



תכנות אינטרנט

Web Programming

אוהד ברזילי

בית הספר למדעי המחשב
אוניברסיטת תל אביב

על סדר היום

האינטרנט ■

טכנולוגיות צד לקוח ■

טכנולוגיות צד שרת ■

Internet for dummies

- האינטרנט היא רשת מחשבים (= מחשבים + חוטים)
- לכל מחשב ניתנת כתובת (IP Address) המורכבת מ-4 מספרים בין 0 ל-255
- כל תוכנית הרצה במחשב מקבלת מספר שלוחה (port) בין 0 ל-65,535
- ניתן ליצור קשר עם מחשב המחובר לרשת ע"י ציון כתובתו ופניה אליו בהודעה המתאימה לפי הפרוטוקול
- בדרך כלל ע"י שימוש בתוכנית המממשת את הפרוטוקול

Internet != Web

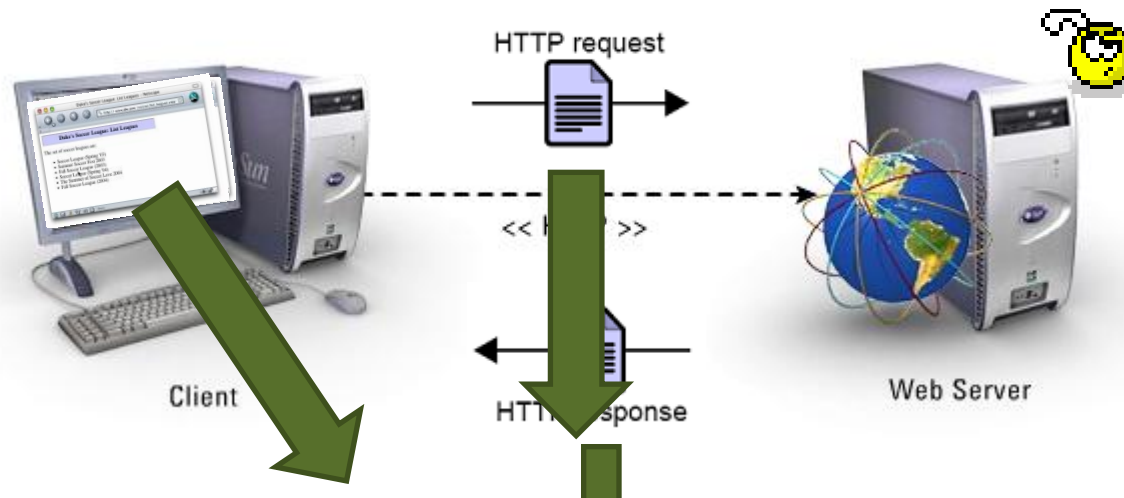
■ האינטרנט (תקשורת בין-רשתית) הומצא בשנות ה-60 של המאה הקודמת וחידושו בכך שהוא מאפשר לשני מחשבים לתקשר ביניהם בין אם הם מחוברים ישירות ובין אם הם מחוברים דרך מחשבי ביניים

■ יישומים: email, ftp, VoIP, Web

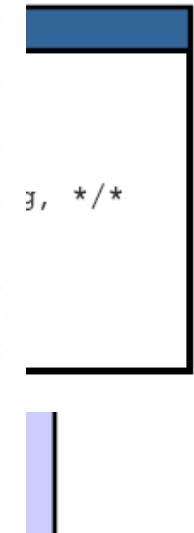
■ ה-**Web** הומצא רק ב-1989. זוהי רשת של מסמכים (אתרים) המכילים קישורים (לינקים) זה לזה. המסמכים עשויים להיות מאוחסנים במחשבים שונים המחוברים לאותה אינטרנט, וניתנים לצפייה בעזרת תוכנה מיוחדת (הדפדפן)

■ רוב ההתייחסויות בעברית לאינטרנט הם בעצם התייחסויות ל**Web** (מה המקבילה בעברית?)

פרוטוקול HTTP בגלישה בדפדפן



Request line
Request headers
Body
M
body



תפקידי הדפדפן

■ תקשורת

- ניהול פרוטוקול HTTP

■ הצגה על המסך (rendering)

- HTML

- CSS

■ הרצת קוד

- ע"י virtual machine (java, flash, Silverlight)

- הרצת קוד JavaScript

Overview of Client Side Web Technologies

HTML

שפת תגיות להצגת עמודי אינטרנט בדפדפן ◀

עמוד HTML הוא עץ עם שורש יחיד אשר לצאצאיו יכולים להיות מאפיינים (attributes, כולל ארועים) ◀

מרכיבי העמוד ◀

link, base, script, title, meta : **head** ◀

body : כותרות, פסקאות, עיצוב טקסט, רשימות, טבלאות, תמונות, קישורים, טפסים ◀

```
<HTML>
  <HEAD>
    <TITLE>
      Hello World!
    </TITLE>
  </HEAD>
  <BODY>
    <H1 id="greetings">Hi</H1>
    <P>Minimal <b>hello world</b> page.</P>
  </BODY>
</HTML>
```

ניתן לציין כל מרכיב בעמוד ע"י ◀

במזהה ייחודי, כדי לאפשר

גישה אליו משפת JavaScript

או CSS

עבודה עם מסגרות ◀

עבודה עם CSS ◀



CSS

- ▶ **Cascading Style Sheets**
- ▶ Semantic tags vs. Design tags
 - ▶ `<p>`, `<h1>` vs. ``, ``
- ▶ No Model-View Separation
- ▶ <http://www.csszengarden.com/>



