Практическая работа №6
Исследование свойств и методов объекта document

Цель работы: изучение основных методов и свойств объекта document и особенностей работы с ними

Задачи:
1. Изучить теоретический материал
2. Выполнить практические задания
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Теоретический материал

Рассмотрим простой пример создания и размещения JavaScript на веб-странице. Этот скрипт помещает текст на веб-страницу. Скрипт:

```
<SCRIPT LANGUAGE="javascript">
document.write("<FONT COL-OR='RED'>Это красный текст</FONT>")
</SCRIPT>
```

Разбор скрипта

Вторая строка скрипта выглядит следующим образом:

```
document.write("<font color='red'>Красный текст</font>")
```

То есть, целиком находится на одной линии и должна сохранять свою форму. Регистр для JavaScript важен.

<SCRIPT LANGUAGE="JavaScript"> Это код HTML, который дает браузеру понять, что с этого места начинается JavaScript. Часть LANGUAGE(язык)="JavaScript" указывает, какой именно язык используется.

Основная часть скрипта:

```
document.write("<FONT COLOR='RED'>Это красный текст</FONT>")
```

Указывается DOCUMENT (документ HTML) и те изменения, которые в нем произойдут — что-то будет написано (WRITE). То, что будет написано, находится в скобках. Термин DOCUMENT представляет собой object (объект). Слово WRITE (писать), отделенное точкой, называется method (методом объекта). Текст в скобках называется instance (примером метода), он передает то, что происходит, когда метод воздействует на объект. Текст внутри скобок находится в кавычках. Кавычки обязательны! Текст в кавычках представляет собой простой HTML. Дальше идут одинарные кавычки. Если поставить двойные, JavaScript решит, что это конец строки, и получится, что только часть текста будет применена к объекту, что вызовет ошибку. Внутри двойных кавычек ставятся одинарные.

Дата и время

Существует семь методов объекта: getDay(), getDate(), getMonth(), getYear(), getHour(), getMinute(), и getSecond() (получитьДень, Число, Месяц, Год, Час, Минуту, Секунду). Для воздействия им нужен объект.

Пример скрипта:

```
<SCRIPT LANGUAGE="JavaScript"> //Скрипт отмечает точную дату и время вашего прибытия на страницу
Now = new Date();
document.write("Сегодня " + Now.getDate()+ "-" + Now.getMonth() + "-" + Now.getFullYear() + " Вы зашли на мою страницу ровно в: " + Now.getHours() + ":" + Now.getMinutes() + " и " + Now.getSeconds() + " секунд.");
</SCRIPT>
```

Строка document.write не должна прерываться.

Разбор скрипта

Строка document.write уходит далеко за границы экрана. Эту форму необходимо сохранить, иначе браузер выдаст сообщение об ошибке.

// ? - Двойная дробь указывает на однострочный комментарий внутри скрипта.

Метод getDay(), который отвечает за день недели, выражается цифрой от единицы до семи. Начнем с месяца. Как уже говорилось выше, getMonth() — это метод, отвечающий за месяц. Метод (method) воздействует на объект (object). Метод документа — write.

getMonth() является методом объекта Date. Date занимает отдельное место в команде:

Now = new Date();

Устанавливаем объект, с которым будет работать метод getMonth().Нужно создать объект. Наш объект называется Now (сейчас). Его можно назвать любым именем. Это не имеет значения, если объект получил оригинальное имя, которое больше нигде в JavaScript не встречается. Точка с запятой в конце указывает на то, что строка JavaScript закончена. Необходимо вывести месяц на странице, значит, должна быть команда document.write(). Известно, что текст в скобках будет виден на странице, поэтому напишем все это, следуя логике:

• Сначала пишем <SCRIPT LANGUAGE="javascript">.

• Потом вставляем комментарий о том, для чего предназначен скрипт.

• Прежде чем приступить к getMonth(), необходимо создать объект.

