

# **Internet Fundamentals**

# **Lecture-19**

**Servers**

**- Apache Tomcat Server**

**Server-side scripts**

**- Java Server Pages**

## Objectives

What is **Request/Response** protocol for a **client-side** script

What is **Request/Response** protocol for **server-side** script

How does server interact with **multiple concurrent clients**

How can html **meta-tags** control request/response interaction

How does effect depend on file **extension** - html versus jsp

## Topics

**I:** client-side, server-side, .jsp, JavaScript form screening, JSP results, multi-threaded server responses, Html meta-tag refresh, need for .jsp versus .html extension, JSP's conditional generation of Html, forms, buttons, onClick event, JavaScript function, reset, role of names on forms to identify form elements, table definition, textfields, alert pop-up window, configuring for Apache Tomcat server, Class & ClassPath system environment variables, <script & <% tags for JavaScript & JSP

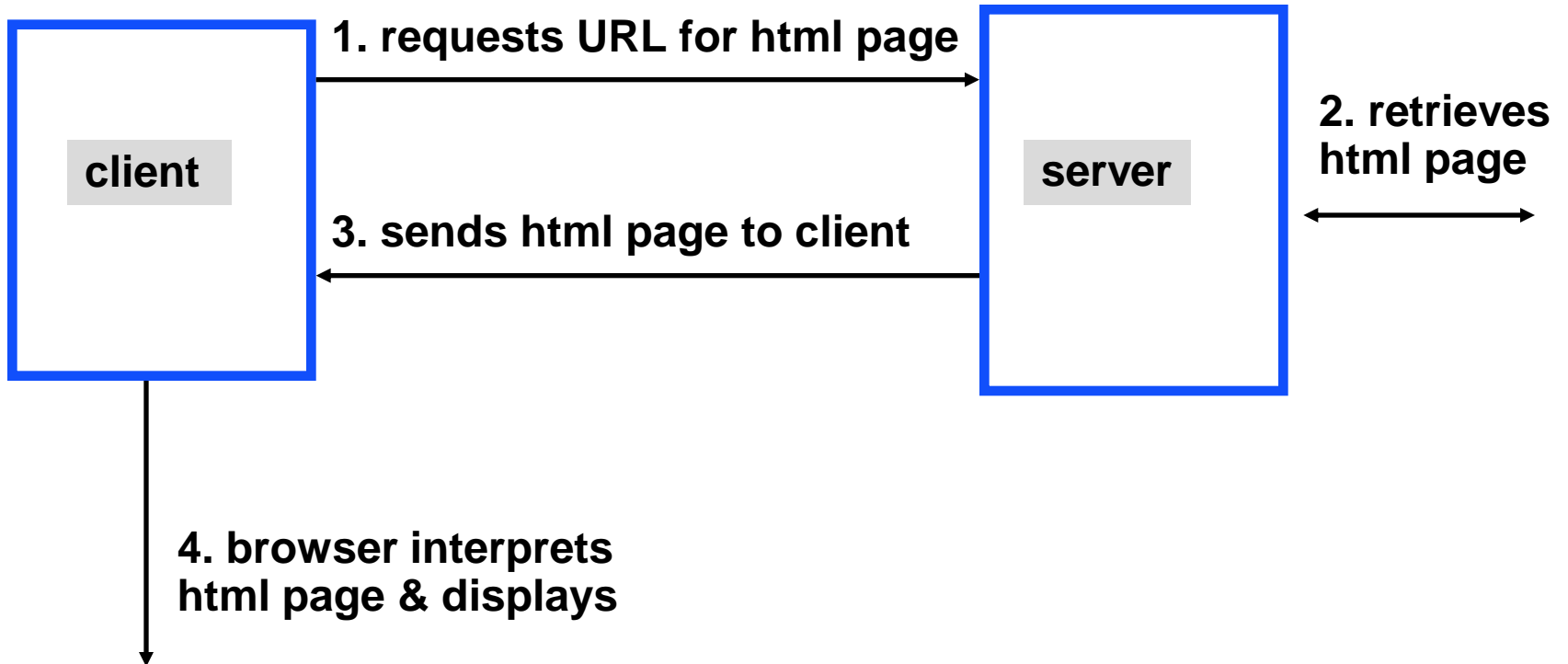
**II:** form - events - extend form, \myjspapp\chapter02\example14.html, qualified names for elements, submitting form data, request/response cycle[38], GET method: processing by server-side program, submit button, method identification, how element names are used, how JSP program accesses form element values, run: chapter03\example6a.html example - note query string sent to server - in address field of browser - after submit clicked - un-highlight beforehand so visible, focus attention, request.getParameter method, JSP Html dynamic output, dynamic html example: 12a

