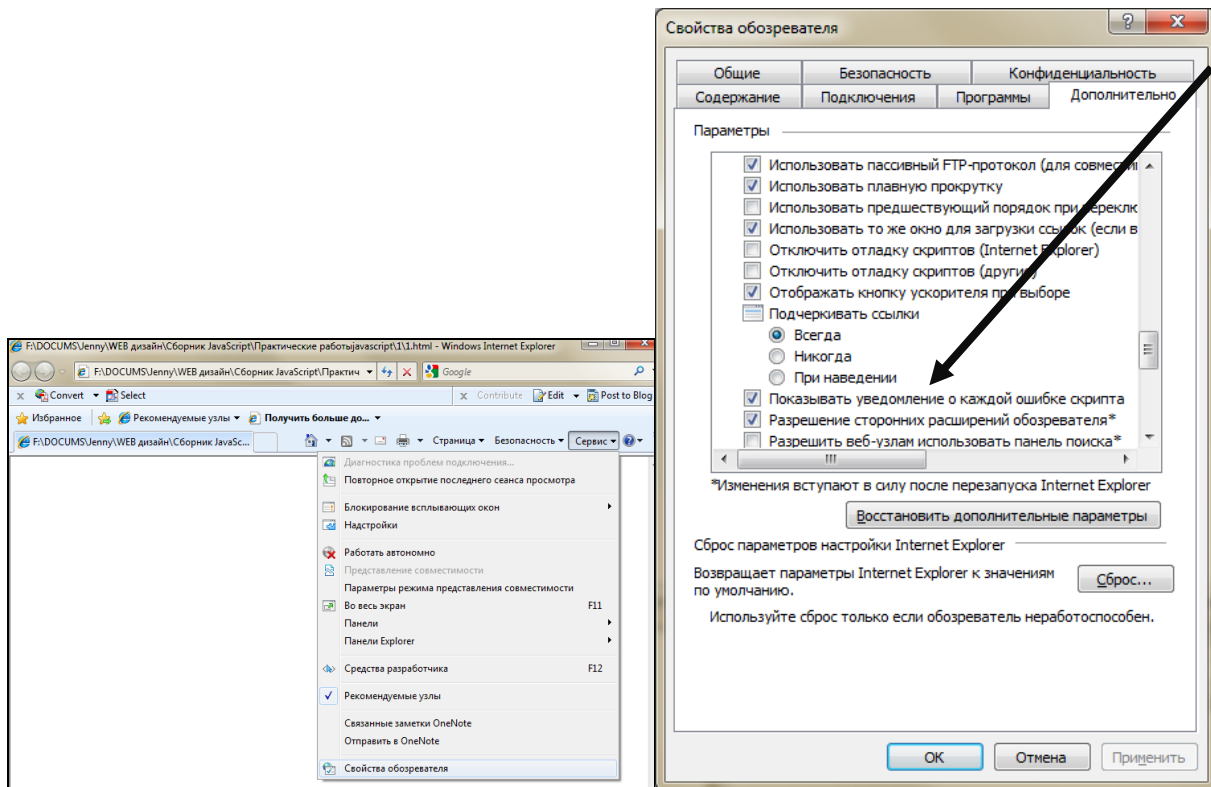
1. При загрузке документа;
2. Сразу после загрузки документа;
3. В ответ на действия пользователя.

Чаще всего мы будем пользоваться третьим способом.

JavaScript

Отладка скрипта

Для того чтобы браузер выводил на экран сообщения об ошибке его необходимо настроить следующим образом:



JavaScript Синтаксис языка

- JavaScript –зависит от регистра. Имена JavaScript и Javascript - разные имена!! Все ключевые слова используют только нижний регистр.
- Требования к именам переменных такие же, как в Паскале.
- Операторы разделяются точкой с запятой, которую можно опустить, если оператор заканчивается символом новой строки (Enter).
- Комментарии:

```
// однострочный комментарий,
/*
..многострочный комментарий
*/
```

Типы данных

Переменные не имеют строгой типизации. Объявляются с помощью оператора var, который можно опускать, за исключением объявления локальных переменных в теле функции. Возможно объявление с одновременной инициализацией, например:

```
var s = 123 //объявляется целочисленная переменная x, имеющая десятичное значение 123
var d=3.14 //объявляется переменная с плавающей точкой имеющая десятичное значение
var str1='Строковая переменная'
var p=true //объявляется логическая переменная
```

Тип переменной может изменяться в процессе выполнения программы. Если в выражении содержатся и числовые и строковые переменные, то числовые переменные автоматически приводятся к строковому виду.

Математические операции

Оператор или операция	Действие	Пример	Значение, которое примет X
+	Сложение	x=100+5 str2='Начало' str1=str2+' конец'	105 Начало конец
-	Вычитание	X=100-5	95
*	Умножение	X=2*3	6
/	Деление	X=12/2	6
%	Остаток от деления (аналогично mod)	X=16%3	1
++	Значение увеличивается на 1	X=2; X++;	3
--	Значение уменьшается на 1	X=2; X--;	1

Операции сравнения

Операция	Действие	Пример	Аналогично, в Паскале
==	Равно	X==10	X=10
!=	Не равно	X!=5	X<>5
>	Больше	X>0	X>0
<	Меньше	X<4	X<4
>=	Больше либо равно	X>=Y	X>=Y
<=	Меньше либо равно	X<=5	X<=5

JavaScript Логические операции

Операция	Действие	Пример	Аналогично, в Паскале
&&	Аналогично логической операции and	X>=2 && y>=2	(X>=2)and(Y>=2)
	Аналогично логической операции or	x>0 y>0	(x>0)or(y>0)
!	Аналогично логической операции not	!(1 < x && x < 10)	Not((1 < x) and (x < 10))

Операторы присваивания

Оператор	Действие	Пример	Значение, которое примет X	Аналогично, в Паскале
=	Присваивает значение переменной	X=1000;	1000	X:=1000;
+=	Увеличивает значение переменной на указанную величину	X=1000; X+=100;	1100	X:=1000; X:=X+100;
-=	Уменьшает значение переменной на указанную величину	X=1000; X-=12;	988	X:=1000; X:=X-12;
=	Умножает значение переменной на указанную величину	X=1000; X=2;	2000	X:=1000; X:=X*2;
/=	Делит значение переменной на указанную величину	X=1000; X/=2;	500	X:=1000; X:=X/2;
%=	Делит значение переменной на указанную величину и возвращает остаток	X=1000; X%=5;	0	X:=1000; X:=X mod 5;

Условный оператор

JavaScript	Pascal
if (a==2) z=2; else z=3	if a=2 then z:=2 else z:=3;
if (x>=2 && x<=6) {y=0; z=1} else {y=1; z=0}	if (x>=2)and(x<=6) then begin y:=0; z:=1; end else begin y:=1; z:=0; end;

Отличия от Паскаля.

