

Основы JavaScript

(!) В качестве онлайн-справочников можно пользоваться <http://javascript.ru/> или <https://learn.javascript.ru/>

(!) В качестве редактора кода можно воспользоваться Visual Studio или Code Studio (Создать файл, выбрать нужный тип .html или .js) или ресурсом <https://jsfiddle.net/>

Web-документ, отображаемый браузером, является результатом интерпретации кода на разных языках:

- html используется для описания структуры,
- css – описания внешнего вида,
- javascript - описания поведения документа, его реакции на действия пользователя.

В браузер встроен интерпретатор JavaScript.

Основные способы добавления кода JavaScript на страницу:

1. В разделе <BODY>...</BODY>

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

2. В разделе <HEAD>...</HEAD> в случае, когда 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. Код javascript может храниться во внешнем файле, тогда данный файл может быть подключен к странице с помощью тега script в разделе <HEAD>...</HEAD>. Расширение данного файла – .js

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

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

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

Существует несколько способов определения момента запуска сценария, например

- при загрузке документа,
- после загрузки документа,
- в ответ на действия пользователя.

Основы синтаксиса языка

- JavaScript является регистрозависимым языком. 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

JavaScript является языком без строгой типизации. Объявляются с помощью оператора var, который можно опускать, за исключением объявления локальных переменных в теле функции. Возможно объявление с одновременной инициализацией.

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

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;

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

- alert()
- confirm()
- prompt()

Также для вывода отладочной информации можно воспользоваться методом **console.log()**, аргументом которого являются данные для вывода. Открытие консоли выполняется через открытие инструмента разработчика в браузере (обычно горячая клавиша – F12).

Метод document.write()

JavaScript это объектно-ориентированный язык. Основной единицей является объект, который объединяет в себе данные (свойства) и средства обработки этих данных (методы). Значения свойств объектов можно изменять.

Про JavaScript говорят, что в нем всё является объектом. Например,

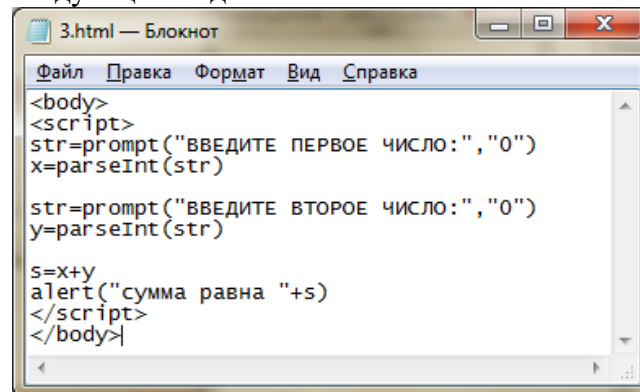
- окно, в котором открывается документ,
- сам документ,
- все элементы документа
- свойства этих элементов.

Можно выделить специальные встроенные объекты. Для упорядочивания огромного количества объектов существует объектная модель документа – Document Object Model (DOM). Данная модель является структурой организации объектов на странице.

Объект document соответствует всему HTML-документу. Одним из методов объекта document является метод document.write("строка html-кода"), который позволяет выводить строку в окно документа. Также существует метод document.writeln ("строка html-кода").

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.

