



*Кафедра «Автоматизированные станочные системы»
Dept. of Automated Manufacturing Systems*

ДИНАМИЧЕСКИЕ WEB-СТРАНИЦЫ СЕРВЕРНЫЕ СЦЕНАРИИ

Динамический HTML (Dynamic HTML, или DHTML) - это развитие традиционного языка HTML, позволяющее создавать интерактивные, "живые" HTML-документы.

Один из методов реализации идеи динамичности заставил разработчиков "дополнить" HTML и встроить в браузеры средства обработки специальных языков программирования, таких как JavaScript и VBScript.

Сами сценарии (скрипты), т. е. программы, написанные на одном из этих языков, можно размещать вместе с основным содержанием и разметкой HTML-документа.

Программы на этих языках передаются по сети Интернет в виде исходных текстов, а транслируются и выполняются после загрузки в конкретный браузер

Серверная и локальная обработка данных

Программа, выполняющая генерацию HTML-кода, может выполняться как на сервере, так и на машине клиента.

Для серверной обработки применяются такие языки, как [Perl](#), [CGI](#), [PHP](#), [Ruby](#), [Python](#). На выходе они генерируют HTML-документ, передаваемый клиенту.

Для локальной обработки чаще всего применяется язык JavaScript. Он также на выходе создает HTML-документ

Языки генерации HTML-страниц, как правило, являются интерпретируемыми. Поэтому программы на таких языках называются **сценариями** или **скриптами**.

Сценарий или **скрипт** - программа на специализированном языке программирования, выполняющаяся в режиме интерпретации.

Под **Web-приложением** подразумевается программа в виде загрузочного модуля, выполняемая либо под управлением браузера (**клиентское Web-приложение**), либо под управлением Web-сервера (**серверное Web-приложение**).

Под **Web-активностью** понимают возможность организации интерактивного взаимодействия с пользователем.

ДИНАМИКА ≠ АКТИВНОСТЬ

Динамика позволяет "оживить" представление информации.

Активность есть набор методов и средств, которые позволяют обеспечить "подстройку" сайта под конкретного пользователя и его нужды.

Различают **клиентскую** и **серверную активность**.

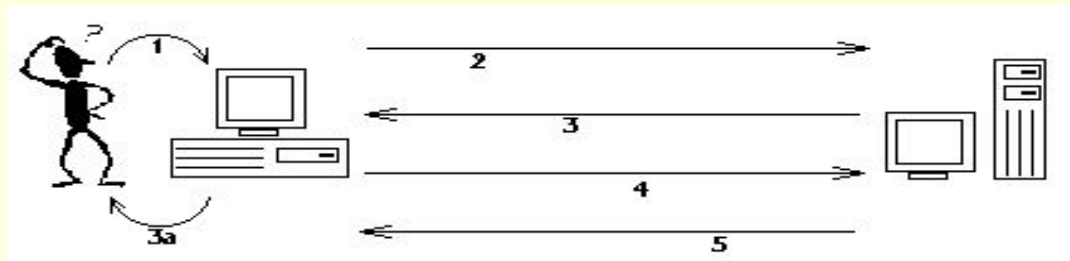
Клиентская активность позволяет за счет внедрения в Web-страницы сценариев и некоторых приложений поддерживать диалог с пользователем исключительно средствами Web-браузера.

При **серверной активности** введенные пользователем данные передаются Web-серверу, обрабатываются им, и результат обработки пересылается обратно пользователю.

Типичные схемы взаимодействия пользователя с Web-сервером:

- Взаимодействие с пассивным Web-сервером.
- Взаимодействие с активным Web-сервером.

