Кафедра «Автоматизированные станочные системы»

Dept. of Automated Manufacturing Systems

BOM- И DOM-ОБЪЕКТНЫЕ МОДЕЛИ БРАУЗЕРА

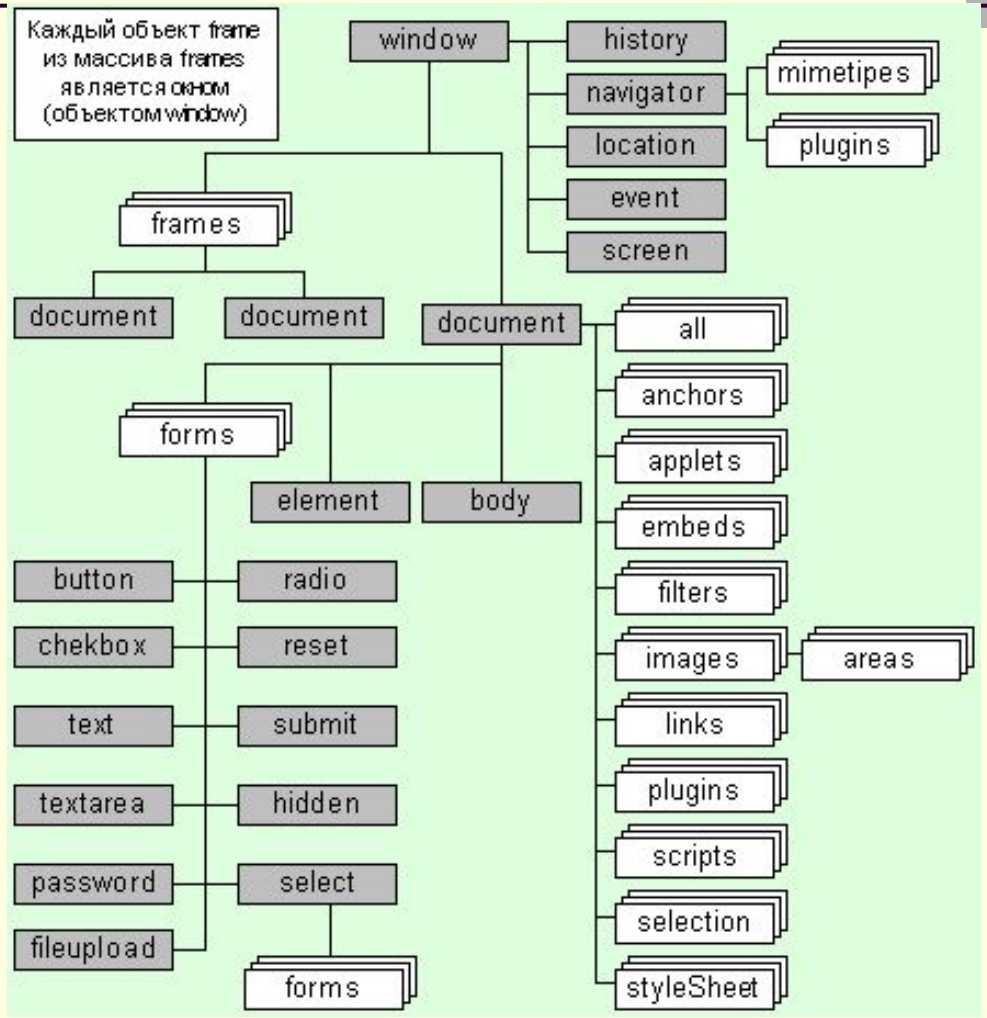
Для обращения из программы на JavaScript и других языках к отдельным элементам как Web-страницы, так и окна браузера используются два набора встроенных объектов со свойствами, методами и событиями.

За окно браузера отвечает так называемая BOM-модель (Browser Object Model), а за Web-страницу – DOM-модель (Document Object Model).

Коллекция – динамический массив, хранящий ту или информацию или ссылки на объекты. Например, все теги документа хранятся в коллекции тегов, все изображения – в коллекции images и т.д.

