



# Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

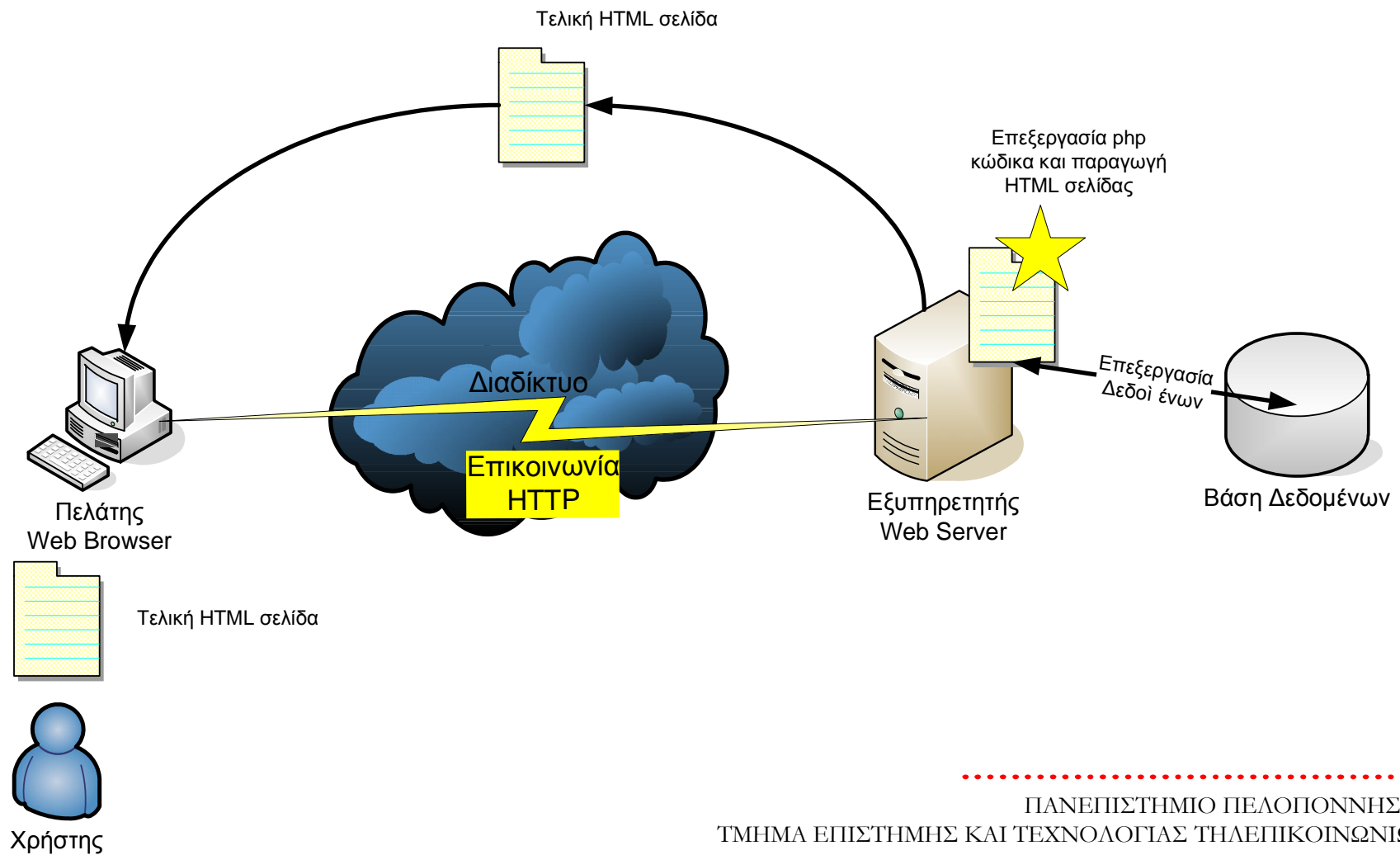
## 8<sup>η</sup> Διάλεξη: Προγραμματισμός στην πλευρά του εξυπηρετητή: Τεχνολογία Java Server Pages (JSP)