Задание 1. Требуется найти сумму двух чисел, введенных пользователем. Создайте html страницу, содержащей следующий код



```
<body>
<script>
str=prompt("ВВЕДИТЕ ПЕРВОЕ ЧИСЛО:", "0")
x=parseInt(str)

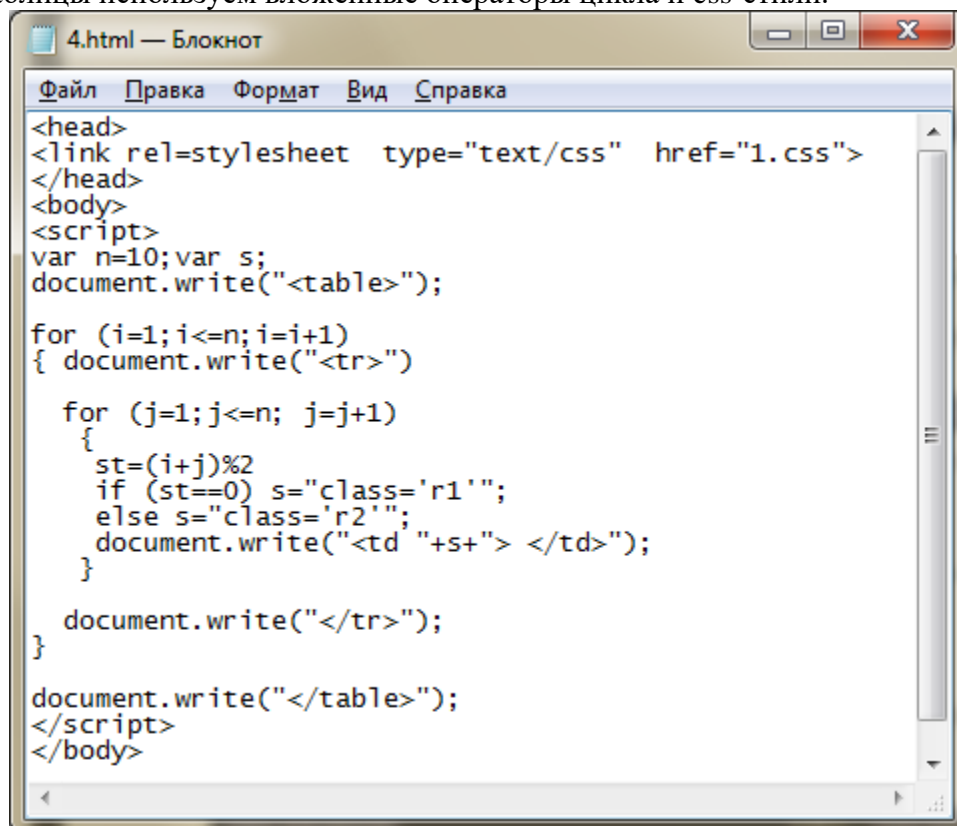
str=prompt("ВВЕДИТЕ ВТОРОЕ ЧИСЛО:", "0")
y=parseInt(str)

s=x+y
alert("сумма равна "+s)
</script>
</body>
```

Проверьте работу кода. Что будет если ввести строку вместо одного из чисел? Ответ аргументируйте.

Задание 2. Создайте новый html-документ.

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 и css-стили.



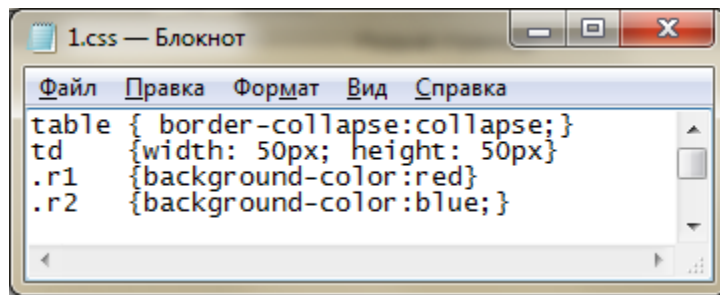
```
<head>
<link rel=stylesheet type="text/css" href="1.css">
</head>
<body>
<script>
var n=10;var s;
document.write("<table>");

for (i=1;i<=n;i=i+1)
{ document.write("<tr>")

  for (j=1;j<=n; j=j+1)
  {
    st=(i+j)%2
    if (st==0) s="class='r1'";
    else s="class='r2'";
    document.write("<td "+s+"> </td>");
  }

  document.write("</tr>");
}

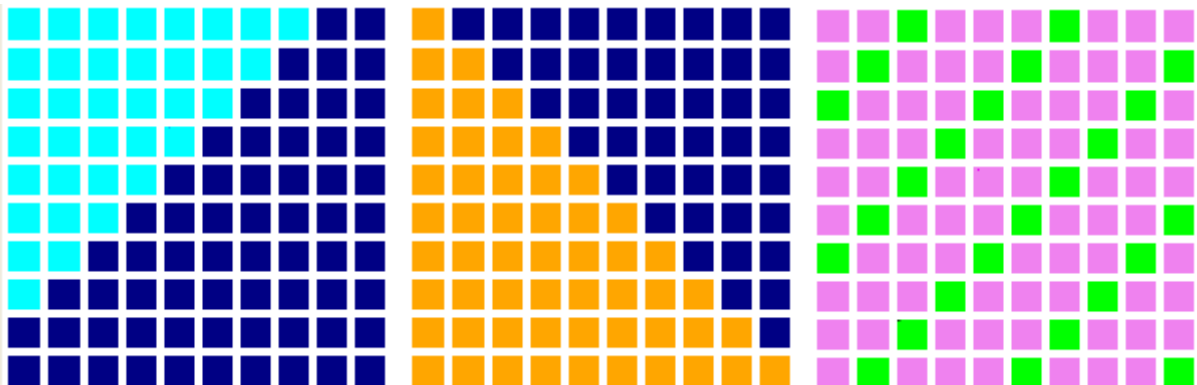
document.write("</table>");
</script>
</body>
```