BOM - объектная модель браузера

Иерархическая структура объектной модели BOM



Основные объекты браузера

Объект document. Обладает коллекциями и свойствами, представляющими все содержимое HTML-документа. Кроме того, предоставляет методы и события для работы с документами.

Объект event. Глобальный объект, позволяющий программе обращаться к параметрам события.

Объект history. Содержит информацию об URL-адресах страницы, посещенных клиентом, хранящуюся в журнале браузера. Позволяет сценарию перемещаться по журналу.

Объект location. Содержит информацию о текущем URL. Предоставляет методы, позволяющие перезагрузить страницу.

Объект MimeType. Предоставляет информацию о типе данных MIME текущей страницы.

Объект navigator. Этот объект представляет собой браузер и содержит информацию о его производителе, версии и возможностях.

Объект screen. Предоставляет сценарию информацию о разрешающей способности и графических возможностях монитора клиента.

Объект Selection. Возвращает активный выделенный участок на экране. Предоставляет доступ к выделенным элементам, в том числе к тексту страницы.

Объект Style. Предоставляет доступ к отдельным стилевым свойствам элемента, которые заранее заданы таблицей стилей или строковым стилевым тегом на странице.

Объект Stylesheet. Этот объект представляет все стили одной таблицы стилей из коллекции stylesheets.

Объект TextRange. Этот объект представляет текстовый поток HTML-документа. Он может быть использован для чтения и записи текста в пределах страницы.

Объект window. Этот объект ссылается на текущее окно, которое может быть как окном верхнего уровня, так и кадром, созданным с помощью тега <FRAMESET> в другом документе.

Коллекции браузера

Имя	Описание
all	Коллекция всех тегов и элементов в теле документа
anchors	Коллекция всех тегов <a> документа
applets	Коллекция всех объектов в документе, включая встроенные элементы управления, изображения, апплеты, внедренные модули и т. д.
areas	Коллекция всех активных областей, входящих в состав карты-изображения
cells	Коллекция всех ячеек <TH> и <TD> в строке таблицы
elements	Коллекция всех управляющих и прочих элементов формы
embeds	Коллекция всех тегов <EMBED> в документе
filters	Коллекция всех объектов-фильтров для элемента
forms	Коллекция всех форм на странице
frames	Коллекция всех кадров, определенных в теге <FRAMESET>

Коллекции браузера (продолжение)

Имя	Описание
images	Коллекция всех изображений на странице
imports	Коллекция всех иллюстрированных таблиц стилей, определенных для Stylesheet
links	Коллекция всех ссылок и блоков <AREA> на странице
mimeTypes	Коллекция всех типов документов и файлов, поддерживаемых браузером
options	Коллекция всех пунктов в списке <SELECT>
plugins	Псевдоним для коллекции всех тегов <EMBED> на странице
rows	Коллекция всех строк в таблице, включая <THEAD>, <TBODY> и <TFOOT>
scripts	Коллекция всех блоков <SCRIPT> на странице
stylesheets	Коллекция всех объектов стилевых свойств документа

DOM - объектная модель документа

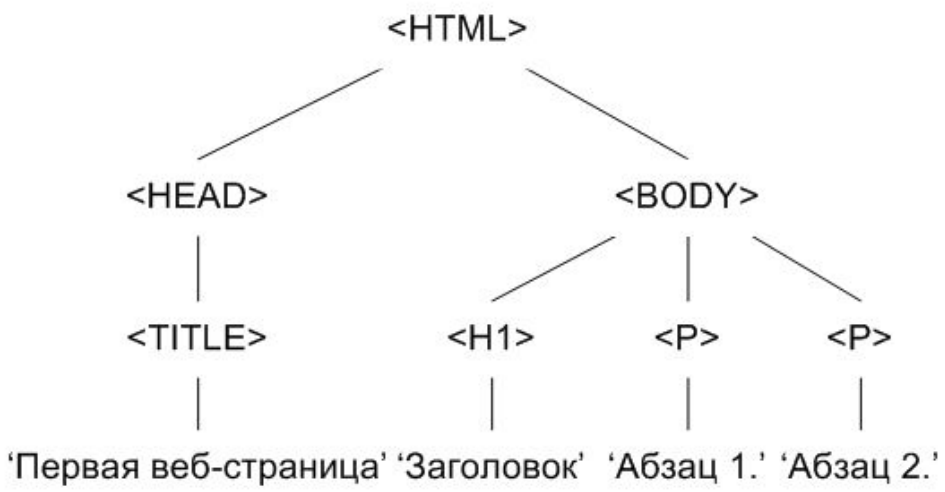
Объектная модель документа (Document Object Model – DOM) является стандартом и регламентирует способ представления содержимого документа (в частности Web-страницы) в виде набора объектов. Под содержимым понимается все, что может находиться на странице: рисунки, ссылки, абзацы, текст и т.д.