Δρ. Απόστολος Γιάμας

Λέκτορας (407/80)

gkamas@uop.gr

# Προγραμματισμός στην πλευρά του εξυπηρετητή (server side)





## Τι είναι Servlet Java Server Pages (JSP)

- Η τεχνολογία JavaServer Pages (JSP) είναι server-side τεχνολογία
- Βασίζεται στην τεχνολογία των Java Servlets
- Η κύρια διαφορά τους είναι πως ενώ ο κώδικας των Servlets είναι δομημένος σε κλάσεις που γράφονται, μεταγλωττίζονται και εκτελούνται όπως κάθε κλάση Java, οι JSP σελίδες αποτελούνται από στατικό HTML και δυναμικά παραγόμενο HTML περιεχόμενο τα οποία είναι διαχωρισμένα με ειδικά tags
- Με απλά λόγια, η JSP τοποθετεί Java κώδικα μέσα σε HTML έγγραφα



## Τι είναι Servlet Java Server Pages (JSP)

- Μια JSP σελίδα έχει extension \*.jsp (αυτό πληροφορεί τον web server ότι η σελίδα πρέπει να φορτωθεί από έναν JSP container για να διερμηνεύσει και να εκτελέσει τον JSP κώδικα
- Ο JSP container διερμηνεύει τα JSP tags και τα scriptlets, παράγει το δυναμικό περιεχόμενο και το στέλνει πίσω στον client ως HTML ή XML σελίδα.

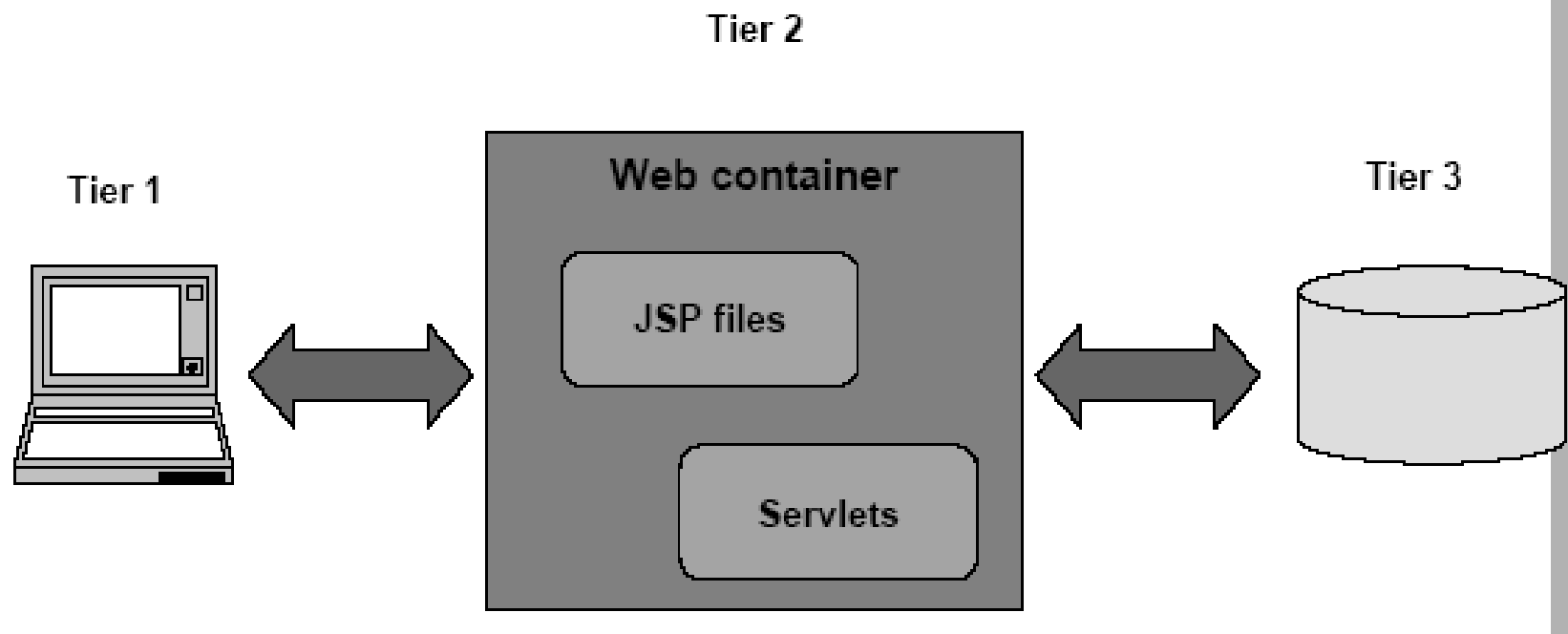
```
<HTML> <BODY>
```

```
Hello! The time is now <%= new java.util.Date() %>
```