1. Then отсутствует.
2. Точка с запятой перед else ставится, если else находится на той же строке.
3. Все условие берется в скобки. Простые условия в скобки можно не брать.
4. Вместо операторных скобок (begin end) с той же целью используются фигурные скобки.

Ввод/вывод данных. Диалоговые окна

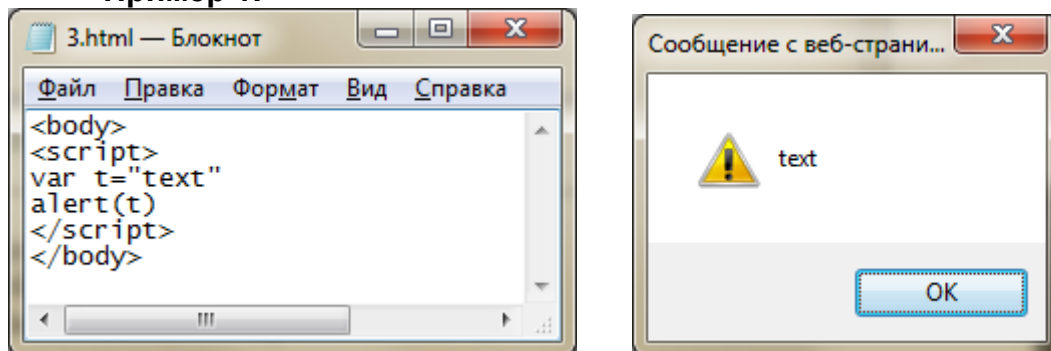
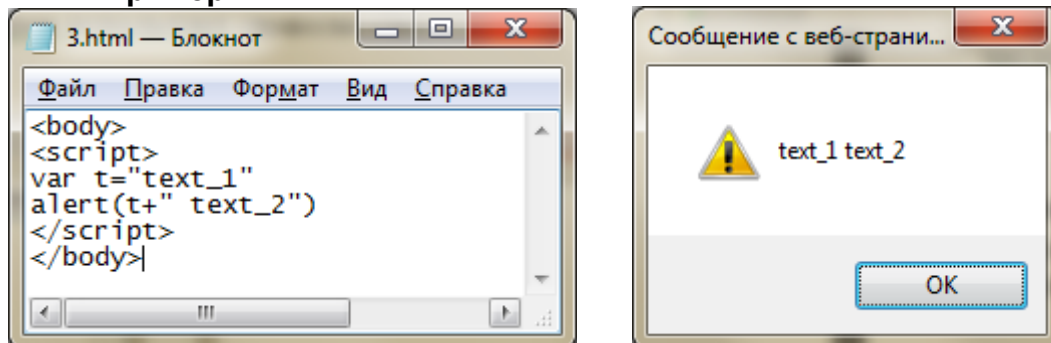
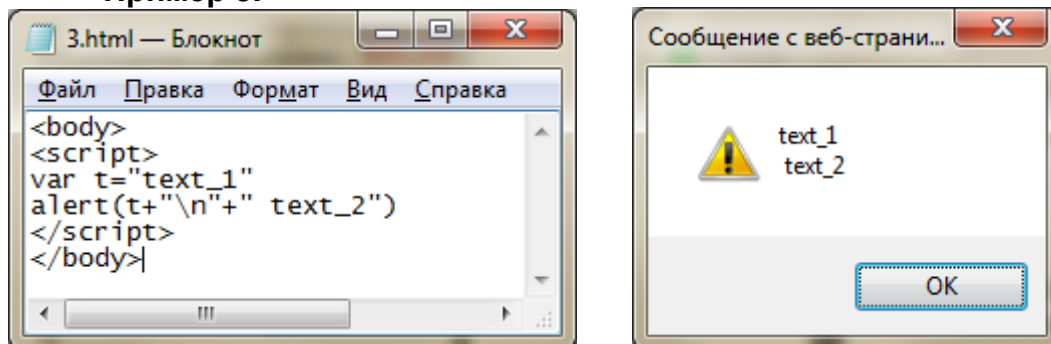
В JavaScript существует 3 функции (метода), позволяющие пользователю выводить диалоговые окна:

- alert
- confirm
- prompt

alert (“строка”)

Метод alert используется для вывода простейшего диалогового окна, содержащего текст сообщения и единственную кнопку "Ok". Программа выводит сообщение и ожидает нажатия кнопки. После нажатия на кнопку, программа начинает выполняться дальше.

Текст сообщения может сцепляться с любой текстовой переменной с помощью знака «+». Чтобы текст выводился в несколько строк используют символы «\n»

Пример 1.**Пример 2.****Пример 3.**

Confirm (“строка”)

Метод `confirm` используется в тех случаях, когда пользователь должен сделать выбор.