JavaScript

שפת תכנות מבוססת עצמים, עשויה להיות מוטמעת בקוד HTML אשר רצה בדפדפן

זוהי **שפת תכנות מלאה** עם תמיכה בתנאים, לולאות, משתנים, מערכים, מחרוזות, פונקציות, עצמים, חריגים, ביטויים רגולרים וארועי דפדפן

"מחלקות" ספרייה:

Object, String, RegExp, Number, **Function**, Math, Date, Boolean, Array, Arguments, Error

התחביר שלה פשוט ומתירני. למשל:

אין צורך להצהיר ב JavaScript על טיפוס המשתנים

Lambda notation

JSON

```
<html>
  <script>
    var x = 3;
    var y = 47;
    alert( " 3 x 47 = " + (x*y) );
  </script>
</html>
```

בגלל המאבק בין הדפדפנים השונים קיימות מספר גרסאות

Javascript 1.0, 1.1, ... (Netscape/Mozilla)

JScript 1.0, 2.0, ... (Microsoft)

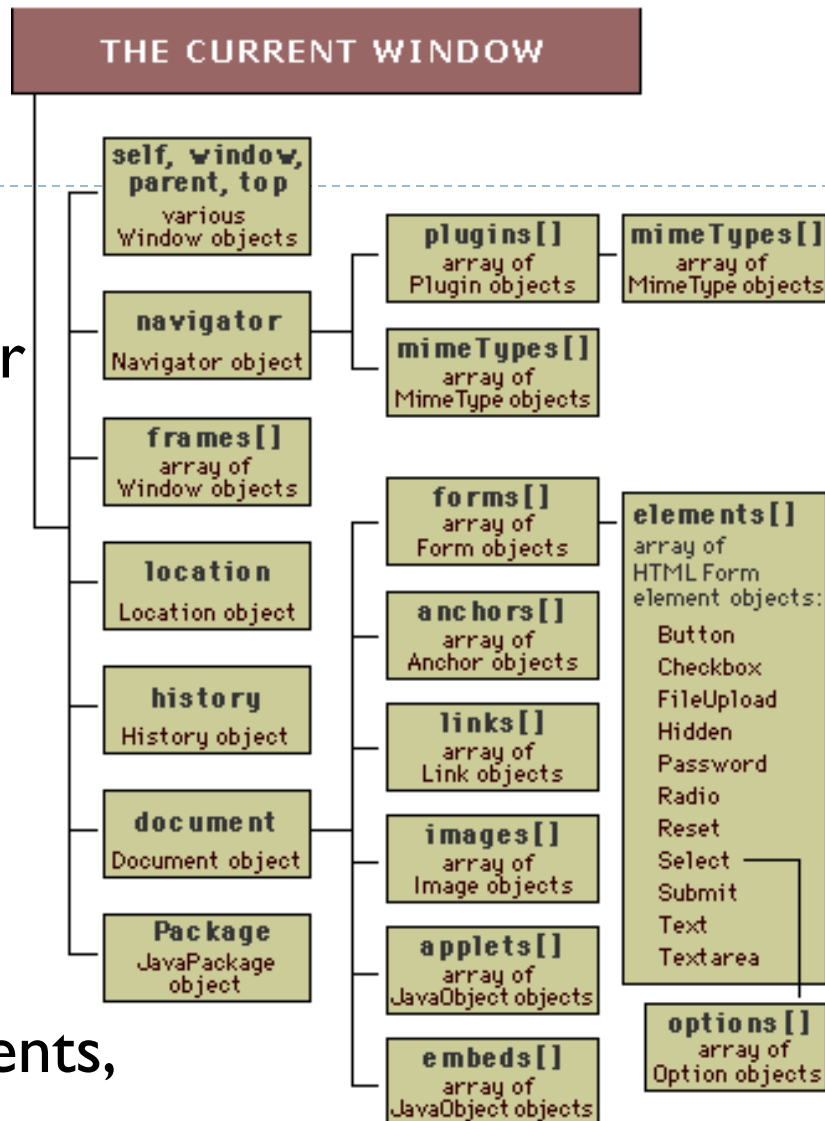
ECMAScript v1, v2, ... (International Standard)

```
<h1 onMouseOver="style.color='red' ;"
  onMouseOut="style.color='blue' ;">Hello </h1>
```



DOM

- ▶ **D**ocument **O**bject **M**odel
- ▶ JavaScript meets the browser
- ▶ Data structure and API that enables programmatic manipulation of HTML page
- ▶ Gives programmatic means to access and manipulate events, components, forms, data, style, etc.



DOM Example

```
<html>
  <head>
    <script type="text/javascript">
      function getValue() {
        var x = document.getElementById("myHeader");
        alert(x.innerHTML);
      }
    </script>
  </head>

  <body>
    <h1 id="myHeader" onclick="getValue()">This is a header</h1>
    <p>Click on the header to alert its value</p>
  </body>
</html>
```



Learn JavaScript

▶ good tutorials:

- ▶ http://www.cs.tau.ac.il/~ohadbr/systems/spring2007/ajax_sivan/ (heb)
- ▶ <http://updatepanel.net/2007/12/01/javascript-tutorial-videos-at-yahoo-ui-library-yui-theater/> (video)
- ▶ http://www.hunlock.com/blogs/Essential_Javascript_-_A_Javascript_Tutorial

▶ Nice references and examples:

- ▶ http://www.w3schools.com/js/js_examples.asp
- ▶ <http://www.quirksmode.org/js/contents.html>



מה ללמוד JavaScript?
בשביל מה למדתי Java?

