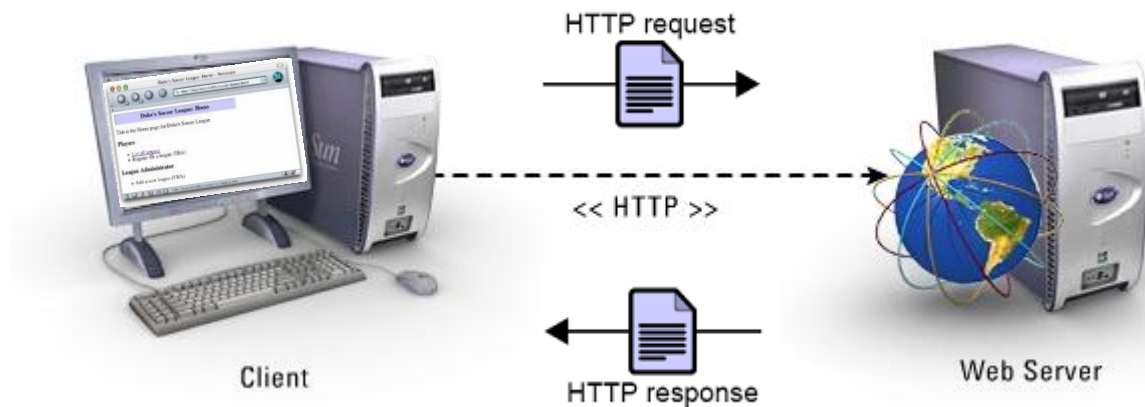


פיתוח מערכות תוכנה מבוססות Java

Web Applications II

אוהד ברזילי

בשבוע שעבר התחלנו בסקירת יישומי Web



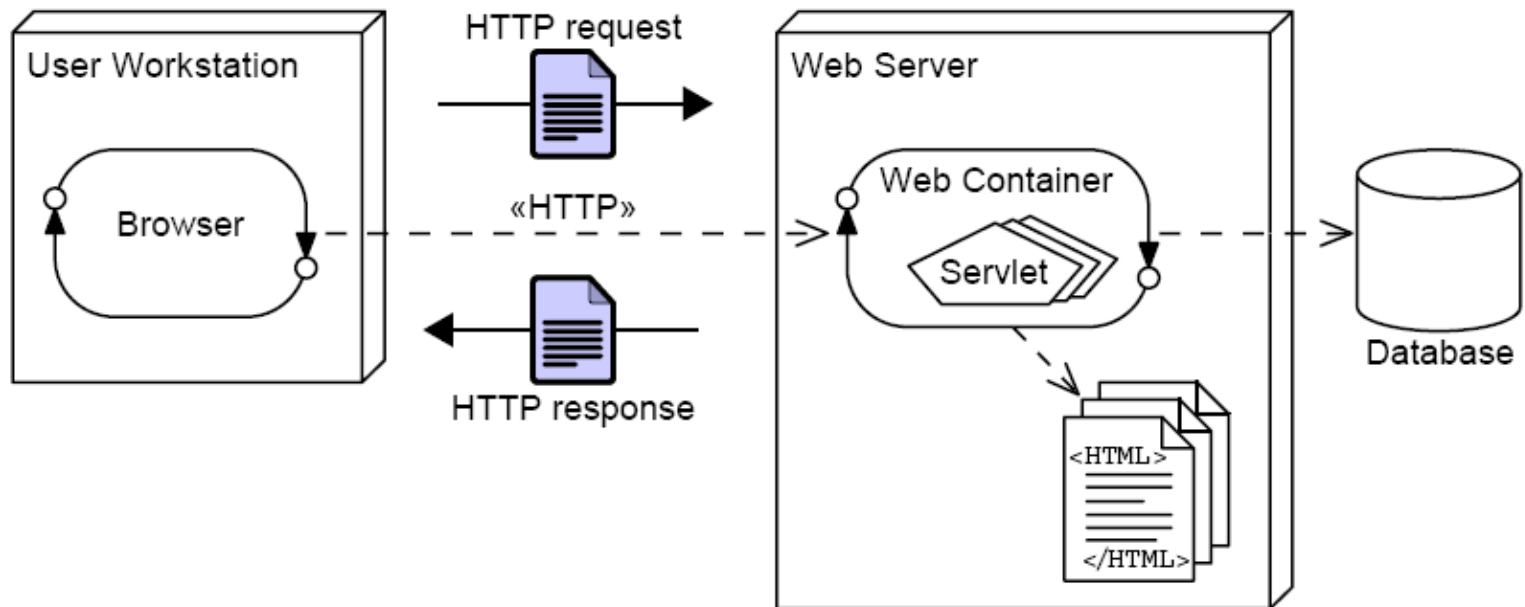
- כדי להבין נושא זה היטב, חשוב להבין עבור כל טכנולוגיה האם היא שייכת ל client (רצה על הדפדפן) או ל server (הקוד רץ על מחשב מרוחק – אתר האינטרנט)
- היום נסקור כמה מהטכנולוגיות הקיימות וננסה להבין אילו בעיות הן פותרות ובאיזו צורה

Web Server Side programming

USING SERVLETS



ארכיטקטורת Web Container



- ה- web container עושה מיפולציה על פורמט הודעת GET - במקום המיועד להזנת **שם הקובץ**, הוא מצפה לקבל **שם מחלקה**

תפקידי ה Web Container

■ טיפול בהבטי תקשורת:

■ יצירת שקע והמתנה ללקוחות

■ ניהול חוטים:

■ יצירת Thread נפרד לטיפול בלקוחות נכנסים (accept)

■ ניהול השרותונים

■ הפניית הבקשות לשרותון המתאים

■ אתחול המופעים לפי דרישה

■ שמירת מידע רלוונטי בין קריאות (session, context)

תפקידי ה Web Container

- טיפול בפרוטוקול HTTP
 - יצירת מחלקות העזר (`HttpServletRequest`, `HttpServletResponse`)
 - ניתוח כותרות ה HTTP (`request`)
 - ניתוח את המחרוזת במקום המיועד לשם הקובץ וחילוץ שם ה Servlet המבוקש ואת שמות המשתנים וערכיהם (לפי התווים: `'/'`, `'='`, `'&'`, `'?'`)
 - חילול כותרות ה HTTP (`response`)
- מיפוי מרחב השמות הלוגיים והפיזיים
- בקרת תצורה

הרעיון

■ הפרדה בין:

■ **המסגרת (framework):** החוזרת על עצמה

■ "ההבטים": טיפול בלקוח חדש, יצירת חוט, יצירת שקע ועוד...

■ **הלוגיקה העסקית:** מה התוכנית עושה

■ קוד Java שיכול לעשות כל דבר

■ את השרת (web container) יכתוב משהו אחר

■ אנחנו נטען לתוכו מחלקות Java בשם servlets (שרותונים?)