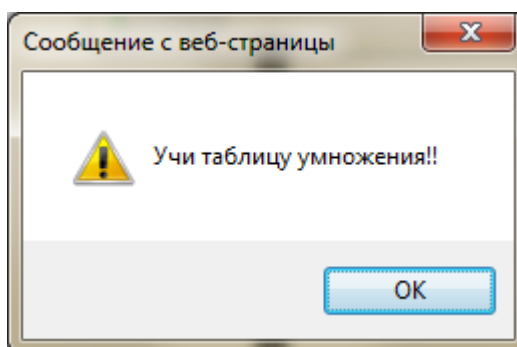
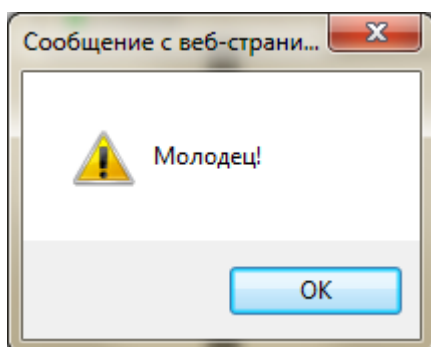
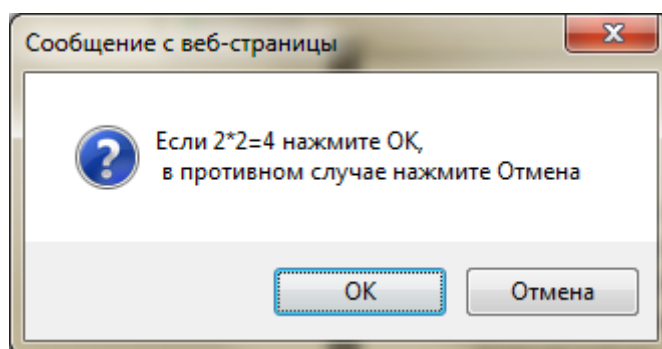
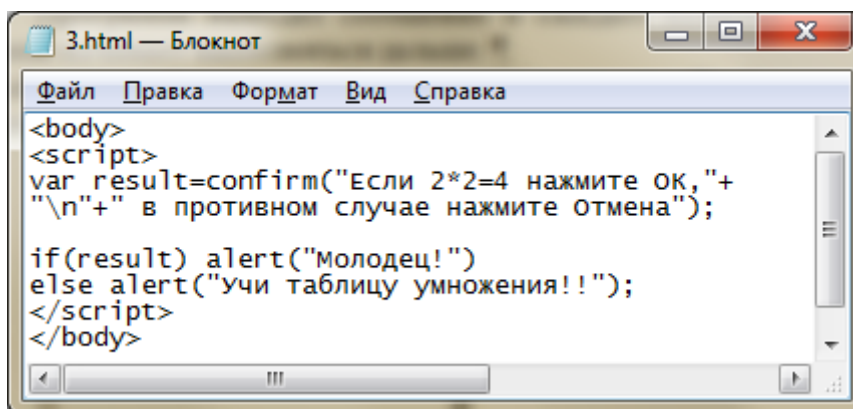
Метод `confirm` позволяет пользователю вывести диалоговое окно, содержащее текст вопроса и кнопки "ОК" и "Отмена".

Функция `confirm` возвращает логическое значение в зависимости от нажатой пользователем кнопки:

- "ОК" соответствует значению `true`,
- "Отмена" - значению `false`.

Как правило, результат работы функции присваивают логической переменной, для дальнейшего анализа, как это показано в примере.

Пример.



prompt (“строка1”, “строка2”)

Метод prompt используется в тех случаях, когда пользователю нужно ввести значение в переменную.

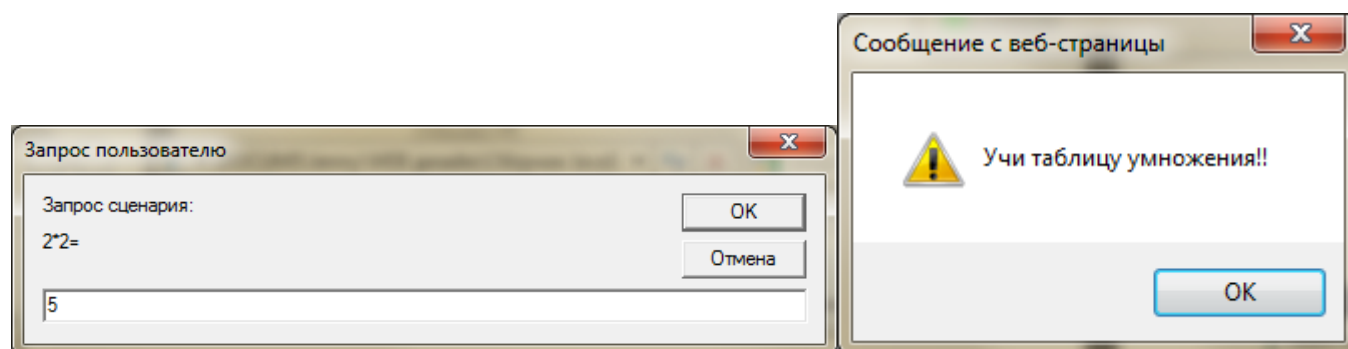
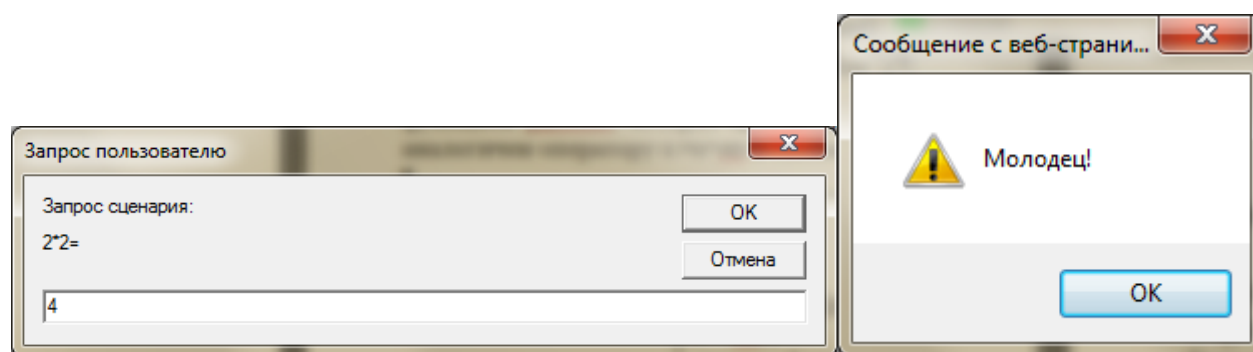
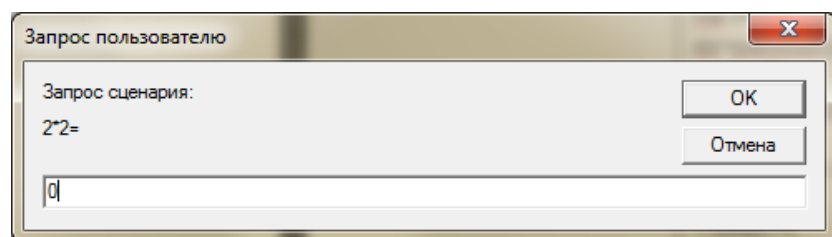
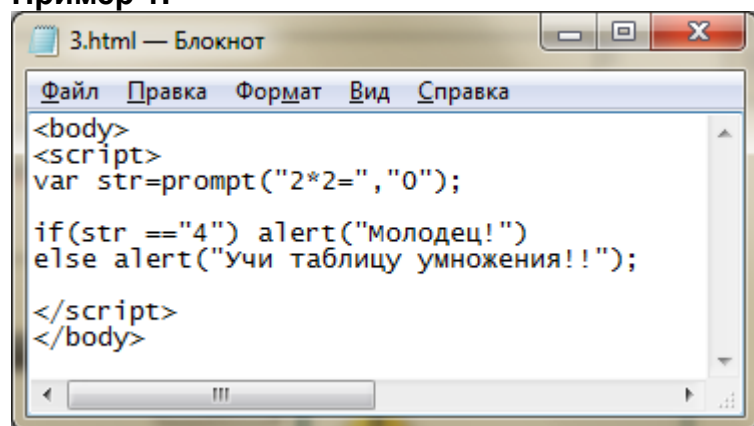
В окно выводится сообщение «строка1», в поле ввода помещается умалчиваемое значение «строка2».