המחלקה URL

גלישה באינטרנט מתוך קוד Java

כדי להקל על כתיבת לקוחות בפרוטוקולים סטנדרטיים (http, ftp וכיו"ב) מספקת Java את המחלקה `java.net.URL`

```
// Create some URL objects
URL url = null, url2 = null, url3 = null;
try {
    url = new URL("http://www.oreilly.com"); // An absolute URL
    url2 = new URL(url, "catalog/books/javanut4/"); // A relative URL
    url3 = new URL("http:", "www.oreilly.com", "index.html");
} catch (MalformedURLException e) { /* Ignore this exception */
}

// Read the content of a URL from an input stream
InputStream in = url.openStream();
```

המחלקה URL

■ ע"י שימוש במחלקה `URLConnection` ניתן לקבל פרטים נוספים על ההודעה (HTTP Message Parsing) כגון:

```
// For more control over the reading process, get a URLConnection object
URLConnection conn = url.openConnection();

// Now get some information about the URL
String type = conn.getContentType();
String encoding = conn.getContentEncoding();
Date lastModified = new Date(conn.getLastModified());
int len = conn.getContentLength();

// If necessary, read the contents of the URL using this stream
InputStream in = conn.getInputStream();

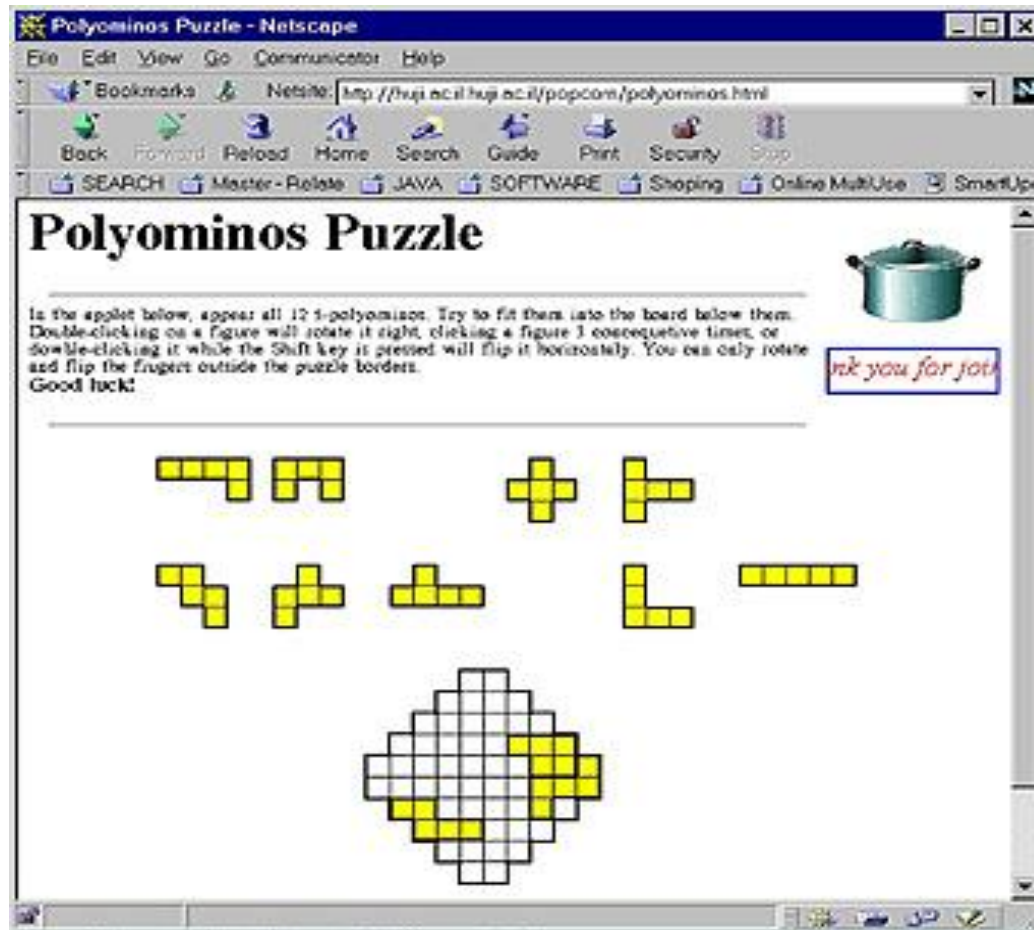
Scanner s = new Scanner(in);
while(s.hasNextLine())
    System.out.println(s.nextLine());
}
```


אבל לאיפה נעלם הדפדפן?

