JavaScript. DOM

Объектная модель документа

Объектная модель документа

- Объектная модель документа - это совокупность объектов, обеспечивающих доступ к содержимому Web-страницы и ряду функций Web-браузера. Доступ к объектам позволяет управлять содержимым Web-страницы уже после ее загрузки.
- Объектная модель представлена в виде иерархии объектов: имеется объект верхнего уровня и подчиненные ему объекты.

Объектная модель документа

- Подчиненные объекты имеют свои подчиненные объекты. Кроме того, все объекты имеют свойства, а некоторые еще и методы.
- Доступ к подчиненным объектам - путем указания пути от верхнего объекта к подчиненному через точку:
- **<Объект верхнего уровня>. <Подчиненный объект>.{Свойство или метод}**
- Часто объект верхнего уровня (и даже подчиненный объект) можно не указывать.

Объектная модель документа

- **window** — это объект самого верхнего уровня, представляющий сам Web-браузер.
- Помимо уже упомянутого объекта самого высокого уровня — **window** в объектной модели имеются следующие основные:
- **event** предоставляет информацию, связанную с событиями.
- **frame** служит для работы с фреймами (коллекция **frames**);

Объектная модель документа

- **history** предоставляет доступ к списку истории Web-браузера;
- **navigator** содержит информацию о Web-браузере;
- **location** содержит URL-адрес текущей Web-страницы;
- **screen** служит для доступа к характеристикам экрана компьютера пользователя;

Объектная модель документа

- **document** служит для доступа к структуре, содержанию и стилю документа:
  - **all** — коллекция всех элементов;
  - **anchors** — коллекция "якорей", заданных тегом **<a>**;
  - **forms— коллекция всех форм;**
  - **elements — коллекция элементов формы;**
  - **frames — все фреймы;**
  - **images — коллекция всех изображений;**

Объектная модель документа

- **links — коллекция ссылок;**
- **scripts — коллекция скриптов;**
- **styleSheets — коллекция стилей.**

- Далее рассмотрим основные свойства и методы корневого объекта **window**, браузера, документа и формы. Примеры позволят освоить управление окнами, изменение содержимого тегов документа и проверку заполнения полей форм.

Window

- Работая в сценарии с объектом Window, вы напрямую манипулируете всем браузером.
- Объект Window предоставляет доступ извне к некоторым свойствам программы просмотра:
- `<SCRIPT LANGUAGE="JavaScript">`
- `for (props in window)`
- `sProps += props+": "+window[props]+ "\n";`
- `alert(sProps);`
- `</SCRIPT>`

Window

- Некоторые свойства возвращают ссылку на другие объекты:
- **parent** — ссылка на родительское окно;
- **self** — ссылка на текущее окно;
- **top** — ссылка на самое верхнее родительское окно;
- **window** — то же, что **self**, ссылка на себя;
- **opener** — ссылка на окно, открывшее данное;
- **name** — имя окна или фрейма.

Window

- Метод **open()** объекта **window** позволяет открыть дополнительное окно и поместить в него Web-страницу:
- **[<Переменная> =]window.open(<URL>, [<Имя окна>], [<Свойства окна>]);**
- Здесь используются следующие компоненты:
- **<URL>** — URL-адрес страницы, которая будет загружена в новое окно;
- **<Имя окна>** — имя нового окна;

Window

- **<Свойства окна>** — определяет отображаемые элементы в новом окне;
- **<Переменная>** — ссылка на объект вновь созданного окна, которую можно использовать для работы с ним.
- Свойства окна, передаваемые методу **open():**
- **width** и **height** задают ширину и высоту окна в пикселах (минимум 100);

Window

- **left** и **top** указывают горизонтальную и вертикальную координаты левого верхнего угла создаваемого окна;
- **fullscreen — {yes | no | 1 | 0}** — включает (**yes** или **1**) или отключает (**no** или **0**) полноэкранный режим для нового окна;
- **resizable — {yes | no | 1 | 0}** — включает или отключает возможность изменения размера создаваемого окна;

Window

- **location — {yes | no | 1 | 0}** — указывает наличие или отсутствие адресной строки;
- **menubar — {yes | no | 1 | 0}** — включает или отключает отображение строки меню;
- **scrollbars — {yes | no | 1 | 0}** — задает, отображать или нет полосы прокрутки;
- **status — {yes | no | 1 | 0}** — указывает на наличие или отсутствие строки состояния;

Window

- **titlebar — {yes | no | 1 | 0}** — включает или отключает отображение заголовка у создаваемого окна;
- **toolbar — {yes | no | 1 | 0}** — включает или отключает отображение панели инструментов.
- Рассмотрим пример создания нового окна.

