B211 Internet Computing

# Server-Side Scripting Languages



## Lecture Outline

- Choosing Server-Side Scripting Technologies
- The Growth of Server-Side Scripting Technologies

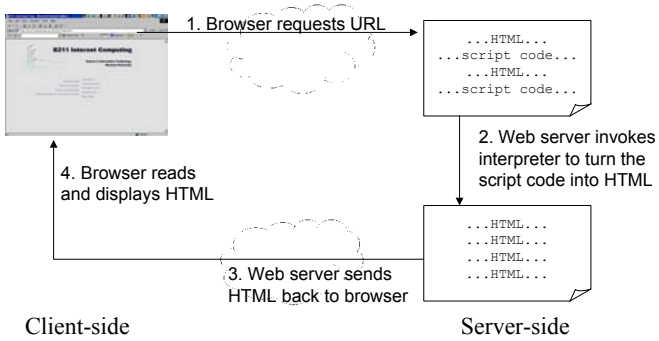


## Server-side Scripting

- Today, most of the powerful web applications uses server-side processing rather than client-side processing.
- Some common server-side technologies:
  - CGI with Perl
  - PHP (PHP Hypertext Preprocessor)
  - Microsoft's ASP (Active Server Pages)
  - JSP (Java Server Pages) and Java Servlets
  - Macromedia's (previously Allaire's) ColdFusion
  - Apache's XSSI (eXtended Server-Side Includes)



## How Server-side Scripting Works






## How Server-side Scripting Works

- All the technologies have slight variations to the above, but the basis is:
  - Processing of script code is done on the server-side, and
  - a document gets sent to the client that is static, or contains client-side code.



## Language and Technology

- Note that the list above are more than just scripting languages. They are technologies consisting of:
  - Methods of being accessed and executed.
  - Methods of communicating between scripts and web servers.
  - Objects to use in the languages.
  - Database handling
  - etc.

## Language and Technology

- Some technologies allows you to use multiple languages. Eg.
  - You can write CGI program in any language as long as the language has CGI support and the web server can run the programs. Most commonly people use Perl, Python, and even Java and C/C++.
  - ASP.NET can be written in any .NET language (currently about 40+ supported).



## Recognizing Which Server-side Technology

- Most server-side technologies makes use of standard file extensions for their script files. You can recognise them when looking at the URL in your web browser's URL box.

| Language | Standard file extensions |
|---|---|
| Perl/CGI | .cgi or .pl |
| PHP | .php |
| ASP | .asp |
| ColdFusion | .cfm or .cfml |
| JSP | .jsp |
| XSSI | .shtml |



## Choosing Server-side Technologies

- The choice in using which server-side technologies is not a straight-forward one.
- Most of the technologies support very powerful functionalities.



## Choosing Server-side Technologies

- The following are common reasons developers adopt the different options:
  - ASP
    - familiarity with Microsoft development platform.
    - take advantage of MS-Windows technologies (eg. COM objects and ActiveX).
    - Commercial support by Microsoft Developer Network.
  - Perl/CGI
    - familiarity with Perl, and advantages of Perl language (web communication is text processing)
    - Perl's rich development community
    - open-source and completely free
  - PHP
    - open-source and free
    - Tight integration with MySQL (an open-source, free database)



## Choosing Server-side Technologies

  - JSP
    - can use the full Java language, and it's libraries.
  - ColdFusion
    - Simple tag syntax (ColdFusion Markup Language CFML)
    - Powerful IDE (integrated development environment)
  - XSSI
    - Natural extension to the Apache web server
- Developers preferring each platform ALL claim their respective language is easiest to learn!



## Choosing Server-side Technologies

- The following are common reasons developers reject certain server-side scripting technologies:
  - ASP
    - restricted to being served through Microsoft's IIS web server (Note: there are now versions of ASP for other web servers, although not popular)
    - cost of development tools and web server.
  - CGI
    - large process and memory requirements
  - ColdFusion
    - MS-Windows platform.
    - Not built for low level programming - built to work best through using the visual IDE and CFML.
    - cost of package.

# Choosing Server-side Technologies

- JSP
  - Restricted to Java programming language
- XSSI
  - A very thin language, only appropriate for small processing.



# Choosing Server-side Technologies

- I make not absolute claims on which technology is best:
  - …even though I have my favourites for different purposes.
  - The previous 2 pages are only "common" reasons
  - If and when your programming-proficiency level is good enough, you should decide for yourself by having a preliminary look at the details.
  - Don't make your decisions on technologies based on what other people say.