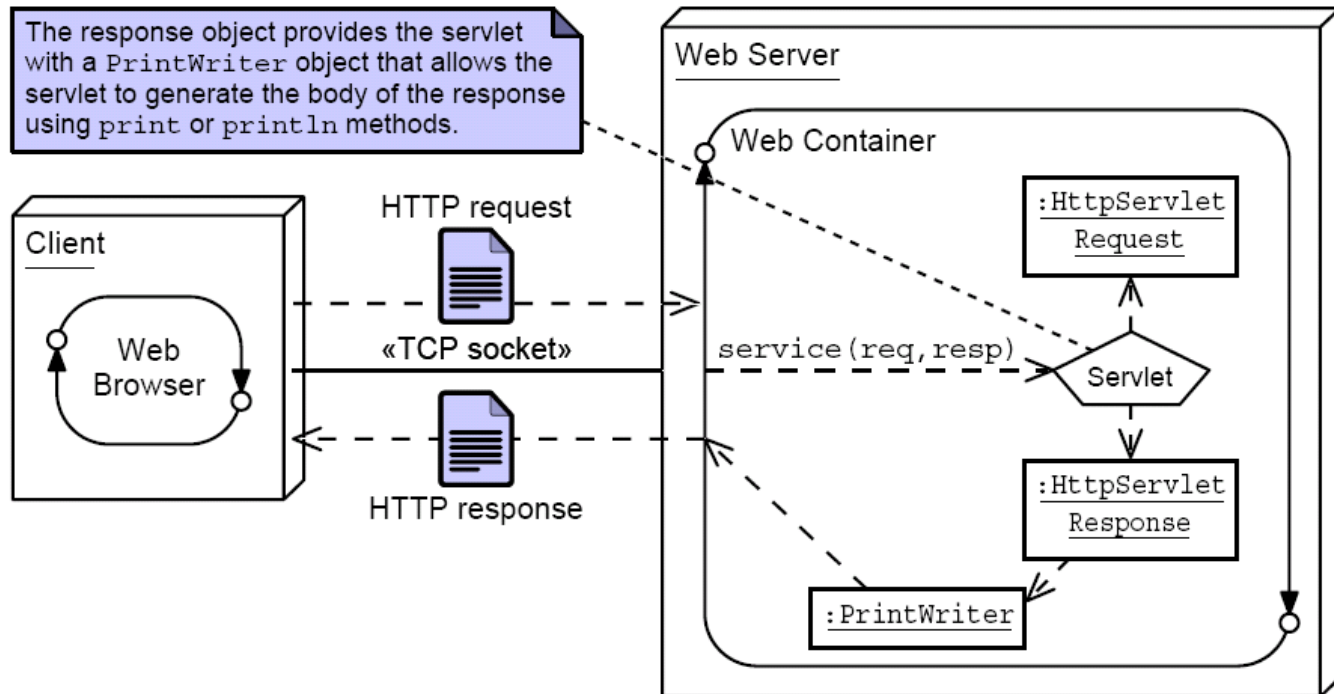
אשר מממשות כל אחת לוגיקה עסקית משלה

■ בפרט נוכל להתקין על אותו שרת כמה שרותים במקביל

■ כמו כן, נוכל בהמשך להחליף את השרת שכתבנו בשרת מקצועי

(Tomcat, OC4J ואחרים) מבלי להחליף את השרותונים

ארכיטקטורת Web Container



על המחלקה הזו הוא יפעיל את השרות `doGet`, (במקרה הכללי `service`) וידאג להעביר לה כפרמטרים **מחלקות עזר** שבעזרתם תקרא את הפרמטרים אם הועברו כאלה בשורת הכתובת, ותייצר הודעת תשובה

Hello World Servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

(forms) ם'ספּט

The image displays two windows side-by-side. The left window is jEdit, showing HTML code for a form. The right window is Mozilla Firefox, showing the rendered form titled "Duke's Soccer League: Add a New League".

HTML Code (jEdit):

```
19
20 <form action='add_league.do' method='POST'>
21 Year: <input type='text' name='year' /> <br/><br/>
22 Season: <select name='season'>
23     <option value='UNKNOWN'>select...</option>
24     <option value='Spring'>Spring</option>
25     <option value='Summer'>Summer</option>
26     <option value='Fall'>Fall</option>
27     <option value='Winter'>Winter</option>
28 </select> <br/><br/>
29 Title: <input type='text' name='title' /> <br/><br/>
30 <input type='submit' value='Add League' />
31 </form>
32
33 </body>
```

Rendered Form (Mozilla Firefox):

Duke's Soccer League: Add a New League

This form allows you to create a new soccer league.

Year:

Season:

Title:

Done

העברת פרמטרים ב GET

```
GET /admin/add_league.do?year=2005&season=Winter&title=Westminster+Indoor HTTP/1.1
```

```
Host: localhost:8080
```

```
User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.4
```

```
Gecko/20030624 Netscape/7.1
```

```
Accept:
```

```
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain
```

```
video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
```

```
Accept-Language: en-us,en;q=0.5
```

```
Accept-Encoding: gzip,deflate
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

```
Keep-Alive: 300
```

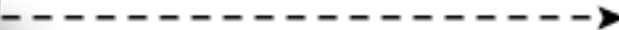
```
Connection: keep-alive
```



Client



Web Server



העברת פרמטרים ב POST

POST /admin/add_league.do HTTP/1.1

Host: localhost:8080

User-Agent: Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1. Gecko/20030624 Netscape/7.1

Accept:

text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plai
0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://localhost:8080/controller/admin/add_league.html

Content-Type: application/x-www-form-urlencoded

Content-Length: 55

year=2005&season=Winter&title=Westminster+Indoor+Soccer



Client



Web Server

GET vs. POST

The HTTP GET method is used when:

- The processing of the request is idempotent
- The amount of form data is small
- You want to allow the request to be bookmarked

The HTTP POST method is used when:

- The processing of the request changes the state of the server, such as storing data in a database
- The amount of form data is large
- The contents of the data should not be visible in the URL (for example, passwords)

Session Management

- פרוטוקול HTTP הוא חסר הקשר אולם ה Web Container מספק אמצעים ליצירת הקשר
- לדוגמא: עגלת קניות וירטואלית

הרעיון: ■

- עבור כל לקוח נשמור מידע נשמר באובייקט Java HttpSession
- בכל פנייה לשרת הלקוח ישלח מזהה ייחודי
- השרת יקבל את המזהה ויקשר את הלקוח עם המידע שנשמר בעבורו

Session Management Strategies

■ עוגיות (Cookies)

- שרת יכול להוסיף לתשובה ללקוח עוגייה – קובץ טקסט עם זוגות `name=value`
- הלקוח ישלח בכל התקשרות עתידית עם השרת את כל העוגיות שהוא קיבל ממנו בעבר
- בפועל עוגיות מכילות רק מזהה ייחודי – שאר המידע נשמר בשרת

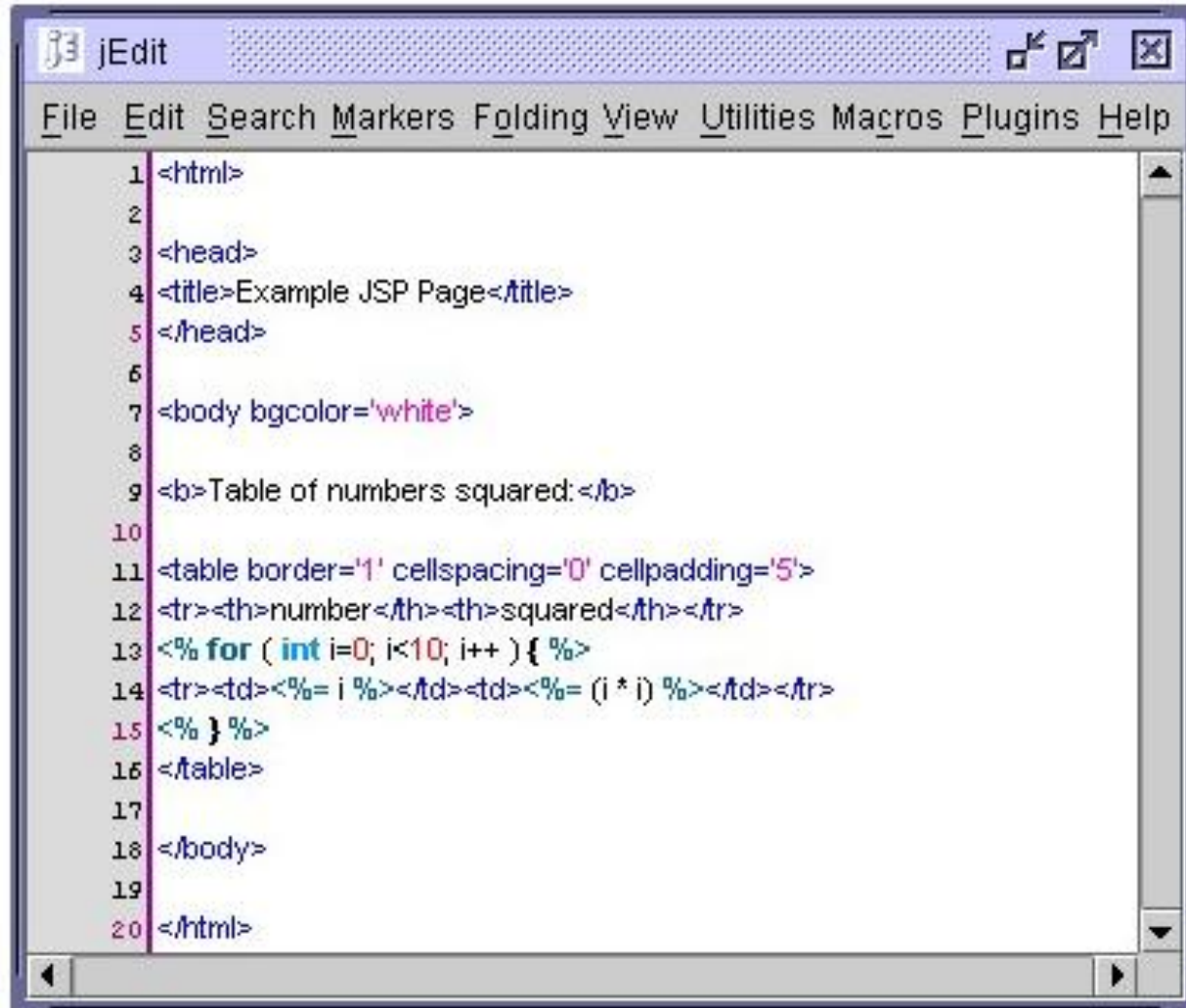
■ שכתוב קישוריות (URL Rewriting)

- כל הקישוריות (`href`) בעמוד ה HTML ישוכתבו להכיל גם את המזהה הייחודי כמשתנה נוסף בשורת הכתובת
- אסטרטגיה זו שימושית במקרה שהדפדפן חוסם עוגיות

Java / HTML

- על אף שרצינו שפלט התוכנית יהיה ב HTML איננו רוצים לערבב קוד Java וקוד HTML
 - Model/View Separation
 - לפעמים נכתבים ע"י מתכנתים שונים
- יש צורך בטכנולוגיה שתאפשר פיתוח אורתוגונלי של המודל ושל ההצגה (גם אם בפועל הקוד שירוך ממזג את שניהם)
- טכנולוגיית JSP מתיימרת להשיג מטרה זו בדיוק:
 - JSP היא שפת תגיות (בדומה ל html) המאפשרת לשלב בתוכה תוכן דינאמי
 - בזמן ריצה קובץ ה JSP עובר "קומפילציה" לקובץ Servlet

גלישה לעמוד JSP



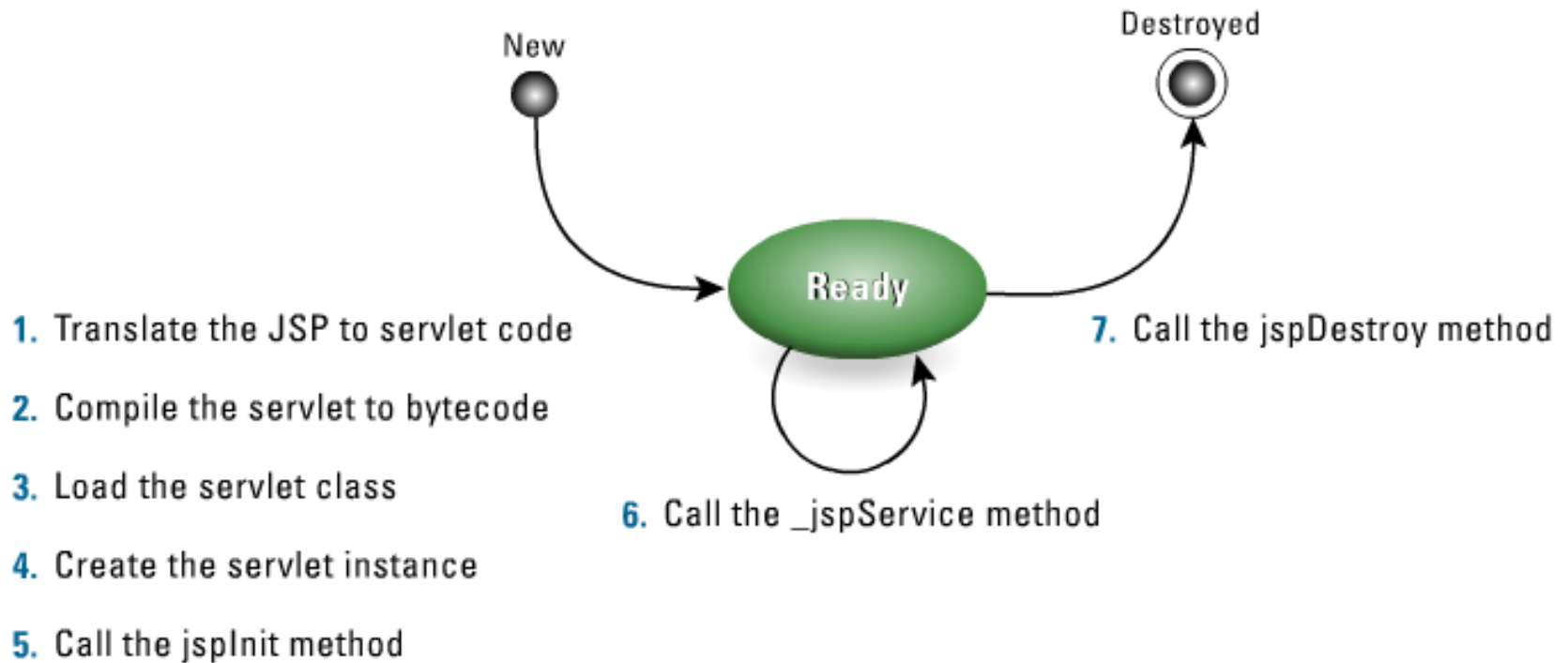
```
1 <html>
2
3 <head>
4 <title>Example JSP Page</title>
5 </head>
6
7 <body bgcolor='white'>
8
9 <b>Table of numbers squared:</b>
10
11 <table border='1' cellspacing='0' cellpadding='5'>
12 <tr><th>number</th><th>squared</th></tr>
13 <% for ( int i=0; i<10; i++ ) { %>
14 <tr><td><%= i %></td><td><%= (i * i) %></td></tr>
15 <% } %>
16 </table>
17
18 </body>
19
20 </html>
```

Servlet הוא JSP