Navigator

- Объект **navigator** предоставляет информацию о самом Web-браузере.
- Свойства объекта **navigator**:
- **appName** — имя Web-браузера;
- **appCodeName** — кодовое имя версии Web-браузера;
- **appVersion** — версия Web-браузера;
- **appMinorVersion** — вторая цифра в номере версии Web-браузера;

Navigator

- **userAgent** — комбинация свойств **appCodeName** и **appVersion**;
- **cpuClass** — тип процессора клиентского компьютера;
- **platform** — название клиентской платформы;
- **systemLanguage** — код языка операционной системы клиента;
- **browserLanguage** — код языка Web-браузера;

Navigator

- **userLanguage** — код языка Web-браузера;
- **onLine** — режим подключения: **true**, если клиент в настоящее время подключен к Интернету, и **false**, если отключен;
- **cookieEnabled** — режим работы **cookie**: возвращает **true**, если прием **cookie** разрешен.

Navigator

- Вывести все поддерживаемые свойства и методы объекта **navigator** позволяет следующий код:
- **for (var p in navigator) {**
- **document.write(p + " ==> " + navigator[p] + "
");**
- **}**

Document

- Объект document предоставляет доступ ко всем элементам Web-страницы.
- Свойства объекта document:
- activeElement — ссылка на активный элемент документа;
- documentElement — ссылка на корневой элемент (тег <html>);
- body — ссылка на все содержимое тега <body>;

Document

- title — название документа, указанное в теге <title>;
- URL — URL-адрес документа;
- referrer — URL-адрес, с которого перешел посетитель по ссылке;
- parentWindow — окно, которому принадлежит документ;
- cookie — объект cookie, который позволяет сохранять данные на компьютере клиента;

Document

- readyState — статус документа. Возвращает следующие строковые значения:
- •complete — документ полностью загружен;
- •interactive — документ загружен не полностью, но доступен для просмотра и управления;
- •loading — документ загружается;
- •uninitialized — документ не доступен;

Document

- location — объект location;
- selection — объект selection;
- fileCreatedDate — дата создания файла документа в виде строки;
- fileModifiedDate — дата последнего изменения файла документа в виде строки;
- fileUpdatedDate — дата последнего изменения файла сервером в кэше компьютера пользователя;

Document

- lastModified — дата и время последнего изменения файла документа в виде строки;
- fileSize — размер файла в виде строки;
- bgColor — цвет фона страницы;
- fgColor — цвет текста страницы;
- linkColor — цвет гиперссылок документа;
- alinkColor — цвет активных гиперссылок;
- vlinkColor — цвет посещенных гиперссылок.

Document

- Методы объекта document:
- getElementById(<Идентификатор>) возвращает ссылку на элемент с указанным идентификатором;
- getElementsByName(<Название элемента>) возвращает ссылку на коллекцию элементов, у которых параметр name равен значению аргумента;

Document

- getElementsByTagName(<Тег>) возвращает ссылку на коллекцию дочерних элементов, созданных с использованием тега, переданного в качестве параметра;
- elementFromPoint(<x>, <y>) возвращает ссылку на элемент, находящийся по координатам <x> и <y>;
- write(<Текст>) записывает текст, переданный как параметр, в текущее место документа;

Document

- Коллекции объекта document:
- all — коллекция всех элементов;
- anchors — коллекция "якорей", заданных тегом <a>;
- forms — коллекция всех форм;
- frames — все фреймы;
- images — коллекция всех изображений;
- links — коллекция ссылок;

Document

- scripts — коллекция скриптов;
- styleSheets — коллекция стилей.
- **Общие свойства и методы элементов Web-страницы**
- Все элементы Web-страницы также имеют свойства и методы. Помимо свойств, специфических для конкретных элементов, все они имеют следующие общие свойства:

