

## JAVA SERVER PAGES

Una pagina JSP (*Java Server Pages*), richiesta da un utente (*client*), viene eseguita dal *server web* nella propria memoria; generalmente il prodotto dell'elaborazione è una pagina html che viene inviata all'utente che ha richiesto l'esecuzione della pagina JSP.

Quando invece un utente chiede una pagina *html* ad un *server*, questo la invia ed è il *browser*, in esecuzione sul pc del *client*, che la esegue e la visualizza, interpretando in modo opportuno i tag in essa contenuti.

L'elaborazione avviene quindi sul lato *client*; le azioni eseguite sul lato *server* sono: la ricezione della richiesta della pagina, la ricerca relativa nel memoria di massa (dischi) del *server*, l'invio al *client* della pagina oppure, se la pagina non è stata trovata, di un messaggio d'errore. Nel *server* non vengono eseguite elaborazioni.

Esempio di esecuzione di una pagina HTML:

Il *client* chiede la pagina:

<http://sdfonline.unimc.it/info/modulo.htm>;

1. il *server web* 'sdfonline.unimc.it' riceve la richiesta;
2. il *server web* ricerca la pagina nella cartella virtuale<sup>i</sup> 'info', che potrebbe corrispondere, ad esempio, alla cartella: 'c:\corso2008\info\';
3. il *server web* invia al *client* la pagina o il messaggio di errore in base all'esito della ricerca;
4. il *browser* sul pc del *client* visualizza la pagina 'modulo.htm' o il messaggio;

Esempio di esecuzione di una pagina JSP:

Il *client* chiede la pagina:

<http://sdfonline.unimc.it/info/modulo.jsp>;

1. il *server web* 'sdfonline.unimc.it' riceve la richiesta;
2. il *server web* ricerca la pagina nella cartella virtuale, 'info';
3. il *server web* esegue la pagina nella propria ram e, se l'elaborazione crea una pagina html, il *server* la invia al *client*;
4. il *browser* sul pc del *client* visualizza la pagina HTML ricevuta.

Quindi: una pagina JSP viene eseguita sul *server*, una pagina HTML viene eseguita sul *client*.

Le esigenze per cui si realizza una scrittura di pagine JSP, o di programmi che vengano eseguiti sul *server*, possono essere molteplici e saranno via via incontrate nel prosieguo del corso.

Spesso, comunque, si ha l'esigenza di inserire degli script<sup>1</sup> nelle pagine HTML in quanto l'HTML non riesce sempre a soddisfare tutte le esigenze incontrate.

Utilizzando solo HTML, ad esempio, non è possibile costruire una pagina che presenti la data e l'ora attuali ed un messaggio di 'Buongiorno' o di 'Buonasera' in base all'ora. Dobbiamo inserire uno script, ad esempio in JS (JavaScript)

(Listato 1)

Pagina con script in linguaggio JavaScript

```
<html>
<head><title>Saluto JS</title></head>
<body>
<script Language="JavaScript">
<!--
    var oggi = new Date();
    var ora  = oggi.getHours();
    var saluto = "";
    if (ora < 12 )
    {
        saluto="Buongiorno";
    }
    else
    {
        saluto="Buonasera";
    };
    document.Write("<h3>data = " + oggi + "</h3>");
    document.Write("<h3>ora = " + ora + "</h3>");
    document.Write("<h3>" + saluto + "</h3>");
//-->
</script>
</body>
</html>
```

La stessa necessità può essere accolta scrivendo una pagina JSP; una pagina cioè che contenga istruzioni scritte in linguaggio Java e che venga eseguita sul server.

Il listato 2 mostra la pagina saluti.jsp che produce gli stessi output ottenibili con script lato *client*.

---

<sup>1</sup> Insieme di istruzioni scritte in un linguaggio

(Listato 2)

saluti.jsp

```
<html>
<head><title>Pagina di saluto</title></head>
<body>
<%@ page import="java.util.*;" %>
<%
    GregorianCalendar gc = new GregorianCalendar();
    int ora = gc.get(gc. HOUR_OF_DAY);
    int min = gc.get(gc.MINUTE);
    String saluto= new String("");
    if (ora < 12 )
    {
        saluto="Buongiorno";
    }
    else
    {
        saluto="Buonasera";
    }
%>
<br>
<h3>ora =<%=ora%>.<%=min%></h3>
<h3><%=saluto%></h3>
</body>
</html>
```

La pagina deve essere memorizzata sul *server*, in una cartella dove è ammessa l'esecuzione di script; è possibile ora richiederne l'esecuzione attraverso un qualunque *browser*:

<http://sdfonline.unimc.it/corso/saluti.jsp>

Analisi dello script `saluti.jsp` (Listato 2).