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



# JSP σε μια 3-tier αρχιτεκτονική



Διαφάνεια 5

Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

# Προϋποθέσεις για να 'τρέξουμε' JSP κώδικα



- Ο web server πρέπει να 'υποστηρίζει' την τεχνολογία JSP
- Δηλαδή πρέπει να υπάρχει εγκατεστημένος ένας JSP container ή application server, π.χ. Blazix, ServletExec, JRun, WebLogic, Tomcat, WebSphere, .....
- Για την εκτέλεση JSP κώδικα στον NetBeans χρησιμοποιείται ο ενσωματωμένος application server Tomcat
- Τα JSP αρχεία, όπως και οι Java Servlets κλάσεις, πρέπει να είναι τοποθετημένα σε ένα συγκεκριμένο directory του application server
- Οι application servers μεταγλωττίζουν τον JSP κώδικα μόνο την πρώτη φορά που εκτελείται (φυσικά, κάθε φορά που τροποποιείται, ο JSP κώδικας επαναμεταγλωττίζεται)



## JSP: Πως δουλεύει

- Βήματα που λαμβάνουν χώρα όποτε ζητείται μια JSP σελίδα:
1. Ο web browser κάνει ένα request μέσω Internet για μια \*.jsp σελίδα
  2. Το request φτάνει στον Web server
  3. Ο Web server αναγνωρίζει ότι ζητήθηκε ένα ειδικό αρχείο (\*.jsp), και περνάει το JSP αρχείο στο JSP Servlet Engine (application server)
  4. Αν το JSP αρχείο ζητείται για πρώτη φορά, τότε διερμηνεύεται (parsing), διαφορετικά η διαδικασία προχωράει στο βήμα 7