Document

- **all** — ссылка на коллекцию дочерних элементов;
- **id** — имя элемента, заданное параметром **id**;
- **className** — имя класса, заданное параметром **class** элемента Web-страницы;
- **sourceIndex** — порядковый номер элемента, который можно использовать для ссылки на элемент из коллекции **all**;

Document

- **tagName** — имя тега элемента;
- **parentElement** — ссылка на элемент-родитель;
- **length** — число элементов в коллекции;
- **height** и **width** — высота и ширина элемента;
- **innerText** — содержимое элемента, исключая теги HTML. Если присвоить свойству новое значение, то содержимое элемента изменится;

Document

- **outerText** — содержимое элемента, исключая теги HTML. Если присвоить свойству новое значение, то содержимое элемента заменится новым, и сам элемент будет заменен;
- **innerHTML** — содержимое элемента, включая внутренние теги HTML. Если присвоить свойству новое значение, то содержимое элемента также изменится;

Document

- **outerHTML** — содержимое элемента, включая теги HTML. Если присвоить свойству новое значение, то содержимое элемента заменится новым, а сам элемент будет заменен;
- а также ряд других, задающих размер и положение.

Document

- Общие методы элементов Web-страницы:
- **getAttribute(<Имя параметра>, true | false)** возвращает значение параметра с именем <Имя параметра> тега текущего элемента. Если второй параметр равен **false**, то поиск параметра тега производится без учета регистра символов;

Document

- **setAttribute(<Имя параметра>, <Значение>, true | false)** присваивает <Значение> параметру с именем <Имя параметра> тега текущего элемента. Если третий параметр равен **false**, то поиск параметра тега производится без учета регистра символов;
- **removeAttribute(<Имя параметра>, true | false)** удаляет параметр тега текущего элемента.

Document

- Если второй параметр равен **false**, то поиск параметра тега производится без учета регистра символов. Возвращает значение **true**, если удаление было выполнено успешно;
- **clearAttributes()** удаляет все параметры тега элемента кроме параметров **id** и **name**;
- **contains(<Имя элемента>)** возвращает **true**, если элемент с этим именем содержится внутри текущего элемента;

Document

- **getElementsByTagName(<Тег>)** возвращает ссылку на коллекцию дочерних элементов, созданных с использованием тега, переданного в качестве параметра.
- Обратиться к элементу документа можно следующими способами:
- по номеру элемента в коллекции (в порядке появления в документе, начиная с нуля).

Document

- Номер элемента может быть указан как в круглых, так и в квадратных скобках:
- **Str = document.all(1).tagName;**
- **Str = document.all[1].tagName;**
- по имени или идентификатору элемента:
- **Str = document.all["<name или id>"].tagName;**
- **Str = document.all("<name или id>").tagName;**

Document

- **Str = document.all.<name или id>.tagName;**
- с помощью метода **item().** Обратите внимание, можно использовать только круглые скобки:
- **Str = document.all.item(1).tagName;**
- **Str = document.all.item("<name или id>").tagName;**
- Обратите внимание: в Internet Explorer версии 8.0 второй способ не работает;

Document

- если идентификатор элемента является уникальным, то к элементу можно обратиться как к отдельному объекту:
- **Str = <name или id>.tagName;**
- если идентификатор элемента не является уникальным, то нужно будет указать второй индекс:
- **Str = document.all("<name или id>", 2).tagName;**

Document

- Обратите внимание: в Internet Explorer версии 8.0 способ не работает.
- Вместо коллекции **all** может быть указана другая коллекция. Например, к изображению можно обратиться следующими способами:
- **Str = document.images.<name или id>.src;**
- **Str = document.images[<Индекс>].src;**
- **Str = document.images.item(<Индекс>).src;**

Document

- Количество элементов в коллекции можно узнать с помощью свойства **length**:
- **document.all.tags("H1").length**
- В некоторых Web-браузерах нет коллекции **all**. Вместо нее для поиска элемента по идентификатору следует пользоваться методом **getElementById (<Идентификатор>)**:

Document

- **Str = document.getElementById ("check1"). value;**
- Чтобы получить элементы по названию тега следует воспользоваться методом **getElementsByTagName(<Тег>)**:
- **len = document. getElementsByTagName ("h1").length;**

Document

- Количество тегов **<input>** внутри формы с идентификатором **frm**:
- **var frms = document.getElementById("frm");**
- **var len = frms.getElementsByTagName ("input").length;**

Forms

- С помощью JavaScript можно обрабатывать данные, введенные пользователем в элементы формы. Обработка на стороне клиента позволит снизить нагрузку на Web-сервер за счет отмены отправки данных формы при неправильно введенных значениях.

