

ЛАБОРАТОРНАЯ РАБОТА № 3

ТЕМА: Встроенные объекты

Основной единицей в языке **JavaScript** является объект, который объединяет в себе данные (свойства) и средства обработки этих данных (методы).

Все объекты, которые используются в языке **JavaScript**, делятся на такие большие группы:

- Встроенные объекты базового языка
- Объекты документа.

Объект Math

В языке **JavaScript** существует специальный объект «Math», в котором собраны основные математические функции и константы. Все свойства и методы объекта являются статическими, поэтому сам объект создавать не нужно. В таблицах ниже приведён список некоторых из его методов.

Метод	Описание
abs (число)	Возвращает модуль (абсолютную величину) числа.
sqrt (число)	Возвращает квадратный корень из числа.
pow (основание , степень)	Возвращает результат возведения основания в указанную степень. Например, Math.pow(5, 3) вернёт $5^3=125$.
random ()	Возвращает псевдослучайное число в диапазоне от 0 до 1.
ceil (число)	Производит округление в большую сторону, то есть возвращает наименьшее целое число, большее либо равное аргументу.
floor (число)	Производит округление в меньшую сторону, то есть возвращает наибольшее целое число, меньшее либо равное аргументу.
round (число)	Округляет указанный аргумент до целочисленного значения.
cos (число)	Возвращает косинус числа.
sin (число)	Возвращает синус числа.
exp (число)	Возводит число e (основание натурального логарифма) в указанную степень.
log (число)	Возвращает натуральный логарифм числа.

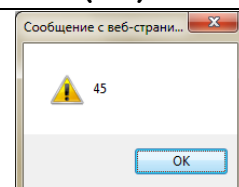
Чтобы задействовать метод (обычно это называют вызовом метода), в операторе **JavaScript** нужно сделать на него ссылку. Для этого используется синтаксис с использованием точки.

Очень часто используется также свойство этого объекта, которое имеет значение числа π , **Math.PI**

Пример

В данном скрипте к объекту **Math** применили метод **floor**. В качестве параметра метода использовали значение переменной **d1**. Для этого использовали конструкцию **Math.floor(d1)**. Этот метод отбрасывает дробную часть у аргумента. Это значение присваивается переменной **d2** и выводится в окно с помощью метода **alert(d2)**.

```
<script>
d1=45.6;
d2=Math.floor(d1);
alert(d2);
</script>
```



Объект Date

Объект **Date** предназначен для манипуляций с датами и временем. Его примитивным значением является число, равное количеству миллисекунд относительно базового времени, равного полуночи 1 января 1970 г.

День состоит из 86400000 миллисекунд.

Объект и экземпляр объекта

Объект -- это шаблон. Экземпляр объекта -- это рабочая копия.

Например, объектом является комплект документации на заводе, по которой изготавливаются телевизоры. Сам телевизор является экземпляром этого объекта. Все телевизоры, которые сходят с конвейера, имеют одни и те же свойства изображения и одни и те же методы управления этими свойствами.

Создание экземпляра объекта Date

Для создания экземпляра объекта используется ключевое слово **new**.

Создание экземпляра объекта с текущей датой и временем: **new Date ()**

```
var a = new Date ();
```

В переменной «a» текущая дата и время.

Создание экземпляра объекта с датой и временем, заданными аргументами конструктора:

```
new Date (год, месяц, день [, часы [, минуты [, секунды [, мс] ] ] ] ] )
```

Здесь:

- год — числовое выражение, задающее полный номер года (например, 1988, а не 88);
- месяц — числовое выражение, задающее номер месяца (0 = январь, 1 = февраль, ..., 11 = декабрь);
- день — числовое выражение, задающее номер дня в месяце от 1 до 31;
- часы — необязательное числовое выражение, задающее номер часа от 0 до 23;
- минуты — необязательное числовое выражение, задающее номер минуты от 0 до 59;
- секунды — необязательное числовое выражение, задающее номер секунды от 0 до 59;
- мс — необязательное числовое выражение, задающее номер миллисекунды от 0 до 999.

Например,

```
var c = new Date (1958, 4, 21, 10, 15);
```

В переменной «c» дата – 21 мая 1958 года и время 10 часов 15 мин. Нумерация месяцев начинается с 0.

Методы объекта Date

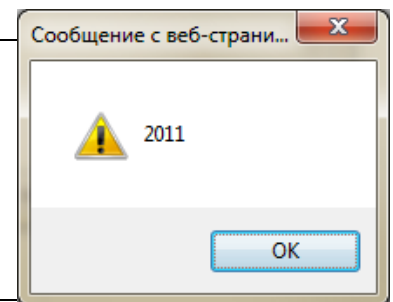
Метод это действие которое выполняется для объекта или с объектом. По своей сути это команда, но ее действия связаны с определенным объектом. У каждого объекта может быть много методов.