<http://localhost:8080/myapp/chapter03/example6a.html>

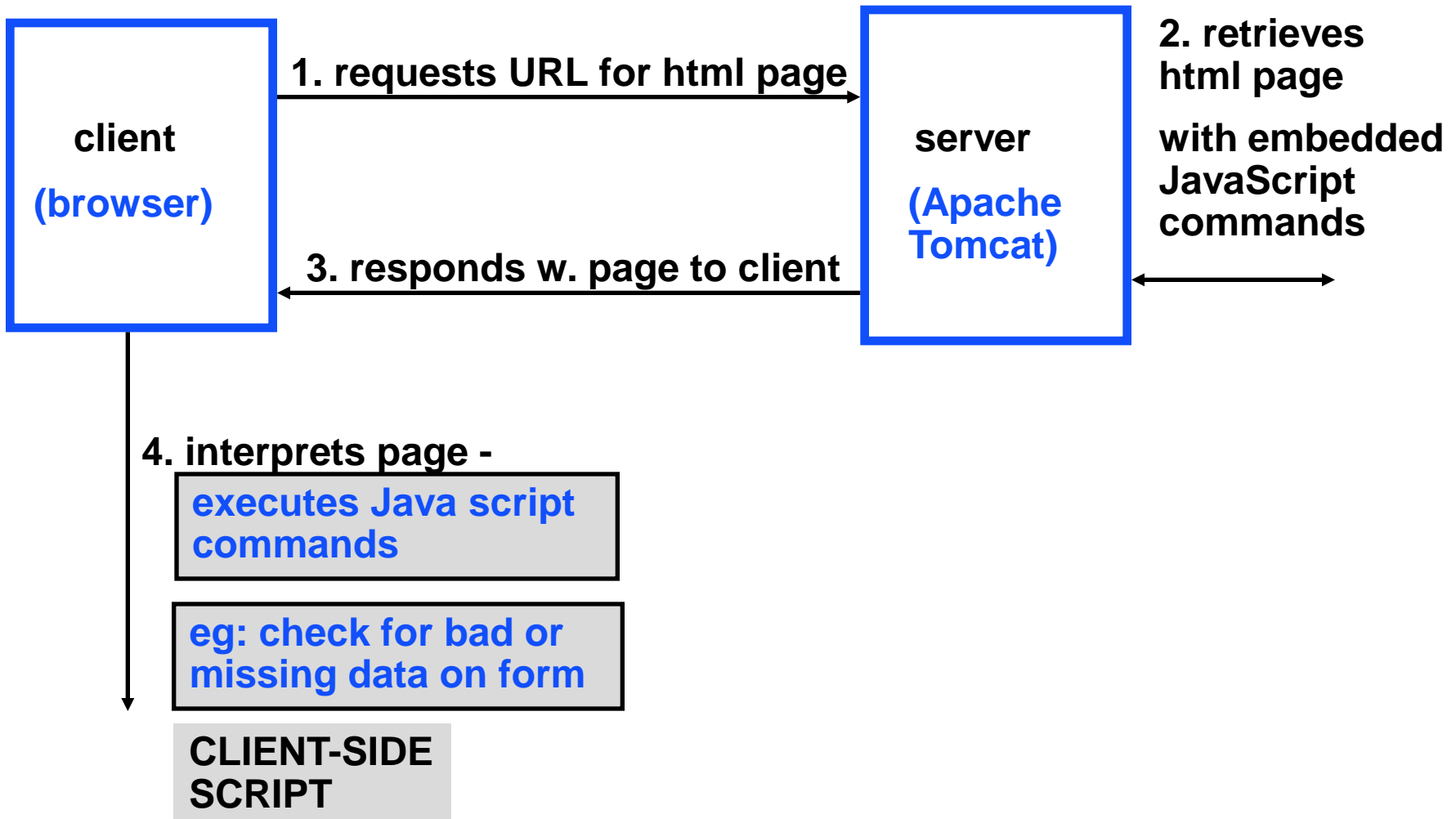
<http://localhost:8080/myapp/chapter03/example10a.jsp> - contrast Netscape & IE

<http://localhost:8080/myapp/chapter03/example12a.html> & 12b

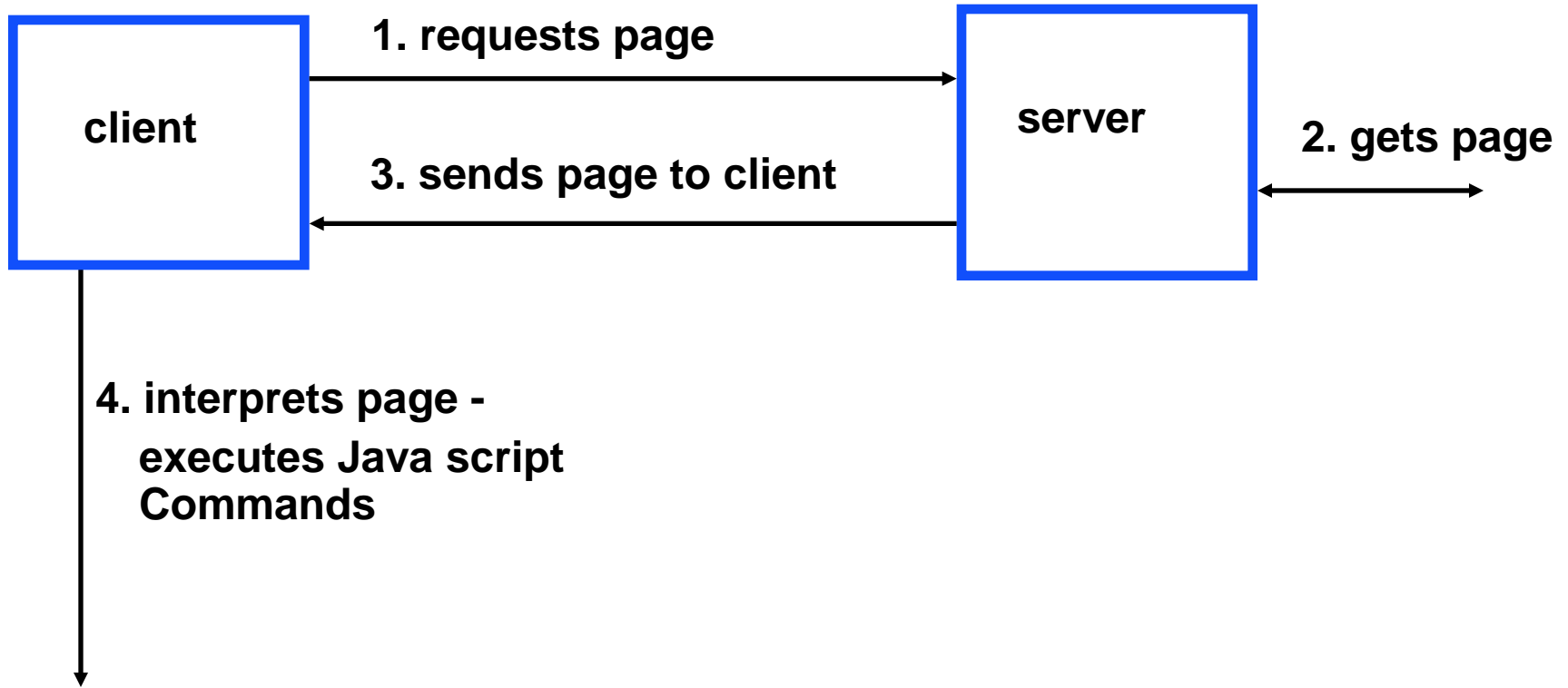
## Typical html **Request/Response** cycle