• Теперь можно вставлять утверждение document.write.

• Текст в скобках после document.write оформляем по правилам.

o Текст, видимый на странице, должен быть окружен двойными кавычками.

o Сочетание текста и команд требует знака «плюс» + между элементами.

o Объект и метод разделены точкой, так что команда напечатать месяц выглядит так: Now.getMonth().

o Now.getMonth() — это не текст, который дол-жен быть виден на странице, а команда, которая указывает месяц. Поэтому не надо ставить ее в кавычки.

• Заканчиваем командой </SCRIPT>.

Результат:

```
<SCRIPT LANGUAGE="javascript">
//Скрипт напечатает на странице номер месяца
Now = new Date();
document.write("Сейчас месяц " + Now.getMonth());
</SCRIPT>
```

Элементы даты разделены дефисами. Дефис должен быть виден на странице, поэтому его следует ставить в кавычки. Все части связаны между собой плюсами.

Команда onMouseOver

События (event) и обработчики событий (event handler) относятся к JavaScript, но они скорее «встроены» в HTML-код, а не существуют самостоятельно. Они входят в структуру документа HTML, не требуя команд <SCRIPT> и </SCRIPT>. Сами они не скрипты, а скорее область взаимодействия между вашей страни-цей и читателем. События — это то, что происходит. Они доба-вят динамики сайту. Среди разнообразных обработчиков событий выберем один, самый популярный, — onMouseOver (навести мышь).

Скрипт

```
<A HREF="http://www.rumart.net" onMouse-Over="window.status='Бесплатныйхостинг'; return true">Ссылка</A>
```

Разбор скрипта

onMouseOver (обратите внимание на заглавные буквы) представляет собой обработчик событий (Event Handler) гипертекстовой ссылки. Он используется внутри гиперссылки. Формат ссылки остается без изменений. onMouseOver ставится сразу же после адреса URL. Событие (Event) приводится в действие, когда браузер распознает onMouseOver="". Объектом является window (окно); status (статус) представляет собой property (свойство) окна. Это небольшой участок окна, где должен разместиться следующий текст. Свойство выражается существительным и существует как небольшая часть элемента, стоящего перед точкой. Если у window есть изменяемое свойство status, значит, можно изменить и другие свойства окна. После window.status

следует знак равенства = и то, что должно произойти. В данном случае это текст в одинарных кавычках. Он появится в строке состояния, когда вы наведете курсор на ссылку. Обратите внимание на точку с запятой в конце строки. return true дают скрипту указание проверить, есть ли строка состояния. Если отчет (return) соответствует действительности (true), тогда происходит событие. Обратите внимание, что текст в строке состояния уже не изменяется и не изменится, сколько раз вы не наводили бы на нее курсор.

В HTML цветом фона страницы управляет команда BGCOLOR. То же самое и здесь, только обязательно соблюдайте регистр. В JavaScript он пишется bgColor. Создадим ссылку, которая меняет фон страницы с помощью обработчика onMouseOver.

1. Во-первых, раз это ссылка, то сохраним ту же схему, которой пользовались.
2. Заменим window на document.
3. Меняем фоновый цвет объекта, потому заменим status на bgColor.
4. Больше нам текст не нужен, поэтому заменим его цветом. Возьмем белый.
5. Нам надо, чтобы новый цвет оставался независимо от того, сколько раз мы будем наводить курсор на ссылку, потому вставляем return true после точки с запятой.

Скрипт:

Не щелкать

Команды и эффекты

1. Команда onClick (на щелчок)

onMouseOver запускает событие, если навести курсор на ссылку. Следовательно, щелкнув по ссылке, можно запустить событие onClick.

Чтобы продемонстрировать действие команды, воспользуемся методом alert:

alert('текст, который появится в окне')

Таким образом, получаем:

 Жмите сюда

2. Команда onFocus (на фокус)

Это команда, которая вызывает действие, когда пользователь «фокусируется» на элементе страницы. Это приемлемо для форм: флажков (checkbox) и текстовых полей (textbox).

