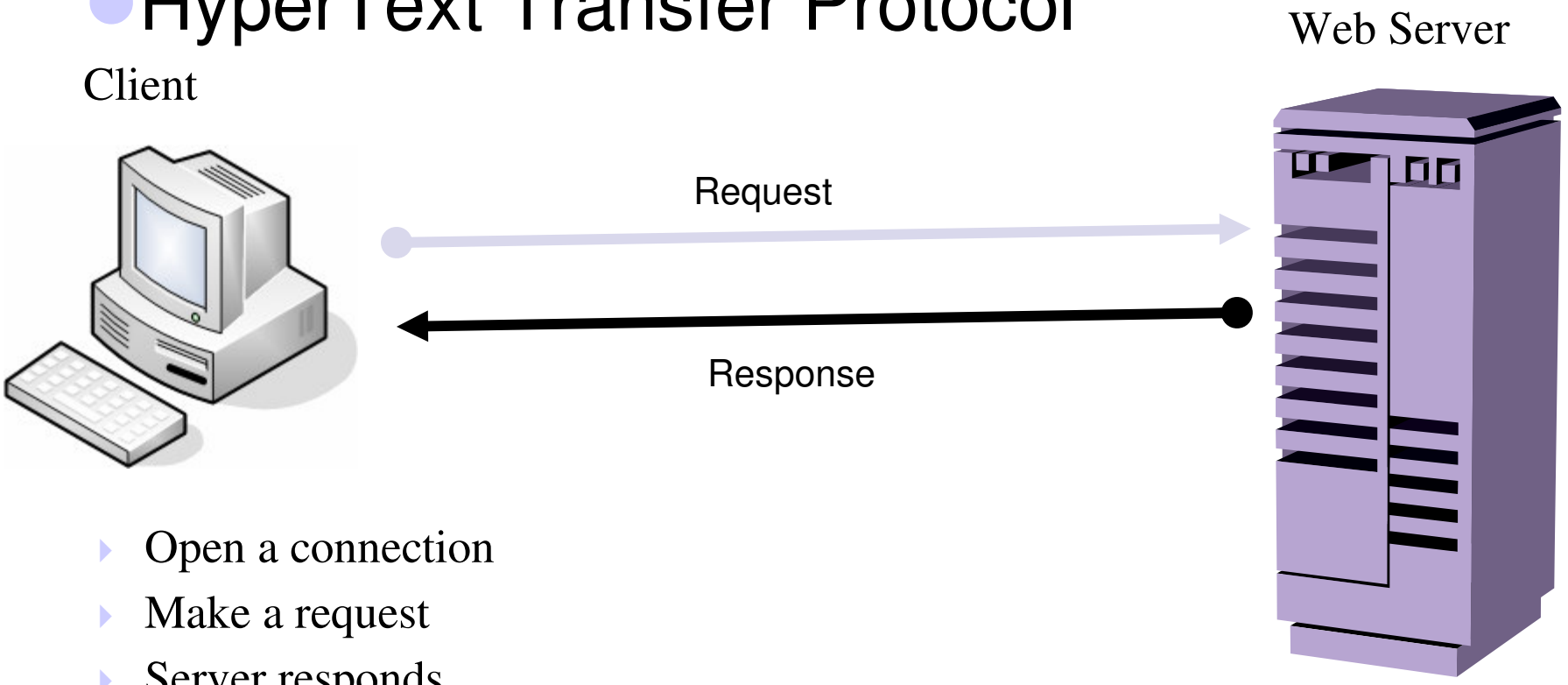
HTTP

- HyperText Transfer Protocol



- Open a connection
- Make a request
- Server responds
- Close connection

HTTP

- Stateless
  - If you view 10 web pages, your browser makes 10 independent HTTP request
  - Restart web server?
- Anonymous

Anatomy of a Typical HTTP Session (1)

- User types www.yahoo.com into browser
- Browser translates www.yahoo.com into an IP address and tries to open a TCP connection with port 80 of that address
- Browser sends the following byte stream:
  - Get / HTTP/1.0

Anatomy of a Typical HTTP Session (2)

- Yahoo responds with a set of headers indicating
  - Which protocol is actually being used
  - Whether or not the file requested was found
  - How many bytes are contained in that file
  - Kind of information (MIME: Multipurpose Internet Mail Extensions)
- Yahoo's server sends a blank line to indicate the end of the headers
- Yahoo sends the contents of its index root
- The TCP connection is closed

Anatomy of a Typical HTTP Session (3)

```
bash-2.03$ telnet www.yahoo.com 80
Trying 216.32.74.53...
Connected to www.yahoo.akadns.net.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Content-Length: 18385
Content-Type: text/html

<html><head><title>Yahoo!</title><base href=http://www.yahoo.com/>...
```

Stateless Connection

- When the connection is over, it is over
- Shopping at an e-commerce site (Amazon) ?
- Engineering Challenge: Creating a stateful application on top of a fundamentally stateless protocol

Where can you store state?