Этот метод позволяет вывести диалоговое окно запроса на ввод данных. Результат работы функции присваивают переменной строкового типа.

Если введенные данные нужно использовать в арифметических выражениях, необходимо выполнить преобразование введенной строки к числовому типу. Это можно сделать при помощи следующих функций:

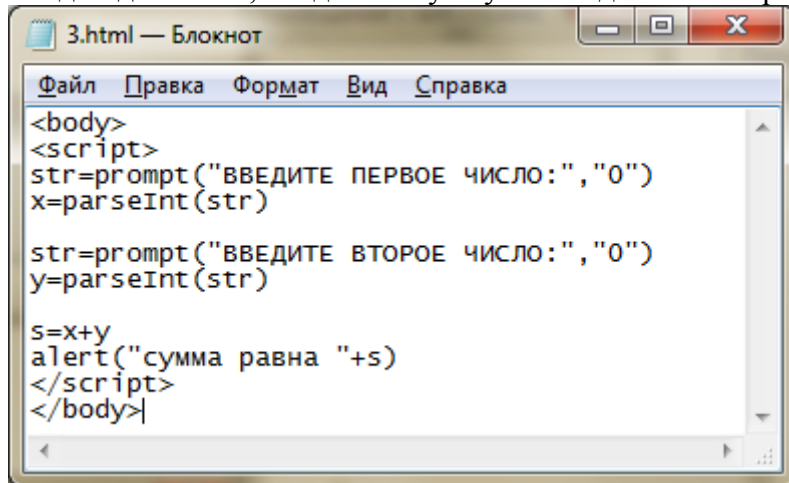
- parseInt("строка") - преобразует строку в целое число;
- parseFloat("строка") - преобразует строку в число с плавающей точкой.

Пример 1.

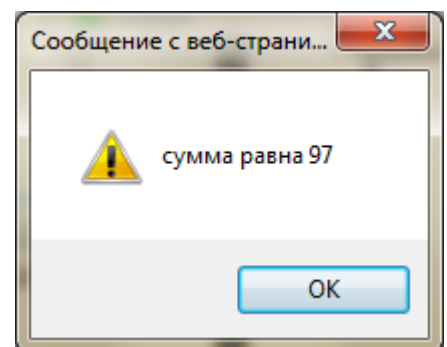
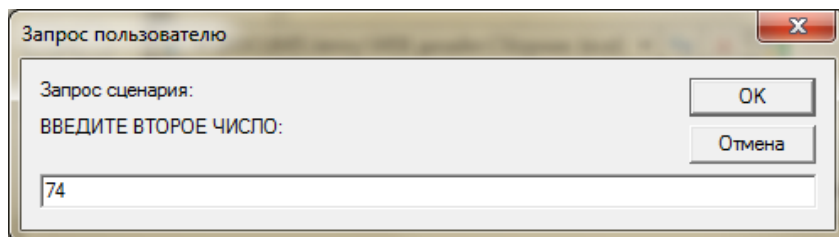
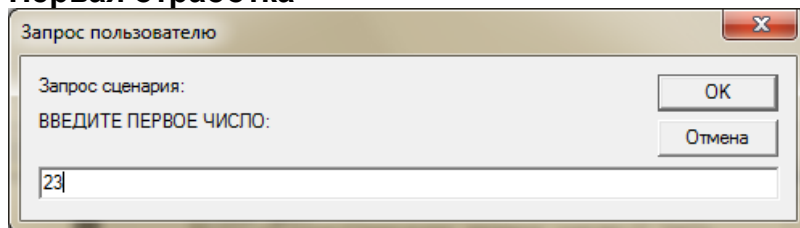
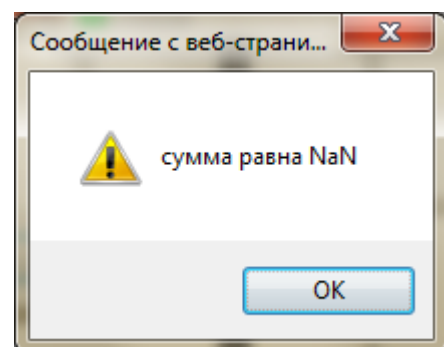
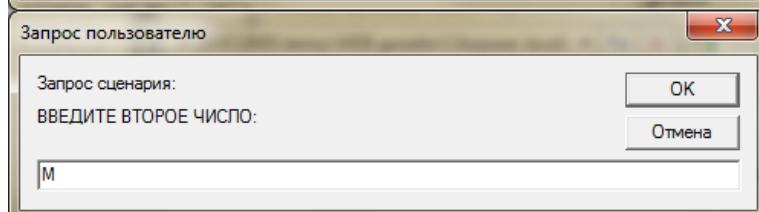
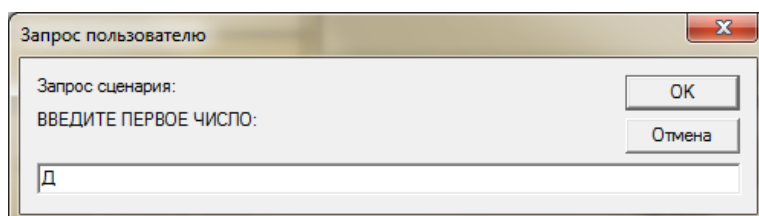