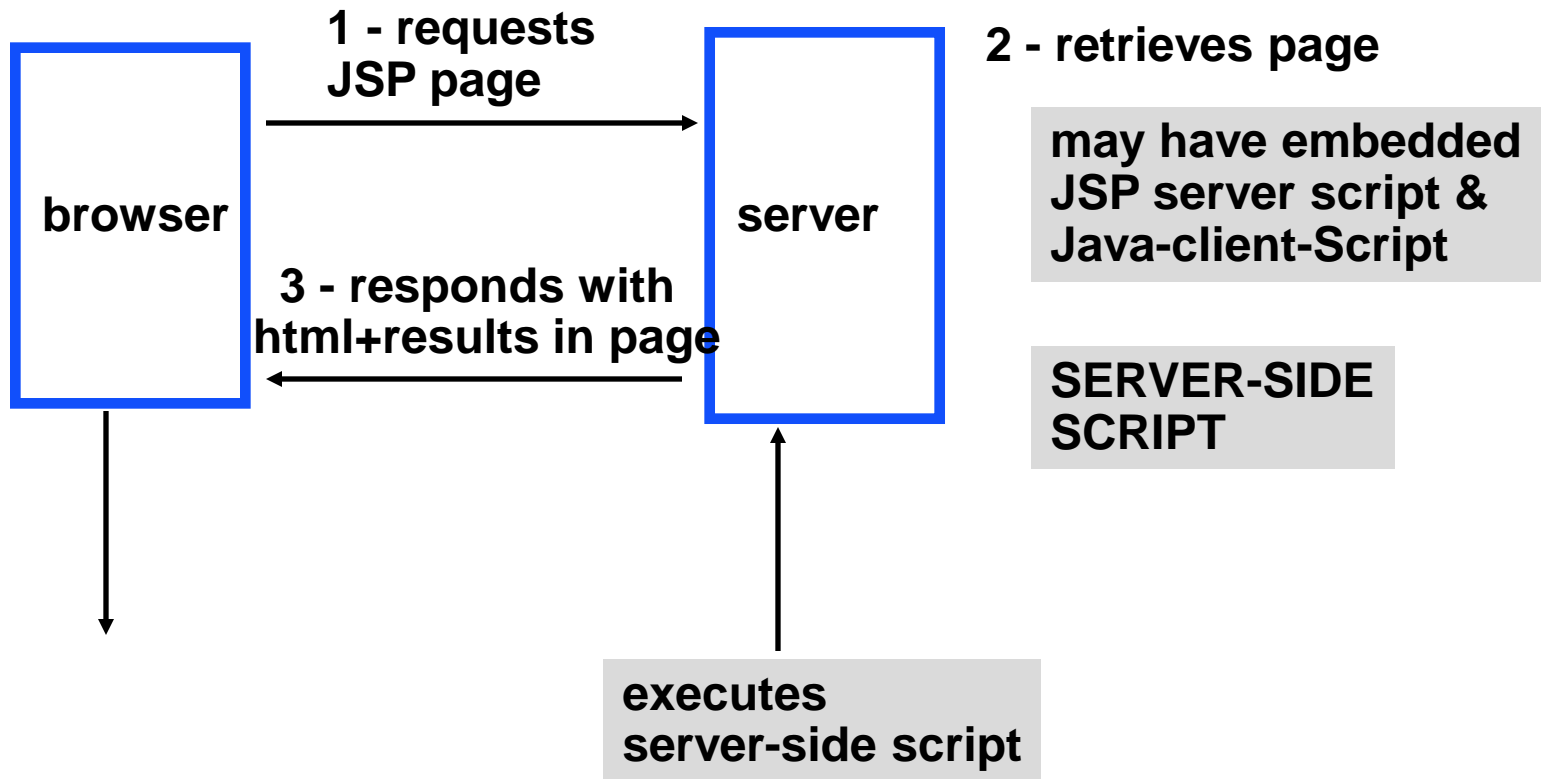
## Request/Response for page - JavaScript commands