```
jEdit
File Edit Search Markers Folding View Utilities Macros Plugins Help
10 public class HelloServlet extends HttpServlet {
11
12     private static final String DEFAULT_NAME = "World";
13
14     public void doGet(HttpServletRequest request,
15                       HttpServletResponse response)
16         throws IOException {
17         generateResponse(request, response);
18     }
19
20     public void doPost(HttpServletRequest request,
21                       HttpServletResponse response)
22         throws IOException {
23         generateResponse(request, response);
24     }
25
26     public void generateResponse(HttpServletRequest request,
27                                 HttpServletResponse response)
28         throws IOException {
29
30         String name = request.getParameter("name");
31         if ( (name == null) || (name.length() == 0) ) {
32             name = DEFAULT_NAME;
33         }
34
35         response.setContentType("text/html");
36         PrintWriter out = response.getWriter();
37
38         out.println("<HTML>");
39         out.println("<HEAD>");
40         out.println("<TITLE>Hello Servlet</TITLE>");
41         out.println("</HEAD>");
42         out.println("<BODY BGCOLOR='white'>");
43         out.println("<B>Hello, " + name + "</B>");
44         out.println("</BODY>");
45         out.println("</HTML>");
46
47         out.close();
48     }
}
```

```
jEdit
File Edit Search Markers Folding View Utilities Ma
1 <%! private static final String DEFAULT_NAME = "World"; %>
2
3 <html>
4
5 <head>
6 <title>Hello JavaServer Page</title>
7 </head>
8
9 <!-- Determine the specified name (or use default) --%>
10 <%
11     String name = request.getParameter("name");
12     if ( (name == null) || (name.length() == 0) ) {
13         name = DEFAULT_NAME;
14     }
15 %>
16
17 <body bgcolor='white'>
18
19 <b>Hello, <%= name %></b>
20
21 </body>
22
23 </html>
24
```

JSP Page Life Cycle



JSP / Java

■ בעיות:

- כעת במקום לזהם קוד Java ב HTML או מזהמים קוד JSP ב Java
- לתחביר הסטנדרטי של JSP יכולות מוגבלות (בעיקר שילוב קוד HTML ו Java)
- יש צורך בטכנולוגיה שתאפשר פיתוח לוגיקה ב Java ושיבוצה בתוך קובצי JSP ללא זיהום ה JSP

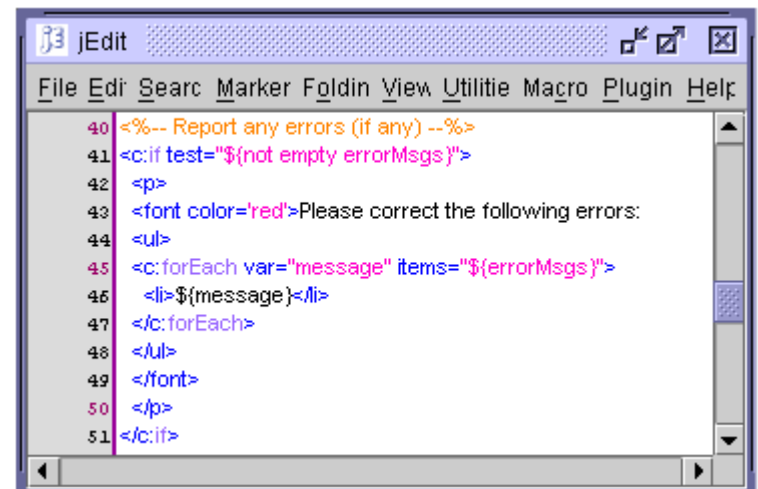
■ תגיות JSP מתיימרות להשיג מטרה זו בדיוק – המתכנתת יכולה להגדיר בעצמה תגיות JSP

- התגיות ימומשו בשפת Java בקובץ נפרד
- הדבר הופך את JSP לשפת תכנות לכל דבר עם מבני בקרה, שימוש ב Java Beans ועוד...

JSP Custom Tags