Пример 2.

Введем два числа, найдем их сумму и выведем ее на экран.



В данном скрипте переменные не описываются, что допускается. Переменная `str` будет иметь строковый тип, так как результат функции `prompt` должен быть строкового типа. Функция `parseInt` преобразовывает переменную `str` строкового типа в переменную `x` числового типа. Переменная `s` в операторе присваивания имеет числовой тип, так как переменные `x` и `y` имеют числовой тип. Переменная `s` в функции `alert` будет преобразована в строковый тип, так как параметр этой функции должен быть строкой.

Первая отработка**Вторая отработка**

Буквы не могут быть преобразованы в числа. Поэтому переменные `x`, `y` и `s` не будут иметь значений.

Когда переменная не имеет значения, то выводится NaN.

Метод document.write()

JavaScript это объектно-ориентированный язык. Основной единицей в объектно-ориентированном языке является объект, который объединяет в себе данные (свойства) и средства обработки этих данных (методы). Если говорить образно, то объекты – это «существительные», свойства объекта – это «прилагательные», а методы объекта – это «глаголы». Значения свойств объектов можно изменять.

Про JavaScript говорят, что в нем все объект. А именно: объектами являются окно, в котором открывается документ, сам документ, все элементы документа и даже свойства этих элементов. Есть также специальные встроенные объекты. Для упорядочивания огромного количества объектов создатели браузеров придумали объектную модель документа. Эта модель является структурой организации объектов на странице.

Объект document соответствует всему HTML-документу.

Изучим один метод этого объекта, позволяющий динамически формировать документ.

Метод document.write(“строка html-кода”) - выводит строку в окно документа.

Метод document.writeln(“строка html-кода”) - выводит строку в окно документа, в конце выводится символ "пробел".

Метод, применяемый к объекту, пишется после имени объекта через точку.

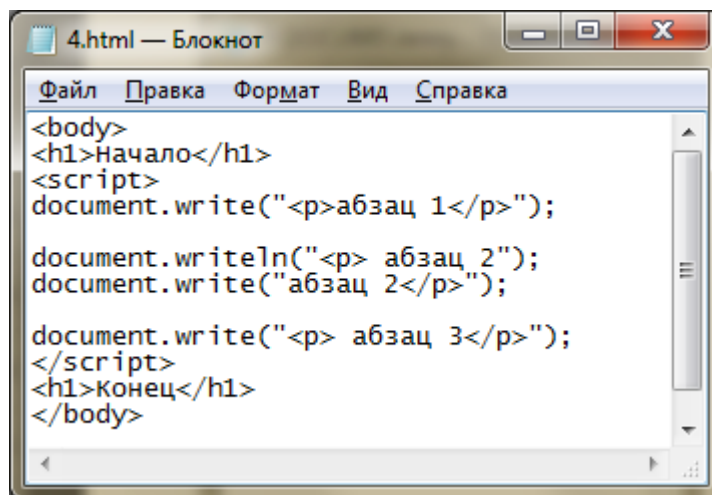
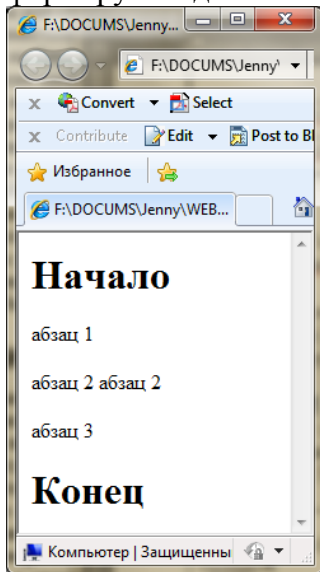
Содержимое строки должно быть в кавычках или это может быть объединение (сумма) нескольких строк или строковых переменных.

Строка должна содержать элементы разметки страницы (теги и их содержимое).

Метод выполняется в процессе загрузки документа.

Пример 1.

В данном примере заголовки (Начало и Конец) находятся в документе, а текст абзацев формируется динамически и затем выводится в документ.