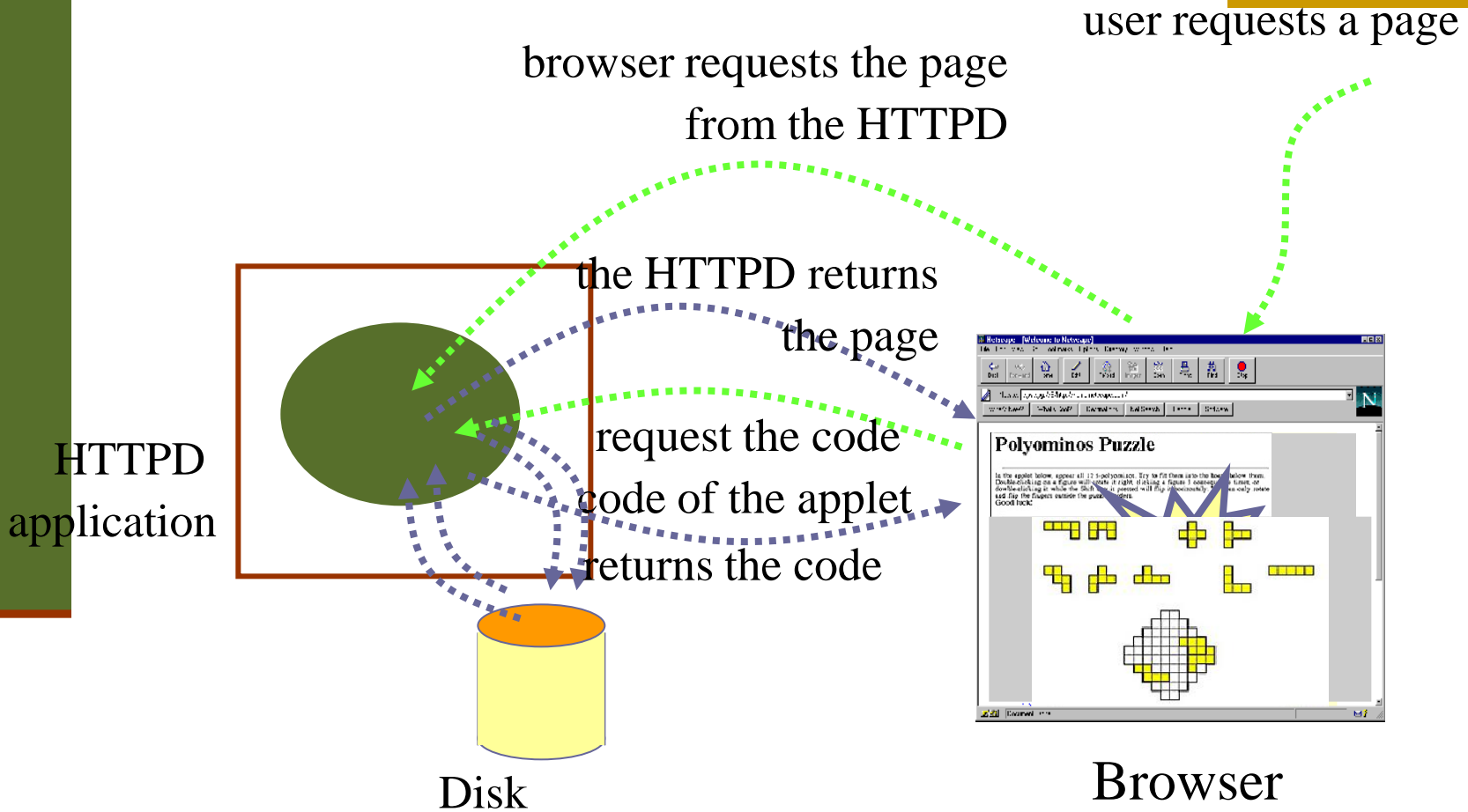
Java Applets

- שיבוץ תוכניות Java בעמודי HTML
- אם עמוד התשובה מכיל תוכנית Java
 - הדפדפן טוען אותה **אוטומטית** מהשרת
 - הדפדפן מריץ אותה
 - הדפדפן מציג 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