- Log file on the web server?
  - HTTP is anonymous
  - The server only knows IP address of client
  - Proxy?
- Rewriting hyperlinks
  - Sending extra information back to the server
  - http://www.amazon.com/exec/obidos/ASIN/1588750019
  - http://www.amazon.com/exec/obidos/ASIN/1588750019/103-9609966-7089404

  - All the hyperlinks contain, at the end, this same session ID.
  - HTTP does not place a priori limit on the length of a URI
    - 255 byte limit, error 414: request-URI Too Long

Cookies

- Write some information out to an individual user that will be returned on that user's next request
- Server side connections can use it to both store and retrieve information on the client side.
- Distributed database management system

Cookies: Problems

- Limit: 20 cookies, max 4 kb
- Cookie information will be passed back up to server on every page load.
  - Overhead: suppose 80 kb for 20 cookies + dialup connection
- They aren't portable for the user
- Security (privacy problem): some users have disabled them
  - Using unique identifier for the data rather than the data

Cookies: Example

```
bash-2.03$ telnet www.eveandersson.com 80
Trying 64.94.245.206...
Connected to www.eveandersson.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Set-Cookie: ad_browser_id=3291092; Path=/; Expires=Fri, 01-Jan-2010 01:00:00
GMT
Set-Cookie:   ad_session_id=3291093%2c0%2c6634C478EF46FC%2c10622158;   Path=/;
Max-Age=86400
Set-Cookie: last_visit=1071622158; path=/; expires=Fri, 01-Jan-2010 01:00:00
GMT
Content-Type: text/html; charset=iso-8859-1
MIME-Version: 1.0
Date: Thu, 03 Feb 2005 00:49:18 GMT
Server: AOLserver/3.3.1+ad13
Content-Length: 8289
Connection: close

<html>
  <head>
...
```

Server-Side Storage

- DBMS
- ACID test:
  - Atomicity: all committed or all rolled back
  - Consistency: DB is transformed from one valid state to another valid state.
  - Isolation: the result of a transaction are invisible to other transactions until the transaction is complete.
  - Durability: once committed, the result are permanent.

Why DBMS?

- Declarative query language (SQL)
- Isolation of important data from programmers mistakes
- Good performance with many thousands of simultaneous users
  - IBM DB2
  - Oracle
  - Microsoft SQL server
  - Open-source PostgreSQL

Building Internet Application

- Develop a data model
- Develop a collection of legal transactions: insert, update
- Design the page flow
  - How user interact with the system?
- Implement the individual pages
  - HTML
  - ASP (Active Server Page)
  - Java Server Page
  - Servlet
  - …