```
<%-- Report any errors (if any) --%>
<%
    // Retrieve the errorMsgs from the request-scope
    List errorMsgs = (List) request.getAttribute("errorMsgs");
    if ( (errorMsgs != null) && !errorMsgs.isEmpty() ) {
%>
<p>
<font color='red'>Please correct the following errors:
<ul>
<%
    Iterator items = errorMsgs.iterator();
    while ( items.hasNext() ) {
        String message = (String) items.next();
%>
<li><%= message %></li>
<%
    } // END of while loop over errorMsgs
%>
</ul>
</font>
</p>
<%
    } // END of if errorMsgs is not empty
%>
```

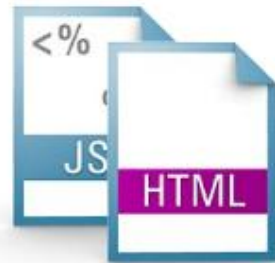
■ בעזרת תגיות משתמש
ניתן להשתמש
במעצבים ללא ידע
בתכנות לצורך כתיבת
עמודי JSP



```
jEdit
File Edit Search Marker Foldin View Utilitie Macro Plugin Help
40 <%-- Report any errors (if any) --%>
41 <c:if test="{not empty errorMsgs}">
42 <p>
43 <font color='red'>Please correct the following errors:
44 <ul>
45 <c:forEach var="message" items="{errorMsgs}">
46 <li>${message}</li>
47 </c:forEach>
48 </ul>
49 </font>
50 </p>
51 </c:if>
```

Job Roles

■ ארכיטקטורת Web Applications מחזקת את הצורך לחלק את משימות הפיתוח בין מתכנתים שונים



- Web Designer – Creates view elements (web pages and layouts)



- Web Component Developer – Creates controller elements



- Business Component Developer – Creates model elements



- Data Access Developer – Creates database access elements

Copyright 2005 Sun Microsystems, Inc. All Rights Reserved.

מה חסר ב Servlet API?

- Binding request parameters to Java types
- Validating data
- Making calls to business logic
- Making calls to the data layer
- Rendering presentation layer (HTML, and so on)
- Providing internationalization and localization



Frameworks

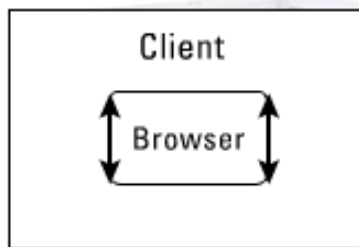
Web Application Framework Goal

- ביצוע אוטומטי של פעילויות חוזרות שכיחות
- הגדרת פתרון ארכיטקטוני
- ב- Struts2 למשל: רעיון ה- Action
- האם אפשר ללא Framework?
- בוודאי אולם יש לך מספר השלכות
- לכתובת framework in house יש חסרונות כלכליים ועסקיים

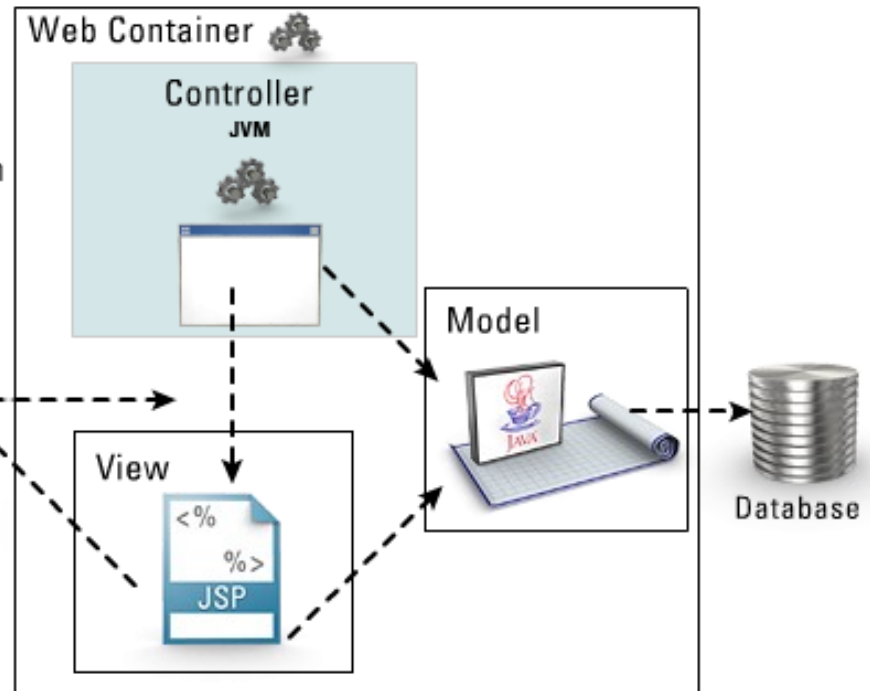
Model 2 Framework

Popular Model 2 Frameworks:

- Struts, part of the Jakarta project
- JavaServer™ Faces technology from Sun
- Velocity from Apache



<< HTTP >>



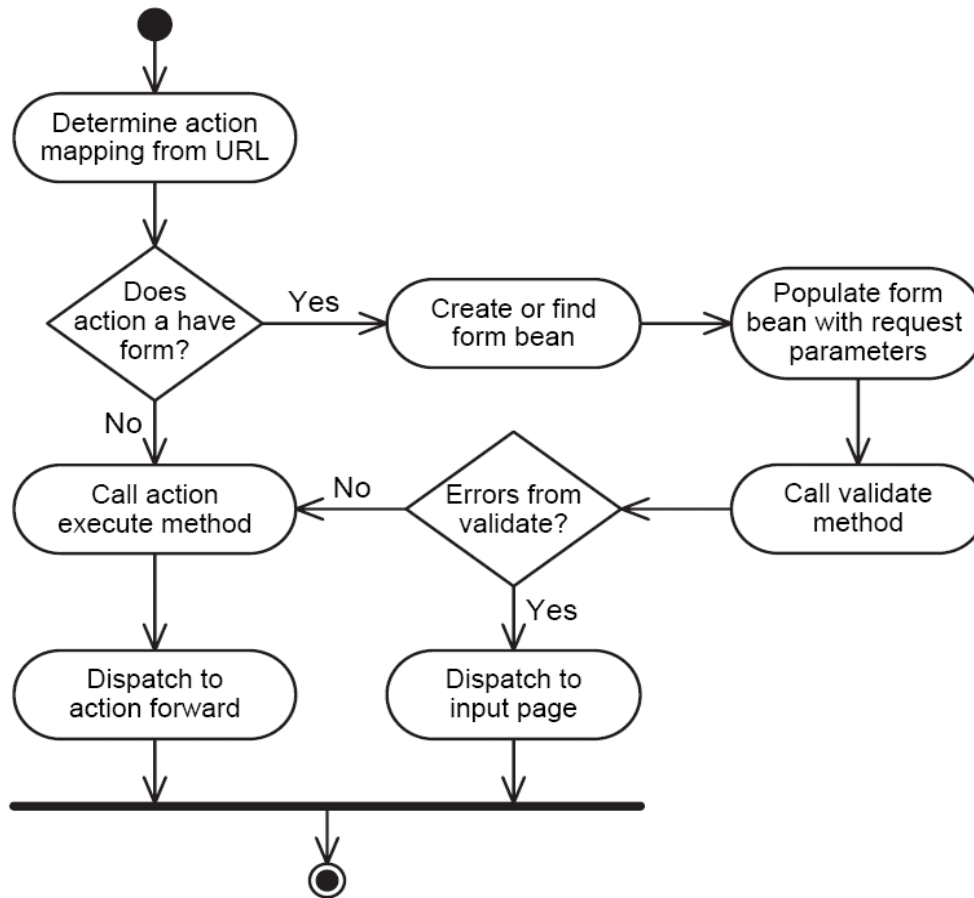
This combination provides the most advantages:

- Fast
- Powerful
- Easy to create
- Cross-platform
- Scalable
- Maintainable

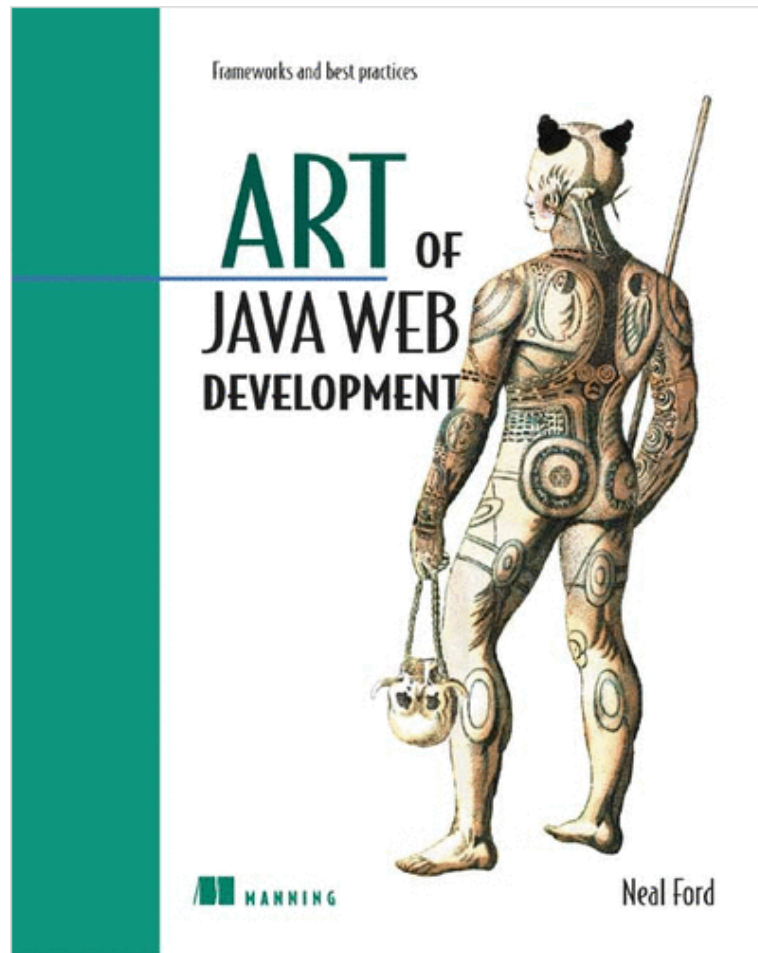
MVC Using Struts

- מימוש רעיון ה MVC בסביבת Web Container מכיל פעולות שחוזרות על עצמן בכל יישום:
 - ה- Controller מבצע את בדיקת תקינות הנתונים
 - ה- Controller מפנה את הבקשה עם תוצאות הבדיקה לעמוד ההצגה המתאים
- היינו מעוניינים לבצע אוטומטיזציה של פעולות אלו, בדרך זו ה Controller יתרכז רק בלוגיקה העסקית וההפרדה בין View ובין Controller תאכף ע"י הסביבה
- Struts Framework מספקת את המחלקה Action אשר עוטפת את פעולת מחלקות ה- Controller
 - בדרך זו ה Controller אפילו לא יודע את שם המחלקה שאליה מופנית ההודעה לאחר ביצוע הלוגיקה העסקית

Struts activity diagram



Java Web Frameworks



קיימות כמה Frameworks המספקות תשתית לפיתוח יישומי אינטרנט מעל Servlets

ה Frameworks השונות מסייעות לאכוף הרגלי עבודה נכונים (כגון הפרדה בין מודל ותצוגה)

הספר

Art of Java Web Development
Struts, Tapestry, Commons, Velocity, JUnit, Axis, Cocoon, InternetBeans, WebWork
Neal Ford

סוקר חלק מהסביבות השונות, עומד על ההבדלים בינהן ומשווה בין קוד "ידני" ובין הקוד שנכתב בעזרתן

כלים וסביבות

- כאשר טכנולוגיה הופכת לפופולרית פועלים מספר כוחות:
 - כח אדם פחות מיומן צריך **כלים תומכים** כדי להשתלב במאמץ הפיתוח
 - מתגלים **שימושים חדשים** לטכנולוגיה אשר לא נתמכים בה היטב מכיוון שלא נלקחו בחשבון בתחילה
- כדי לענות על הדרישה מתפתחים כלים וסביבות ייעודיות אשר מנסים לפשט את משימות הפיתוח
- הצד השני של המטבע: כאשר טכנולוגיה הופכת מורכבת מדי היא מהווה **חסם כניסה** למתכנתים חדשים – וקמות במקומה שפות פשוטות **שמחליפות** את הטכנולוגיה הקיימת
- לאיזו טכנולוגיה עולה דרוש פיתוח של כלים חדשים היום?
- אילו יכולות על הכלים לספק?

JSP vs. ASP

- JSP אינו רעיון חדש
- קדם לו ASP של מייקרוסופט (1996) שהיה מבוסס על שפות תסריט כגון VBScript (בעיקר) ו-Jscript

