

13. Назначение и применение JavaScript, общие сведения

Цель: познакомиться с понятием объектной модели применительно к JavaScript, размещением кода на HTML странице, иерархией классов.

13.1 Теоретические сведения

JavaScript — это язык управления сценариями просмотра гипертекстовых страниц Web на стороне клиента. Если быть более точным, то JavaScript — это не только язык программирования на стороне клиента. Liveware, прародитель JavaScript, является средством подстановок на стороне сервера Netscape. Однако наибольшую популярность JavaScript обеспечило программирование на стороне клиента.

Основная идея JavaScript состоит в возможности изменения значений атрибутов HTML-контейнеров и свойств среды отображения в процессе просмотра HTML-страницы пользователем. При этом перезагрузки страницы не происходит.

13.1.1 Понятие объектной модели применительно к JavaScript

Для создания механизма управления страницами на клиентской стороне было предложено использовать объектную модель документа. Суть модели в том, что каждый HTML-контейнер — это объект, который характеризуется тройкой:

- свойства
- методы
- события

Объектную модель можно представить, как способ связи между страницами и браузером. Объектная модель - это представление объектов, методов, свойств и событий, которые присутствуют и происходят в программном обеспечении браузера, в виде, удобном для работы с ними кода HTML и исходного текста сценария на странице.

Объекты с одинаковым набором свойств, методов и событий объединяются в классы однотипных объектов. Классы - это описания возможных объектов. Сами объекты появляются только после загрузки документа браузером или как результат работы программы. Об этом нужно всегда помнить, чтобы не обратиться к объекту, которого нет.

13.1.1.1 Свойства

Многие HTML-контейнеры имеют атрибуты. Например, контейнер якоря <A...>... имеет атрибут HREF, который превращает его в гипертекстовую ссылку.

Не у всех атрибутов можно изменять значения. Например, высота и ширина графической картинке определяются по первой загруженной в момент отображения страницы картинке. Все последующие картинки, которые заменяют начальную, масштабируются до нее. Справедливости ради следует заметить, что в Microsoft Internet Explorer размер картинки может меняться.

Для общности картины свойствами в JavaScript наделены объекты, которые не имеют аналогов в HTML-разметке. Например, среда исполнения, называемая объектом Navigator, или окно браузера, которое является вообще самым старшим объектом JavaScript.