Проверьте работу кода.

Задание 3. Дано два числа x , y и знак арифметической операции (+, -, *, /). Найдите $x+y$, $x-y$, $x*y$, x/y , в зависимости от введенного знака. В случае ошибки в знаке или деления на 0 вывести сообщение об ошибке.

Задание 4. Создайте таблицы посредством JavaScript-код и закрасьте их соответствующим образом:



Задание 5. Пусть есть массив объектов:

```
1 var users = [{
2   name: "Вася",
3   surname: 'Иванов',
4   age: 20
5 }, {
6   name: "Петя",
7   surname: 'Чапаев',
8   age: 25
9 }, {
10  name: "Маша",
11  surname: 'Медведева',
12  age: 18
13 }];
```

Сортировка по нужному полю объектов в массиве может выполняться так:

```
1 // по полю name (Вася, Маша, Петя)
2 users.sort(function(a, b) {
3   return a.name > b.name ? 1 : -1;
4 });
5
6 // по полю age (Маша, Вася, Петя)
7 users.sort(function(a, b) {
8   return a.age > b.age ? 1 : -1;
9 });
```

Напишите более универсальную функцию, которая позволяла бы сортировать по указанному полю.

Расширьте массив новыми элементами и протестируйте работу написанной вами функции.

Задание 6.

1. Создайте функцию `filter(arr, func)`, которая получает массив `arr` и возвращает новый, в который входят только те элементы `arr`, для которых `func` возвращает `true`.
2. Создайте набор «готовых фильтров»: `inBetween(a,b)` – «между a,b», `inArray([...])` – «в массиве [...]».

Использование должно быть таким:

- `filter(arr, inBetween(3,6))` – выберет только числа от 3 до 6,
- `filter(arr, inArray([1,2,3]))` – выберет только элементы, совпадающие с одним из значений массива.

Пример, как это должно работать:

```
1  /* .. ваш код для filter, inBetween, inArray */
2  var arr = [1, 2, 3, 4, 5, 6, 7];
3
4  alert(filter(arr, function(a) {
5    return a % 2 == 0
6  })); // 2,4,6
7
8  alert( filter(arr, inBetween(3, 6)) ); // 3,4,5,6
9
10 alert( filter(arr, inArray([1, 2, 10])) ); // 1,2
```

Задание 7. Создайте функцию `find(arr, value)`, которая ищет в массиве `arr` значение `value` и возвращает его номер, если найдено, или `-1`, если не найдено. Например:

```
1  arr = ["test", 2, 1.5, false];
2
3  find(arr, "test"); // 0
4  find(arr, 2); // 1
5  find(arr, 1.5); // 2
6
7  find(arr, 0); // -1
```

Задание 8. Напишите функцию, которая позволяет, используя функцию `sort`, «перетрясти» элементы массива в случайном порядке.

```
1  var arr = [1, 2, 3, 4, 5];
2
3  arr.sort(ваша функция);
4
5  alert( arr ); // элементы в случайном порядке, например [3,5,1,2,4]
```

Задание 9. Анаграммы – слова, состоящие из одинакового количества одинаковых букв, но в разном порядке.

Например:

```
1  воз - зов
2  киборг - гробик
3  корсет - костер - сектор
```

Напишите функцию `aclean(arr)`, которая возвращает массив слов, очищенный от анаграмм.

Например:

```
1  var arr = ["воз", "киборг", "корсет", "зов", "гробик", "костер", "сектор"];
2
3  alert( aclean(arr) ); // "воз,киборг,корсет" или "зов,гробик,сектор"
```

Из каждой группы анаграмм должно остаться только одно слово (не важно какое).