- la parte di codice è racchiusa fra i simboli '`<%`' e '`%>`'; inserire del codice fra questi simboli significa richiedere l'esecuzione del codice sul *server*;
- `<%=ora%>`, `<%=saluto%>`: le variabili '`ora`' e '`saluto`' sono precedute dal simbolo uguale, ciò significa che i contenuti delle variabili verranno inseriti nella pagina html generata.
- i tag html vengono trasferiti nella pagina html che viene creata nella memoria del server

Il *server web* svolge, in risposta alla richiesta della pagina JSP, le seguenti azioni:

1. ricerca la pagina;
2. esegue la pagina nella propria RAM (trattandosi di una pagina JSP);
3. costruisce una pagina HTML con i tag in essa contenuti ed esegue le istruzioni racchiuse fra '<%>' e '%>'.

| Listato pagina JSP                                                                                                                                                                                                                    | Elaborazione                                                                                                                                                                                                                                                                                                            | Pagina HTML costruita                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <pre>&lt;html&gt; &lt;head&gt;&lt;title&gt; Pagina di saluto &lt;/title&gt;/head&gt; &lt;body&gt; &lt;%@ page import="java.util.*;" %&gt;</pre>                                                                                       | direttiva per importare le classi di java.util                                                                                                                                                                                                                                                                          | <pre>&lt;html&gt; &lt;head&gt;&lt;title&gt; Pagina di saluto &lt;/title&gt;&lt;/head&gt; &lt;body&gt;</pre> |
| <pre>&lt;% GregorianCalendar gc = new GregorianCalendar(); int ora = gc.get(gc.HOUR); int min = gc.get(gc.MINUTE); String saluto= new String(""); if (ora &lt; 12 ) { saluto="Buongiorno"; } else { saluto="Buonasera"; } %&gt;</pre> | <p>Viene creata un'istanza della classe GregorianCalendar, che viene chiamata gc; La classe GregorianCalendar è contenuta in java.util e permette la gestione della data corrente; viene usato il metodo 'get' per estrarre l'ora ed il minuto</p> <p>saluto ='Buongiorno'</p> <p>oppure</p> <p>saluto ='Buonasera'</p> |                                                                                                             |
| </h3><br><h3>ora =                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                         | </h3><br><h3>ora =                                                                                          |
| <%=ora%>.<%=min%>                                                                                                                                                                                                                     | visualizza nella pagina html creato il valore di 'ora' e 'minuti'                                                                                                                                                                                                                                                       | 10.58 AM                                                                                                    |
| </h3><h3>                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                         | </h3><h3>                                                                                                   |
| <%=saluto%>                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                         | Buongiorno                                                                                                  |

|                             |  |                             |
|-----------------------------|--|-----------------------------|
| </h3><br></body><br></html> |  | </h3><br></body><br></html> |
|-----------------------------|--|-----------------------------|

In definitiva il *server* crea la seguente pagina HTML, che viene inviata al *client*:

(Listato 3)

```
<html>
<head><title>Pagina di saluto</title></head>
<body>
<br>
<h3>ora = 10.58 AM</h3>
<h2>Buongiorno</h3>
</body>
</html>
```

Il *browser* del *client* visualizza la pagina sul video:



La cartella 'c:\corso2000\corso', sul *server*, contiene la pagina 'saluti.jsp', ma non contiene la pagina inviata al *client*; tale pagina non esiste:

- è generata dal *server web* durante l'esecuzione della pagina JSP;
- è chiamata pagina dinamica.

Il linguaggio utilizzato nelle pagine JSP è JAVA.

Altri metodi della classe `GregorianCalendar`

```
gc.get(gc.YEAR)
gc.get(gc.MONTH)
gc.get(gc.WEEK_OF_YEAR)
gc.get(gc.WEEK_OF_MONTH)
gc.get(gc.DATE)
gc.get(gc.DAY_OF_MONTH)
gc.get(gc.DAY_OF_YEAR)
gc.get(gc.DAY_OF_WEEK)
gc.get(gc.DAY_OF_WEEK_IN_MONTH)
```

```
gc.get(gc.HOUR)
gc.get(gc.HOUR_OF_DAY)
gc.get(gc.MINUTE)
gc.get(gc.SECOND)
gc.get(gc.MILLISECOND)
```

Per poter usare determinate classi, occorre importare le librerie che le contengono attraverso 'import' nella direttiva @page nel seguente modo:

```
<@page import="aaa.bbb.ccc">
```

Le librerie sono organizzate in modo gerarchico e vengono memorizzate nell'elaboratore con l'installazione di Java.

