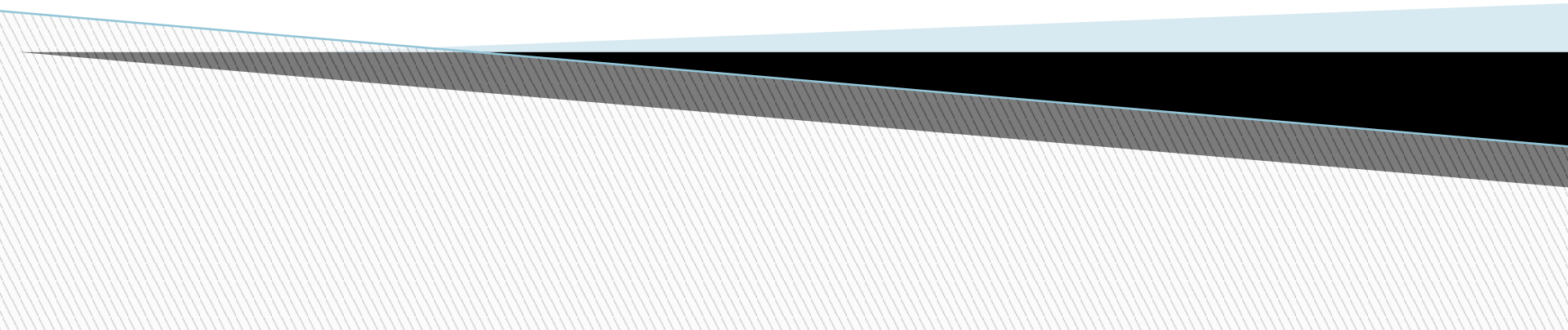


Обзор технологий разработки клиентских веб-приложений

ст.преподаватель кафедры ПБИ
Еремеев Алексей Александрович



Введение

Как правило, *веб-приложение* - приложение, в котором клиентом выступает *браузер*, а сервером - *веб-сервер*.

Помимо клиента и сервера в веб-сеансе могут участвовать и другие программы, которые и являются объектом *веб-программирования*.

Результатом работы веб-приложения является *веб-страница*, отображаемая в окне браузера. При этом само *веб-приложение* может выполняться как на компьютере клиента, так и на компьютере сервера.

Программы, выполняющиеся на клиент-машине

Одним из типов программ, предназначенных для выполнения на клиент-машине, являются сценарии, например, JavaScript (VBScript). Исходный текст сценария представляет собой часть веб-страницы, поэтому сценарий JavaScript передается клиенту вместе с документом, в состав которого он входит. Обработывая HTML-документ, браузер обнаруживает исходный текст сценария и запускает его на выполнение.

Одно общее требование: эти программы должны быть лишены возможности обращаться к ресурсам компьютера, на котором они выполняются.