Метод	Описание
getTime()	Возвращает примитивное значение объекта.
getFullYear()	Возвращает номер года по местному времени
getMonth()	Возвращает месяц по местному времени.
getDate()	Возвращает день месяца по местному времени.
getDay()	Возвращает день недели по местному времени.
getHours()	Возвращает часы по местному времени.
getMinutes()	Возвращает минуты по местному времени.
getSeconds()	Возвращает секунды по местному времени.

Пример 1

Выведем в окно номер текущего года.

```
<body>
<script>
var today= new Date();
var god=today.getFullYear();
alert(god);
</script>
</body>
```

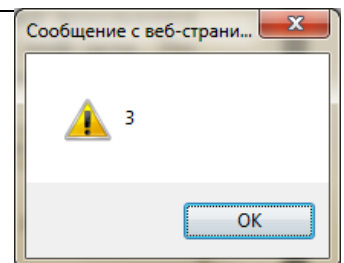


В данном скрипте создается экземпляр объекта **Date**. Он имеет имя **today** и содержит текущую дату. Для применения метода **getFullYear()** к объекту **today** используется конструкция **today.getFullYear()**. Она возвращает значение года из этой даты. Это значение присваивается переменной **god** и выводится в окно с помощью метода **alert(god)**.

Пример 2

Выведем в окно номер дня недели для даты 21 мая 1958 года.

```
<body>
<script>
var dr = new Date(1958, 4, 21);
dn=dr.getDay();
alert(dn);
</script>
</body>
```



В данном скрипте создается экземпляр объекта **Date**. Он имеет имя **dr** и содержит дату 21 мая 1958 года. Для применения метода **getDay()** к объекту **dr** используется конструкция **dr.getDay()**. Она возвращает значение дня недели для этой даты. Это значение присваивается переменной **dn** и выводится в окно с помощью метода **alert(dn)**. Выводится число 3, то есть среда.

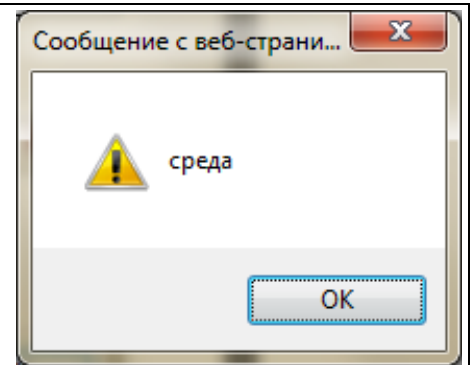
Объект Array

Объект **Array** предназначен для хранения массивов данных. Массив -- это упорядоченный набор элементов. Доступ к отдельному элементу производится по имени и индексу (номеру). Нумерация элементов в JavaScript начинается с нуля.

Пример 3

Выведем в окно названия дня недели для даты 21 мая 1958 года. Для хранения названий дней недели создадим **dayn** – экземпляр объекта **Array**. В качестве индекса для выбора из массива названия дня недели используется вычисленное значение переменной **dn**.

```
<body>
<script>
var dr = new Date(1958, 4, 21);
dn=dr.getDay()
var dayn=new Array('воскресенье',
'понедельник', 'вторник', 'среда',
'четверг', 'пятница', 'суббота')
alert(dayn[dn])
</script>
</body>
```



Ссылка на объект документа. Метод getElementById

Для размещения значений, возвращаемых описанными методами на Web-странице, необходимо уметь обратиться к элементам этой страницы (документа).

Наилучший способ заключается в том, чтобы обратиться к объекту по имени. Поэтому нужно каждому элементу (объекту) присвоить имя. Для этого используется атрибут **id** (или **name**).

Примеры:

```
<p id="first">
<img id='logo1' name="logo1" src='1.gif'>
```

Рекомендуется назначать одно и то же значение атрибутам **id** и **name**.

Для обращения к элементу, который имеет имя, в **JavaScript** используется следующая конструкция:

```
document.getElementById("Имя_объекта")
```

Например, если мы хотим обратиться к элементу с именем **"first"**, то будем использовать следующую ссылку:

```
document.getElementById("first")
```

Обратите внимание, что все ключевые слова чувствительны к регистру. В названии метода используются три буквы в верхнем регистре.

Метод **getElementById()** относится к объекту **document**. С его помощью мы можем найти любой поименованный элемент документа. Для разделения элементов иерархической ссылки используется точка.