Διαφάνεια 7

Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

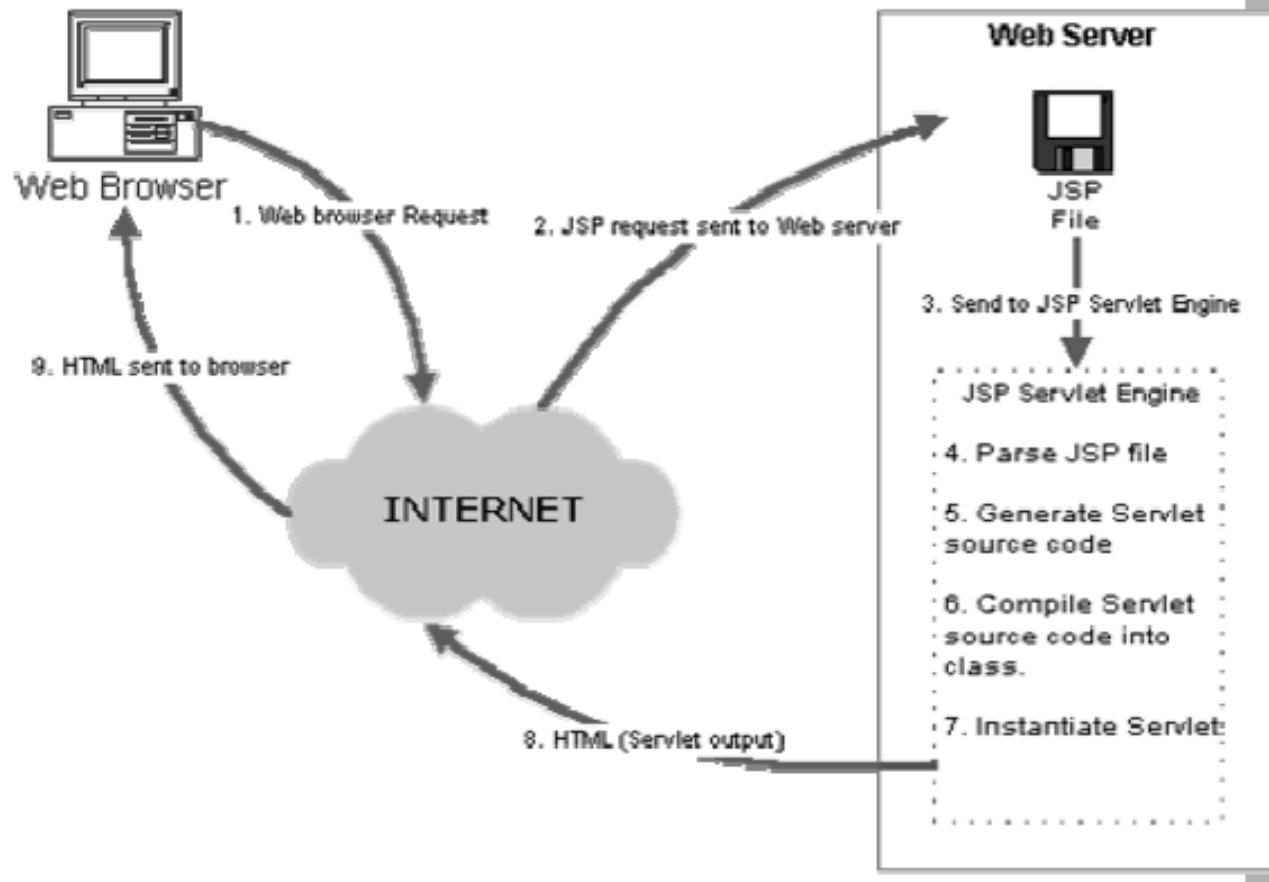
## JSP: Πως δουλεύει



5. Στη συνέχεια παράγεται ένας Servlet από το JSP αρχείο. Όλο το στατικό HTML που περιέχεται στη JSP σελίδα μετατρέπεται σε 'out.println' εντολές
6. Ο πηγαίος κώδικας του Servlet που προκύπτει μεταγλωττίζεται σε ένα class αρχείο
7. Ο Servlet εκτελείται, καλούνται οι κατάλληλες μέθοδοι
8. Ο HTML κώδικας που προκύπτει από το Servlet output στέλνεται στον web browser
9. Ο HTML κώδικας εμφανίζεται στον web browser του χρήστη



# JSP: Πως δουλεύει



Διαφάνεια 9

Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

# Hello World με JSP



```
<html>
<head>
<title>Hello World JSP page</title>
</head>
<body>
<h3>Αυτή είναι απλή HTML. Ακολουθεί δυναμικά
παραγόμενη HTML από JSP...</h3>
<h4>Hello World!!! The date is: <%=
java.text.DateFormat.getDateInstance(java.text.Dat
eFormat.MEDIUM,
java.text.DateFormat.MEDIUM).format(new
java.util.Date())%> </h4>
</body>
</html>
```

Διαφάνεια 10

Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

# JSP tags



- Στην JSP χρησιμοποιούνται 5 κύρια tags (ετικέτες):
- Declaration tag
- Expression tag
- Directive tag
- Scriptlet tag
- Action tag



## JSP tags: Expression tag ( `<%= %>` )

- Αυτή η ετικέτα επιτρέπει την ενσωμάτωση Java εκφράσεων (expressions)
- Το ερωτηματικό ( ; ) ΔΕΝ προστίθεται στο τέλος του κώδικα που τοποθετείται ανάμεσα σ' αυτές τις ετικέτες

```
<HTML>
```

```
<BODY>
```

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

```
</BODY>
```

```
</HTML>
```



