

Лекция № 2. Объектная модель браузера

УЧЕБНЫЕ ВОПРОСЫ

1. Понятие объектной модели браузера WOM
2. Основные объекты WOM

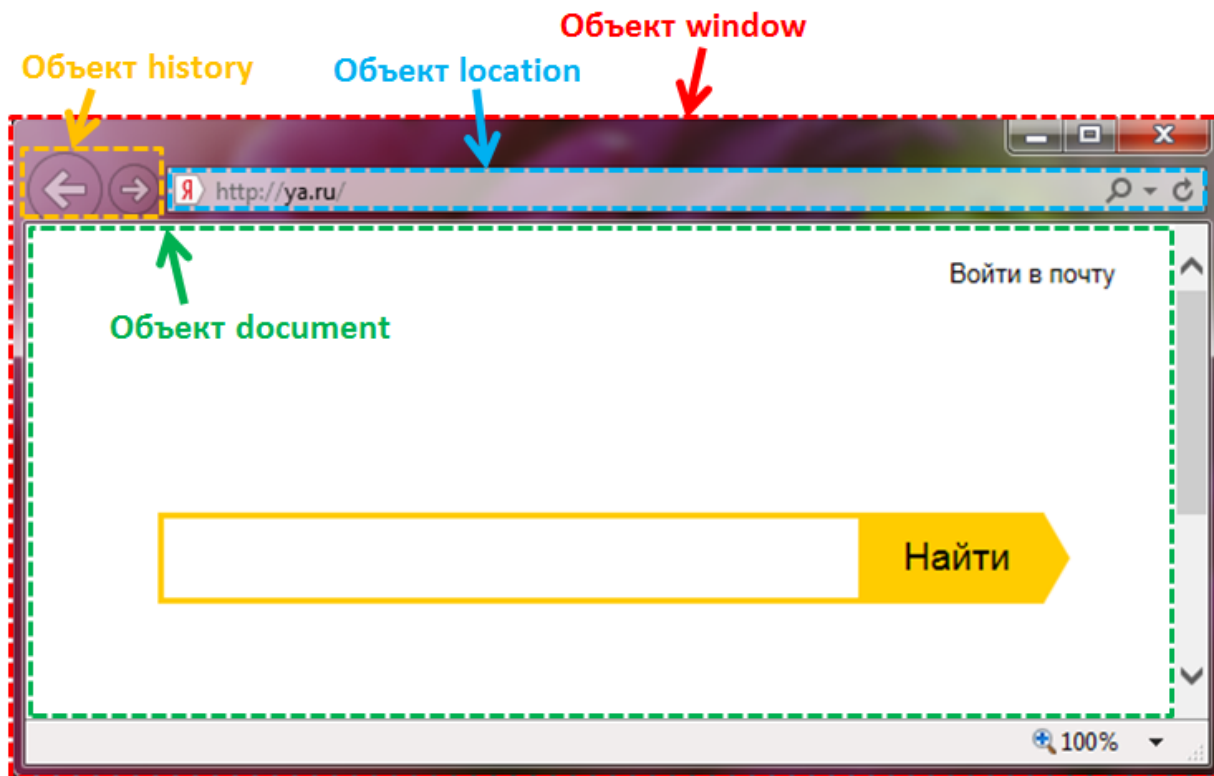
Вопрос №1. Понятие объектной модели браузера
ВОМ

Веб-страницы бывают *статическими* и *динамическими*, последние отличаются тем, что в них используются *сценарии* (программы) на языке JavaScript.

Объектная модель браузера (BOM – Browser Object Model) – это набор объектов, с помощью которых разработчик может взаимодействовать с элементами веб-страницы и самим браузером в сценариях JavaScript.

На самом верху этой модели находится **глобальный объект window**. Он представляет собой одно из окон или вкладку браузера с его панелями инструментов, меню, строкой состояния, HTML страницей и другими объектами.