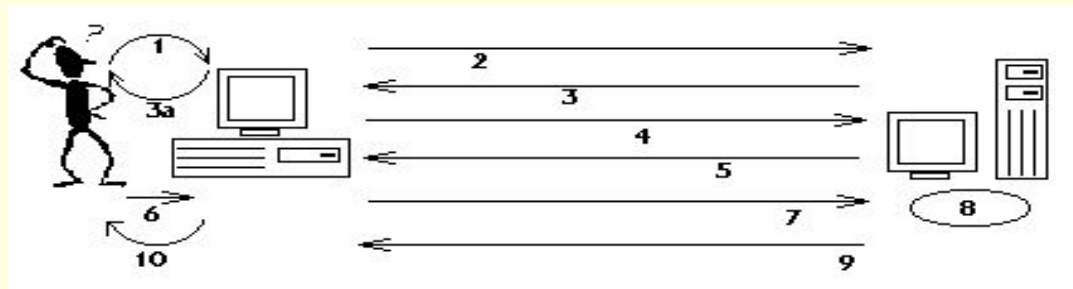
Сеанс взаимодействия с пассивным Web-сервером



Этапы взаимодействия с пассивным сервером:

1. Пользователь передает браузеру адрес URL необходимой ему Web-страницы.
2. Браузер обращается к серверу за нужной страницей.
3. Происходит загрузка страницы и ее содержимое отображается на экране пользователя (За).
4. В процессе интерпретации страницы браузер запрашивает объекты, ссылки на которые та содержит (рисунки, аудиофайлы, видеоклипы и т.п.)
5. Запрошенные объекты передаются с сервера и отображаются браузером пользователя.

Сеанс взаимодействия с активным Web-сервером



Этапы взаимодействия с активным сервером:

Действия, выполняемые на этапах 1..5 аналогичны приведенным в ранее рассмотренной схеме.

6. Пользователь вводит необходимую информацию посредством элементов пользовательского интерфейса (формы), расположенных на текущей Web-странице.

7. Введенные данные пересылаются браузером серверу.

8. На стороне сервера исполняется серверный сценарий либо серверное приложение.

9. Результат работы приложения/сценария в виде HTML-страницы пересылается браузеру.

10. HTML-страница отображается на экране пользователя.

Серверные сценарии

Основные отличия серверных сценариев от клиентских:

- интерпретатором серверного сценария является Web-сервер, а не браузер клиента;
- пользователю доступен исходный текст клиентских скриптов, в то время, как ему виден лишь результат работы серверных;
- серверные сценарии не зависят от клиентского программного обеспечения;
- возможности серверных сценариев существенно шире, чем клиентских, в частности, с их помощью можно осуществлять доступ к базам данных.

CGI-сценарии

Хронологически первой технологией создания серверных сценариев была технология CGI-сценариев или CGI-скриптов. В качестве языка программирования при разработке CGI-скриптов используется язык Perl, отдаленно напоминающий язык C.

Возможности CGI-сценариев:

- обработка пользовательских данных;
- доступ к файловой системе Web-сервера;
- работа с базами данных;
- динамическая генерация Web-страниц

Технология ASP

Технология **активных серверных страниц (ASP - Active Server Pages)** была предложена сравнительно недавно специалистами компании Microsoft и завоевала широкую популярность, благодаря своей простоте, логичности и удобству применения.

Отличия технологии ASP от CGI-сценариев:

- активная страница представляет собой Web-страницу со встроенными в текст сценариями, интерпретируемыми сервером;
- в зависимости от работы сценария, клиенту передается тот или иной документ, который может содержать теги HTML, расположенные на данной активной странице, либо сформированный самим сценарием;
- сценарии ASP составляются на языках JavaScript и VBScript;
- имеется набор объектов, содержащих свойства и методы, очень удобные для использования на стороне сервера;

Технология ASP удачно сочетается с другой технологией от Microsoft - ADO (от ActiveX Data Objects), что позволяет за короткий срок создавать сложные Интернет-приложения с базами данных.

Технология CGI-сценариев больше ориентирована на Web-сервера по ОС семейства UNIX (например, Web-сервер Apache).

ASP изначально предназначалась для Web-серверов от Microsoft (продукты MS Internet Information Server и MS Personal Web Server).