## JSP tags: Declaration tag ( `<%! %>` )

- Χρησιμοποιείται για δηλώσεις μεταβλητών και μεθόδων
- Οι δηλώσεις τοποθετούνται ανάμεσα στις ετικέτες `<%!` και `%>`
- Ο κώδικας που τοποθετείται ανάμεσα σε αυτές τις ετικέτες πρέπει να τελειώνει με ερωτηματικό ( ; )



## JSP tags: Declaration tag ( `<%! %>` )

```
<HTML>
<BODY>
<%!
java.util.Date d = new java.util.Date();
String getDate() {
return "The current date is: " + d;
}
%>
<h2> <%= getDate() %> </h2>
</BODY>
</HTML>
```

## JSP tags: Directive tag ( `<%@ directive ... %>` )



- Μια JSP directive (οδηγία) παρέχει ειδικές πληροφορίες σχετικά με τη JSP σελίδα στην JSP Engine
- Υπάρχουν τρεις κυρίως τύποι οδηγιών (directives):
  - `page` - πληροφορίες για την επεξεργασία της JSP σελίδας
  - `include` - αρχεία που θα χρησιμοποιηθούν από αυτή τη JSP σελίδα
  - `Tag library` - βιβλιοθήκη ετικετών (tag library) που θα χρησιμοποιηθεί από αυτή τη JSP σελίδα
- Οι οδηγίες (directives) δεν παράγουν κάποιο ορατό αποτέλεσμα όταν φορτώνεται η σελίδα, αλλά αλλάζουν τον τρόπο που η JSP Engine επεξεργάζεται τη σελίδα



## JSP tags: Directive tag: page directive

- Αυτή η οδηγία (directive) έχει 11 προαιρετικά ορίσματα που παρέχουν στη JSP Engine ειδικές πληροφορίες για την επεξεργασία της JSP σελίδας, π.χ.:

<b>language</b>	Ποια γλώσσα χρησιμοποιεί η σελίδα	<code>&lt;%@ page language = "java" %&gt;</code>
<b>extends</b>	Η κλάση από την οποία κληρονομεί (superclass) ο Servlet που προκύπτει από τη μεταγλώττιση της σελίδας από τη JSP engine	<code>&lt;%@ page extends = "com.taglib... %&gt;</code>
<b>import</b>	Εισαγάγει τις κλάσεις ή ολόκληρα πακέτα (packages) που θα χρησιμοποιηθούν από το JSP κώδικα (επιτρέπει επανάχρηση κώδικα)	<code>&lt;%@ page import="java.util.*, java.text.*" %&gt;</code>
<b>contentType</b>	Ορίζει το mime type και το character set για τη JSP σελίδα (κωδικοποίηση του output της σελίδας).	





# JSP tags: Directive tag: page directive

```
<html>
<head>
<title>Hello World JSP page</title>
</head>
<body>
<h3>Αυτή είναι απλή HTML. Ακολουθεί δυναμικά παραγόμενη
HTML από JSP...</h3>
<%@ page contentType="text/html;charset=windows-1253"
language="java" import="java.util.*, java.text.*"%>
<h4>Hello World!!! The date is: <%=
DateFormat.getDateInstance(DateFormat.MEDIUM,
DateFormat.MEDIUM).format(new Date())%> </h4>
</body>
</html>
```

## JSP tags: Directive tag: include directive



- Επιτρέπει να συμπεριλάβουμε τα περιεχόμενα μιας HTML ή JSP σελίδας μέσα σε μια JSP σελίδα
- Συνήθως χρησιμοποιείται για να συμπεριλάβουμε navigation bars, headers και footers που είναι κοινά σε πολλές σελίδες ενός site.

```
<%@ include file = "myfolder/privacy.html" %>
```

```
<%@ include file = "navigation.jsp" %>
```



## JSP tags: Directive tag: tag lib directive

- Ένα tag lib είναι μια συλλογή από προσαρμοσμένες ετικέτες (custom tags) που μπορούν να χρησιμοποιηθούν σε αυτή τη JSP σελίδα
- `<%@ taglib uri = "tag library URI" prefix = "tag Prefix"%>`