В отличие от объектной модели браузера (BOM), которая уникальна для каждого браузера, объектная модель документа является стандартна и должна поддерживаться всеми браузерами.

В DOM документ представляется в виде древовидной структуры. Это обеспечивает унифицированный способ навигации по документу.

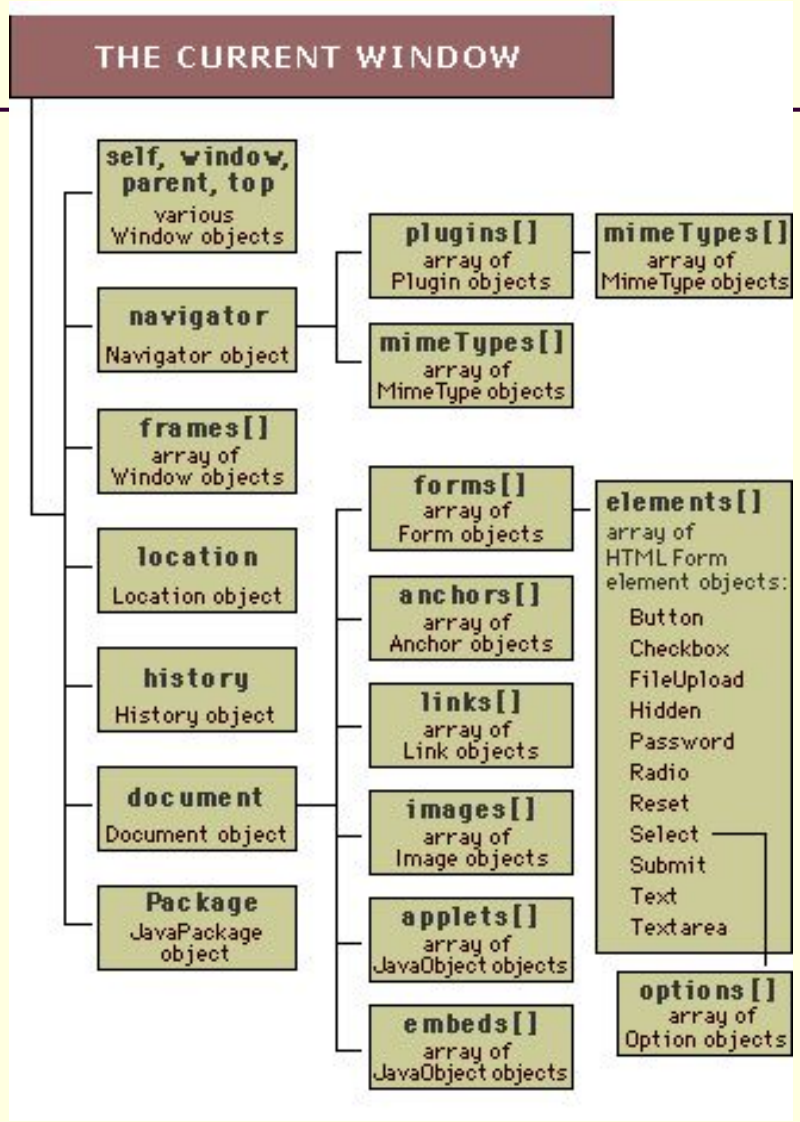
Представление HTML-документа в виде древовидной структуры

```
<html>
  <head>
    <title>Первая веб-страница</title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <p>Абзац 1.</p>
    <p>Абзац 2.</p>
  </body>
<html>
```



Иерархия объектов DOM-модели

Язык JavaScript имеет доступ ко всем этим объектам (фактически, они являются встроенными объектами Java).