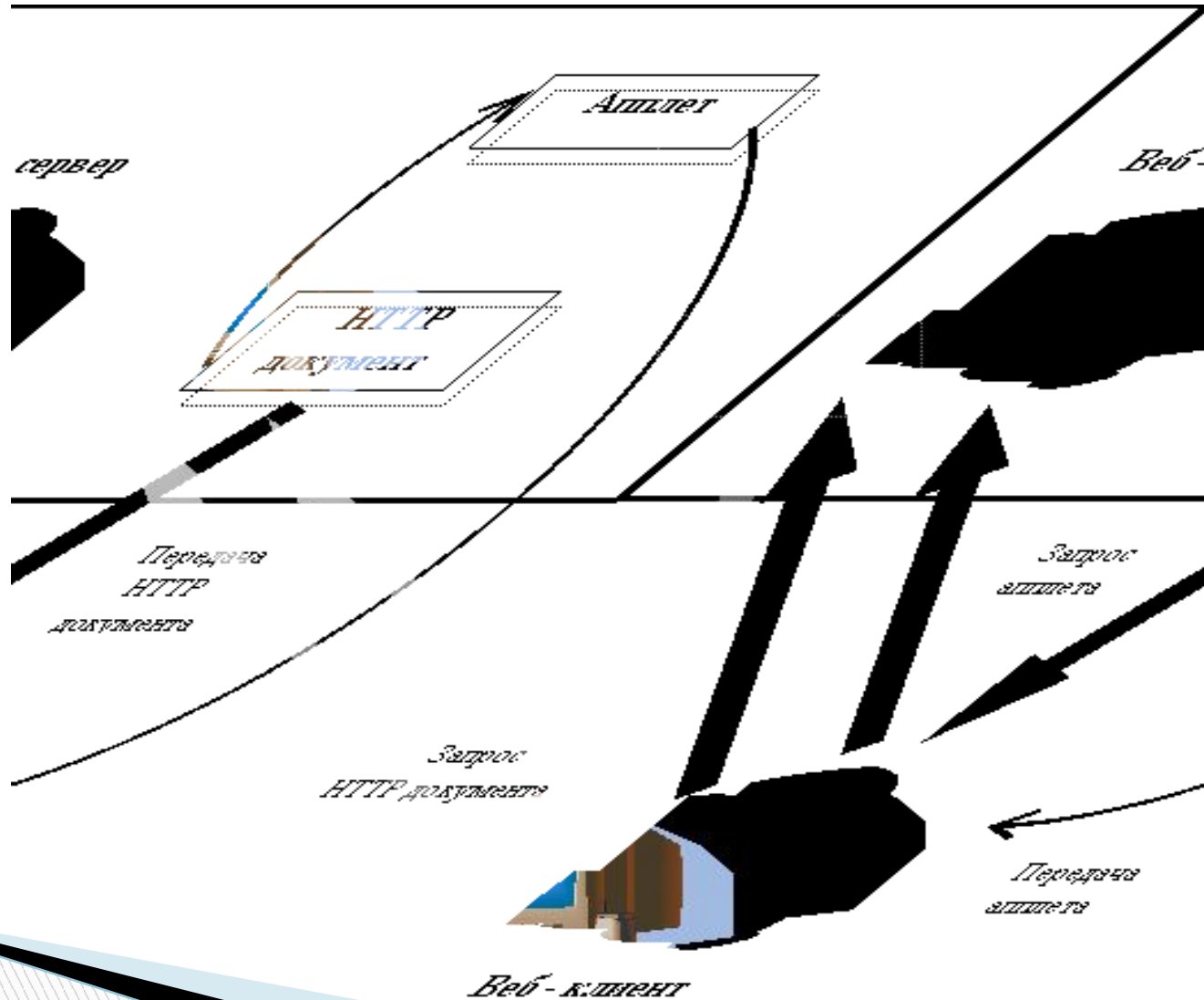
Языки, предназначенные для создания программ, выполняющихся на клиент-машине, должны быть абсолютно непригодны для написания вирусов и подобных программ.

Насыщенные интернет-приложения

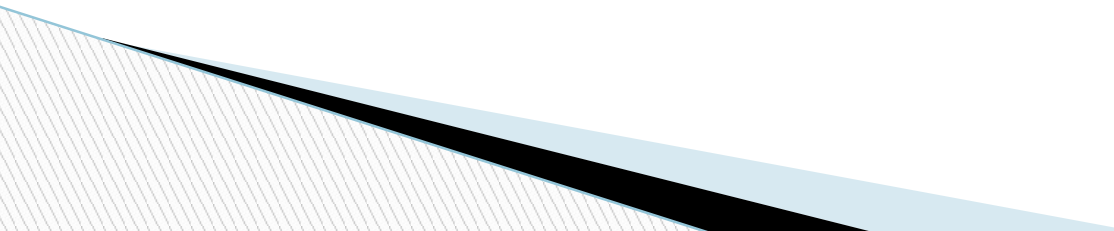
Подход заключается в использовании Adobe Flash или Java-апплетов для полной или частичной реализации пользовательского интерфейса, поскольку большинство браузеров поддерживает эти технологии (как правило, с помощью плагинов).