Доступ к этим различным объектам окна браузера осуществляется с помощью следующих основных объектов: **navigator, history, location, screen, document** и т.д.



Так как данные объекты являются *дочерними* по отношению к объекту **window**, то обращение к ним происходит как к *свойствам* объекта **window**.

Например, для того чтобы обратиться к объекту **screen**, необходимо использовать следующую конструкцию:

window.screen.

Но если мы работаем с текущим окном, то "**window.**" можно опустить. Например, вместо **window.screen** можно использовать просто **screen**.

Из всех этих объектов, наибольший интерес и значимость для разработчика представляет объект **document**, который является корнем объектной модели документа (DOM – Document Object Model). Данная модель в отличие от объектной модели браузера **стандартизована** в спецификации и **поддерживается** всеми браузерами.

Объект `document` представляет собой HTML документ, загруженный в окно (вкладку) браузера. С помощью свойств и методов данного объекта Вы можете получить доступ к содержимому HTML-документа, а также изменить его содержимое, структуру и оформление.

Основная задача при создании динамических веб-страниц в основном сводится к **отбору** нужных объектов (элементов) и **выполнению** над ними различных действий.

Результаты этих действий **сразу отображаются** на экране пользователя, а точнее в тех местах, за которые эти объекты отвечают.

Вопрос №2.

Основные объекты ВОРМ

Основные объекты Browser Object Model:

window,

navigator,

history,

location,

screen,

document.

Объект window

window – самый главный объект в браузере, который отвечает за одно из окон (вкладок) браузера. Он является корнем иерархии всех объектов доступных веб-разработчику в сценариях JavaScript.

Объект window

Объект window кроме глобальных объектов (document, screen, location, navigator и др.) имеет **собственные свойства** и методы, которые предназначены для:

- открытия нового окна (вкладки);
- закрытия окна (вкладки) с помощью метода close();
- распечатывания содержимого окна (вкладки);
- передачи фокуса окну или для его перемещения на задний план (за всеми окнами);
- управления положением и размерами окна, а также для осуществления прокручивания его содержимого;
- изменения содержимого статусной строки браузера;
- взаимодействия с пользователем
- выполнения определённых действий через определённые промежутки времени и др.

Объект window

Взаимодействие с пользователем осуществляется посредством следующих окон:

alert (для вывода сообщений),

confirm (для вывода окна, в котором пользователю необходимо подтвердить или отменить действия),

prompt (для получения данных от пользователя);

Объект window

Если в браузере открыть несколько вкладок (окон), то браузером будет создано столько объектов window, сколько открыто этих вкладок (окон). Т.е. каждый раз открывая вкладку (окно), браузер создаёт новый объект window связанный с этой вкладкой (окном).

Объект window

Примеры:

Вызвать **метод write** объекта document, который расположен в текущей вкладке (окне) браузера:

```
window.document.write("Строчка текста");
```

или

```
document.write("Строчка текста");
```

Вызвать **метод alert** для текущей вкладки (окна) браузера:

```
window.alert("Сообщение");
```

или

```
alert("Строчка текста");
```

Объект navigator

navigator – информационный объект, который предназначен для предоставления **подробной информации о браузере**, который пользователь использует для доступа к сайту или веб-приложению. Кроме данных о браузере, в нём ещё содержится сведения о операционной системе, сетевом соединении и др.

Доступ к данному объекту осуществляется как к **свойству** объекта window, т.е. через точку или без нее.

Объект navigator имеет **свойства** и **методы**. Очень часто они используются для того чтобы узнать, какие функции поддерживаются браузером, а какие нет.

Объект navigator

Свойства объекта navigator:

- `appName` – кодовое имя браузера;
- `appName` – имя браузера;
- `appVersion` — версия браузера;
- `cookieEnabled` - позволяет определить включены ли cookie в браузере;
- `geolocation` - используется для определения местоположения пользователя;
- `language` - язык браузера;
- `online` - имеет значение `true` или `false` в зависимости от того находится ли браузер в сети или нет;

Объект navigator

Свойства объекта navigator:

- platform - название платформы, для которой скомпилирован браузер;
- product - имя движка браузера;
- userAgent - возвращает заголовок user agent, который браузер посылает на сервер.