Пример:

<form>

<input type="text" size="30" onFocus="window.status='Текст в строке состояния';">

</form>

Посмотрите результат в браузере (щелкните в поле ввода и по-смотрите на строку состояния):

3. Команда onBlur (на потерю фокуса)

onBlur позволяет сообщить пользователю о том, что он изменил свой ответ.

Пример:

<form>

<input type="text" size="45" value="Впишите свое имя и щелкните по другой строке" onBlur="alert('Вы изменили ответ — уверены, что он правильный?');">

</form>

4. Команда onChange (на изменение)

Ее главная задача — проверка. Этот обработчик события проверяет, сделал ли пользователь то, что вы от него просили.

Пример:

<form>

<input TYPE="text" size="45" value="Измените текст и щелкните по другой стро-ке"onChange="window.status='Текст был изменен';">

</form>

5. Команда onSelect (на выделение)

Эта команда работает так же, как и три предыдущие, отмечая, что в поле ввода произошли изменения, — в данном случае был выделен текст.

6. Команда onSubmit (на отправку)

Она позволяет вызвать какое-либо действие при нажатии кнопку Submit(отослать, отправить).

```
<form>
<input TYPE="submit"
onSubmit="parent.location='thanksalot.html'";>
</form>
```

*parent.location=" означает ссылку.

Пример

Создадим форму, которая будет взаимодействовать с пользователем. Форма должна иметь три элемента:

• поле ввода с просьбой ввести имя;

• два поля для флажков с вопросом о том, что больше нравится пользователю, мороженое или шоколад;

• кнопку отправки данных.

С каждым элементом должно произойти следующее:

• При вводе имени в строке состояния должны появиться слова: «Впишите сюда свое имя».

• Два поля с флажками должны отослать в строку состояния слова: «Вы выбрали...» и выбор пользователя.

• При нажатии на кнопку должно выскочить окно предупрежде-ния, благодарящее пользователя за участие в опросе.

Можно сделать это таким способом:

Можно сделать это таким способом:

Имя:

Что вам больше нравится: Шоколад Мороженое

Подача запроса

Скрипт:

```
<FORM> Name: <INPUT TYPE="text" SIZE="30" onFo-cus="window.status='Введите свое имя';">
Что вам больше нравится:
<INPUT TYPE="checkbox" onClick="window.status='Вы выбрали шоколад'";> Шоколад
<INPUT TYPE="checkbox" onClick="window.status='Вы выбрали мороженое'";> Мороженое
<INPUT TYPE="submit" onClick="alert('Спасибо за участие в опросе')";">
</FORM>
```

Запрос пользователю

Достаточно часто при работе скриптов возникает необходимость запроса данных у пользователя. Например, необходимо сделать запрос имени пользователя и записать его имя в переменную. Как только переменная будет присвоена, можно ввести ее в строку document.write, которая напечатает это имя на странице.

Скрипт

```
<SCRIPT LANGUAGE="javascript">
/* Скрипт предназначен для того, чтобы получить от пользователя информацию и поместить ее на страницу */
var user  name = prompt ("Напишите свое имя","Здесь");
document.write("Привет, " + user_name + "! Добро пожаловать!");
</SCRIPT>
```

Разбор скрипта

/* */ - комментарии внутри скрипта. Двойную дробь необходимо ставить в начале каждой новой строки. Эти же команды годятся для пространных комментариев.

Создание переменной

Переменные имеют первостепенное значение в JavaScript. Мы вводим переменную, которая должна представлять окончательный результат метода. Возьмем, например, переменную d. Тогда надо будет только один раз написать getDate() и назначить методу переменную d. И на протяжении всего оставшегося скрипта будем просто писать d там, где необходимо поставить дату.