Синтаксис языка

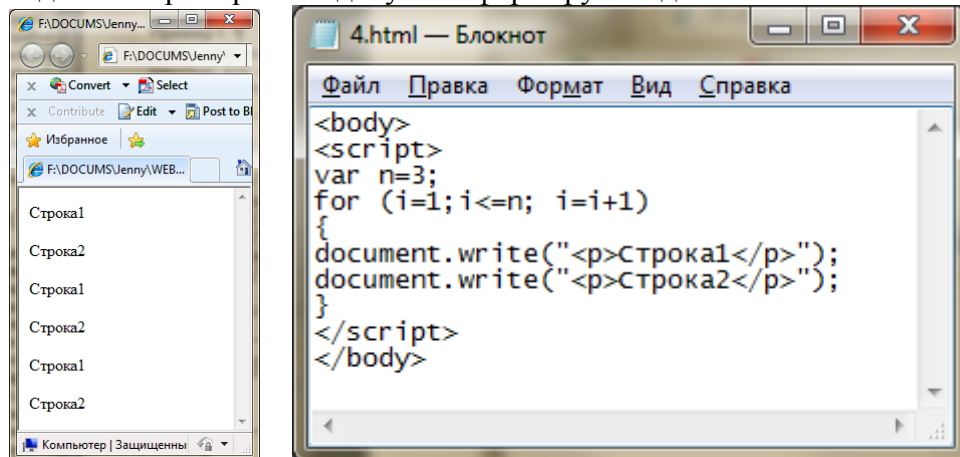
Оператор цикла for

Отличия от Паскаля.

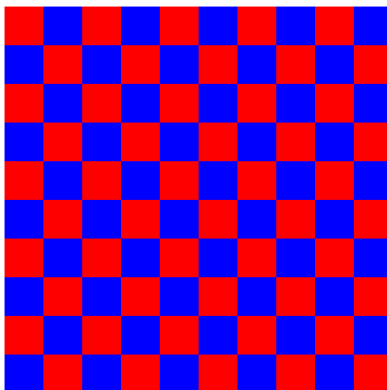
1. Возможен цикл с любым шагом.
2. Отсутствует do.
3. Начальное и конечное значение параметра, а также шаг указываются в скобках в заголовке цикла.
4. Вместо операторных скобок (begin end) с той же целью используются фигурные скобки.

Пример 2.

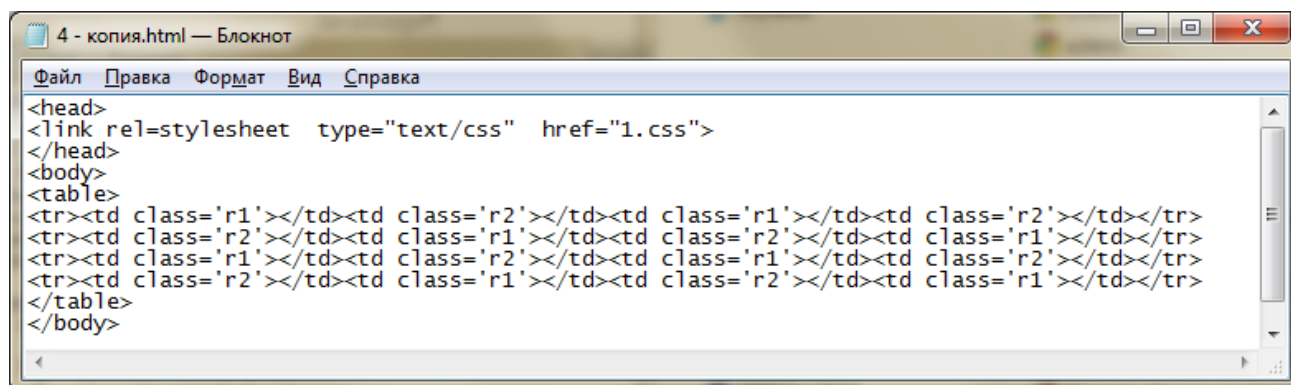
В данном примере весь документ формируется динамически.

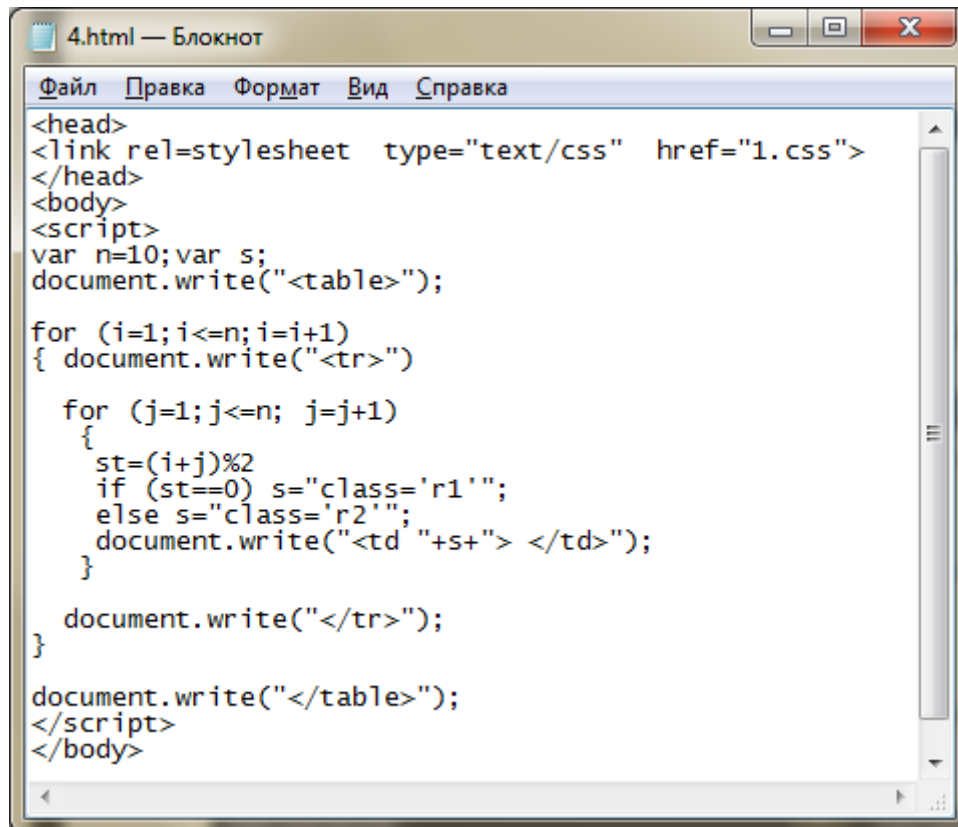
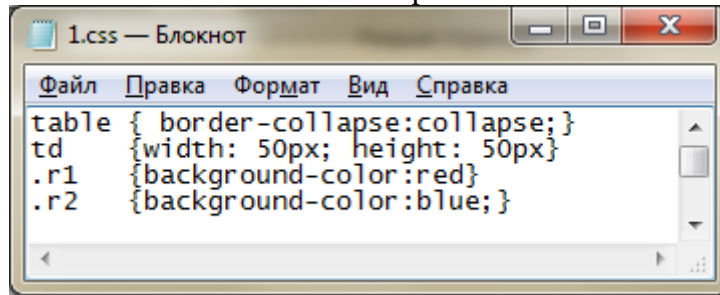
**Пример 3.**