```
<html>  
<body>  
<%= "Hello World!" %>  
</body>  
</html>
```

- החל מ-2002 מחליפה ASP.net את ASP עם תמיכה מלאה של כל שפות .NET.

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

PHP

PHP: Hypertext Preprocessor ■

התחילה כשפת סקריפט פשטנית בשנת 1995 ■

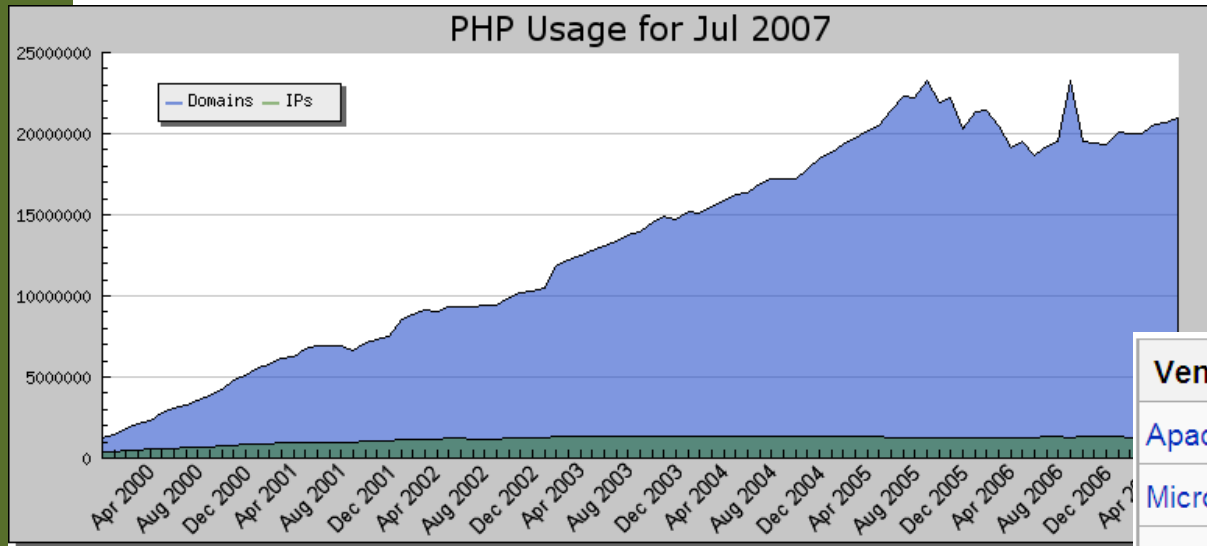
כרגע פעילות גרסאות 4 ו-5 אשר מוסיפות לשפה יכולות OO (כמעט מלאות), תמיכה ב XML, ב web services ועוד...

ניתנת להורדה כחלק מחבילת xAMP ■

(**A**pache, **M**ySQL and **P**HP) ■

כנראה השפה הפופולרית ביותר לצד השרת (40% משרתי Apache :<http://www.appideas.com/technology.html>) ■

PHP, ASP, JSP and Web Servers



Vendor	Product	Web Sites Hosted	Percent
Apache	Apache	83,206,564	50.22%
Microsoft	IIS	58,540,275	35.33%
Google	GWS	10,075,991	6.08%
Oversee	Oversee	1,926,812	1.16%
lighttpd	lighttpd	1,495,308	0.9%
nginx	nginx	1,018,503	0.61%
Others	-	9,432,775	5.69%
Total	-	165,696,228	100.00%

<http://trends.builtwith.com/framework/PHP>

Scripting languages and Server Side

- בחלק מהמקרים הלוגיקה של השרת הדינאמי פשוטה למדי ומסתכמת בשליפה והצגה של מידע מתוך בסיס נתונים
- במקרים כאלה ניתן לשקול שימוש בשפות תסריט שאינן יעודיות לשימוש בשרתים
- דוגמאות:
 - Perl
 - Ruby
 - python

Battle of the Frameworks!

