Overview

- Web
- HTML review
- Introduction of JSP and Servlet
HTTP is a very simple protocol of communication, the protocol provides a schema to perform the communication, which is “Request” and “Response”.

Every connection between C/S includes 4 steps:
1. connection
2. request
3. response
4. close
The Common Gateway Interface, or CGI, is a standard for external gateway programs to interface with information servers such as HTTP servers.

CGI may be a piece of program which is written by C/C++, Fortran, Perl, TCL etc. It is executable, can handle and response to the request from the client.
1. The User Agent issues a request to access the CGI program.

2. The Web Server determines how to process the request by searching for either:
   > The cgil-bin directory directive in the obj.conf file.
   > The .cgi MIME type definition in the mime.types file.

3. An external process executes the CGI program.

4. The CGI program runs and writes the output as HTML or other document type.

5. The server sends the output from the CGI program to the User Agent.
HTML page call the CGI program by certain methods. The two common methods are:

1. GET: CGI may obtain data from the environment variable QUERY_STRING which contain the data was sent by client.

When the client only expect to obtain data from server but not change anything on the server, the GET method is preferred.

```html
<form name="getform" action="http://www.name.com/cgi/your.cgi" method="GET">
```


The request from client

- 2. POST: Web server delivers data to CGI through STDIN.
- When the client wants to change the data on the server or the size of data exceeds 1024 bytes, the POST method is considered.

- `<form name="postform" action="http://www.name.com/cgi/your.cgi" method="POST">`
Review of HTML

- **HTML**, which stands for **Hyper Text Markup Language**, is the predominant **markup language** for **web pages**. It provides a means to create **structured documents** by denoting structural **semantics** for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows **images and objects** to be embedded and can be used to create **interactive forms**.
The form contains 5 tags:

- `<form></form>`
- `<input>`
- `<select></select>`
- `<option>`
- `<textarea></textarea>`

Notice: the last four tags only can be used in `<form></form>` tag.
HTML FORM

The Java 2 Enterprise Edition (J2EE) has taken the once-chaotic task of building an Internet presence and transformed it to the point where developers can use Java to efficiently create multitier, server-side applications.

In late 1999, Sun Microsystems added a new element to the collection of Enterprise Java tools: JavaServer Pages (JSP). JavaServer Pages are built on top of Java servlets and designed to increase the efficiency in which programmers, and even nonprogrammers, can create web content.
JavaServer Pages is a technology for developing web pages that include dynamic content. Unlike a plain HTML page, which contains static content that always remains the same, a JSP page can change its content based on any number of variable items, including the identity of the user, the user's browser type, information provided by the user, and selections made by the user.
When a user asks for a JSP page, the server executes the JSP elements, merges the results with the static parts of the page, and sends the dynamically composed page back to the browser, as illustrated in figure 1–1.
Figure 1-1. Generating dynamic content with JSP elements.
Embedding Dynamic Elements in HTML Pages

- Instead of embedding HTML in programming code, JSP lets you embed special active elements into HTML pages. These elements look similar to HTML elements, but behind the scenes they are actually componentized Java programs that the server executes when a user requests the page. Here's a simple JSP page that illustrates this:
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
  <body bgcolor="white">

  <jsp:useBean id="clock" class="java.util.Date" />
  <c:choose>
    <c:when test="${clock.hours < 12}">
      <h1>Good morning!</h1>
    </c:when>
    <c:when test="${clock.hours < 18}">
      <h1>Good day!</h1>
    </c:when>
    <c:otherwise>
      <h1>Good evening!</h1>
    </c:otherwise>
  </c:choose>

  Welcome to our site, open 24 hours a day.

  </body>
</html>
Figure 1-2. The output of a simple JSP page

Good evening!

Welcome to our site, open 24 hours a day.
In simple terms, a servlet is a piece of code that adds new functionality to a server (typically a web server), just like CGI and proprietary server extensions such as NSAPI and ISAPI. But compared to other technologies, servlets have a number of advantages:
Platform and vendor independence

Integration

Efficiency

Scalability

Robustness and security