## JSP tags: Scriptlet tag ( `<% ... %>` )

- Java κώδικας τοποθετημένος ανάμεσα στις ετικέτες `<%` και `%>` ονομάζεται **Scriptlet**
- Αυτός ο κώδικας έχει πρόσβαση σε οποιαδήποτε μεταβλητή, μέθοδο ή **java bean** έχει προηγουμένα οριστεί
- Ο κώδικας που τοποθετείται ανάμεσα σε αυτές τις ετικέτες πρέπει να ακολουθείται από ερωτηματικό ( ; )



## JSP tags: Scriptlet tag ( `<% ... %>` )

```
<%! int n = 5; %>
<table border=2>
<%
for ( int i = 0; i < n; i++ ) {
%>
<tr>
<td>Number</td>
<td><%= i+1 %></td>
</tr>
<%
}
%> </table>
```



## Scriptlets που παράγουν HTML

- Τα scriptlet δεν παράγουν από μόνα τους HTML. Για να παράγει HTML, ένα scriptlet πρέπει να χρησιμοποιήσει τη μεταβλητή 'out' (της κλάσης `javax.servlet.jsp.JspWriter`), η οποία δεν χρειάζεται να δηλωθεί.

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

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

```
Hello! The time is now <% out.println(d.toString()); %>
```

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

- Άλλες χρήσιμες μεταβλητές
- `request`: επιτρέπει πρόσβαση σε πληροφορίες σχετικά με τη συνομιλία από τον browser στο server, π.χ. διάβασμα παραμέτρων που στάλθηκαν από HTML φόρμα στη JSP σελίδα
- `response`: επιτρέπει πρόσβαση σε πληροφορίες σχετικά με τη συνομιλία από το server στον browser



## JSP tags: Action tag

- Υπάρχουν τρεις κύριοι ρόλοι για τα action tags :
- 1. Ενεργοποίηση της χρήσης των server side Javabeans
- 2. Μεταφορά ελέγχου ανάμεσα σε σελίδες
- 3. Υποστήριξη applets ανεξαρτήτως browser
- Javabeans: Το Javabean είναι μια ειδική κατηγορία κλάσης (class) με ένα αριθμό μεθόδων. Μια JSP page μπορεί να καλέσει αυτές τις μεθόδους και να αφήσει τον περισσότερο κώδικα ενσωματωμένο σε αυτά τα Javabeans. Για παράδειγμα, αν θέλατε να έχετε μια feedback form που αυτοματοποιεί την αποστολή e-mail. Έχοντας μια JSP σελίδα με μια φόρμα, όταν ο επισκέπτης πατάει το submit button που στέλνει δεδομένα σε ένα Javabean που αναλαμβάνει να στείλει το e-mail. Με αυτόν τρόπο, η JSP σελίδα δεν θα έχει κώδικα που αναλαμβάνει την αποστολή e-mails (JavaMail API) και το ίδιο Javabean μπορεί να χρησιμοποιηθεί και από άλλη σελίδα (επανάχρηση κώδικα)
- Για να χρησιμοποιήσουμε ένα Javabean σε μια JSP σελίδα, ακολουθείται η σύνταξη:
- `<jsp : usebean id=".." scope = "application" class = "com..." />`

# Διάβασμα παραμέτρων HTML φόρμας από JSP σελίδα (I)



```
<html>
<head>
<title>Hello form...</title>
</head>
<body>
<form method="POST" action="HelloName.jsp">
<P>Give me your name:
<input type="text" name="name"/>
</P>
<P>
<input type="submit" value="Submit"/>
</P>
</form>
</body>
</html>
```



## Διάβασμα παραμέτρων HTML φόρμας από JSP σελίδα (II)



```
<%  
String name = request.getParameter("name");  
out.println("<h1>Hello " + name + "</h1>");  
%>
```

## Άλλες λειτουργίες



- Χρήση Session (δες παράδειγμα)
- Χρήση cookies (δες παράδειγμα)
- Αλληλεπίδραση με βάση δεδομένων (δες παράδειγμα)