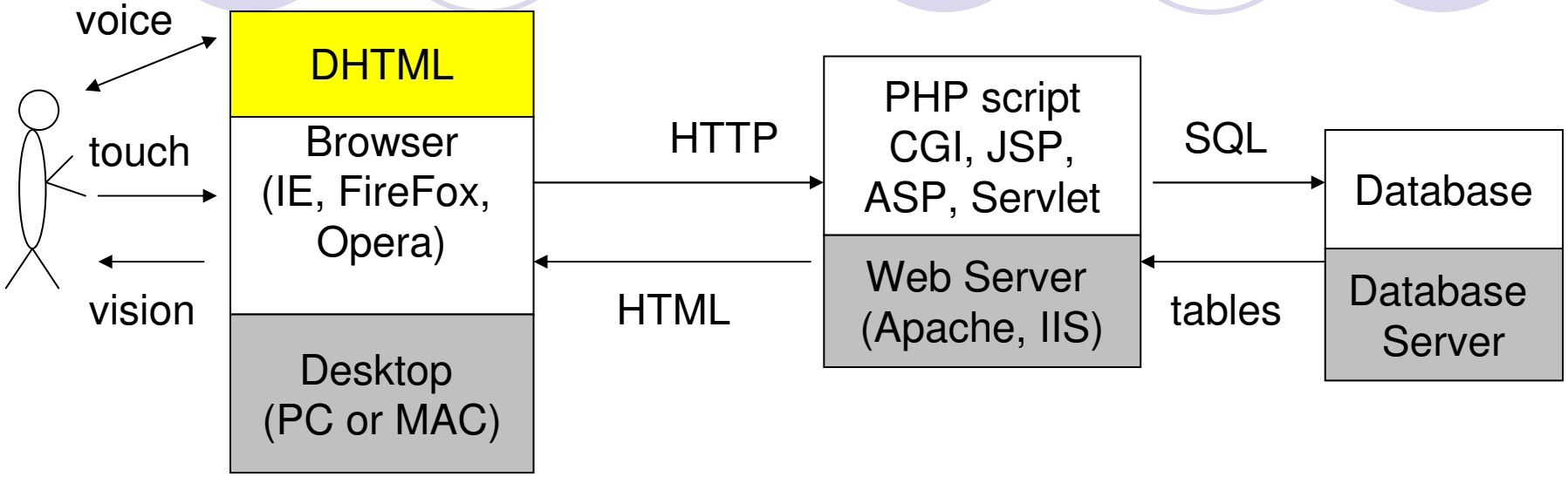
Introduction to HTML

- Hyper Text Markup Language
- An HTML document is just a text document with some special directives, called tags, that a web browser understands.
- Tags are those things in “angle-brackets”, like <HTML>, <HEAD>, etc.
- HTML has no variables or commands.
- HTML is merely a way of formatting a document.
- Intended to be platform- and device-independent

What is Hypertext?

- Text with links to other documents
- What's the big deal?
  - Links didn't exist until the 1960's and were novel well into the 1980's
  - Hypertext only existed on single computers or local area networks until about 1990

3(+1) Tier architecture



Presentation Layer

Application Layer

Persistence Layer

Web Server

- A piece of software
- Listens for HTTP requests
- Sends back HTTP responses
- Apache HTTP Server
- Internet Information Services (IIS)
- Serves up contents (html, images, txt…)
  - Static contents
  - Dynamic contents

Static Contents

- HTTP request comes in
- Sends existing html file back
- http://www.server.com/dir1/file1.html
  - <server root dir>/dir1/file1.html

Dynamic Contents

- HTTP request comes in
- Generates HTML page
- Sends generated HTML back
- http://forum.cs.umd.edu/forumdisplay.php?f=17

Comparing Static & Dynamic Contents

- Static Contents
  - Faster responses
  - Less CPU usage
- Dynamic Contents
  - Less file management
  - Easier to update contents

Web Applications

- A web application is an **application** delivered to users from a **web server** over a network such as the **Internet** or an **intranet**.

Advantages

- Only needs a web browser to use the application (Thin Client)
- Easy to distribute and update application

Three-Tiered Architecture

1. Web Browser
2. Dynamic Content Engine
3. Database

First Tier – Web Browser

- Sends Requests to middle tier
- ttp://www.amazon.com/index.jsp?item=5

- Displays HTML responses

Second Tier– Dynamic Content Engine

- Processes requests
  - http://ww.amazon.com/index.jsp?item5
  - “Runs” index.jsp with parameter item = 5
- Makes queries to the database
- Generates HTML with information from database
- Sends back response

Third Tier - Database

- Stores data
  - e.g. Amazon.com's database stores
    - Items for sale
    - Customer information

Dynamic Content Engines

- Java Server Pages (JSP) and Servlets
- Active Server Pages (ASP)
- PHP
- CGI

Technology Stacks

- L.A.M.P.
  - Linux Operating System
  - Apache HTTP Server
  - MySQL Database
  - PHP, Python or Perl Scripting Language
- J2EE
- .NET