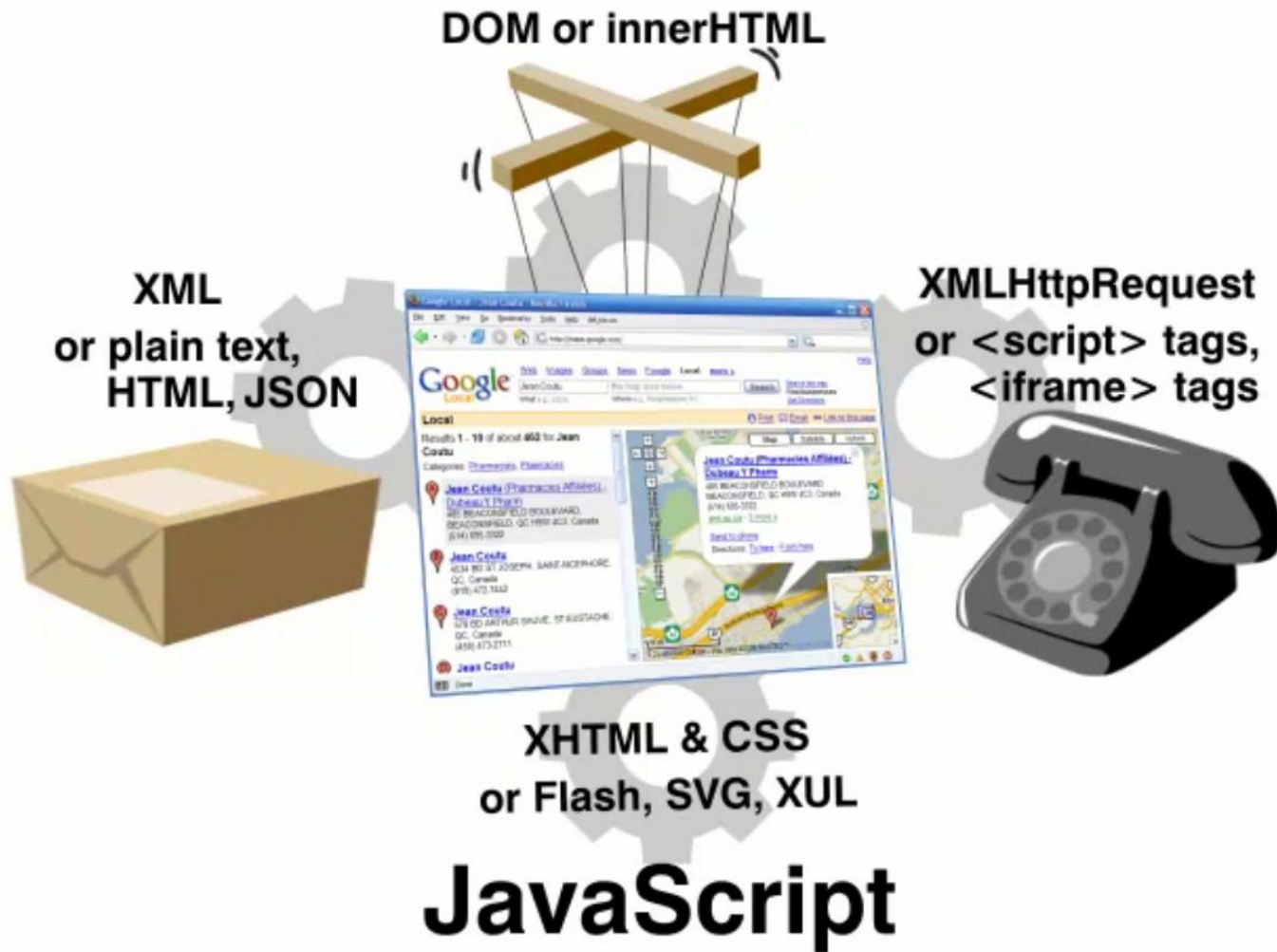
- A good framework makes web development easier, faster, and more consistent, but which one should you choose?
 - **Seam** or **Struts 2**?
 - **Zend Framework** or **CakePHP**?
 - **Laszlo** or **Flex 3**?
 - **jQuery** or **Prototype**?
 - **Wicket** or **GWT**?
 - **Rails 2** or **Merb**?
 - **Adobe AIR** or **Silverlight**?
 - Maybe you're even considering **Second Life**
- Not only you should choose a framework you should choose the underlying technology
- (<http://phillyemergingtech.com/abstracts.php#frameworks>)

Higher Level Frameworks and tools

- <http://wordpress.org/>
- <http://www.joomla.org/>
- <http://drupal.org/>

CLIENT SIDE WEB PROGRAMMING

חוב משבוע שעבר - AJAX



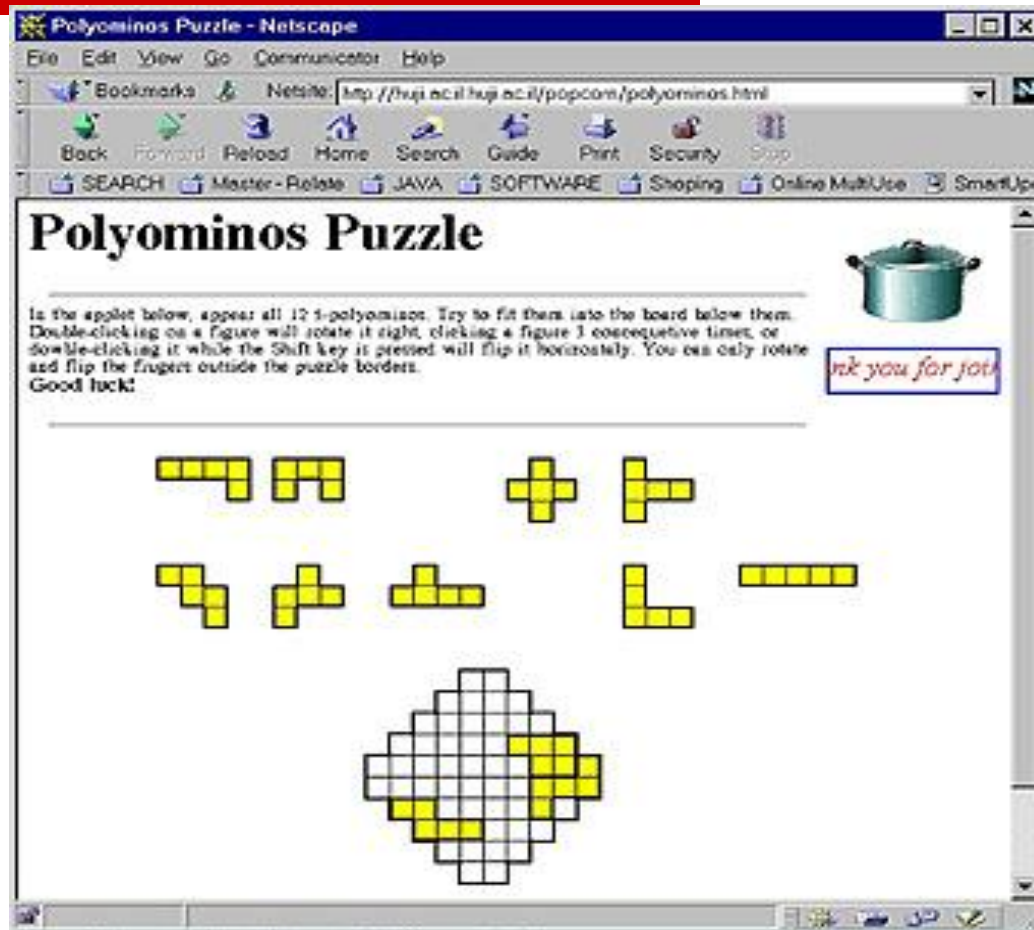
אבל אני יודע Java...

- שיבוץ תוכניות Java בעמודי HTML

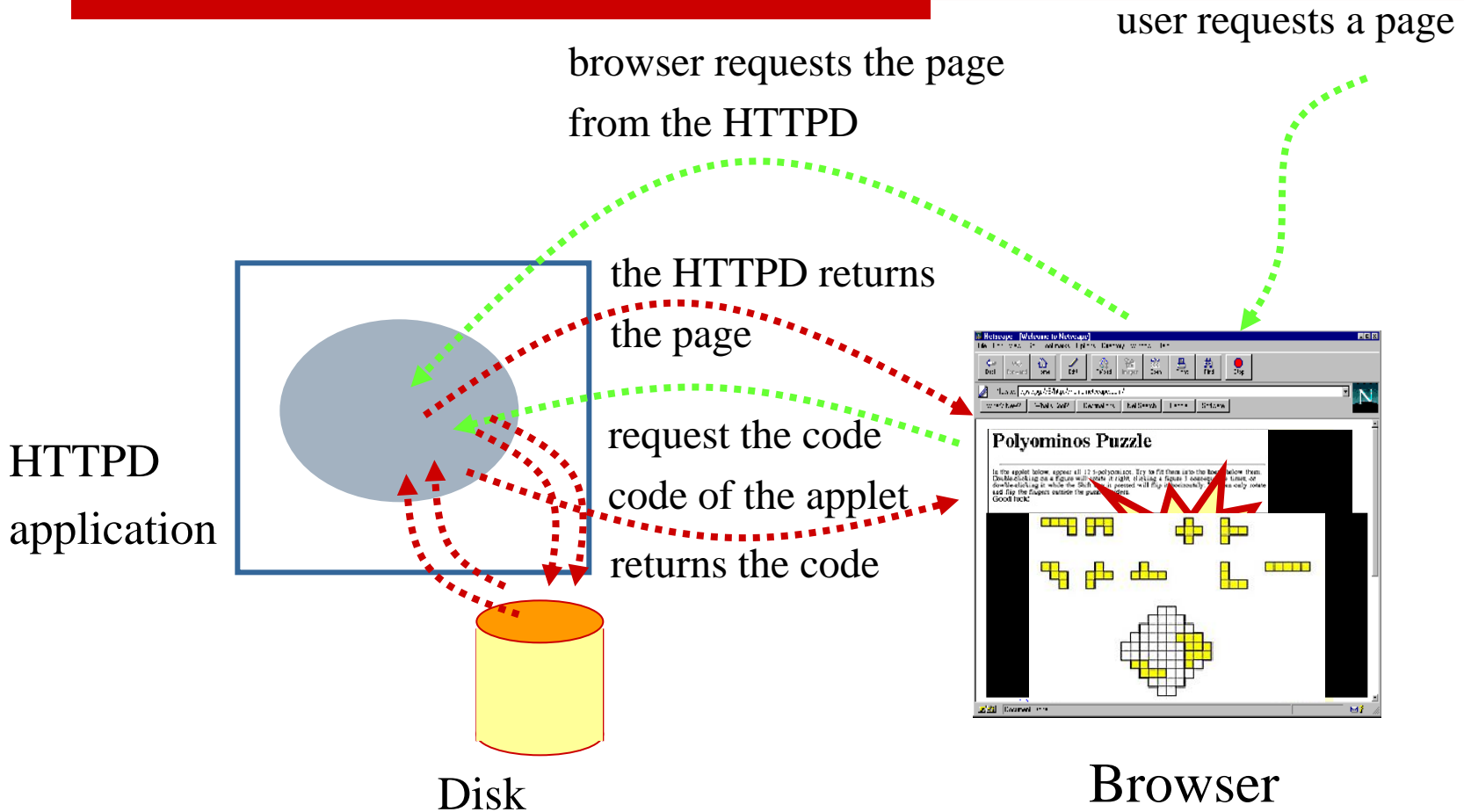
- נוסף לדפדפן תפקיד נוסף, אם עמוד התשובה מכיל תוכנית Java
 - הדפדפן יטען אותה **אוטומטית** מהשרת
 - הדפדפן יריץ אותה
 - הדפדפן יציג את הפלט שלה (בדרך כלל גרפי) בתוך העמוד

- לשם נוחות העבודה הוגדרה מחלקת Java בשם `Applet` (יישומון) המתקשרת עם הדפדפן בצורה נוחה

Applets - Mobile Code



Mobile Code



דוגמא

נכתוב מחלקה היורשת מ Applet

נשבץ אותה בתוך עמוד HTML