Διαφάνεια 26

Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου



## Εφαρμογές JavaServer Pages

- Η τεχνολογία είναι ευρέως χρησιμοποιούμενη στο χώρο του WWW
- Μπορούμε εύκολα να διασυνδέσουμε JSP σελίδες σε βάσεις δεδομένων για να επιτύχουμε ανάκτηση, προσθήκη ή ενημέρωση δεδομένων, με την ίδια περίπου μέθοδο που διασυνδέσαμε Java Servlets με ΒΔ
- Χρήση JSP για έλεγχο και παρακολούθηση συνόδων (sessions tracking & control)
- Χρήση JSP για αυτοματοποιημένη αποστολή e-mail, αρκεί να γνωρίζουμε τον SMTP server του δικτύου όπου εκτελείται η JSP σελίδα



## Συμπεράσματα: JSP vs Servlets

- Η JSP δεν προσφέρει κάτι που δεν μπορούμε να επιτύχουμε με Java Servlets, άλλωστε βασίζεται στην τεχνολογία των Servlets
- Η διαφορά έγκειται κυρίως στη διαφορετική προσέγγιση, καθώς στους Servlets ή παραγωγή στατικού ή δυναμικού HTML είναι ενσωματωμένη σε κώδικα Java
- Αντίθετα, στη JSP το στατικό HTML περιεχόμενο είναι ξεκάθαρα διαχωρισμένο από το δυναμικό HTML που παράγεται από JSP (Java) κώδικα. Αυτό συνιστά έναν πιο «καθαρό» σχεδιασμό καθώς διαχωρίζει τον απλό HTML κώδικα από την προγραμματιστική λογική και αφήνει τον web designer να επικεντρωθεί στην παρουσίαση και μορφοποίηση του περιεχομένου (HTML)



## Πλεονεκτήματα των JSPs

- Είναι ευκολότερη και ταχύτερη η ανάπτυξή τους, κυρίως για μικρά projects
- Με τους servlets είναι δυσκολότερη η παραγωγή στατικού HTML κώδικα, χρειάζονται πολλές `out.println` εντολές, π.χ.  

```
out.println("<body>");  
out.println("<h2>Hello " + user + "</h2>");
```
- Δίνουν έμφαση στην μορφή και παρουσίαση της σελίδας (HTML) και αφήνουν την προγραμματιστική λογική σε Java κώδικα
- Αν έχουμε έτοιμο JSP κώδικα, μπορεί εύκολα να ενσωματωθεί σε HTML σελίδες από web authors που δε γνωρίζουν προγραμματισμό



## Μειονεκτήματα των JSPs

- Εύρεση και διόρθωση λαθών (debugging) είναι πολύπλοκη
- Μεγάλες ποσότητες ενσωματωμένου κώδικα σε scriplets μπορεί να οδηγήσουν σε σελίδες που είναι δύσκολο να συντηρηθούν
- Η απόδοση JSPs είναι κατώτερη των servlets (το compilation 200 JSPs θα δημιουργήσει 200 servlets). Αν χρησιμοποιούσαμε τεχνολογία Servlets θα μπορούσαμε να ενσωματώσουμε περισσότερη λειτουργικότητα σε λιγότερους servlets.



# JSP Tutorials /Servlet Tutorials

## — Java

— <http://java.sun.com/j2se/1.5.0/docs/api/index.html>

## — JSP

— <http://www.jsptut.com/Index.html>

— <http://www.caucho.com/resin-3.0/jsp/tutorial/index.xtp>

— <http://www.roseindia.net/jsp/jsp.shtml>

## — Servlet:

— <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>

— <http://www.novocode.com/doc/servlet-essentials/>

— [http://www.itl.cs.tu-bs.de/doc/JSDK2.0/doc/servlet\\_tutorial.html](http://www.itl.cs.tu-bs.de/doc/JSDK2.0/doc/servlet_tutorial.html)

— [http://www.cs.bham.ac.uk/~tmw/servlet\\_tutorial/index.shtml](http://www.cs.bham.ac.uk/~tmw/servlet_tutorial/index.shtml)