Обращение к свойствам объекта

Каждый объект обладает некоторыми характеристиками, которые называются свойствами. Например: пусть объект текстовое поле описывается так:

```
<input type='text' id='entry' name='entry' value='User Name?'>
```

Для получения доступа к свойству объекта используется тот же тип синтаксиса с точками. Ссылка на данное свойство состоит из ссылки на данный объект плюс еще одно расширение, указывающее на нужное свойство. Например, чтобы обратиться к свойству **value** элемента текстовое поле, который имеет имя **'entry'** нужно записать выражение вида:

```
document.getElementById("entry").value
```

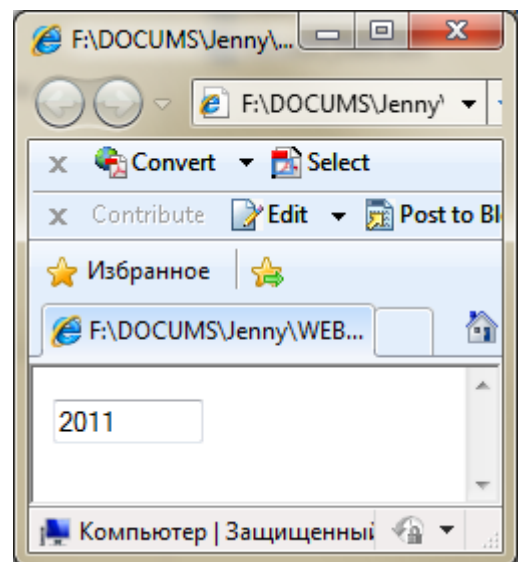
Или можно сопоставить наш объект с переменной **ob** и присвоить свойству **value** этого объекта новое значение **'Введите имя '**.

```
var ob=document.getElementById("entry")
ob.value='Введите имя ';
```

Пример 4

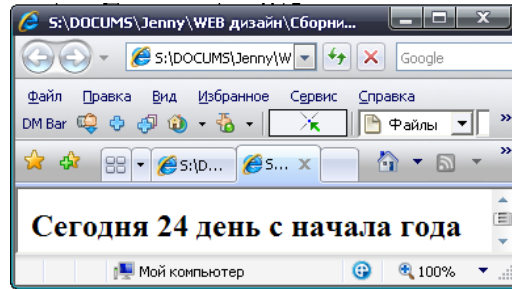
Выведем год извлеченный из текущей даты с помощью метода не в окно, как в примере 1, а в текстовое поле формы. Это текстовое поле имеет имя **id='f1'**. Наш скрипт оформлен в виде функции **gd()**, которая вызывается при загрузке документа. В этой функции с помощью метода **getElementById** определена переменная **p1**, которая есть наш объект текстовое поле. У этого объекта есть свойство **value**, которому мы присваиваем значение **god**.

```
<html>
<head>
<script>
function gd()
{
var today= new Date();
var god=today.getFullYear();
p1=document.getElementById('f1');
p1.value=god;
}
</script>
</head>
<body onLoad="gd()">
<form>
<input type='text' id='f1' size=8>
</form>
</body>
</html>
```



Пример 5

Создать HTML-страницу, которая при загрузке сообщает, сколько дней прошло от начала года.



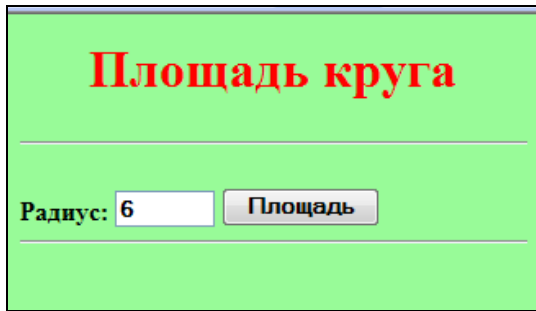
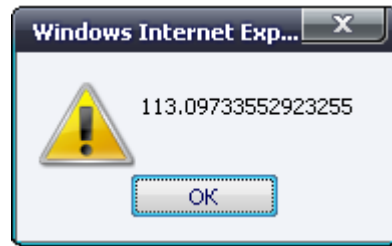
Для решения задачи нужно определить:

- количество миллисекунд от 1 января 1970 г до текущей даты
- количество миллисекунд от 1 января 1970 г до 1 января текущего года.,
- из первого вычесть второе и разделить на 86400000 (количество миллисекунд в сутках).

```
<html>
<head>
</head>
<body>
<script>
// переменная today имеет значение текущее время и дата
var today = new Date();
// переменная y- текущий год
var y = today.getFullYear();
// переменная yearBegin имеет значение начало текущего года
var yearBegin = new Date(y,0,1);
// переменная d - количество миллисекунд от начала года.
var d = today.getTime() - yearBegin.getTime();
// Переведем d в сутки
d =d/ (86400000);
// прибавим 1, иначе 1 января будет нулевым днем
d=d+1;
// отбросим дробную часть
d=Math.floor(d)
// печать результата
document.write("<h2>Сегодня "+d+" день с начала года</h2>");
</script>
</body>
</html>
```

Пример 6

Напишите приложение для вычисления площади круга. Радиус для вычисления вводить в текстовое поле формы. Площадь должна вычисляться при нажатии кнопки.

