**Практическая работа 14**

**Предопределенные объекты**

**Цель:** Научить работать с предопределенными объектами: Date, Array, Math

**Задания:**

1. Ознакомиться с теоретическим материалом.
2. Создайте документ, печатающий текущую дату и время.
3. Создайте документ, создающий на экране изображение работающих часов.
4. Создайте документ, использующий массив.

**Необходимые приборы:** ПК, текстовый редактор Блокнот, браузер

**Методические рекомендации к выполнению лабораторной работы:**

**Методические рекомендации к выполнению задания 1**

В JavaScript разрешено пользоваться некоторыми заранее заданными объектами. Примерами таких объектов могут служить Date, Array или Math.   
Объект Date позволяет работать как со временем, так и с датой. Например, можно легко определить, сколько дней еще остается до следующего рождества. Или можете внести в HTML-документ запись текущего времени.

Рассмотрим пример, который высвечивает на экран текущее время. Сперва мы должны создать новый объект Date. Для этого используем оператором *new*:

today= new Date()

Здесь создается новый объект Date, с именем *today*. Если при создании этого нового объекта Date не указаны какие-либо определенные дата и время, то будут предоставлены текущие дата и время. То есть, после выполнения команды today= new Date() вновь созданный объект *today* будет указывать именно те дату и время, когда данная команда была выполнена.

Объект Date предоставляет нам кое-какие методы, которые теперь могут применяться к нашему объекту *today*. Например, этометоды - getHours(), setHours(), getMinutes(), setMinutes(), getMonth(), setMonth() итакдалее.   
Обратите пожалуйста внимание, что объект Date лишь содержит определенную запись о дате и времени. Он не уподобляется часам, автоматически отслеживающим время каждую секунду, либо миллисекунду.   
Чтобы зафиксировать какое-либо другие дату и время, мы можем воспользоваться видоизмененным конструктором (это будет метод Date(), который при создании нового объекта Date вызывается через оператор new):

today= new Date(1997, 0, 1, 17, 35, 23)

При этом будет создан объект Date, в котором будет зафиксировано первое января 1997 года 17:35 и 23 секунд. Таким образом, выбираются дата и время по следующему шаблону:

Date(year, month, day, hours, minutes, seconds)

Для обозначения января необходимо использовать число 0, а не 1. Число 1 будет обозначать февраль, ну и так далее.

Массивы играют в программировании очень важную роль. Массив может быть полезен там, где имеется много взаимосвязанных переменных. При этом к каждой из них можно получить доступ, воспользовавшись общим названием и неким номером. Допустим, есть массив в именем *names*. В этом случае мы можем получить доступ к первой переменной с именем name, написав *names[0]*. Вторая переменная носит *name[1]* и так далее. Начиная с версии 1.1 языка JavaScript (Netscape Навигатор 3.0), можно использовать объект Array. Можно создать новый массив, записав myArray= new Array(). После этого можно начать заносить в массив значения:

myArray[0]= 17;

myArray[1]= "Stefan";

myArray[2]= "Koch";

Массивы JavaScript обладают большой гибкостью. Например, нет нужды беспокоиться о размере массива - он устанавливается динамически. Если написать myArray[99]= "xyz", размер массива будет установлен 100 элементов.

Не имеет значения, заносятся ли в массив числа, строки, либо другие объекты. Если необходимо в скрипте выполнять математические расчёты, то некоторые полезные методы для этого можно найти в объекте Math. Например, имеется метод random(). Напишем функцию, позволяющую генерировать случайные числа. Теперь, чтобы работать на всех без исключения платформах, нам не нужно ничего, кроме метода random().

Если вызвать функцию Math.random(), то получите случайное число, лежащее в диапазоне между 0 и 1. Один из возможных результатов вызова document.write(Math.random()) (при каждой новое загрузке данной страницы здесь будет появляться другое число):

*0.9070647660301312*

**Методические рекомендации к выполнению задания 2**

1 Создайте документ, печатающий текущую дату и время.