Методы объекта navigator:

- javaEnabled – позволяет узнать, включён ли в браузере Java;
- sendBeacon - предназначен для отправки небольшого количества информации на веб-сервер без ожидания ответа.

Объект history

history – объект, который позволяет получить **историю переходов** пользователя по ссылкам в пределах одного окна (вкладки) браузера. Эти переходы браузер сохраняет в **сессию** истории текущего окна (вкладки).

Данный объект отвечает за кнопки **forward** (вперёд) и **back** (назад). С помощью методов объекта **history** можно **имитировать нажатие** на эти кнопки, а также переходить на определённое количество ссылок в истории вперёд или назад.

Объект history

Каждый элемент в сессии состоит из:

- data - это некоторые данные, которые можно связать с определённым элементом истории.
- title - название пункта истории.
- url - URL адрес страницы.

Объект history

Свойства, методы и события объекта history

- Свойство `history.length` - возвращает **количество элементов** в сессии истории текущего окна (вкладки). Другими словами, данное свойство возвращает **количество переходов**
- Свойство `history.state` - позволяет получить данные (`data`) текущего элемента истории. Эти данные можно **добавить** к элементу истории с помощью методов `history.pushState()` и `history.replaceState()`.

Объект history

Свойства, методы и события объекта history

- Метод `history.go()`. Он позволяет переместить пользователя на некоторое количество страниц вперёд или назад по истории. Метод `history.go()` имеет один обязательный параметр - это число, на которое надо переместить пользователя вверх или вниз по истории.
- Метод `history.back()` - осуществляет перемещение пользователя на одну страницу назад по истории, т.е. он программно 'имитирует' нажатие кнопки "Назад" в браузере.

Объект history

Свойства, методы и события объекта history

- Метод `history.forward()` - осуществляет перемещение пользователя на одну страницу вперёд по истории, т.е. он программно "имитирует" нажатие кнопки "Вперёд" в браузере.
- Метод `history.pushState()` - позволяет добавить новую запись в сессию истории текущего окна (вкладки). Добавление записи (элемента) осуществляется в конец сессии истории.

Объект location

location – объект, который отвечает за адресную строку браузера. Данный объект содержит свойства и методы, которые позволяют:

- получить текущий адрес страницы браузера,
- перейти по указанному URL,
- перезагрузить страницу и т.п.

Объект location

Свойства объекта location:

- hash - устанавливает или возвращает якорную часть (#) URL;
- host - устанавливает или возвращает имя хоста и номер порта URL;
- hostname - устанавливает или возвращает имя хоста URL;
- href - устанавливает или возвращает содержимое URL;
- origin - возвращает протокол, имя хоста и номер порта URL;
- pathname - устанавливает или возвращает часть URL, содержащей путь;

Объект location

Свойства объекта location:

- port - устанавливает или возвращает номер порта URL;
- protocol - устанавливает или возвращает протокол URL;
- search - устанавливает или возвращает часть URL, содержащей строку с параметрами (?параметр1=значение1&параметр2=значение2&...);

Объект location

Свойства объекта location:

<http://itchief.ru/search?q=bootstrap#part2>

1 2 3 4 5

location.href

- 1 – location.protocol
- 2 – location.hostname
- 3 – pathname.hostname
- 4 – search.hostname
- 5 – hash.hostname

Объект location

Изменение URL или какой либо её части с помощью свойств объекта location приводит к **немедленному переходу** к этому URL в текущем окне, или в том окне или вкладке браузера, для которого этот объект был вызван.

Объект screen

screen – объект, который предоставляет информацию об экране пользователя.