## Request/Response for page - with JavaScript commands

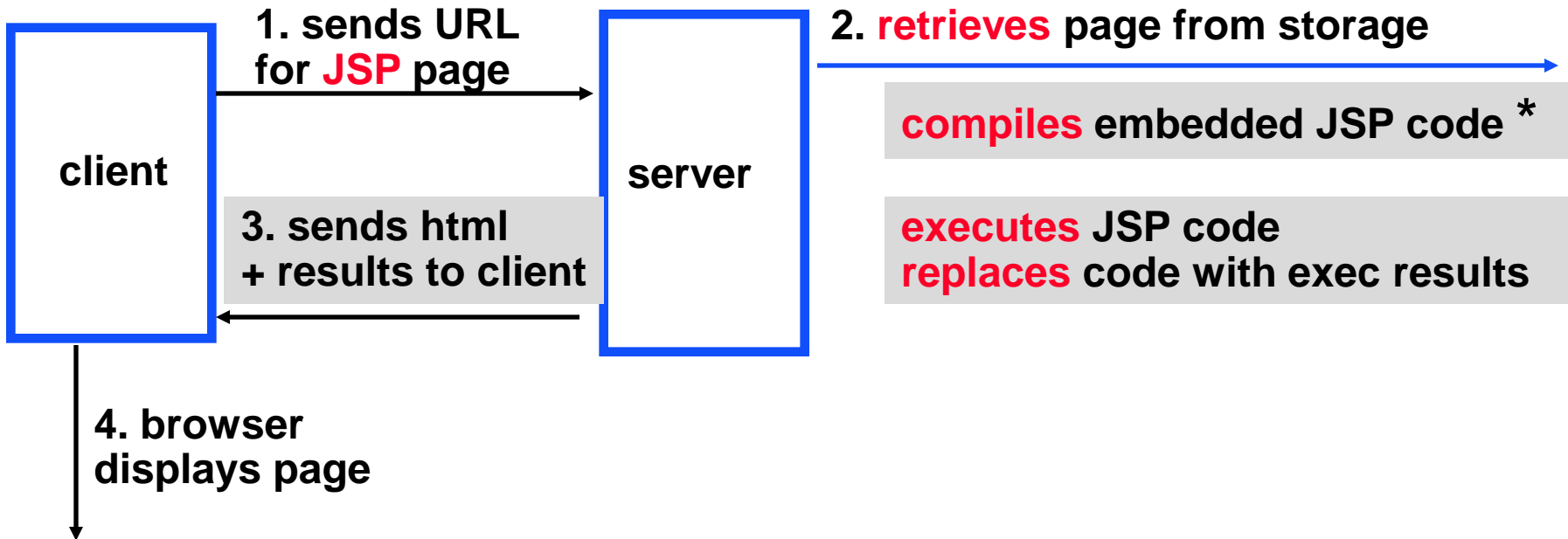


# Request/Response for file with **Java Server Page** parts

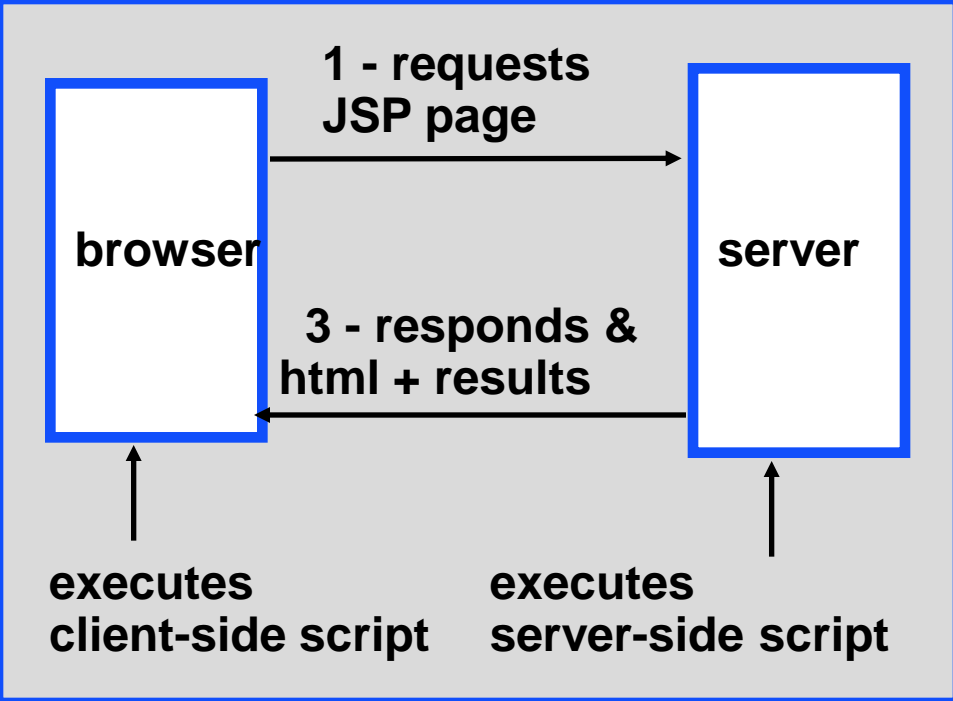




## Request/Response for Java Server Page



\* compiled first time only - thereafter uses compiled copy  
experiment on effect of extensions like .jsp or .html

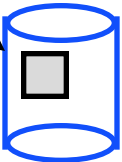


2 - retrieves page

using URL addr & server configuration

Context path = **"/myapp"**  
docBase = **"c:\myjspapp"**

<http://localhost:8080/myapp/chapter02/example2a.html>  
<http://localhost:8080/c:\myjspapp\chapter02\example2a.html>



**Experiment: JSP versus JavaScript - client v. server execution**

1. Start Apache Tomcat server  
listening on port 8080 localhost

2. Request a **Java Server page** from server  
- source file **will have Results**

3. Request **html page** with JavaScript  
- source page **will have the JavaScript**  
[at least in Explorer - not Netscape]

2. C:\myjspapp\chapter01\example2.jsp

3. C:\myjspapp\chapter02\example2a.html – prev slide

# Request for JSP

.jsp page  
retrieved is:

C:\myjspapp\chapter01\example2.jsp

```
<html>

<head>
<title> current server time </title>
</head>

<font face = "Arial" size =4>

The current Date and time on the web server are:
<BR>

<%= new java.util.Date() %>

</font>

</body>
</html>
```

jsp instruction

- executed on "server-side"
- result replaces code

embedded jsp instruction

The current Date and time on the web server are:  
Wed Nov 27 20:27:02 EST 2002

1. Request this page from server which is Listening on port 8080

2. Contents sent by server - **after executing jsp code** located in requested file

1. "source" as shown in browser

```
<html>

<head>
<title> current server time </title>
</head>

<font face = "Arial" size =4>

The current Date and time on the web server are:
<BR>

Wed Nov 27 20:27:02 EST 2002

</font>

</body>
</html>
```