13.1.1.2 Методы

В терминологии JavaScript методы объекта определяют функции изменения его свойств. Например, с объектом «документ» связаны методы open(), write(), close(). Эти методы позволяют сгенерировать или изменить содержание документа. Приведем простой пример:

```
function hello()
{
  id=window.open("", "example", "width=400, height=150");
  id.focus(); id.document.open();
  id.document.write("<H1>Привет! </H1>");
  id.document.write("<HR><FORM>");
  id.document.write("<INPUT TYPE=button VALUE='Закрыть окно' ");
  id.document.write("<onClick='window.opener.focus(); window.close(); '>");
  id.document.close();
}

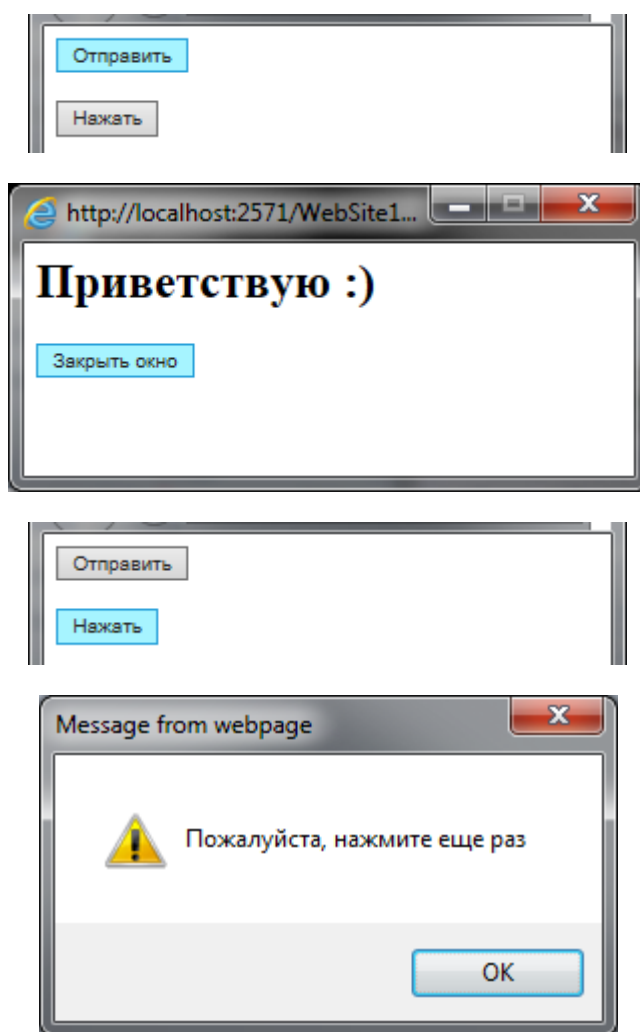
<html>
<title></title>
<head>
<script language="JavaScript" type="text/JavaScript">
function hello() {
  var id;
  id=window.open("", "example", "width=400, height=150");
  id.focus();
  id.document.open();
```

```

id.document.write("<h1><form color='olive' face='Lucida Console' size='5'>Приветствую
:)</form></h1>");
id.document.write("<input type='button' value='Закрыть окно' onclick='window.close();'> </form>");
id.document.close();
}
</script>
</head>
<body>
<form name="f2">
<input type="button" onclick="hello();" value="Отправить"/><br /><br />
<input type="button" value="Нажать"
onclick="window.alert('Пожалуйста, нажмите еще раз'); "><br /><br />
</form>
</body>
</html>

```

В этом примере метод open() открывает поток записи в документ, метод write() осуществляет эту запись, метод close() закрывает поток записи в документ. Все происходит так же, как и при записи в обычный файл. Если у окна есть поле статуса (обычно в нем отображается уровень загрузки документа), то при незакрытом потоке записи в документ в нем будет «метаться» прямоугольник продолжения записи, как это происходит при загрузке документа.



13.1.1.3 События

Кроме методов и свойств объекты характеризуются событиями. Собственно, суть программирования на JavaScript заключается в написании обработчиков этих событий.

Обработчики событий указываются в тех контейнерах, с которыми эти события связаны. Например, контейнер BODY определяет свойства всего документа, поэтому обработчик события завершения загрузки всего документа указывается в этом контейнере как значение атрибута onLoad.

13.1.2 Размещение кода на HTML-странице

Главный вопрос любого начинающего программиста: «Как оформить программу и выполнить ее?». Попробуем на него ответить, как можно проще, но при этом не забывая обо всех способах применения JavaScript-кода.

Во-первых, исполняет JavaScript-код, браузер. В него встроен интерпретатор JavaScript. Следовательно, выполнение программы зависит от того, когда и как этот интерпретатор получает управление. Это, в свою очередь, зависит от функционального применения кода. В общем случае можно выделить четыре способа функционального применения JavaScript:

- гипертекстовая ссылка (схема URL);
- обработчик события (handler);
- подстановка (entity) (в Microsoft Internet Explorer реализована в версиях от 5.X и выше);
- вставка (контейнер SCRIPT).

13.1.2.1 URL-схема JavaScript

Схема URL (Uniform Resource Locator) — это один из основных элементов Web-технологии. Каждый информационный ресурс в Web имеет свой уникальный URL. URL указывают в атрибуте HREF контейнера A, в атрибуте SRC контейнера IMG, в атрибуте ACTION контейнера FORM и т.п. Все URL подразделяются на схемы доступа, которые зависят от протокола доступа к ресурсу, например, для доступа к FTP-архиву применяется схема ftp, для доступа к Gopher-архиву — схема gopher, для отправки электронной почты — схема smtp. Тип схемы определяется по первому компоненту URL: <http://intuit.ru/directory/page.html>

13.1.2.2 Обработчики событий

Такие программы, как обработчики событий (handler), указываются в атрибутах контейнеров, с которыми эти события связаны.