# Choosing Server-side Technologies

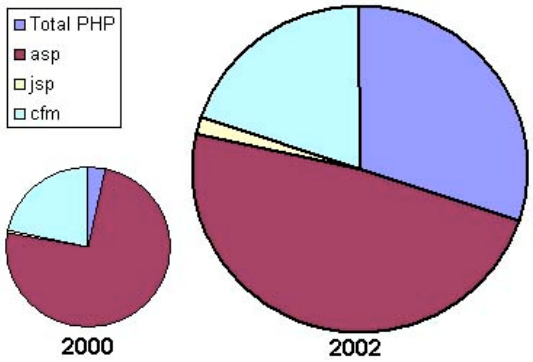
- In a lot of cases, your decision my be based on more than just which technology is technically better. Eg.
  - What is my programming team familiar with?
  - How much time do we have to learn a new technology?
  - What kind of documentation and support is available?
  - What software is available within the organisation? How much money is there to buy extra software?
  - Who is going to maintain the software once it is implemented and deployed?



# The Growth of Server-Side Scripting Technologies

- PHP and ASP are the two largest growing web-server scripting technologies around.
- References:
  - http://www.php.net/usage.php - using statistics on the number of domains and IP addresses that have PHP installed
  - http://php.weblogs.com/popularity - using search engine indices on the number of web pages with PHP terms in it






Source:http://php.weblogs.com/popularity (accessed 5/5/02)



| | October 12, 2000 | March 4, 2002 | Growth | Share |
|---|---|---|---|---|
| php | 84,296 | 4,409,034 | 5130% | |
| php3 | 49,906 | 2,308,381 | 4525% | |
| phtml | 23,268 | 831,815 | 3475% | |
| **Total PHP** | 157,470 | 7,549,230 | 4694% | 30% |
| **asp** | 3,166,710 | 11,958,185 | 278% | 48% |
| **jsp** | 24,435 | 413,827 | 1594% | 2% |
| **cfm** | 936,223 | 4,950,133 | 429% | 20% |

Source:http://php.weblogs.com/popularity (accessed 5/5/02)

## PHP

- PHP stands for "***PHP: Hypertext Preprocessor***"
- PHP is "*a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML*."



## Why Another Language?

- At the time it was developed (1994-1996) many web developers found that existing tools and languages were not ideal for the specific task of embedding code in markup.
- At the time, Microsoft's ASP (Active Server Pages) provides some of the facilities the developers were looking for, but:
  - Was a proprietary system geared towards VBScript and Microsoft IIS, and
  - Ports to other environments were very unstable and cumbersome.



## Why Another Language?

- It was created specifically for developing dynamic web-based sites and applications.
- PHP was not created as a general purpose programming language, although it can be used as such.



## PHP vs CGI

- At the time before PHP3 was released in 1997, the most popular method for creating dynamic web content using free software was using writing CGI scripts (especially using Perl).
- PHP code was developed for embedment within HTML. It gives
  - quicker response time,
  - improved security, and
  - better transparency to the end users.



## PHP as a Language

- PHP3 (and subsequently PHP4) makes use of Perl, Java, and C concepts.
  - Programmers in any of these languages should not have any problems picking up PHP.
- It also removes some of the less structured features of the languages, to enable developers to pick up PHP easily
  - Eg. obscure syntax and symbols in Perl.



## Active Server Pages (ASP)

- ASP is Microsoft's version of web server-side scripting technology.
- ASP pages generally have the extension ".asp".
- ASP code are embedded in HTML using <% … %> tags.
  - In a similar way approach to <? … ?> tags in PHP.

## ASP as Server-side Scripts

- ASP is only a technology, and is not a language.
- ASP scripts are actually written in a chosen language. The two languages supported in ASP is VBScript and JScript.
    - We can also use other languages like PerlScript and Python by installing scripting engines for those languages on the server.



## ASP in Web Servers

- ASP must be served through a Microsoft web server
    - IIS (Internet Information Server), or
    - PWS (Personal Web Server)



## From ASP to ASP.NET

- ASP.NET is the web server-side scripting tool for the Microsoft's .NET environment.
- Although the name "ASP" is retained, ASP.NET was actually build from grown-up, and therefore is fundamentally different from "classic" ASP.
    - ASP.NET was built to specifically overcome the weaknesses of ASP, and to integrate with the new .NET environment (such as support for Web Services).
    - However, ASP.NET has some of the superficial similarities to "classic" ASP, such using <% … %> tags.



## Co-existence of ASP and ASP.NET

- Developers do not have to choose between ASP and ASP.NET on one server.
    - ASP and ASP.NET pages can be installed and used on the same server.
    - ASP.NET pages uses the default extension ".aspx", to differentiate them from the ".asp" pages.



## Non-Microsoft ASP

- There are some non-Microsoft ports of the ASP architecture to other platforms.
- Eg:
    - ChiliASP
        - » Commercial ASP port to Apache, iPlanet and Zeus Web servers running on a variety of platforms.
    - Apache ASP
        - » Free ASP port to the Apache web server.
        - » Uses Perl as the language for scripts.
- See for other examples:
    - http://directory.google.com/Top/Computers/Programming/Internet/ASP/Tools/



## Further Reading

- A overview on various server-side scripting options:
    - http://hotwired.lycos.com/webmonkey/99/46/index1a.html
- PHP
    - http://www.php.net
- Microsoft's ASP.NET site:
    - http://www.asp.net/