Forms

- **Коллекция *Forms*.**
- Все формы документа доступны через коллекцию **forms**. Например, чтобы получить значение текстового поля с именем **text1** (входящего в состав формы **form1**), можно воспользоваться следующей строкой кода:
- **document.forms["form1"].text1.value**
- Обратиться к форме можно и как к любому элементу документа:

Forms

- **document.form1.text1.value**
- К отдельной форме можно также обратиться по индексу:
- **document.forms[0].text1.value**
- Если элемент управления находится внутри тега **<form>**, то ссылку на саму форму нужно обязательно указывать, иначе Web-браузер будет искать элемент в теле документа, и в итоге вернет значение **null**.

Forms

- Получить доступ к элементу, вне зависимости от того находится он внутри формы или нет, позволяет метод **getElementById()** объекта **document**:
- **document.getElementById("text1").value**
- Все элементы формы доступны через коллекцию **elements**:
- **document.forms["form1"].elements["text1"].value**

Forms

- **document.forms["form1"].elements[0].value**
- **document.forms[0].elements[0].value**
- **document.form1.elements[0].value**
- Объект формы поддерживает следующие свойства:
- **length** — количество элементов формы;
- **action** — URL-адрес программы обработки формы;

Forms

- **elements** — ссылка на коллекцию **elements**;
- **encoding** — MIME-тип передаваемых данных;
- **method** — режим пересылки данных формы на Web-сервер;
- **enctype** — метод кодирования данных формы;
- **name** — имя формы;

Forms

- **target** — имя фрейма, в который будет загружен документ, являющийся результатом обработки данных формы Web-сервером.
- Объект формы поддерживает следующие методы:
- **submit()** выполняет отправку данных формы серверной программе. Аналогично нажатию кнопки **Submit**;

Forms

- **reset()** очищает форму, то есть все элементы формы получают значения по умолчанию. Аналогично нажатию кнопки **Reset**.
- Объект формы поддерживает следующие события:
- **onsubmit** наступает при отправке данных формы;
- **onreset** возникает при очистке формы.

Поля форм

- Элементы управления имеют свои свойства, методы и события. Рассмотрим основные типы элементов формы по отдельности.
- **Текстовое поле и поле ввода пароля.**
- Текстовое поле и поле для ввода пароля имеют одинаковые свойства:
- **value** — значение элемента формы;
- **defaultValue** — начальное значение, заданное параметром **value**;

Поля форм

- **disabled** — запрет элемента формы: если задано значение **true**, то поле является неактивным (отображается серым цветом);
- **form** — ссылка на форму, в которой находится элемент;
- **maxLength** — максимальное количество символов, которое может быть введено в поле;
- **name** — имя элемента;

Поля форм

- **type** — тип элемента формы;
- **readOnly** — запрет редактирования: если задано значение **true**, текст в поле **нельзя** редактировать, если **false** — можно.
- Методы тоже одинаковы:
- **blur()** убирает фокус ввода с текущего элемента формы;
- **focus()** помещает фокус на текущий элемент формы;

Поля форм

- **select()** выделяет текст в поле.
- Обоими элементами поддерживаются следующие события:
- **onblur** происходит при потере фокуса элементом формы;
- **onchange** наступает после изменения данных в поле, при переводе фокуса ввода на другой элемент либо при отправке данных формы. Наступает перед событием **onblur**;

Поля форм

- **onfocus** возникает при получении фокуса ввода элементом формы.
- Можно использовать стандартные события мыши и клавиатуры
- **Поле для ввода многострочного текста.**
- **<textarea>** поддерживает те же свойства, методы и события, что и простое поле ввода, за исключением свойства **maxLength**. Кроме того, поддерживается еще одно свойство:

Поля форм

- **wrap** — режим переноса слов. Может принимать следующие значения:
- **off** — не переносить слова;
- **physical** — слова переносятся как на экране, так и при передаче данных серверу;
- **virtual** — слова переносятся только на экране, но не при передаче данных серверу.