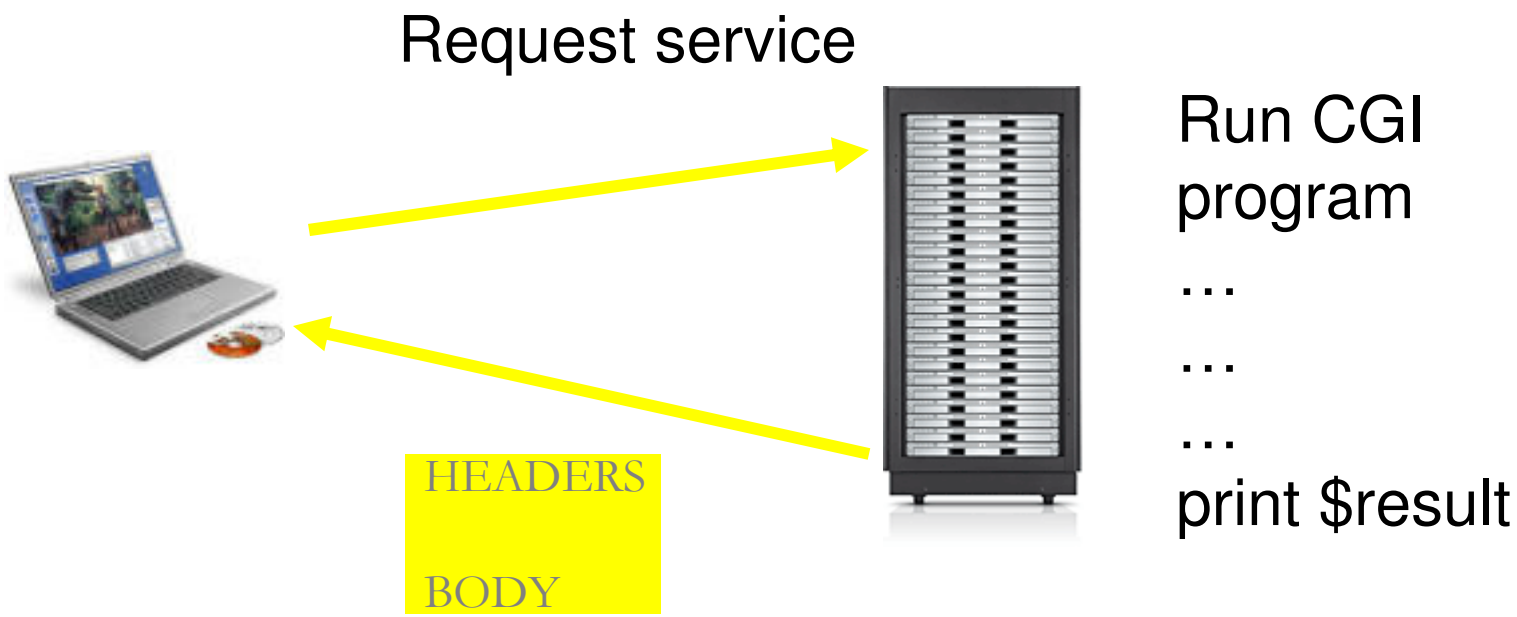
CGI

- Common Gateway Interface
- Invented in 1993 by NCSA for HTTP web server
  - Client requests program to be run on server-side
  - Web server passes parameters to program through UNIX shell environment variables
  - Program spawned as separate process via fork
  - Program's output => Results
  - Server passes back results (usually in form of HTML)

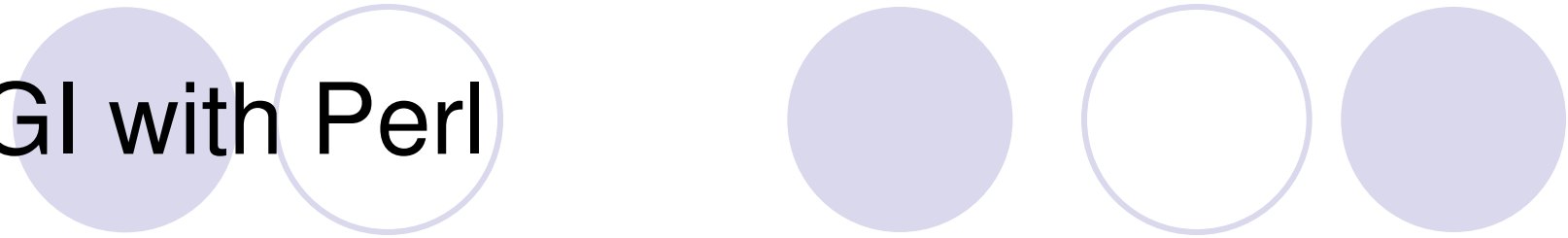
CGI

- Good for interfacing external applications with information servers
- In fact it is a standard that enables clients and servers to exchange data.
- it is language independent
- CGI programs are most often written in PERL, C/C++, VB, Java, or UNIX shell scripts.