<script language="JavaScript">

<!-- hide

now= new Date();

document.write("Time: "+ now.getHours()+":" + now.getMinutes() + "<br>");

document.write("Date: " + (now.getMonth() + 1) + "/" + now.getDate() + "/" +

(now.getYear()));

// -->

</script>

Результат будет выглядеть примерно следующим образом:

Time: 20:40

Date: 10/24/2009

Здесь используются такие методы, как getHours(), чтобы вывести на экран время и дату, указанные в объекте Date с именем *now*. Помните также, что нужно увеличивать на единицу значение, получаемое от метода getMonth().

В данном скрипте не выполняется проверки на тот случай, если количество минут окажется меньше, чем 10. Это значит, что выводится запись времени примерно в следующем виде: *14:3*, что на самом деле должно было бы означать *14:03*.

**Методические рекомендации к выполнению задания 3**

1 Создайте документ, создающий на экране изображение работающих часов:

Начало формы

<html>

<head>

<script Language="JavaScript">

<!-- hide

var timeStr, dateStr;

function clock() {

now= new Date();

// время

hours= now.getHours();

minutes= now.getMinutes();

seconds= now.getSeconds();

timeStr= "" + hours;

timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;

timeStr+= ((seconds < 10) ? ":0" : ":") + seconds;

document.clock.time.value = timeStr;

// дата

date= now.getDate();

month= now.getMonth()+1;

year= now.getYear();

dateStr= "" + month;

dateStr+= ((date < 10) ? "/0" : "/") + date;

dateStr+= "/" + year;

document.clock.date.value = dateStr;

Timer= setTimeout("clock()",1000);}

// -->

</script>

</head>

<body onLoad="clock()">

<form name="clock">

Время:

<input type="text" name="time" size="8" value=""><br>

Дата:

<input type="text" name="date" size="8" value="">

</form>

</body>

</html>

2 Просмотрите результат.

Примерный вид результата:

Время: 

Дата: 

Конец формы

Здесь для ежесекундной коррекции времени и даты используется метод setTimeout(). Фактически это сводится к кому, что каждую секунду создается новый объект Date, в который заносится текущее время.

Можно видеть, что функции clock() вызываются программой обработки события onLoad, помещенной в тэг <body>. В разделе body HTML-страницы имеется два элемента формы для ввода текста. Функция clock() записывает в оба эти элемента в корректном формате текущие время и дату. Для этой цели используются две строки *timeStr* и *dateStr*. Существует проблема с индикацией, когда количество минут меньше 10 - в данном скрипте эта проблема решается с помощью следующей строки: timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;

Как видим, количество минут заносится в строку *timeStr*. Если у нас менее 10 минут, то мы еще должны приписать спереди 0. Эта строка в скрипте может показаться немного странной, и ее можно было бы переписать в более знакомом Вам виде:

if (minutes < 10) timeStr+= ":0" + minutes

else timeStr+= ":" + minutes;

**Методические рекомендации к выполнению задания 4**

1. Создайте документ, использующий массив:

<scriptlanguage="JavaScript">

<!-- hide

myArray= new Array();

myArray[0]= "first element";

myArray[1]= "second element";

myArray[2]= "third element";

for (var i= 0; i< 3; i++) {

document.write(myArray[i] + "<br>");}

// -->

</script>

Данный скрипт печатает следующий текст:

first element

second element

third element

Первым делом создается здесь новый массив с именем myArray. Затем заносятся в него три различных значения. После этого мы запускаем цикл, который трижды выполняет команду document.write(myArray[i] + "<br>");. В переменной *i* ведется отсчет циклов от 0 до 2. Заметим, что в цикле мы пользуемся конструкцией *myArray[i]* . И поскольку *i* меняет значения от 0 до 2, то в итоге мы получаем три различных вызова document.write(). Иными словами, мы могли бы расписать этот цикл как:

document.write(myArray[0] + "<br>");

document.write(myArray[1] + "<br>");

document.write(myArray[2] + "<br>");

**Вопросы для самоконтроля:**

1. Примерами каких объектов служат объекты Date, Array, Math?
2. Для чего предназначен объект Date?
3. Что называют массивом?
4. Для чего предназначены массивы?
5. Каким образом осуществляется доступ к массиву?
6. Каким образом производится установка размера массива?
7. Для чего предназначен объект Math?
8. Перечислите методы объекта Date.
9. Каким образом производится создание объектов Date, Array, Math?
10. Какой объект позволяет работать со временем?