Создадим таблицу 10 на 10. Высота и ширина ячейки 50 пикселей. Закрасим таблицу в шахматном порядке. Таблицу будем формировать динамически, с помощью метода write. Для закрашки таблицы используем вложенные операторы цикла.



Фактически, нужно сформировать документ аналогичный данному. Правда, в этой таблице всего четыре строки и четыре столбца.



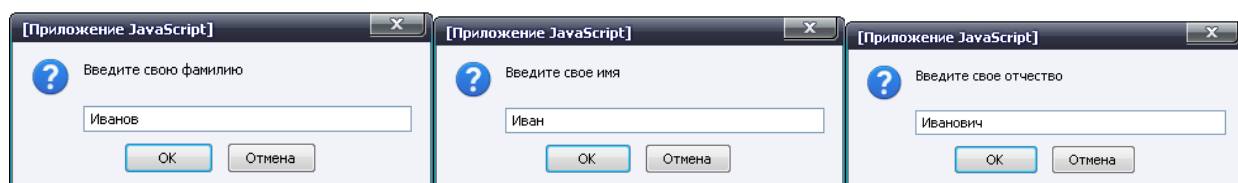


До цикла и после цикла выводятся теги `<table>` и `</table>`. Во внешнем цикле в документ выводятся теги `<tr>` и `</tr>`, формирующие строку таблицы. Во внутреннем цикле в документ выводятся теги `<td>` и `</td>`, формирующие ячейки таблицы. От четности `st` зависит значение текстовой переменной `s`, которая определяет класс ячейки. Ячейка класса `r1` имеет красную заливку, ячейка класса `r2` заливается синим цветом.

Вариант 1

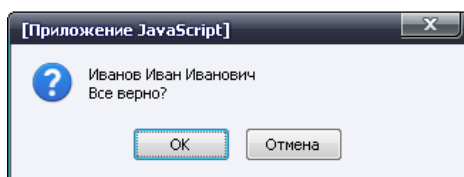
Задание 1

Введите свою фамилию, имя и отчество.

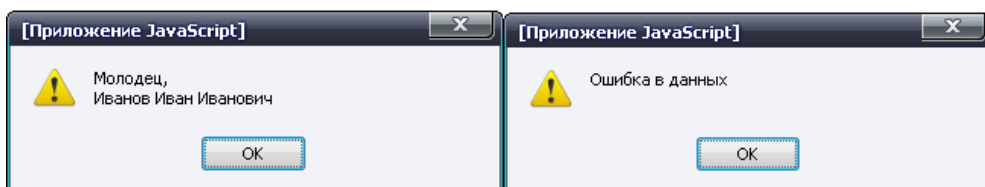


JavaScript

Запросите подтверждение.

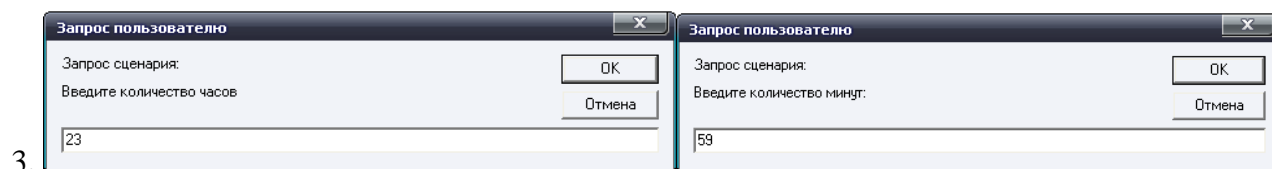
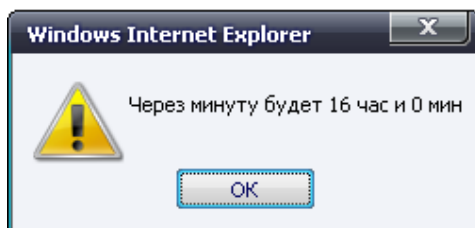
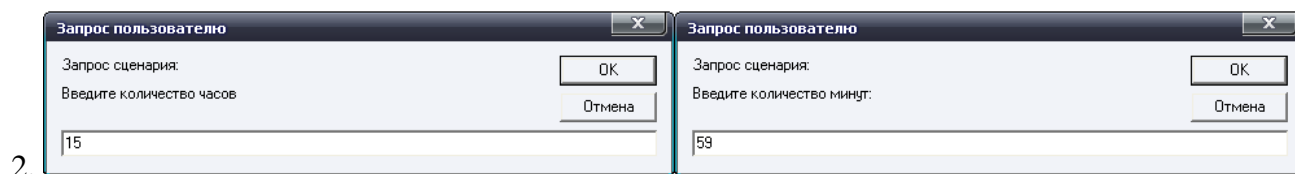
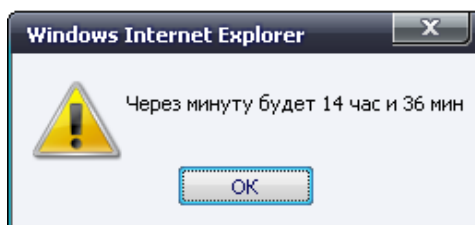
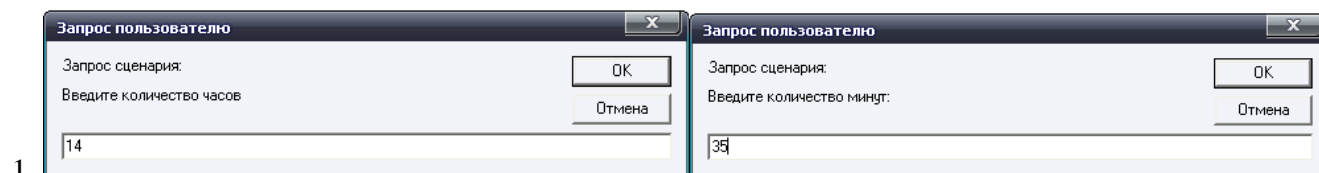