2. Note how Date's **Result** replaces original JSP in page sent to browser

## Request for Java Script page

requested source page is same as displayed in browser

```
<HTML>
<HEAD>
<TITLE>Client-side script </TITLE></HEAD>
<BODY>
THE TIME ON THE CLIENT IS:

Current time is:
<%= new java.util.Date( ) %>

<script language="JavaScript" >
  document.write (new Date() )
</script>

</BODY>
</HTML>
```

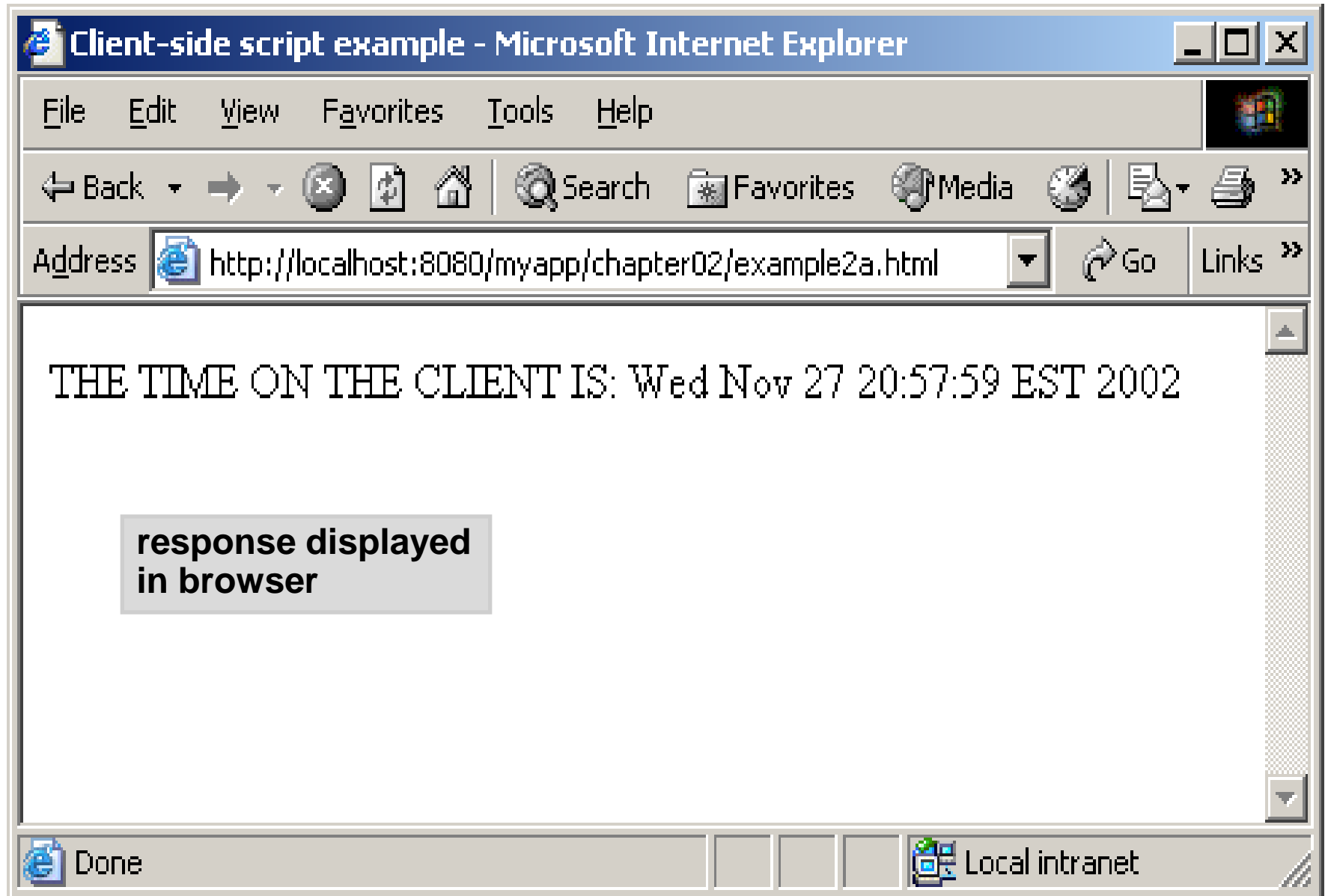
Why not executed on server ?

sent to browser and executed on browser

C:\myjspapp\chapter02\example2a.html

Request from server - versus drag-and-drop in browser - observe address bar

## JavaScript Request



**Source shows only JavaScript code - not the computed date**



How can a server interface with multiple client browsers simultaneously ?