Вот строка из скрипта, которая назначает переменную:

var user name = prompt ("Напишите свое имя","Здесь")

Переменная была создана по следующей схеме:

• var (от variable, переменная) объявляет, что следующим словом будет имя переменной.

• user name (имя пользователя) — имя переменной.

• Регистр имеет значение для JavaScript.

• Знак равенства = указывает на то, что переменная будет равна результату следующей команды.

• В нашем случае переменная будет представлять результат, полученный с помощью окна запроса.

Команда Prompt

Формат запроса:

var variable_name = prompt("Текст в окне","Текст в строке ввода")

Имя переменной включено в схему скрипта, иначе вы получили бы запрос, но полученные им данные никуда бы не пошли.

Правила:

• Чтобы строка ввода оставалась чистой, ничего не пишите между второй парой кавычек.

• Если вы не укажете в скобках второй пары кавычек, в строке появится слово undefined.

• Если вы написали что-либо в строке ввода и пользователь выберет ОК, ничего не меняя, на странице появится то, что вы написали.

• Если в строке ввода ничего нет и пользователь выберет ОК, ничего не вписав, на странице появится слово null.

Перейдем к основной части:

```
var user  name = prompt ("Напишите свое имя","Здесь");
document.write("Привет, " + user_name + "! Добро пожаловать!");
```

• Имя переменной user name присвоено результату запроса.

• prompt просит пользователя написать свое имя.

• В строке ввода читаем: "Здесь."

• Точка с запятой в конце строки.

• document.write вызывает текст "Привет, ".

• Знак плюс + отмечает, что все элементы идут друг за другом.

• user_name вводит результат запроса.

• Еще плюс.

• "! Добро пожаловать!" завершает текст.

• Точка с запятой.

Пример

```
<SCRIPT LANGUAGE="javascript">
var name = prompt("Напишите свое имя, пожалуйста","")
var d = new Date();
var y = d.getFullYear();
var da = d.getDate();
var m = d.getMonth() + 1;
```

```
var t = da + '/' + m + '/' + y;
document.write("<TITLE>")
document.write("Привет, "+name+ ". Сегодня " +t+ ". Спасибо, что зашли."); document.write("</TITLE>")
</SCRIPT>
```

Разбор скрипта

• Начинаем со строки <SCRIPT LANGUAGE="javascript">.

• Создаем переменную name для имени пользователя, которое он введет по запросу.

• Необходима дата, поэтому создаем переменную d.

• Известно, что год, месяц и число определяются с помощью d.getЧто-либо(), поэтому создаем переменные для каждого из этих элементов, y, m и da.

• «+ 1» после переменной m позволяет получить правильный номер месяца (они нумеруются с 0).

• Полная дата, состоящая из числа, месяца и года, также становится переменной и получает имя t.

• Заголовок создается с помощью команд <title>, написанных в трех строках document.write.

• Видимый текст помещен в двойные кавычки и отделен от команд, отвечающих за дату, знаками плюс.

• Третья строка document.write завершает команду <title>.

• Наконец </SCRIPT>.

Скрипт помещается перед командой <title>.

Имя было вставлено в текст с помощью команды:

<SCRIPT>document.write(name)</SCRIPT>

Концепция свойств

Рассмотрим множество скриптов, составленные по одной схеме: создается переменная для каждой команды объект.свойство, затем переменные помещаются в document.write(). Замечание: 1) заголовки жирным шрифтом не являются частью самих скриптов; 2) текст в скобках после document.write() должен располагаться на одной строке.