После того как клиент получает HTML-код документа, включающего апплет, он генерирует дополнительный запрос серверу. После того как сервер пересылает клиенту код апплета, сам апплет запускается на выполнение.

Передача клиенту Java-апплета



Проблемы насыщенных интернет-приложений

- необходимость обеспечения безопасной среды выполнения;
 - для исполнения кода должно быть разрешено исполнение сценариев;
 - потеря в производительности (т.к. выполняется на клиентской стороне);
 - требуется много времени на загрузку.
- 

Java-апплеты

Java-апплет - это программа, написанная на языке Java и откомпилированная в байт-код. Апплеты используются для предоставления интерактивных возможностей веб-приложений, которые не возможны в HTML. Так как байт-код Java платформо-независим, то Java-апплеты могут выполняться браузерами на многих операционных платформах.

Код апплета загружается с веб-сервера, и браузер

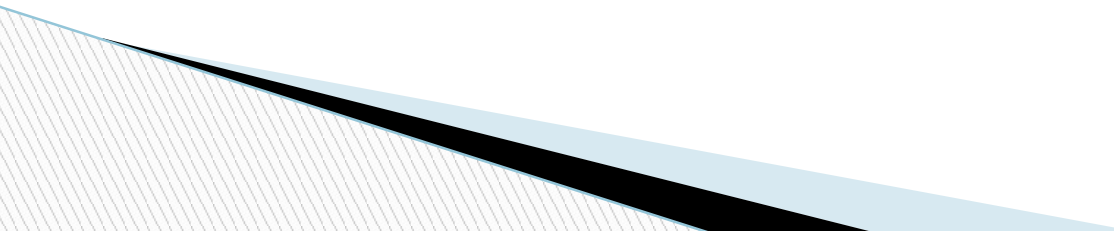
- либо вставляет апплет в веб-страницу;
- либо открывает отдельное окно с собственным пользовательским интерфейсом апплета.

Апплет может быть внедрен в веб-страницу с помощью использования HTML тега `<applet>`, или (что рекомендуется) тега `<object>`.

Преимущества *Java*-апплетов

- работают практически на большинстве операционных платформ;
- поддерживаются большинством браузеров;
- кэшируются в большинстве браузеров, что существенно ускоряет их загрузку при возвращении на веб-страницу;
- после первого запуска апплета, когда Java-машина уже выполняется и быстро запускается, выполнение апплетов происходит существенно быстрее;
- загружаются со скоростью сопоставимой с программами на других компилируемых языках, например C++, но во много раз быстрее чем на JavaScript.

Недостатки *Java*-апплетов

- требуется установка Java-расширения, которые доступны по умолчанию не во всех браузерах;
 - проблемы реализации Java-расширений для 64-разрядных процессоров;
 - не могут запускаться до первой загрузки виртуальной Java-машина, что может занимать значительное время;
 - разработка пользовательского интерфейса с использованием апплетов является более сложной задачей по сравнению с HTML;
 - не имеют прямого доступа к локальным ресурсам клиентского компьютера.
- 

Введение в JScript

JavaScript - интерпретируемый язык программирования, стандартизированный международной организацией ECMA в спецификации ECMA-262.

Синтаксис JScript во многом аналогичен языку JavaScript, однако, помимо добавления клиентских скриптов на веб-страницы и некоторых других функций, JScript может использоваться и для других целей, например:

- автоматизация администрирования систем Microsoft Windows;
- создание страниц ASP.

Язык JScript получил дальнейшее развитие в виде языка JScript.NET, который ориентирован на работу в рамках платформы Microsoft.NET

JScript - интерпретируемый, объектно-ориентированный язык. Хотя он имеет существенно меньшее количество возможностей, чем такие объектно-ориентированные языки как C++ и Java.