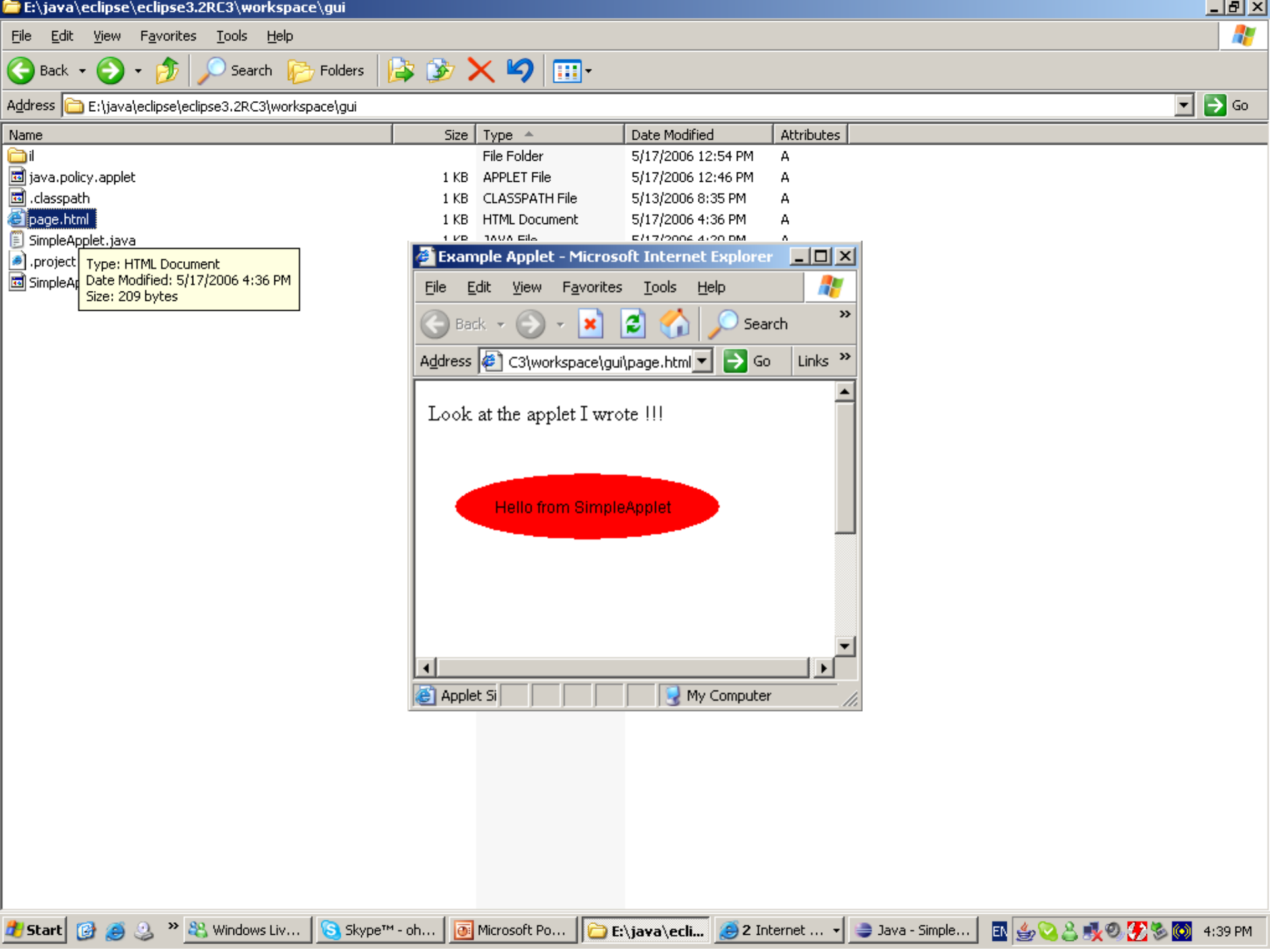
או

■ 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
.project		Type: HTML Document		
SimpleApplet.html		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

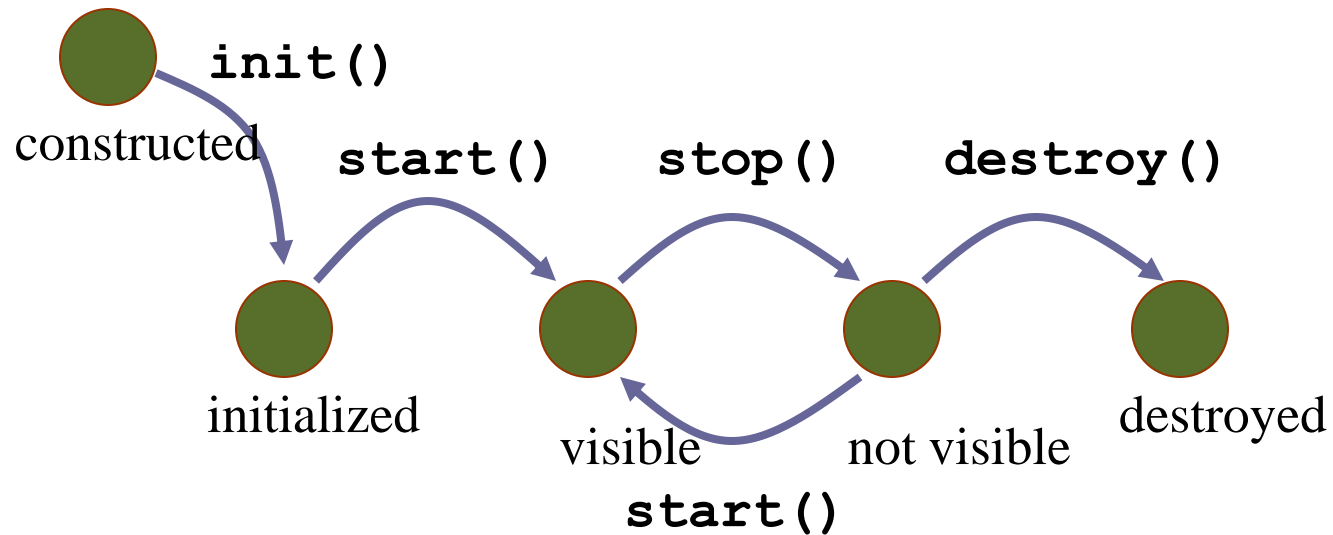
- import declarations
- SimpleApplet
 - paint(Graphics)

- 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?

■ האם גם לא-מתכנתים יכולים לעבוד בקלות עם Applets?

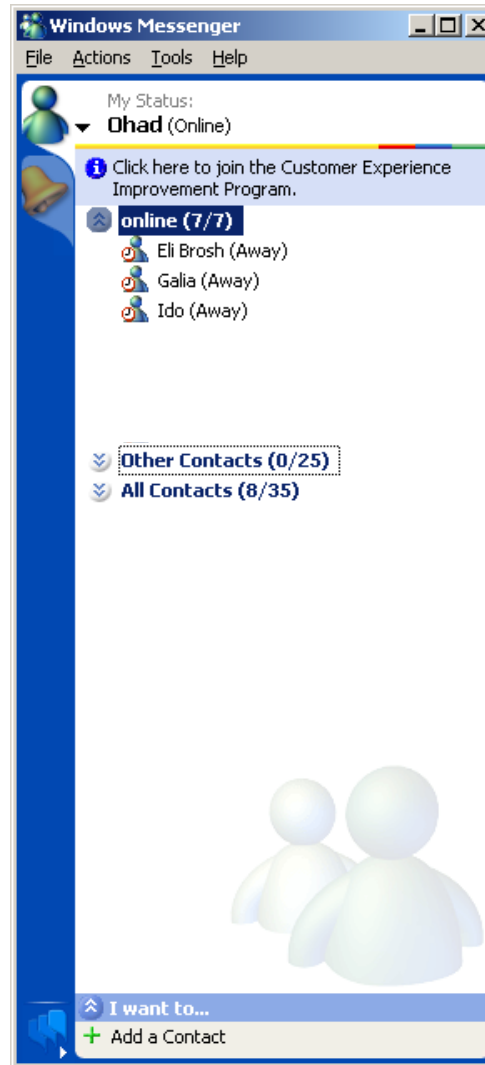
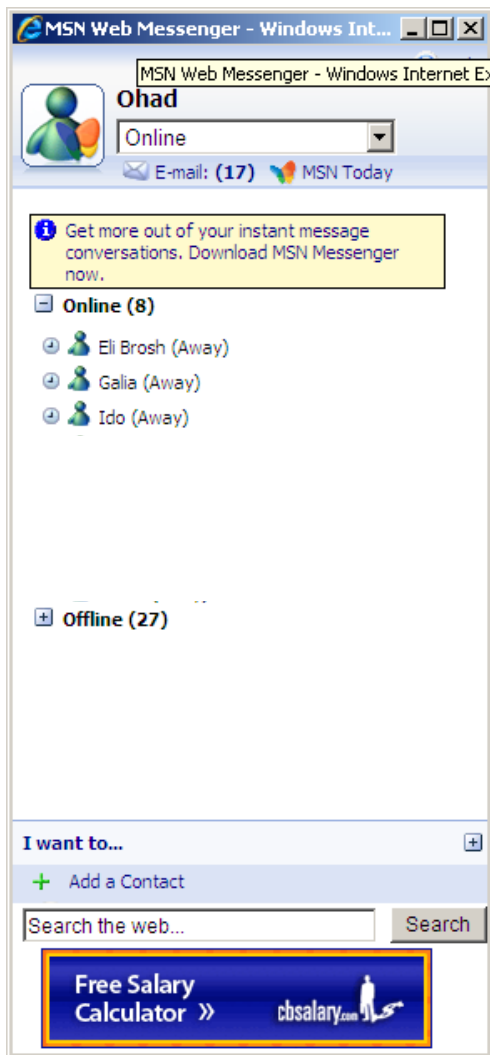
■ מה עם כלי עיצוב?