1. Свойства объекта document

```
<SCRIPT LANGUAGE="javascript">
var bgc = document.bgColor;
var fgc = document.fgColor;
var lc = document.linkColor;
var al = document.alinkColor;
var vlc = document.vlinkColor;
var url = document.location;
var ref = document.referrer;
var t = document.title;
var lm = document.lastModified;
document.write("Цвет фона этой страницы <B>" +bgc+ "</B>.")
document.write("<BR>Цвет текста этой страницы <B>" +fgc+ "</B>.")
document.write("<BR>Цвет ссылок этой страницы <B>" +lc+ "</B>.")
document.write("<BR>Цвет активной ссылки этой страницы <B>" +al+ "</B>.")
document.write("<BR>Цвет посещенной ссылки этой страницы <B>" +vlc+ "</B>.")
document.write("<BR>URL этой страницы <B>" +url+ "</B>.")
document.write("<BR>До этого вы были на странице <B>" + ref+ "</B>.")
document.write("<BR>Заголовок этой страницы <B>" +t+ "</B>.")
document.write("<BR>Последние изменения внесены: <B>" +lm+ "</B>.")
</SCRIPT>
```

2. Свойства объекта history

```
<SCRIPT LANGUAGE="javascript">
```

```
var h = history.length;
document.write("До этой страницы вы посетили" +h+ " страниц.")
</SCRIPT>
```

3. Два свойства объекта location (адрес)

```
<SCRIPT LANGUAGE="javascript">
var hst = location.host
document.write("Страница находится на <B>" + hst + "</B>." )
</SCRIPT>
<SCRIPT LANGUAGE="javascript">
var hstn = location.hostname
document.write("Страница находится на <B>" + hstn + "</B>." )
</SCRIPT>
```

4. Свойства объекта navigator

```
<SCRIPT LANGUAGE="javascript">
var an = navigator.appName;
var av = navigator.appVersion;
var acn = navigator.appCodeName;
var ua = navigator.userAgent;
document.write("Вы пользуетесь <B>" +an+ "</B>, версия " +av+ ".<BR>Кодовое название"
+acn+ ", заголовок " +ua+ "." );
</SCRIPT>
```

Объект navigator имеет четыре свойства.

• appName сообщает название браузера, например, Netscape или Explorer.

• appVersion сообщает версию браузера и платформу, на которой он работает.

• appCodeName сообщает кодовое имя, данное браузеру,

• userAgent сообщает заголовок протокола, используемого браузе-ром во время работы с серверами.

5. Свойства объекта document

```
<SCRIPT LANGUAGE="javascript">
var bgc = document.bgColor;
var fgc = document.fgColor;
var lc = document.linkColor;
var al = document.alinkColor;
var vlc = document.vlinkColor;
var url = document.location;
var ref = document.referrer;
var t = document.title;
var lm = document.lastModified;
document.write("Цвет фона этой страницы <B>" +bgc+ "</B>.")
document.write("<BR>Цвет текста этой страницы <B>" +fgc+ "</B>.")
document.write("<BR>Цвет ссылок этой страницы <B>" +lc+ "</B>.")
document.write("<BR>Цвет активной ссылки этой страницы <B>" +al+ "</B>.")
document.write("<BR>Цвет посещенной ссылки этой страницы<B>" +vlc+ "</B>.")
document.write("<BR>URL этой страницы <B>" +url+ "</B>.")
document.write("<BR>До этого вы были на странице <B>" + ref+ "</B>.")
document.write("<BR>Заголовок этой страницы <B>" +t+ "</B>.")
document.write("<BR>Последние изменения внесены: <B>" +lm+"</B>.")
</SCRIPT>
```

Свойства документа очень популярны в JavaScript.:

• bgColor сообщает цвет фона в шестизначном коде.

• fgColor сообщает цвет текста в шестизначном коде.

• linkColor сообщает цвет ссылки.
• alinkColor сообщает цвет активной ссылки.
• vlinkColor сообщает цвет посещенной ссылки.
• location сообщает URL страницы.
• referrer сообщает, с какой страницы пришел пользователь.
• title сообщает заголовок документа между командами TITLE.
• lastModified сообщает дату, когда были внесены последние из-менения в страницу

Воспользовавшись командой If, можно сказать: "Если время больше 6 вечера, пусть текст будет белый, а фон черный. Если еще нет 6 вечера, то пусть фон будет голубой, а текст зеленый".