С помощью свойств объекта `screen` мы можем получить следующую информацию об экране:

- `width` – ширина экрана в пикселях;
- `height` – высота экрана в пикселях;
- `availWidth` – доступная ширина экрана;
- `availHeight` – доступная высота экрана (например, в Windows, эта высота равна общей высоте из которой нужно вычесть высоту панели задач);

Объект screen

screen – объект, который предоставляет информацию об экране пользователя.

С помощью свойств объекта `screen` мы можем получить следующую информацию об экране:

- `availLeft` – x-координата первого доступного пикселя;
- `availTop` – y-координата первого доступного пикселя;
- `colorDepth` – глубину цвета в битах;
- `pixelDepth` – глубину цвета на пиксель экрана в битах;
- `orientation` – ориентация экрана.

Объект `document`

document – HTML документ, загруженный в окно (вкладку) браузера. Он является **корневым узлом** HTML документа и "владельцем" всех других узлов: элементов, текстовых узлов, атрибутов и комментариев.

Объект **document** содержит **свойства** и **методы** для доступа ко всем узловым объектам.

document как и другие объекты, является **частью объекта** `window` и, следовательно, он может быть доступен как `window.document`.

Объект document

Свойства и методы объекта document

- Свойство `document.characterSet` - возвращает **кодировку**, которая используется для рендеринга текущего документа.
- Свойство `document.referrer` - возвращает строку, содержащую **адрес (URL) страницы**, с которой **пользователь пришёл** на эту страницу. Если текущий документ не был открыт через ссылку (например, с помощью закладки или прямого ввода адреса в адресную строку), то данное свойство вернёт пустую строку.

Объект document

Свойства и методы объекта document

- Свойство `document.readyState` - возвращает строку, содержащую **статус** текущего документа. Данное свойство доступно только для чтения.

В процессе загрузки документ последовательно проходит следующие состояния:

1. **uninitialized** - процесс загрузки ещё не начался;
2. **loading** - идёт процесс загрузки;
3. **loaded** - загрузка HTML кода завершена;
4. **interactive** - документ достаточно загружен для того, чтобы пользователь мог взаимодействовать с ним. Код JavaScript может начать выполняться только на этом этапе;
5. **complete** - документ полностью загружен.

Объект document

Свойства и методы объекта document

- Свойство `document.cookie` - позволяет получить или установить **cookie**, которые будут связаны с текущим документом.
- Свойство `document.URL` - возвращает строку, содержащую полный URL **адрес текущего HTML документа**.
- Свойство `document.domain` - возвращает строку, содержащую **доменное имя сервера**, с которого загружен текущий документ. Если домен текущего документа не может быть определён, то данное свойство вернёт значение `null`.

Объект document

Свойства и методы объекта document

- Метод `document.write()` - предназначен для **вывода** в документ **строки**, указанной в качестве параметра данного метода.

Если данный метод вызывается **в процессе загрузки** документа, то он выводит строку в **текущем** месте.

В том случае, если данный метод вызывается **после загрузки** документа, то он приводит к **полной очистке** этого документа и вывода строки.

Это происходит, потому что после загрузки документа браузер уже **полностью построил DOM** и в документ уже нельзя внести изменения таким способом, а только с помощью **добавления узлов** в объектную модель документа.

Объект document

Свойства и методы объекта document

- Метод `document.writeln()` - полностью **аналогичен методу** `document.write()` за тем исключением, что он добавляет символ перевода строки (`\n`) в конец строки, указанной в качестве параметра.

Объект document

Свойства и методы объекта document

- Метод `document.open()` — открывает **поток для вывода** в текущий документ строк с помощью методов `document.write()` и `document.writeln()`. Данный метод (`document.open()`) обычно применяется **для формирования нового документа**. Вызов данного метода в текущем документе приводит к **его полной очистке**. После того как документ открыт методом `document.open()` и сформирован с помощью методов `document.write()` и `document.writeln()`, его необходимо закрыть методом `document.close()`.
- Метод `document.close()` - предназначен для **закрытия потока**, который Вы открыли с помощью метода `document.open()`.