■ בשל חסרונות אלו (ואולי גם אחרים) טכנולוגיות מתחרות מחליפות את

השימוש ב Applets:

■ Flash, Silverlight, HTML5

הדפדפן הוא הלקוח



בשנים האחרונות
מתפתחת מגמה של
פרוטוקולי תקשורת
מעל HTTP גם
ביישומים שאינם
כוללים גלישה
באינטרנט

מצאו את ההבדלים
בין לקוחות של MSN
Messenger



Server Side Web Programming

Servlets

דינאמיות בצד השרת

■ גישה משלימה סוברת שהשרת הוא זה שצריך להיות דינאמי

■ הרעיון:

■ נכתוב תוכנת מחשב שתתחזה לשרת אינטרנט שיעשה
מניפולציה על פורמט הודעת GET - במקום המיועד להזנת שם
הקובץ, הוא יקבל שם מחלקה

■ השרת יריץ את המחלקה ואת הפלט שלה יחזיר בתוך הודעת
HTTPResponse

■ כך הופך הדפדפן למעין "מסוף טיפוש" – כל הלוגיקה מבוצעת
בשרת. ותפקידו רק להציג אותה על המסך

בעיה: לכתוב שרת כזה זה די מסובך

■ טיפול בהבטי תקשורת:

■ יצירת שקע והמתנה ללקוחות

■ מקביליות:

■ יצירת Thread נפרד לטיפול בלקוחות נכנסים (accept)

■ ניהול מספר מחלקות באותו שרת:

■ הפניית הבקשות למחלקה המתאימה

■ אתחול המופעים לפי דרישה

■ שמירת מידע רלוונטי בין קריאות (session, context)

ממש מסובך...

- טיפול בפרוטוקול HTTP
 - ניתוח כותרות ה HTTP (request)
 - ניתוח את המחרוזת במקום המיועד לשם הקובץ וחילוץ שם המחלקה המבוקשת ואת שמות המשתנים וערכיהם (לפי התווים: '?', '&', '=', '/')
 - חילול כותרות ה HTTP (response)
- מיפוי מרחב השמות הלוגיים והפיזיים
- בקרת תצורה
- ניהול עומסים, אבטחה, התאוששות מפילות...

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?)