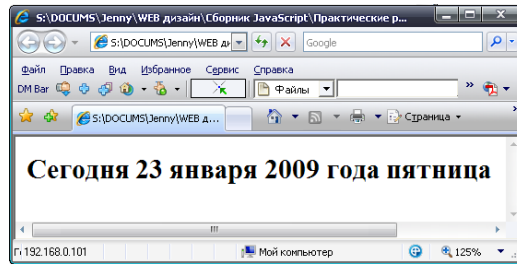



```
<html>
<head>
<style type="text/css">
h1 {color:red;text-align:center;}
body {background-color:PaleGreen; font-weight:bold;}
input {font-weight:bold;}
</style>
<script>
function sqr()
{
// переменная p объект с именем t
var p=document.getElementById('t');
// переменная r содержит значение свойства value этого объекта
r=p.value;
var s= Math.PI*r*r;
alert(s)
}
</script>
</head>
<body>
<h1>Площадь круга</h1>
<hr>
<form>
Радиус:
<input type=text value="1" id='t' name="t" size="5">
<input type=button value="Площадь" OnClick="sqr()">
<hr>
</form>
</body>
</html>
```

По нажатию кнопки "Площадь" начинает работать функция `sqr()`. Она обращается к элементу с именем `t` и присваивает значение свойства `value` этого элемента переменной `r`. Используя это значение в качестве радиуса, и обращаясь к свойству `PI` объекта `Math`, вычисляем площадь круга и выводим ее в окно с помощью метода `alert`.

Вариант 1

1. Создать HTML-страницу, которая при загрузке выводит на страницу текущий день недели, число, месяц и год. Для месяцев и дней недели организовать массивы.



2. Создать HTML-страницу, которая при загрузке запрашивает дату вашего рождения и выводит количество дней, которые Вы прожили в текстовое поле формы, например

Запрос пользователю

Запрос сценария:
введите год рождения

1988

OK Отмена

Запрос пользователю

Запрос сценария:
введите номер месяца рождения

4

OK Отмена

Запрос пользователю

Запрос сценария:
введите день рождения

28

OK Отмена

Я прожил всего 8380 дней

3. Ввести координаты точки в поля формы и определить расстояние от этой точки до начала координат. Расстояние должно вычисляться по нажатию кнопки.

Расстояние от точки до начала координат

Введите координаты точки:

X: 3

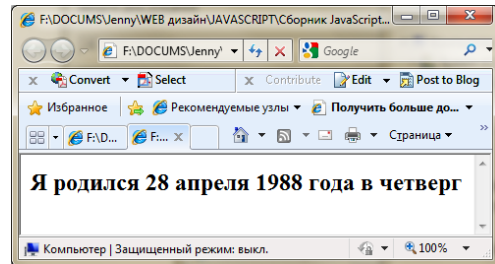
Y: 4

Расстояние

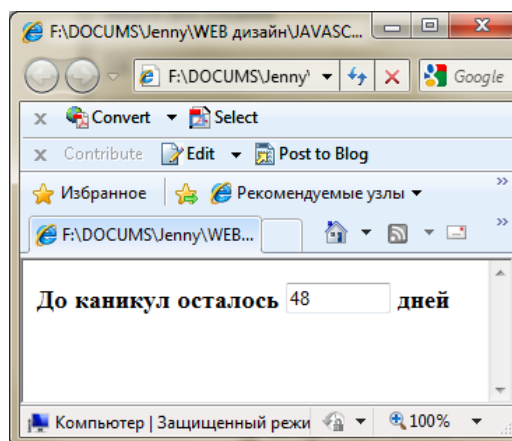


Вариант 2

1. Создать HTML-страницу, которая при загрузке запрашивает дату вашего рождения и выводит на страницу день недели, число, месяц и год этой даты. Для месяцев и дней недели организовать массивы.



2. Создать HTML-страницу, которая при загрузке выводит в текстовое поле формы, сколько дней осталось до каникул, например



3. Ввести длину и ширину прямоугольника в поля формы и определить площадь этого прямоугольника. Площадь должна вычисляться по нажатию кнопки.