```
<html>
  <head>
    <title> Example Applet </title>
  </head>
  Look at the applet I wrote !!!
  <body>
    <applet CODE=AppletSubclass.class
            WIDTH=80 HEIGHT=60>
    </applet>
  </body>
</html>
```

נריץ את היישומון ע"י הצגת העמוד בדפדפן

דוגמא - המשך

יישומון לדוגמא:

```
import java.awt.*;
import java.applet.Applet;

public class AppletSubclass extends Applet {

    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.fillOval(20,20,200,50);
        g.setColor(Color.black);
        g.drawString("Hello from AppletSubclass", 50, 50);
    }
}
```

איך מריצים?

פותחים את דף ה HTML ע"י הדפדפן

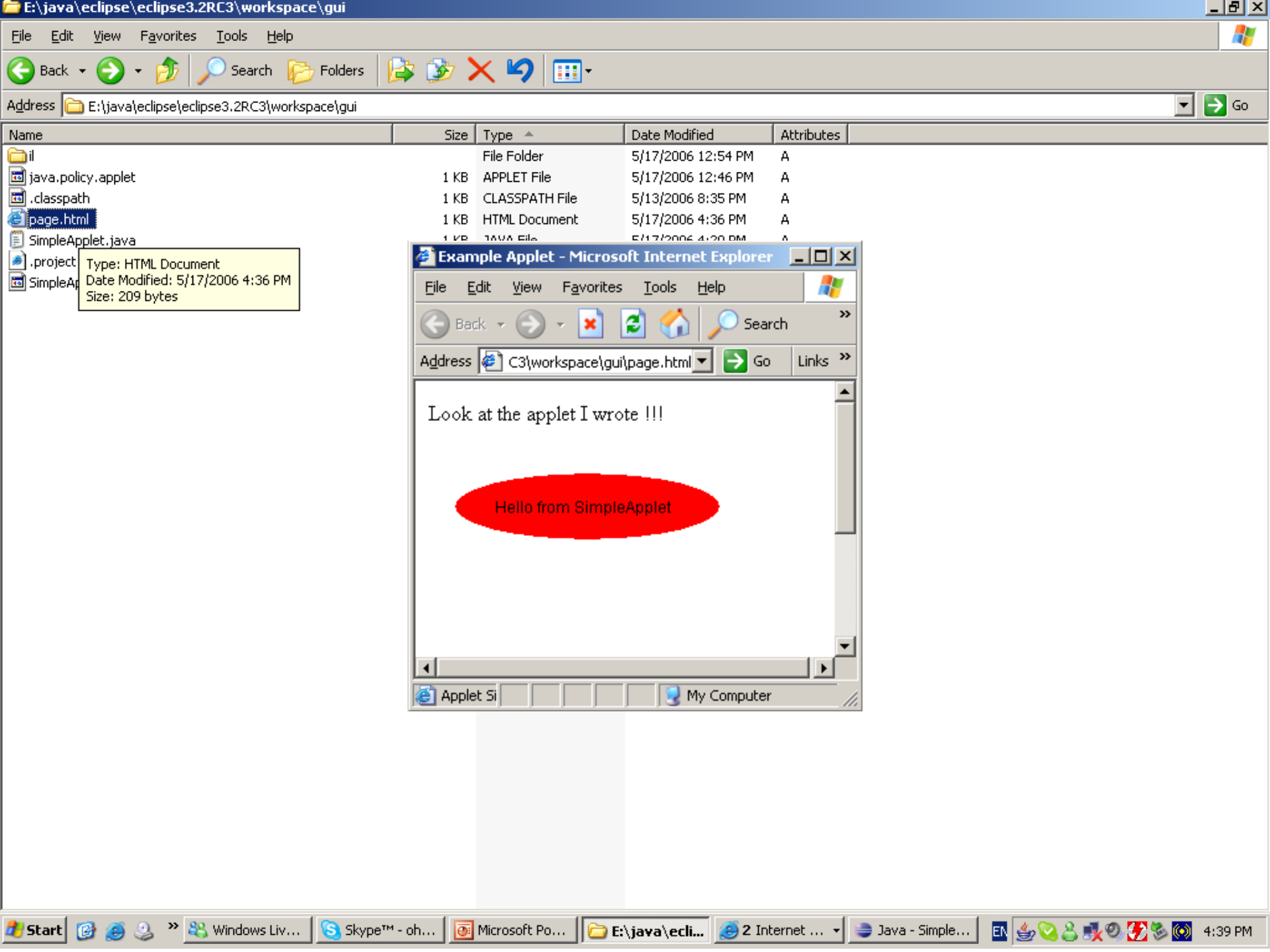
או

פותחים את דף ה HTML ע"י `appletviewer`

או

מריצים את המחלקה כ `Java Applet` מתוך
Eclipse

■ שפותח בשבילנו `appletviewer`



Name	Size	Type	Date Modified	Attributes
------	------	------	---------------	------------

il		File Folder	5/17/2006 12:54 PM	A
java.policy.applet	1 KB	APPLET File	5/17/2006 12:46 PM	A
.classpath	1 KB	CLASSPATH File	5/13/2006 8:35 PM	A
page.html	1 KB	HTML Document	5/17/2006 4:36 PM	A
SimpleApplet.java	1 KB	JAVA File	5/17/2006 4:30 PM	A

Type: HTML Document
Date Modified: 5/17/2006 4:36 PM
Size: 209 bytes

Example Applet - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search

Address C3\workspace\gui\page.html Go Links

Look at the applet I wrote !!!

Hello from SimpleApplet

Applet Si My Computer

Applet Viewer: SimpleApplet.cl...

Applet



Hello from SimpleApplet

Applet started.