CGI



CGI with Perl



- Write a standard Perl Program
- Program's output (to stdout) is sent back as HTTP Response
- You must write out everything
  - Headers
  - Blank Space
  - Body

Perl – a simple example

- "Hello World" in PERL

```
#! /usr/bin/perl
print "Content-type: text/html\n\n";
print "<html><body><h1>Hello World!";
print "</h1></body></html>\n";
```

- Simple concept -- the program executes, and the output is sent to the browser that called it.

Perl – a simple counter

```
#! /usr/bin/perl
open (INPUT,”count.txt”);
@inline= <INPUT>;
$count = $inline[0] + 1;
close INPUT;
open (OUT,”>count.txt”);
print OUT “$count\n”;
close OUT;
print "Content-type: text/html\n\n";
print "<html><body>”;
print “<h1>Let’s Count! "</h1>";
print “This page accessed $count times<p>”;
print “</body></html>\n";
```

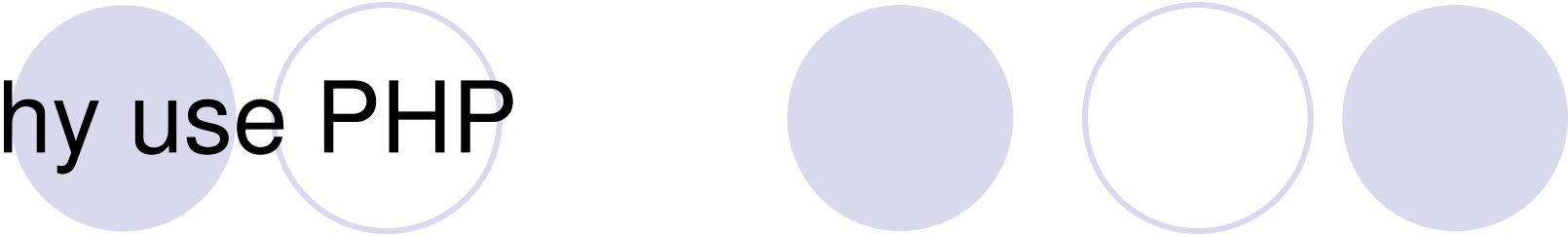
PHP overview

- Open Source server-side scripting language designed specifically for the web.
- In-line scripting
- Conceived in 1994, now used on +10 million web sites. Now in version 5.0
- Outputs not only HTML but can output XML, images (JPG), PDF files and even Flash movies all generated on the fly. Can write these files to the file system.

PHP overview

- Supports a wide-range of databases (inherently or via ODBC).
- Supports OO programming
- Perl- and C-like syntax. Relatively easy to learn.
- Website @ http://www.php.net/

Why use PHP



- **free** software
- **portable** across multiple platforms (e.g. Red Hat Linux to Windows 2000)
- To add dynamic content to your pages
- If you want to make your pages easier to maintain
- There are a lot of open source/free packages/libraries available in PHP.

What is in a PHP file

- PHP files may contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- PHP files have a file extension of ".php", ".php3", or “.phtml“
- Embedding PHP in HTML:

```
<html>
<body>
    <strong>Hello World!</strong><br />
<?
    echo ‘This is a PHP introductory course!’;
?>
</body>
</html>
```

Include mechanism

```
<?php
include '../includes/header.html';
?>
<center>
content of your web page
</center>
<?php
include 'http://cs.ucy.ac.cy/php/footer.html';
?>
```

- Content can be included from a local or remote source via such protocols as HTTP, HTTPS, FTP, and FTPS

HTML Forms

- When a form is submitted to a PHP script, the information from that form is automatically made available to the script
  - There's a few ways to do this
  - Example:

```
<form action="foo.php" method="POST">
Name: <input type="text" name="username"><br>
Email: <input type="text" name="email"><br>
<input type="submit" name="submit" value="Submit">
</form>
```

- <html><body>
- <?php
- **print $_POST['username'];**
- ?>
- </body></html>

HTTP methods

- GET: request a resource by URL
  - Get is idempotent
  - Querying information, not performing any actions on the back-end
- HEAD
  - is just like a GET request, except it asks the server to return the response headers only, and not the actual resource (i.e. no message body).
  - This is useful to check characteristics of a resource without actually downloading it, thus saving bandwidth.