Для организации динамического изменения цветов фона и текста необходимо скрипт писать до тега <Body>, причем в скрипте необходимые изменения нужно оформлять в виде функции. Для вызова функции используется команда onload тега <body>. Для обновления страницы необходимо задать период обновления посредством функции window.setTimeout("название функции",период); внутри скрипта.

Например, ниже представлен скрипт обратного отсчета для оставшихся дней до нового года.

```
<script language="JavaScript">
timeend= new Date();
// IE и FF по разному отрабатывают getYear()
timeend= new Date(timeend.getYear()>1900?(timeend.getYear()+1):(timeend.getYear()+1901),0,1);
// для задания обратного отсчета до определенной даты укажите дату в формате:
// timeend= new Date(ГОД, МЕСЯЦ-1, ДЕНЬ);
// Для задания даты с точностью до времени укажите дату в формате:
// timeend= new Date(ГОД, МЕСЯЦ-1, ДЕНЬ, ЧАСЫ-1, МИНУТЫ);
function time() {
        today = new Date();
        today = Math.floor((timeend-today)/1000);
        tsec=today%60; today=Math.floor(today/60); if(tsec<10)tsec='0'+tsec;
        tmin=today%60; today=Math.floor(today/60); if(tmin<10)tmin='0'+tmin;
        thour=today%24; today=Math.floor(today/24);
        timestr=today +" дней "+ thour+" часов "+tmin+" минут "+tsec+" секунд";
        document.getElementById('t').innerHTML=timestr;
        window.setTimeout("time()",1000);
}
</script>
<body onload="time()">

<h1>Обратный отсчет времени</h1>
<p>До нового года осталось <span id="t" style="font-size:20px"></span></p>
<p>Этот скрипт позволяет показать у себя на сайте время до какого-нибудь события,
до нового года, до экзаменов, до приказа и т.д.</p>
```

Практические задания

Для всех страниц предусмотреть запрос имени пользователя и вывод его имени и времени входа на страницу.

1. Разработать Web-страницу «Калькулятор». Предусмотреть операции сложения, вычитания, деления, умножения. Для каждой кнопки при наведении мыши в строке состояния окна должно отображаться ее значение. Для знаков операций значения отображать словами («+» - сложение и т.п.).
2. Разработать Web-страницу с небольшим фрагментом текста, которая позволяет пользователю задавать цвет фона и цвет текста самостоятельно. Ввод цвета фона организовать с клавиатуры посредством textbox, ввод цвета текста – посредством radioButton. Каждый объект должен быть подписан в строке состояния при наведении мыши.

3. Разработать Web-страницу, отображающую текст на странице в зависимости от положения переключателя. Предусмотреть три фрагмента текста и три переключателя. Текст каждого фрагмента должен быть задан своим цветом.
4. Разработать web-страницу, отображающую текущее время в режиме постоянного обновления. Также динамически, посекундно, должен меняться цвет фона и цвет текста.

Контрольные вопросы

1. Что представляет собой document в JavaScript?
2. С помощью какого метода осуществляется вывод текста? Как задать форматирование текста из скрипта?
3. Какие методы существуют для работы с датой и временем? Как осуществляется доступ к этим методам?
4. Каким образом можно изменить свойство объекта при наведении мыши?
5. С помощью какого метода можно вывести сообщение на экран?
6. Как осуществить запрос данных у пользователя?
7. Каковы свойства объекта navigator?
8. Какие свойства позволяют получить цвет ссылок документа?
9. Как обратится к цвету фона и текста?
10. Для чего предназначен объект history?
11. Как организовать динамическое изменение содержимого страницы?

Содержание отчета

1. Титульный лист
2. Цели, задачи работы
3. Текст задания
4. Листинг
5. Результаты работы
6. Ответы на контрольные вопросы