13.1.2.3 Подстановки

Подстановка (entity) встречается на Web-страницах довольно редко. Тем не менее это достаточно мощный инструмент генерации HTML-страницы на стороне браузера. Подстановки используются в качестве значений атрибутов HTML-контейнеров.

13.1.2.4 Вставка (контейнер SCRIPT - принудительный вызов интерпретатора)

Контейнер SCRIPT — это развитие подстановок до возможности генерации текста документа JavaScript-кодом.

Контейнер SCRIPT выполняет две основные функции:

- размещение кода внутри HTML-документа;
- условная генерация HTML-разметки на стороне браузера.

Размещение кода внутри HTML-документа

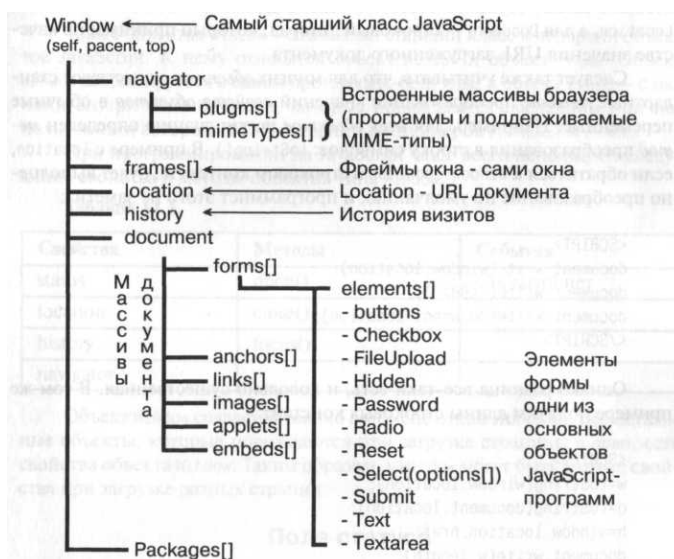
Код можно разместить либо в заголовке документа, внутри контейнера HEAD, либо внутри BODY.

Условная генерация HTML-разметки на стороне браузера

Для генерации HTML-разметки контейнер SCRIPT размещают в теле документа.

13.1.3 Иерархия классов

Объектно-ориентированный язык программирования предполагает наличие иерархии классов объектов. В JavaScript такая иерархия начинается с класса объектов Window, т.е. каждый объект приписан к тому или иному окну. Для обращения к любому объекту или его свойству указывают полное или частичное имя этого объекта или свойства объекта, начиная с имени объекта старшего в иерархии, в который входит данный объект:



13.2 Контрольные вопросы

1. Что такое JavaScript?
2. Что такое объектная модель?
3. Какие объекты в JavaScript наделены свойствами?
4. Назовите основные способы функционального применения JavaScript.
5. Что такое классы объектов?
6. Какие функции выполняют методы open(), write(), close()?
7. Что такое схема URL?
8. Какие методы позволяют установить текущее время?
9. Что такое контейнер SCRIPT?
10. Какие две основные функции выполняет контейнер SCRIPT?

13.3 Задания

1. Напишите пример сценария с использованием методов open(), write(), close() (R1301)
2. Напишите пример сценария с использованием события onClick (R1302)
3. Напишите код, в котором, при нажатии на гипертекстовую ссылку, появляется окно предупреждения. (R1303)
4. Напишите код, в котором, при нажатии кнопки submit, в форме можно заполнить текстовое поле этой же формы. (R1304)
5. Напишите код, в котором, при нажатии на кнопку, происходило бы событие click. (R1305)
6. Напишите код с использованием подстановки. (R1306)
7. Напишите код, который отображает часы в строке статуса, при этом обновление времени происходило бы через каждые 5 секунд. (R1307)
8. Напишите скрипт для осуществления конкатенации трех строк при помощи метода concat(). (R1308)