Возможности JScript

Возможности языка существенно ограничены:

- язык не позволяет разрабатывать самостоятельные приложения;
- сценарии на JScript могут выполняться только при помощи интерпретатора, в частности веб-браузером.
- JScript - язык без строгого контроля типов. Поэтому не требуется объявлять тип переменных явно. Кроме того, во многих случаях JScript исполняет преобразования автоматически, когда они необходимы. Например, при сложении строки и числа, число будет преобразовано в строку.

Краткая характеристика VBScript

Visual Basic Scripting Edition (обычно просто VBScript) - сценарный язык программирования, интерпретируемый компонентом Windows Script Host. Он широко используется при создании скриптов в операционных системах семейства Microsoft Windows.

Синтаксис VBScript является упрощённой версией синтаксиса языка Visual Basic.

Сценарии на языке VBScript чаще всего используются в следующих областях, использующих программные продукты Microsoft:

- ▣ автоматизация администрирования систем Windows;
- ▣ серверный программный код в страницах ASP;
- ▣ клиентские сценарии в браузере Internet Explorer.

Microsoft Silverlight и XAML

Silverlight предоставляет графическую систему, схожую с Windows Presentation Foundation, и объединяет мультимедиа, графику, анимацию и интерактивность в одной программной платформе. Он был разработан чтобы работать с XAML и с языками Microsoft .NET. XAML используется для разметки страниц, использующих векторную графику и анимацию. Текст, содержащийся в Silverlight приложениях, доступен для поисковых систем, так как его код доступен в виде XAML, т.е. имеет текстовый формат как и XML.

Понятие о DOM

DOM (Document Object Model) - объектная модель документа. Это независимый от платформы и языка программный интерфейс, позволяющий программам получать доступ к содержимому документов, а также изменять содержимое, структуру и вид документов.

В рамках DOM любой документ представляется в виде дерева узлов. Каждый узел представляет собой элемент, атрибут, текстовый, графический или любой другой объект. Узлы между собой находятся в отношении "родитель-потомок".