E' possibile anche la forma:

```
<@page import="aa.bb.*">
```

, ad indicare le classi esistenti dal livello 'bb' in poi, nella gerarchia.

Leggere "Cosa sono i Package di Java" nella dispensa 'variabili operatori istruzioni package'

## Esempi di pagine jsp

### 1. stampare in una pagina html gli interi da 1 a 10

```
<html>
<head><title>visualizza interi</title></head>
<body>
<%
    int i;
    for(i=1;i<=10;i++)
    {
        out.print(i);
        out.print("<br>");
    }
%>
</body>
</html>
```

2° versione:

```
<html>
<head><title>visualizza interi</title></head>
<body>
<%
```

```

        int i;
        for(i=1;i<=10;i++)
        {
%>
            <%=i%>
            <br>
        <%
        }
%>
</body>
</html>

```

## 2. generare una tabellina pitagorica

```

<html>
<head><title>visualizza interi</title></head>
<body>
tabellina pitagorica
<br>
<table>
<%
    int i;
    int j;
    int n;
    for(i=1;i<=10;i++)
    {
%>
        <tr>
<%
            for(j=1;j<=10;j++)
            {
%>
                <td>
<%
                    n=i*j;
                    out.println(n);
%>
                </td>
<%
            }
%>
        }
    }
%>
</table>

```

```

        }
    %>
</tr>
<%
    }
%>
</table>
</body>
</html>

```

## Vettori

<http://www.ba.infn.it/~zito/jsem/array.html>  
<http://www.mokabyte.it/2002/06/javabase-4.htm>

esempio:  
 sommare gli elementi di un vettore

```

<html>
<head><title>visualizza interi</title></head>
<body>
<%
    int[] v = new int[5];
    v[0] = 10;
    v[1] = 12;
    v[2] = 14;
    v[3] = 16;
    v[4] = 18;
    int i;
    int s;
    s=0;
    for(i=0;i<5;i++)
    {
        s=s+v[i];
    }
    out.println(s);
%>
</body>
</html>

```

Provare gli esempi precedenti:

1. memorizzare le pagine, utilizzando 'blocco note' in documenti con estensione .jsp
2. inserire queste pagine in una cartella della ROOT
3. lanciare EasyTomcat: doppio click sull'icona sul desktop, oppure rintracciarlo su 'programmi'
4. attivare Tomcat e Mysql agendo sul pulsante start
5. lanciare un browser e digitare:  
[http://127.0.0.1:8080/nome\\_cartella/pagina.jsp](http://127.0.0.1:8080/nome_cartella/pagina.jsp)

### **Creare le seguenti pagine JSP**

1. modificare la pagina del listato 2 visualizzando l'ora nella barra del titolo e provando gli altri metodi di GregorianCalendar
2. modificare la pagina relativa alla tabellina pitagorica, prevedendo:
  - a. la memorizzazione della tabellina in una matrice
  - b. la visualizzazione della matrice

---

<sup>i</sup> Le Java Server Pages (JSP) possono essere eseguite in un server web Apache con un application server Tomcat (o altri). L'application server è il software che permette l'esecuzione di pagine jsp.

Pagine jsp, per poter essere eseguite, debbono essere memorizzate in cartelle, sul server, eseguibili (cartelle virtuali).

Le cartelle eseguibili sono cartelle interne alla cartella C:\Programmi\EasyTomcat\Server\Tomcat\webapps\ROOT (dipende, comunque, dal percorso scelto durante l'installazione). Quindi per poter eseguire pagine jsp, occorre inserirle o direttamente in ROOT, oppure in sottocartelle interne a ROOT.

Per poter lanciare le pagine sul proprio pc, dopo aver installato EasyTomcat (che installa automaticamente Apache, Tomcat, Mysql), basta digitare l'indirizzo:

[http://127.0.0.1:8080/nome\\_cartella\\_interna\\_a\\_ROOT/nome\\_pagina.jsp](http://127.0.0.1:8080/nome_cartella_interna_a_ROOT/nome_pagina.jsp)