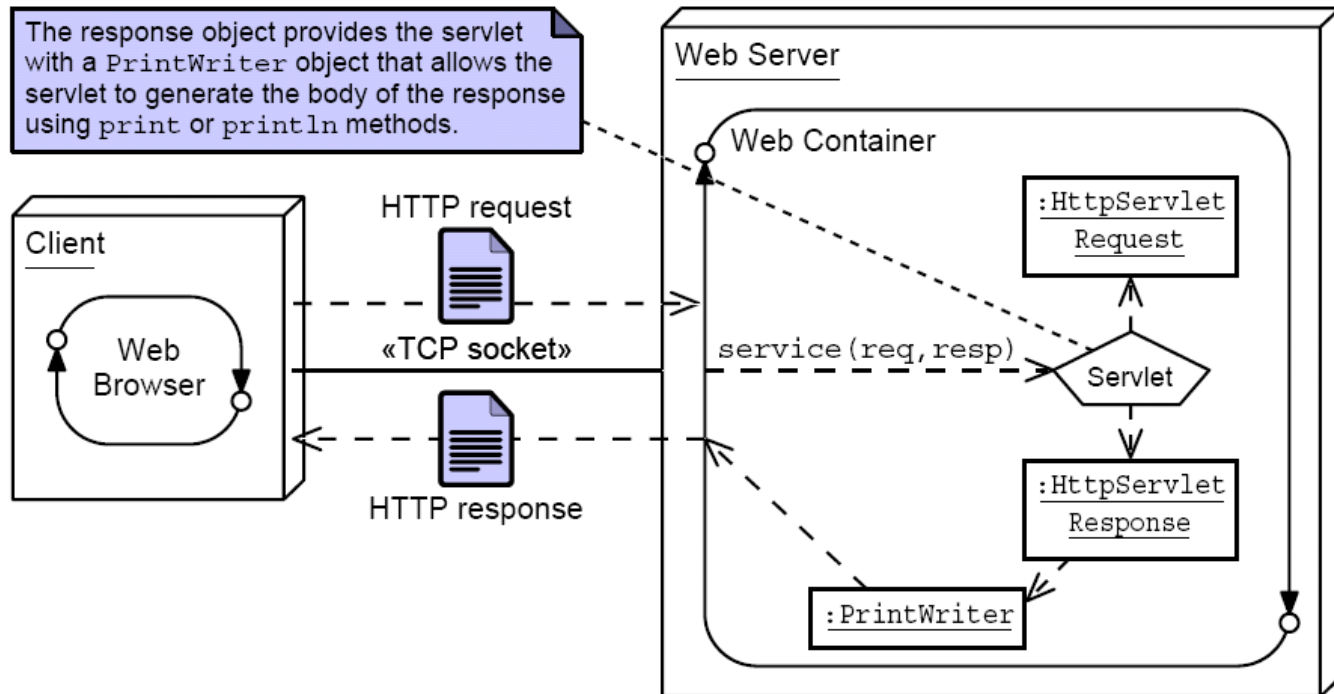
אשר מממשות כל אחת לוגיקה עסקית משלה

■ 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 a code editor titled 'jEdit' showing HTML code for a form. The right window is a Mozilla browser titled 'Duke's Soccer League: Add a New League' showing the rendered form.

HTML Code (Left Window):

```
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 (Right Window):

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

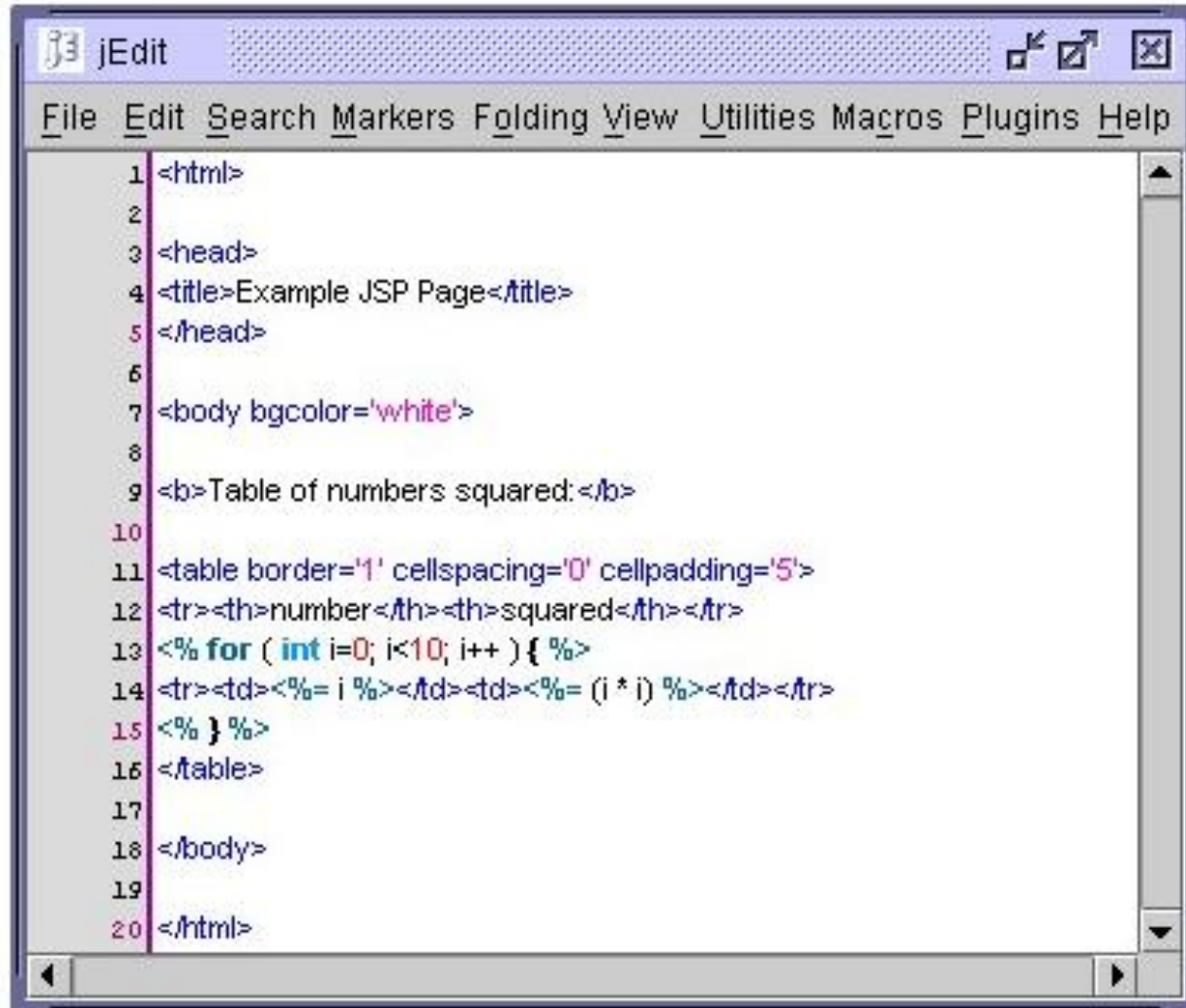
■ שכתוב קישוריות (URL Rewriting)

- 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

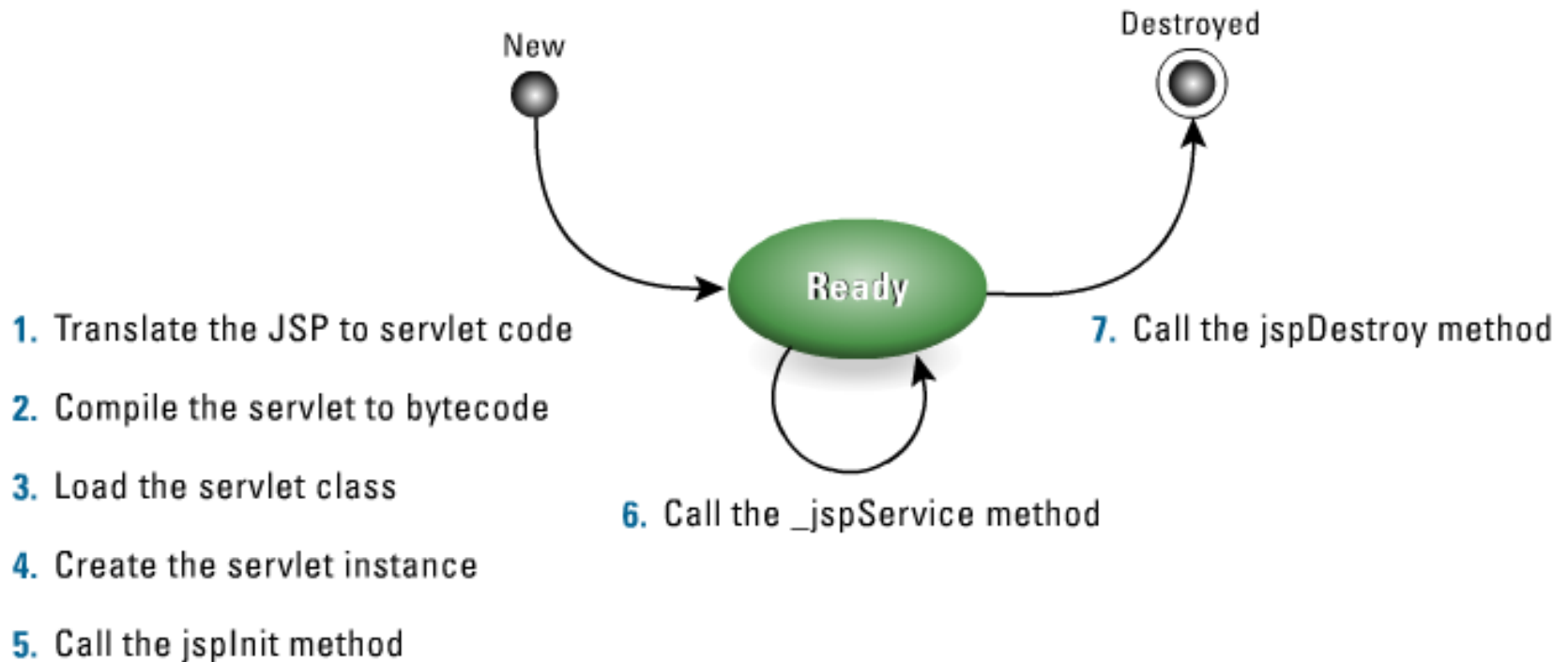
```
10 public class HelloServlet extends HttpServlet {
11
12     private static final String DEFAULT_NAME = "World";
13
14     public void doGet(HttpServletRequest request,
15                       HttpServletResponse response)
16         throws IOException {
17         generateResponse(request, response);
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