Иерархия классов DOM

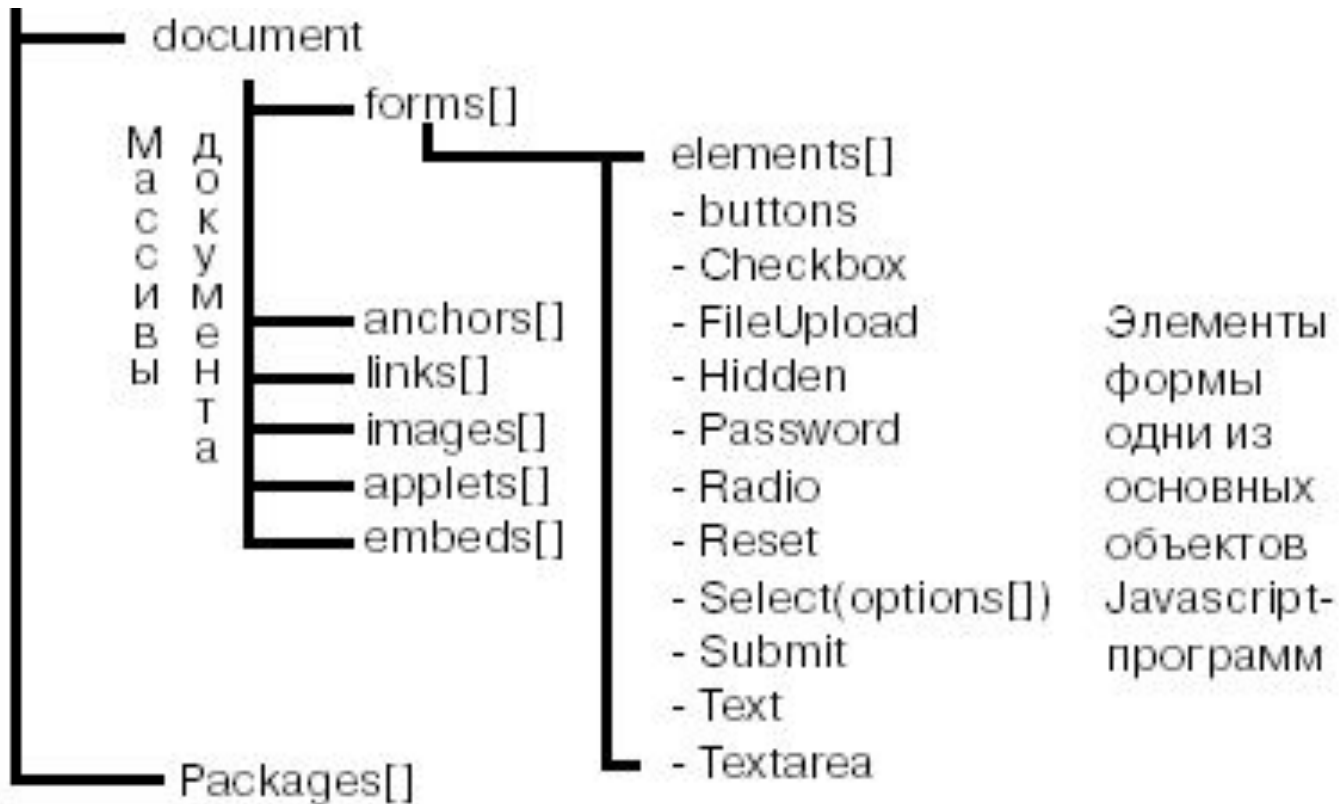
Объектно-ориентированный язык программирования предполагает наличие *иерархии классов объектов*. В JavaScript такая иерархия начинается с *класса объектов window*, т.е. каждый объект приписан к тому или иному окну. Для обращения к любому объекту или его свойству указывают полное или частичное имя этого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

Фрагмент DOM

Window ← Самый старший класс Javascript
(self, parent, top)

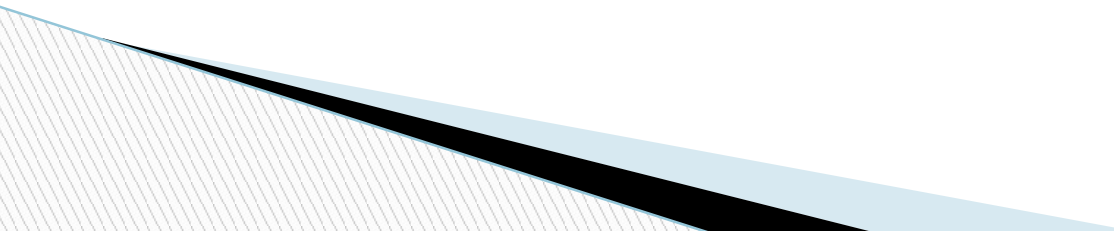


Фрагмент DOM (продолжение)



DHTML

Динамический HTML или DHTML представляет собой набор технологий, которые совместно позволяют создавать интерактивные веб-сайты на основе статического языка разметки (HTML), языка создания клиентских сценариев (JavaScript), языка описания представления документа (CSS) и документной объектной модели (DOM).



Что позволяет DHTML?

DHTML позволяет сценарным языкам изменять переменные языка описания представления документа, таким образом, изменяя вид и поведение прежде статического содержимого *HTML* документа уже после полной загрузки документа и в процессе просмотра его пользователем. Таким образом, динамичность, привносимая *DHTML*, проявляется себя в процессе просмотра страницы, но не имеет никакого отношения к генерации содержимого страницы при каждой ее загрузке.