Поля форм

- **Список с возможными значениями.**
- Свойства объекта списка:
- **disabled** — запрет доступа: если задано значение **true**, то список является неактивным (отображается серым цветом);
- **form** — ссылка на форму, в которой находится элемент;
- **length** — количество пунктов в списке (доступно и для записи);

Поля форм

- **multiple** — разрешение множественного выделения: **true**, если из списка можно выбрать сразу несколько элементов одновременно;
- **name** — имя элемента;
- **options** — ссылка на коллекцию пунктов в списке;
- **selectedIndex** — номер выбранного пункта (нумерация начинается с нуля);

Поля форм

- **size** — число одновременно видимых элементов списка;
- **type** — тип элемента формы (**select-multiple** или **select-one**);
- **value** — значение пункта, выбранного в списке.
- Свойства пункта списка:
- **defaultSelected** — пункт списка, выбранный изначально;

Поля форм

- **index** — номер пункта в списке;
- **selected** — признак выделения: **true**, если пункт выбран в списке;
- **disabled** — если задано значение **true**, то пункт списка является неактивным (отображается серым цветом). Свойство поддерживается Web-браузером Internet Explorer, начиная с версии 8.0;
- **text** — текст пункта списка;

Поля форм

- **value** — значение выбранного пункта.
- Методы:
- **blur()** убирает фокус ввода с текущего элемента формы;
- **focus()** помещает фокус на текущий элемент формы.
- События:
- **onblur** наступает при потере фокуса элементом формы;

Поля форм

- **onchange** происходит после выбора нового пункта списка;
- **onfocus** наступает при получении фокуса ввода элементом формы.
- Кроме перечисленных событий можно использовать стандартные события мыши и клавиатуры.

Поля форм

- **Флажки и переключатели.**
- Свойства:
- **value** — значение элемента формы;
- **checked** —**true**, если флажок или переключатель находится во включенном состоянии;
- **defaultChecked** — флажок или переключатель по умолчанию. Возвращает **true** или **false**;

Поля форм

- **disabled** — признак запрета: если задано значение **true**, то элемент является неактивным (отображается серым цветом);
- **indeterminate** — флажок находится в неопределенном состоянии (закрашивается серым). Возвращает **true** или **false**;
- **form** — ссылка на форму, в которой находится элемент;
- **name** — имя элемента;

Поля форм

- **type** — тип элемента формы.
- Методы:
- **blur()** убирает фокус ввода с текущего элемента формы;
- **focus()** помещает фокус на текущий элемент формы.
- События:
- **onblur** наступает при потере фокуса элементом формы;

Поля форм

- **onclick** возникает при выборе элемента;
- **onfocus** происходит при получении фокуса ввода элементом формы.
- Чтобы найти выбранный элемент-переключатель в группе, необходимо перебрать все переключатели в цикле. Получить значение выбранного переключателя можно через метод **item()**, указав индекс элемента в группе:

Поля форм

- var count = document.frm.radio1.length;
- for (i=0; i<count; i++) {
- if (document.frm.radio1.item(i).checked) {
- value1 = document.frm.radio1.item(i).value;
- break;
- }
- }

Поля форм

- **Кнопки.**
- Кнопки поддерживают следующие свойства:
- **value** — текст, отображаемый на кнопке;
- **disabled** — признак запрета: если задано значение **true**, то кнопка является неактивной (отображается серым цветом);
- **form** — ссылка на форму, в которой находится элемент;
- **name** — имя элемента;

Поля форм

- **type** — тип элемента формы.
- Методы традиционны:
- **blur()** убирает фокус ввода с текущего элемента формы;
- **focus()** помещает фокус на текущий элемент формы.
- События:
- **onblur** наступает при потере фокуса элементом формы;

Поля форм

- **onclick** возникает при нажатии кнопки;
- **onfocus** происходит при получении фокуса ввода элементом формы.
- Кнопка может быть вставлена в Web-страницу не только с помощью тега **<input>**, но и с помощью парного тега **<button>**. При использовании этого тега текст на кнопке можно сделать цветным, а также можно задать клавишу быстрого доступа.

Поля форм

- При одновременном нажатии клавиши, указанной в параметре **accesskey**, и клавиши <Alt> выполняется функция обработки события **onclick():**
- **<button accesskey="т" onclick="f_click();" id="button1" disabled>**