**Ans:** creates multiple threads - one per browser.

**Experiment:** JSP with refresh meta-tag from multiple browsers

**Use URL:** <http://localhost:8080/myapp/chapter02/example12.jsp>

Source contains **Html Meta tag:**

```
<META HTTP-EQUIV = "REFRESH" CONTENT = "5, URL=example12.jsp">
```

and **JSP instruction:**

```
<%= new java.util.Date() %>
```

JSP page requested is:

requested  
every 5 sec

from current  
directory of original  
request

```
<HTML>
<HEAD>
<TITLE> server-side scripts </TITLE>
<META HTTP-EQUIV = "REFRESH" CONTENT = "5, URL=example12.jsp">
</HEAD>
<BODY>
The time on the server is:
<%= new java.util.Date( ) %>
</BODY>
</HTML>
```

JSP result replaces this code  
& is sent to browser

URL: <http://localhost:8080/myapp/chapter02/example12.jsp>

Requested File's address on the server

"http://localhost: 8080 /myapp/ helloWorld.js"

WHERE

WHICH

WHAT

IP address of server

port that server listens on

server uses configuration file to convert this to initial part of path:

remainder of file address path

C:\myjspapp


**What happens if you request an html page that has embedded JSP ?**

-- effect of EXTension on how server handles request:

Effect of suffix:

http://localhost:8080/myapp/chapter01/project3.html versus  
http://localhost:8080/myapp/chapter01/project3.jsp

Source html contains Java Server Page instructions - java inside  
jsp tags: `<% ... %>`



```
<% java.util.Calendar calendar = java.util.Calendar.getInstance( );  
    int hour = calendar.get(calendar.HOUR_OF_DAY);  
    if ( hour<12) { %>  
        <h3>Good morning!</h3>  
    <% } else { %>  
        <h3>Good afternoon!</h3>  
    <% } %>
```

...it would be better to just use print statement – this looks confusing.

```
<HTML>
<HEAD><TITLE>Welcome to JSP</TITLE></HEAD>
<BODY>

<% java.util.Calendar calendar = java.util.Calendar.getInstance();
    int hour = calendar.get(calendar.HOUR_OF_DAY);
    if ( hour < 12) {
%>

    Hour is <br>

    <%= hour %>

    <h3>Good morning!</h3>

    <%}else {%>

    <h3>Good afternoon!</h3>
    <%}%>

</BODY>
</HTML>
```

HTML – green  
JSP – red / blue

JSP starts/stops with <% ... %>

Html can occur anywhere else

Page displayed for  
project3.jsp:

time-dependent jsp output



**Good afternoon!**

**Welcome to Introduction to JSP**

**In this chapter, you:**

**learned about Web client/server architecture**

**learned the difference between static and dynamic Web pages**

**learned how dynamic Web pages are generated in JSP**

**reviewed various server-side processing technologies**

**compared JSP to alternate**

Displayed page  
For project3 .html

IE browser - no result (and no JSP shown)



**Good morning!**  
**Good afternoon!**

**Welcome to Introduction to JSP**  
**In this chapter, you:**

**learned about Web client/server architecture**  
**learned the difference between static and dynamic Web pages**  
**learned how dynamic Web pages are generated in JSP**  
**reviewed various server-side processing technologies**  
**compared JSP to alternate technologies**



Displayed page  
project3 .html

Netscape - no result (JSP shown)

```
<% java.util.Calendar calendar = java.util.Calendar.getInstance(); int hour =  
calendar.get(calendar.HOUR_OF_DAY); if( hour<12){%>
```

**Good morning!**

```
<%} else {%>
```

**Good afternoon!**

```
<%}%>
```

**Welcome to Introduction to JSP**

**In this chapter, you:**

**learned about Web client/server architecture**

**learned the difference between static and dynamic Web pages**

**learned how dynamic Web pages are generated in JSP**

**reviewed various server-side processing technologies**

**compared JSP to alternate technologies**

## **Java Server Pages - II**

## Objectives - Understand

How do you **submit** form data to server using **GET** method?

How does **request/response protocol** operate for forms & jsp pages

What is the **URL** format

How do JSP programs **get data** from **forms**

How to **configure** a (Tomcat) server

How do you **dynamically** construct Html using Java Server Pages

How do you **submit** form data to server using **GET** method?

C:\myjspapp\chapter03\example6a.html

C:\myjspapp\chapter03\getUserInfo.jsp

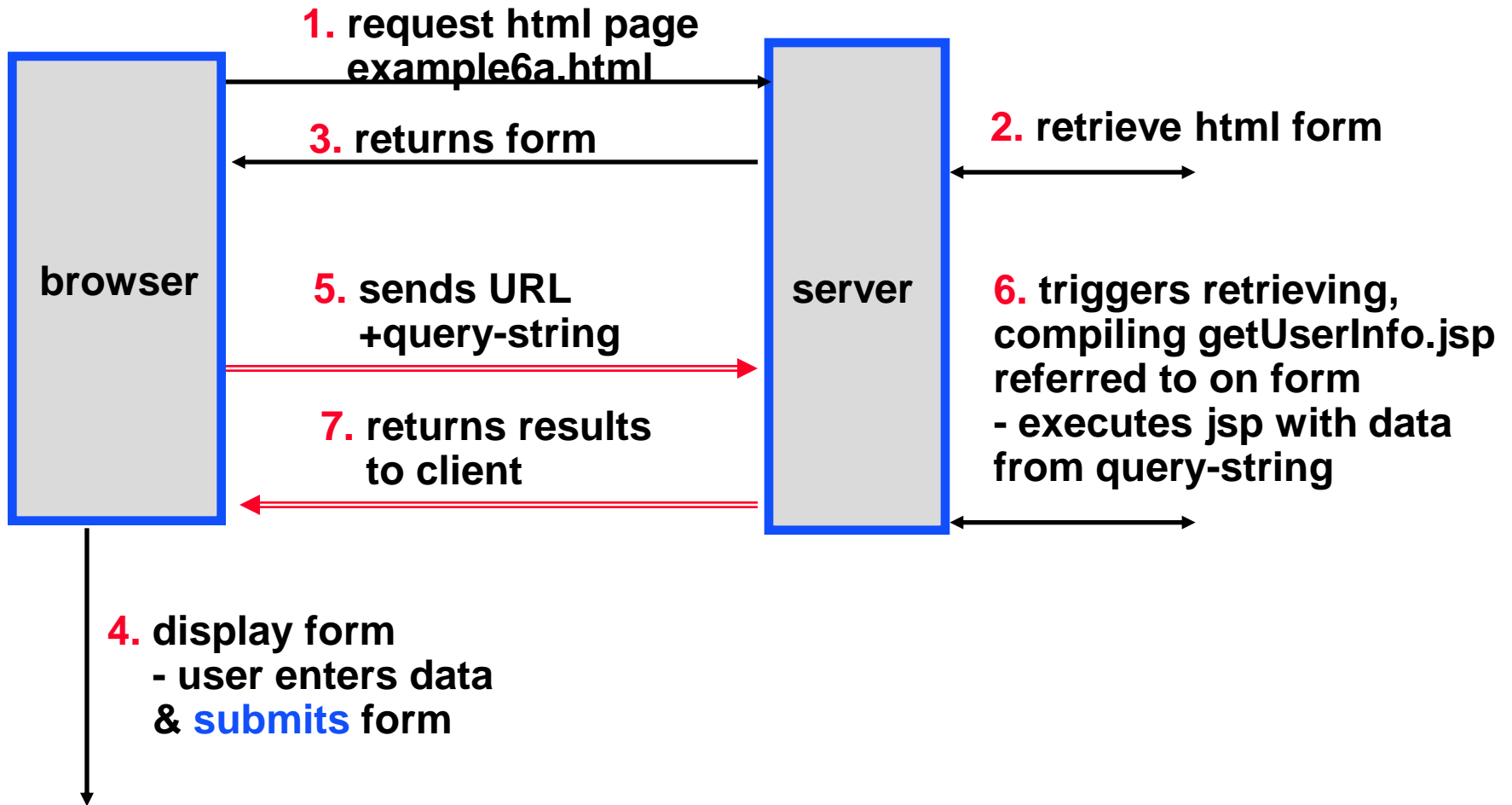
**Experiment:** demonstrates submitting and retrieving form data using GET method