HTTP methods (2)

- POST
  - A POST request is used to send data to the server to be processed in some way, like by a CGI script.
  - There's a block of data sent with the request, in the message body. There are usually extra headers to describe this message body, like **Content-Type:** and **Content-Length:**.
  - The *request URI* is not a resource to retrieve; it's usually a program to handle the data you're sending.
  - The HTTP response is normally program output, not a static file.
  - Using POST will result in a site that breaks the browser Back button.
  - Refresh = resubmit ?

HTTP methods: POST or GET?

- Searching users or content: GET
- Inserting a user or updating a profile :POST
- GET forms are limited in length (how much your browser can send in a URL field)
  - Use POST for complex queries
- POST forms can only be performed by having an HTML button (or by using JavaScript)
  - Use GET for other components
- when you POST data for an insert or update, have your script process the POST, then redirect to a thank-you-page.
  - Refresh = reloading thank-you-page

PHP and MySQL

- PHP and MySQL are a perfect companion
- Largely because they are both free and they have numerous capabilities
- PHP as of version 3 supports inherently MySQL i.e. specialized build-in functions handle the database interactions
- Same goes with ORACLE but not with Microsoft databases (Access, SQL Server)

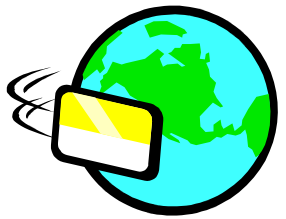
Servlet

- Servlet is Java program that runs as separate thread inside servlet container.
- Servlet container is part of web server
- It interact with web client using response request paradigm
- Runs in a container
  - Contains print statements that output an HTML page:

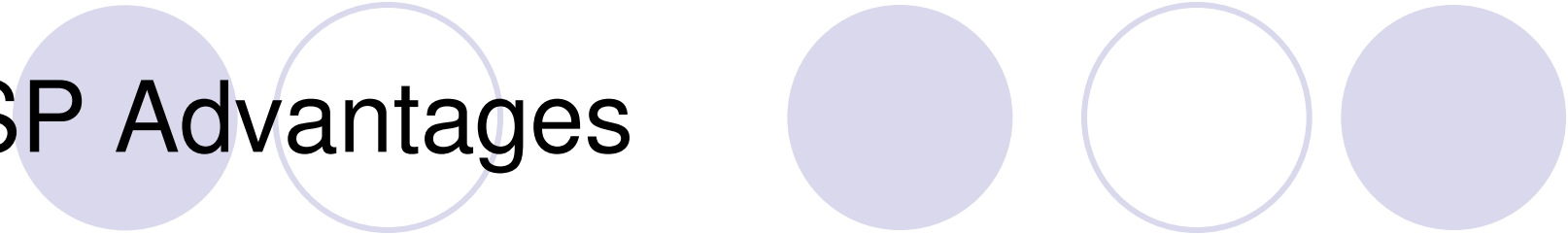
```
out.println("<html>")
```

JSP Application

- JavaServer Pages technology is an extension of servlet technology
  - From Sun Microsystems, as are servlets
  - JSPs can also output HTML
  - Also runs on the web tier server
- Contain some static HTML (e.g., **<BODY>**)
  - Contain some JSP tags and Java code that creates dynamic content
- When JSP is run, it creates a servlet
- JSPs are easier to develop than servlets
- Files have .jsp extension



JSP Advantages



- Performance
  - Runtime characteristics of servlets
  - Automatic recompilation of modified pages
  - Server side processing
- Programming
  - Emphasize use of reusable components
  - Extensible through custom tag libraries

Parts of JSP Pages

Directive

<%@ page import="java.util.", MVCApp.Cart, MVCApp.CartItem" %>

Declaration

<%! Iterator it = null; CartItem ci = null; Vector cpi = null;%>

Raw HTML

<html><head><title>Shopping Cart</title></head></html>

Action

<jsp:usebean id ="Cart" scope = "session" class = "MVCApp.Cart"/>

Scriplets

```
<%
Cpi = cart.getCartItems ( );
it = cpi.iterator();
While (it.hasNext()){ci= (Cart Item)it.next();
%>
```

Parts of JSP Pages

- Expression

<td<% = ci.getTitle() %></td>

<td align =“right”><%=ci.getQuantity()%></td>

- Implicit Objects

<% string action = request.getParameter(“action”) ; %>

Server Side Caching

- Reduces web server load
- Faster response time
- Saves recently or frequently accessed resources
  - file system
  - memory

Questions