Если все верно, то вывести приветствие, если нет, вывести сообщение об ошибке.

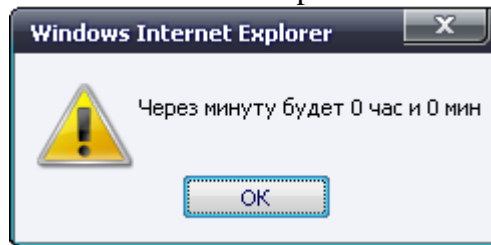


Задание 2

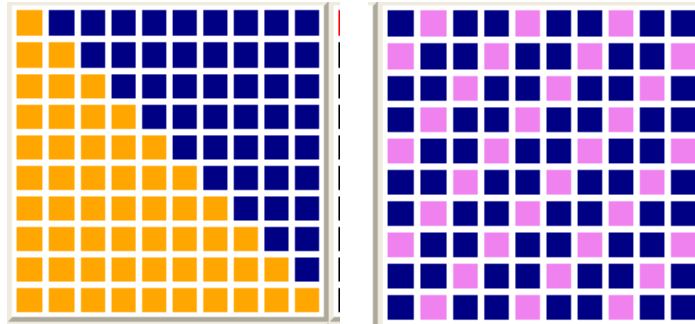
Введите время в часах и минутах. Определите время, которое будет через минуту. Используйте вложенные операторы IF. Возможны три случая:



JavaScript



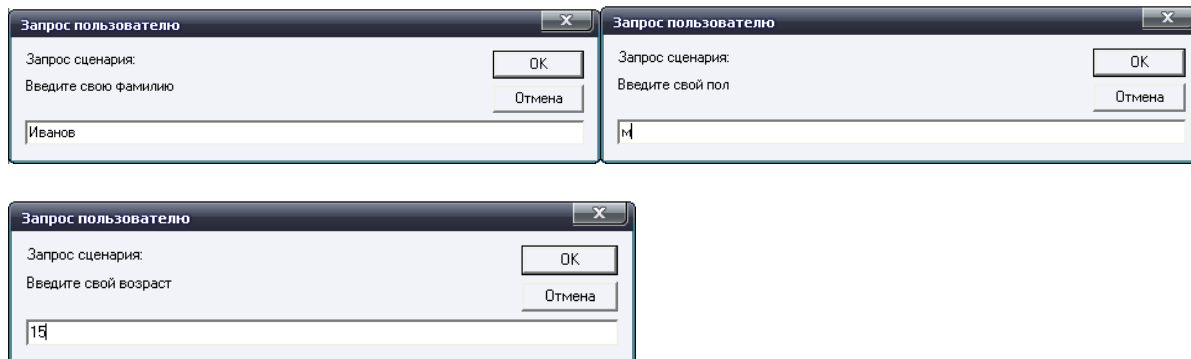
Задание 3



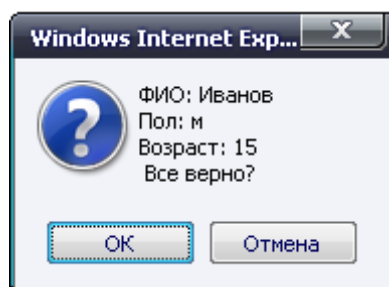
Вариант 2

Задание 1

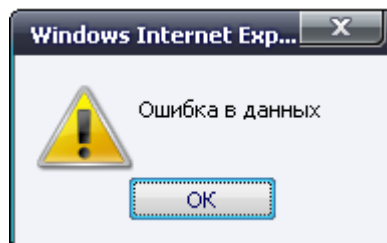
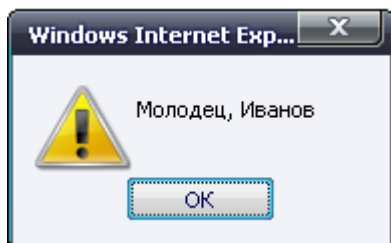
Введите свою фамилию, пол и возраст.

Three separate "Запрос пользователю" (User Prompt) dialog boxes. The first prompts for a surname ("Введите свою фамилию") with "Иванов" entered. The second prompts for gender ("Введите свой пол") with "м" entered. The third prompts for age ("Введите свой возраст") with "15" entered. Each dialog has "ОК" and "Отмена" buttons.

Запросите подтверждение:



Если все верно, то вывести приветствие, если нет, вывести сообщение об ошибке.



Задание 2

Дано два числа x , y и знак арифметической операции (+, -, *, /). Найти $x+y$, $x-y$, $x*y$, x/y , в зависимости от введенного знака. В случае ошибки в знаке или деления на 0 вывести сообщение об ошибке. Примеры ввода данных:

1.

Запрос пользователю

Запрос сценария:

Введите первое число

Запрос пользователю

Запрос сценария:

Введите второе число

Запрос пользователю

Запрос сценария:

Введите знак

Windows Internet Exp...

45/15=3

2.

Запрос пользователю

Запрос сценария:

Введите первое число

Запрос пользователю

Запрос сценария:

Введите второе число

Запрос пользователю

Запрос сценария:

Введите знак

Windows Internet Exp...

ошибка

Задание 3