1. chapter03/example6a.html sends data via **GET**
2. chapter03/getUserInfo.jsp uses **request** object

**request.getParameter ( ... )**

to retrieve named data from form

# Request/Response protocol



understand thoroughly

jsp program executed  
on Submit

GET sends data  
in query-string

example6a.html

```
<HTML>
<HEAD> <TITLE> Submit using GET Method </TITLE> </HEAD>
<BODY> <H1> Please sign </H1>

<form name = "formName" ACTION = "getUserInfo.jsp" Method = "GET" >

Your Name: <input type=text name= firstName size=24 > <br>
Your Major: <input type=text name= major size=24 > <br>

<input type=submit value="submit" >
<input type=reset value="Clear Form" >

</form>

THE TIME ON THE CLIENT IS:
<script language="JavaScript" > document.write (new Date( ) ) </script>
</BODY>
</HTML>
```

triggers submission of query string to server

data  
names

**Query string**  
sent by browser  
to server:

jsp to execute



data for program



<http://localhost:8080/myapp/chapter03/getUserInfo.jsp?firstName=aaa&major=zzz>

**SERVER**

**WHERE**

**PROGRAM**

**WHO**

**DATA FOR PROGRAM**

**WHAT**



retrieves values entered on form  
which are returned in query string

## getUserInfo.jsp

```
<HTML>
<HEAD> <TITLE>  get User information  </TITLE> </HEAD>
<BODY>
<H1> Hi there - how are you?  </H1>

Your major is: <%= request.getParameter("major")    %> <br>
Your name is:  <%= request.getParameter("firstName") %> <br>

<br>

The current Date and time on the web server are: <BR>
<%= new java.util.Date() %>
<br>

</BODY>
</HTML>
```

field names on html form  
that triggered execution of  
getUserInfo.jsp

## getUserInfo.jsp

```
<HTML>
<HEAD> <TITLE> get User information </TITLE> </HEAD>
<BODY> <H1> Hi there - how are you? </H1>

Your major is: <%= request.getParameter("major") %> <br>
Your name is: <%= request.getParameter("firstName") %> <br>

<br>
The current Date and time on the web server are: <BR>
<%= new java.util.Date() %>
<br>
</BODY>
</HTML>
```

<http://localhost:8080/myapp/chapter03/getUserInfo.jsp> ? [firstName=aaa&major=zzz](#)

DATA FOR PROGRAM

## Interpretation of URL

<u>Server IP</u>	<u>port</u>	<u>Requested file on server</u>
http://localhost:8080/ <b>myapp</b> /chapter01/filename		

↓  
replaced with  
**C:/myjspapp**  
because of  
server configuration

so file retrieved is:

**C:/myjspapp\chapter01\filename**

The context prefix *myapp* makes the web site Relocateable.  
You just change *myapp* to point to the directory where site pages begin.

How do you **configure the Tomcat server** ?

## Environment setup

1. control panel > system > advanced  
> environment variables > **system variables**

update **PATH** variable to include JDK

```
%SystemRoot%\system32;%SystemRoot%;  
%SystemRoot%\System32\Wbem;  
C:\jdk1.2.2\bin ← points to JDK bin
```

...on one line

2. similarly **add CLASSPATH** system variable defined as :

```
C:\jdk1.2.2\jre\lib\rt.jar;  
.;  
C:\Program Files\Apache Tomcat 4.0\common\lib\servlet.jar;  
C:\myjspapp\WEB-INF\CLASSES
```

...all on a single line

Assumes jdk 1.2.2 -- same idea for other versions of jdk.