В модели DOM к элементу документа можно обратиться непосредственно по его идентификатору id, воспользовавшись методом getElementById объекта Document.

Например:

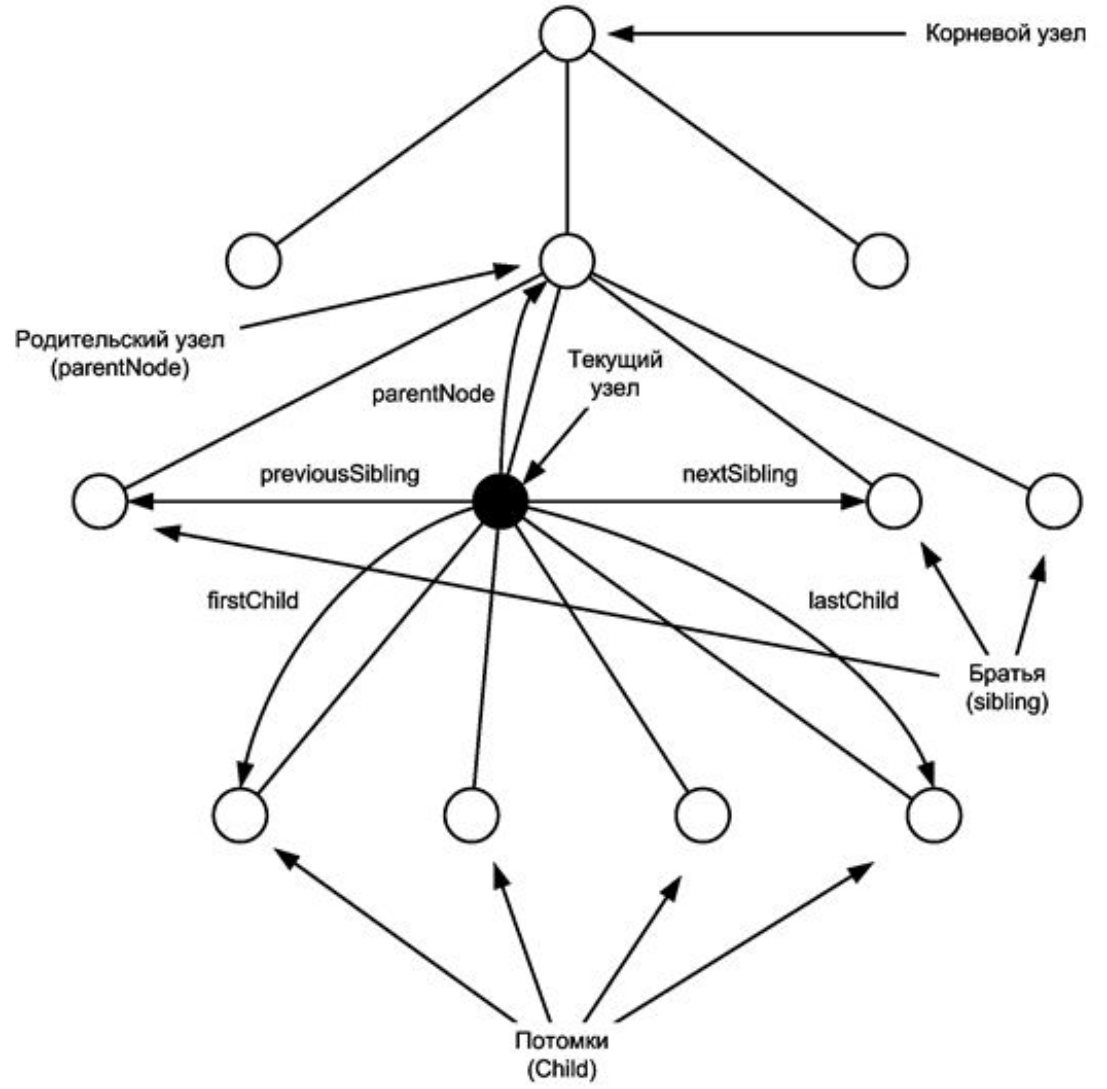
```
<html>
<head>
<title>Основы DOM</title>
</head>
<body>
<h1 id = "head">Основы DOM</h1>
<p>A Text</p>
<script language = "JavaScript">
var a = document.getElementById("head");
</script>
</body>
</html>
```

Для получения коллекции всех элементов, соответствующих какому-либо тегу, используется метод объекта Document getElementsByTagName.