```

Demo.java | SimpleApplet.java
import java.awt.*;

public class SimpleApplet extends Applet {

    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.fillOval(20,20,200,50);
        g.setColor(Color.black);
        g.drawString("Hello from SimpleApplet", 50, 50);
    }
}

```

Outline

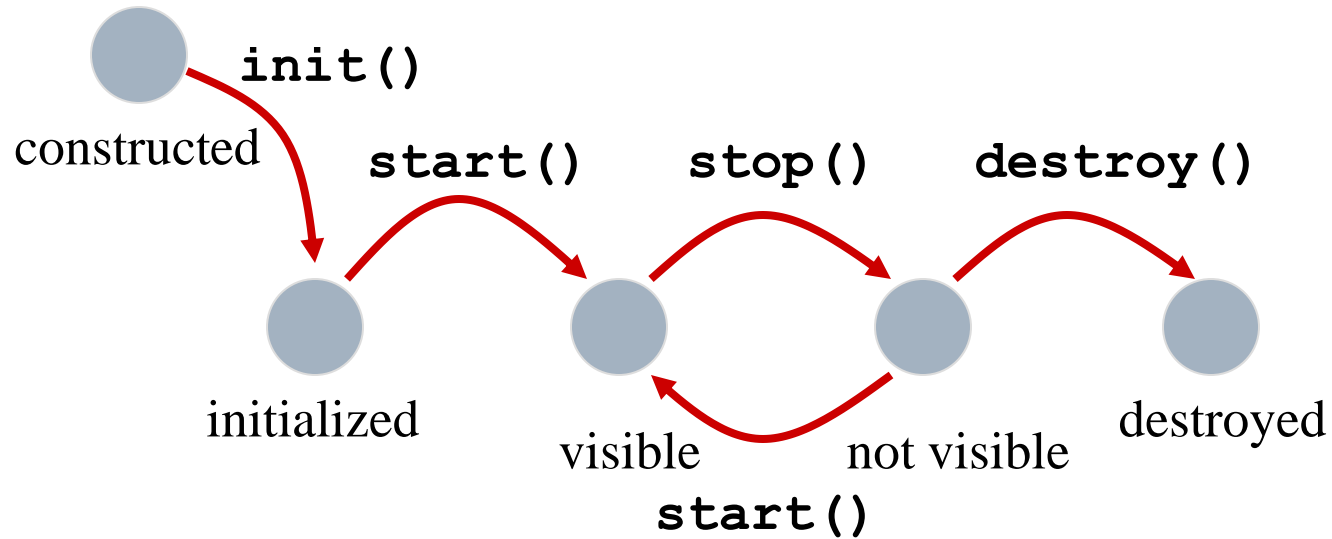
- import declarations
- SimpleApplet
 - paint(Graphics)

- Open Type Hierarchy F4
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Build Path
- Source Alt+Shift+S
- Refactor Alt+Shift+T
- Import...
- Export...
- References
- Declarations
- Refresh F5
- Run As**
 - 1 Java Applet Alt+Shift+X, A
 - Run...
- Debug As
- Team
- Compare With
- Replace With
- Restore from Local History...
- Properties Alt+Enter

Javadoc Declaration Console

let [Java Applet] H:\Program Files\Java\jre1.5.0_06\bin\javaw.exe (May 17, 2006 4:20:10 PM)

מחזור החיים של יישומון



הדפדפן כ- container

- המחלקה `Applet` היא דוגמא לרעיון ה- `Component`
- `Component` (רכיב) – היא מחלקה אשר לא מיועדת להרצה בצורה עצמאית, אלא מיועדת להיות מורצת ע"י מחלקה מסוימת אחרת המכונה מיכל או סביבה (`container, framework`)
- הדבר מפשט את כתיבת ה `Component` מכיוון שאנו יודעים את האילוצים שתחתם יקראו השרותים השונים
- זהו אחד הרעיונות המרכזיים בפיתוח תוכנה מודרני והוא מהווה בסיס למבחר טכנולוגיות בשנים האחרונות:
COM, CORBA, EJB, Servlets ■

חסרונות ה Applet

- וירוסים – sandbox model
- היישומונים מבוססים על מחלקות ה GUI של Java
 - למה דווקא GUI ?
 - למה דווקא AWT/Swing ?
- אם נרצה, ניתן לעקוף אילוצים אלו ב SWT ע"י מחלקות המקשרות בין AWT ו- SWT אולם זהו פתרון עבור SWT בלבד
- כדי לפתור את הבעיה באופן כללי פיתחה Sun את טכנולוגית ה Java Web Start – הפותרת אילוצים אלו

Java Web Start

הדפדפן טוען את המחלקה אבל לא מריץ אותה

■ מקל על בעיות התאימות של הדפדפנים

■ קיים מנגנון לאימות הספק ע"י certification authority

■ קיים מנגנון למתן הרשאות גישה נוספות

■ קיים מנגנון להטמנת קבצים ומעקב אחר גרסאות

לצורך הגדרת הרשאות פיתחה Sun פרוטוקול מבוסס

XML בשם JNLP (Java Network Launching Protocol)

המאפשר לשלוט בקלות על פעולות התוכנית

Java Web Start

דוגמאות:

■ סביבת פיתוח UML מבוססת Web Start:

<http://argouml.tigris.org/>

■ תוכניות דוגמא:

<http://java.sun.com/javase/technologies/desktop/javawebstart/demos.html>

דוגמאות מודרכות:

■ הסבת תוכנית סייר קבצים standalone ל Web Start:

<http://www-128.ibm.com/developerworks/opensource/library/os-jws/>

■ מדריך הפצת תוכנה עם JNLP ו WebStart באתר Sun:

<http://java.sun.com/developer/technicalArticles/Programming/jnlp/>

Adobe Technologies

חברת Adobe מציעה מבחר טכנולוגיות פרטיות
מבוססות Flash לצד הלקוח

מאוד פופלרית

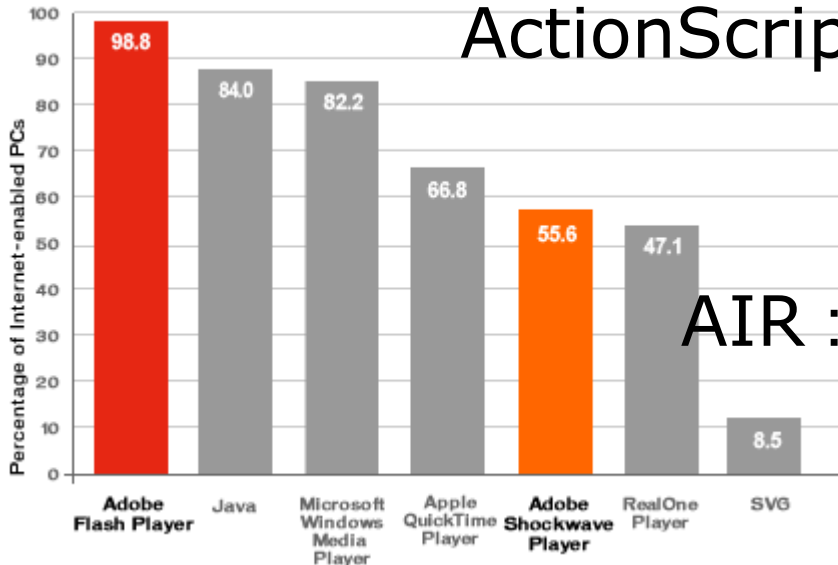
הורחבה במספר כיוונים:

הוספת לוגיקה תכנותית: ActionScript

פיתוח יישומים: Flex

ספריות, כלי פיתוח

דואליות Web/Desktop AIR:



Microsoft Silverlight

- Microsoft Silverlight is a cross-browser, cross-platform, and cross-device plug-in for delivering the next generation of .NET based media experiences and rich interactive applications for the Web.

(<http://silverlight.net/>)

Google Web Toolkit (GWT)

- Google Web Toolkit makes it easier to write high-performance AJAX applications
- You write your front end in the **Java** programming language and **GWT compiles your source** into highly optimized JavaScript.
- Writing web apps today is a tedious and error-prone process. You spend 90% of your time working around **browser quirks**, and JavaScript's lack of modularity makes sharing, testing, and reusing AJAX components difficult and fragile

(<http://code.google.com/webtoolkit/>)

Google Web Toolkit (GWT)

- **High performance JavaScript.** GWT produces AJAX apps that:
 - Load faster than hand-written JavaScript apps
 - Use smaller, more compact, cacheable code
 - Automatically support IE, Firefox, Mozilla, Safari, and Opera
 - Use the browser's "back" button correctly
- **Better development tools.** Since you're writing in Java, you can use:
 - IDEs that you love like Eclipse, IntelliJ, and NetBeans
 - Full-featured debugging, with variable watches and breakpoints
 - Unit tests (based on JUnit) both in a debugger and in a browser
- **Google APIs and reusable UI components.** GWT comes equipped with useful libraries:
 - Built-in UI components serve as cross-browser building blocks for your app
 - RPC helps with client-server interaction
 - JavaScript Native Interface (JSNI) simplifies integrating GWT code with existing JavaScript code

Google's Dual Mode

- Applications: Picasa + Picasa Web
- Infrastructure Technology: Google Gears
 - LocalServer Cache and serve application resources (HTML, JavaScript, images, etc.) locally
 - Database Store data locally in a fully-searchable relational database
 - WorkerPool Make your web applications more responsive by performing resource-intensive operations asynchronously
- <http://code.google.com/apis/gears/>