## Environment setup

1. start > all programs > Apache Tomcat 4.0 > **Edit Configuration file**

right after:

```
<!-- Tomcat Root Context -->  
<!--  
<Context path="" docBase="ROOT" debug="0"/>  
-->
```

add:

```
<Context path="/myapp"  
docBase="c:\myjspapp"  
debug="0"  
reloadable="true" />
```

2. gets expanded  
to *this*

1. *this*

<http://localhost:8080/myapp/chapter01/filename>

## Environment setup

1. start > all programs > Apache Tomcat 4.0 > Edit Configuration file

add:

```
<Context path="/myapp"  
docBase="c:\myjspapp"  
debug="0"  
reloadable="true" />
```

If user wants: [C:\myjspapp\chapter03\example6.html](http://localhost:8080/myapp/chapter03/example6.html)

then asks for: <http://localhost:8080/myapp/chapter03/example6.html>

## What are the advantages of using a client-side versus server-side script ?

Client-side script can prevent bad data being sent to server which would waste **user's time, server's time, and waste communication resources**

Server-side script allows **protected** server-side access to data on the server side.



## Different ways JSP can output HTML:

```
<HTML>
<HEAD> <TITLE> jsp output </TITLE> </HEAD>
<BODY>
```

```
<%= "<font size =6 color = red> Hello </font>" %>
```

1. as quoted html

```
<br><br>
```

```
<font size =4 color = blue> How are you? </font>
```

2. as direct html

```
<br><br>
```

```
<%
```

```
out.println ( "<font size = 8 color = green> Goodbye </font>" );
```

3. in print statement

```
%>
```

```
</BODY>
```

```
</HTML>
```

<http://localhost:8080/myapp/chapter03/example10a.jsp>  
example10a.jsp

## **Form**

**=> name & font fields**

**=> JSP program**

**=> returns dynamic html**

**chapter03/example12a.html**

**http://localhost:8080/myapp/chapter03/example12a.html**

**http://localhost:8080/myapp/chapter03/getFontEffect.jsp?Name=Mimi&font=4**

```
<HTML>
<HEAD> <TITLE> font effect </TITLE> </HEAD>
<BODY>
<H1> Please sign </H1>

<form name = "formName" ACTION = "getFontEffect.jsp" Method ="GET" >

Font size:  <input type=text name= font  size=24> <br>
Type name:  <input type=text name= Name    size=24> <br>

<input type=submit value="submit"    >
<input type=reset  value="Clear Form" >

</form>

</BODY>
</HTML>
```

## JSP page triggered by form:

chapter03/getFontEffect.jsp

```
<HTML>
<HEAD> <TITLE> get User information </TITLE> </HEAD>
<BODY> <H1> Hi there - how are you? </H1>

<%
out.print ("<font size = ");
out.print (request.getParameter("font"));

out.print (" color = blue>");

out.print (request.getParameter("Name"));

out.print ("</font>");
%>

</BODY>
</HTML>
```

This builds font statement:

```
<font size = font.value color = blue> Name.value </font>
```

## Variation [uses Name as color choice]

```
<HTML>
<HEAD> <TITLE> get User information </TITLE> </HEAD>
<BODY> <H1> Hi there - how are you? </H1>
```

used as color  
[blue, red...]

```
<%
```

```
out.print("<font size = ");
out.print( request.getParameter("font") );

out.print(" color = ");
out.print( request.getParameter("colorName") );
out.print(">");
```

make the  
font tag prefix

```
out.print( request.getParameter("Name") );
```

font text

```
out.print("</font>");
```

font tag suffix

```
%>
```

```
</BODY>
</HTML>
```

used as text  
[double use]

<http://localhost:8080/myapp/chapter03/example12a1.html>

[chapter03/getFontEffect2.jsp](http://localhost:8080/myapp/chapter03/getFontEffect2.jsp)

## JSP control structure and Html

**Experiment: Retrieve font values from an HTML form & return name in font depending on hour of day**

1. **chapter03/example12b.html => returns form**
2. **chapter03/ifThenElseHtml.jsp => returns time-sized name**

requested JSP program

```
<HTML>
<HEAD> <TITLE> font effect </TITLE> </HEAD>
<BODY>
<H1> Please sign </H1>

<form name = "formName" ACTION = "ifThenElseHtml.jsp" Method = "GET" >

Your Name:  <input type=text name= Name    size=24> <br>
Select font:  <input type=text name= font    size=24> <br>

<input type=submit value="submit"    >
<input type=reset  value="Clear Form" >

</form>

</BODY>
</HTML>
```



# Please sign

Your Name:

Select font:

submit

Clear Form



```
<HTML>
<HEAD> <TITLE> time-based display </TITLE> </HEAD>
<BODY> <H1> Hi there - how are you? </H1>
```

```
<%
java.util.Calendar date = java.util.Calendar.getInstance();
int hour = date.get(date.HOUR_OF_DAY);
if (hour <= 15)
{
out.print ("<font size = ");
out.print (request.getParameter("font"));
out.print (" color = blue>");
out.print (request.getParameter("Name"));
out.print ("</font>");
}else
{
out.print ("<font size = 10");
out.print (" color = red>");
out.print (request.getParameter("Name"));
out.print ("</font>");
}
%>
<BR><BR>
<a href = "example12b.html" > Link Illustration </a>
</BODY>
```

```
<HTML>
<HEAD> <TITLE> time-based display </TITLE> </HEAD>
<BODY> <H1> Hi there - how are you? </H1>
```

```
<%
java.util.Calendar date = java.util.Calendar.getInstance();
int hour = date.get(date.HOUR_OF_DAY);
if (hour <= 15)
{
out.print ("<font size = ");
out.print (request.getParameter("font"));
out.print (" color = blue>");
out.print (request.getParameter("Name"));
out.print ("</font>");
}
else
{
out.print ("<font size = 10");
out.print (" color = red>");
out.print (request.getParameter("Name"));
out.print ("</font>");
}
%> <BR><BR>
<a href = "example12b.html" > Link Illustration </a>
</BODY>
```