Например, команда
var a = document.getElementsByTagName("TD")
присвоит переменной a коллекцию всех элементов <td>. Имя элемента (тега) следует писать прописными буквами ("TD").

Чтобы воспользоваться преимуществом древовидной структуры, принятой в DOM для представления документа, следует использовать навигационные атрибуты.

Навигационные атрибуты объекта Node



Навигационные атрибуты объекта Node

Атрибут	Описание
firstChild	Возвращает первый узел-потомок
lastChild	Возвращает последний узел-потомок
previousSibling	Возвращает предыдущий соседний узел, имеющий с текущим одного родителя
nextSibling	Возвращает следующий соседний узел, имеющий с текущим одного родителя
parentNode	Возвращает родительский узел
ownerDocument	Возвращает корневой узел документа, содержащий текущей узел
nodeName	Возвращает имя узла
nodeValue	Возвращает значение узла в текстовом формате
nodeType	Возвращает тип узла в виде числа

Динамическая генерация Web-страниц средствами DHTML на основе DOM

Часто требуется динамически формировать Web-страницы, например, в случае создания чатов, форумов, либо динамически создаваемых Web-страниц, содержимое которых хранится в базе данных. Использование DOM позволяет решить такую задачу.

Для создания объектов у объекта Document имеются следующие методы

Метод	Описание
createElement(имя_элемента)	Создает новый узел элемента с указанным именем
createTextNode(текст)	Создает текстовый узел с указанным текстом
createAttribute(имя_атрибута)	Создает новый узел атрибута с указанным именем

Вновь созданные объекты добавляются в структуру документа при помощи методов объекта Node.

Метод	Описание
appendChild(новый_узел)	Добавляет объект Node в конец списка узлов-потомков
cloneNode(потомок-опция)	Создает объект Node, идентичный указанному в аргументе. В качестве аргумента можно использовать и все узлы-потомки одновременно
hasChildNodes()	Возвращает true, если узел имеет потомков
insertBefore(новый_узел, текущий_узел)	Вставляет объект Node в список потомков перед узлом, указанным в качестве второго параметра
removeChild(узел-потомок)	Удаляет узел-потомок, указанный в качестве параметра
replaceChild(новый_потомок, старый_потомок)	Заменяет старого потомка на нового

Пример динамической генерации документа средствами DOM:

```
<html>
<head>
<title>Пример динамической генерации документа</title>
</head>
<body>
<script language = "JavaScript">
var newText;
var newElem;
newText = document.createTextNode
("Пример динамического создания HTML-документа");
newElem = document.createElement("H1");
newElem.appendChild(newText);
document.body.appendChild(newElem);
newText = document.createTextNode("Абзац");
newElem = document.createElement("P");
newElem.appendChild(newText);
document.body.appendChild(newElem);
</script>
</body>
</html>
```

Для чтения и установки атрибутов используются следующие методы объекта Element.

Метод	Описание
`getAttribute (имя_атрибута)`	Возвращает значение атрибута
`setAttribute(имя_атрибута, значение)`	Устанавливает значение атрибута
`removeAttribute (имя_атрибута)`	Устанавливает значение атрибута по умолчанию, удаляя текущее значение

Как правило, программы для Web-страниц управляются событиями. Чтобы узнать, какое событие произошло, в DOM имеется объект события event. Объект event является локальным и его следует явным образом передавать в обработчик события

Свойство	Описание
bubbles	Указывает возможность «всплывания» события (передачи управления вверх по иерархической структуре)
cancelable	Указывает возможность отмены действия события, заданного по умолчанию
currentTarget	Указывает событие, обрабатываемое в данный момент
eventPhase	Указывает фазу возбуждения события
target	Указывает элемент, вызвавший событие
timestamp	Указывает время возникновения события
type	Указывает имя события

События, связанные с мышью, генерирует объект mouse.