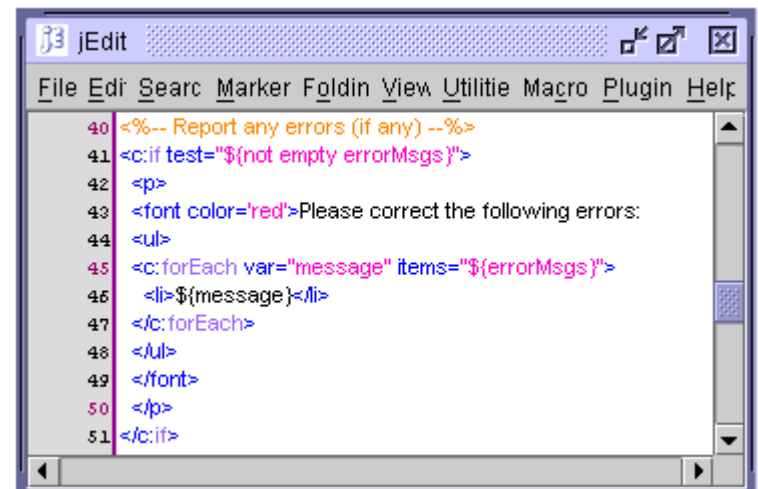
■ תגיות JSP מתיימרות להשיג מטרה זו בדיוק – המתכנתת יכולה להגדיר בעצמה תגיות JSP

- 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

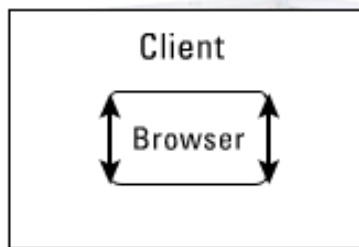


```
jEdit
File Edit Search Marker Foldin View Utilitie Macro Plugin Help
40 <!-- Report any errors (if any) --%>
41 <c:if test="{not empty errorMsgs}">
42 <p>
43 <font color='red'>Please correct the following errors:
44 <ul>
45 <c:forEach var="message" items="{errorMsgs}">
46 <li>${message}</li>
47 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

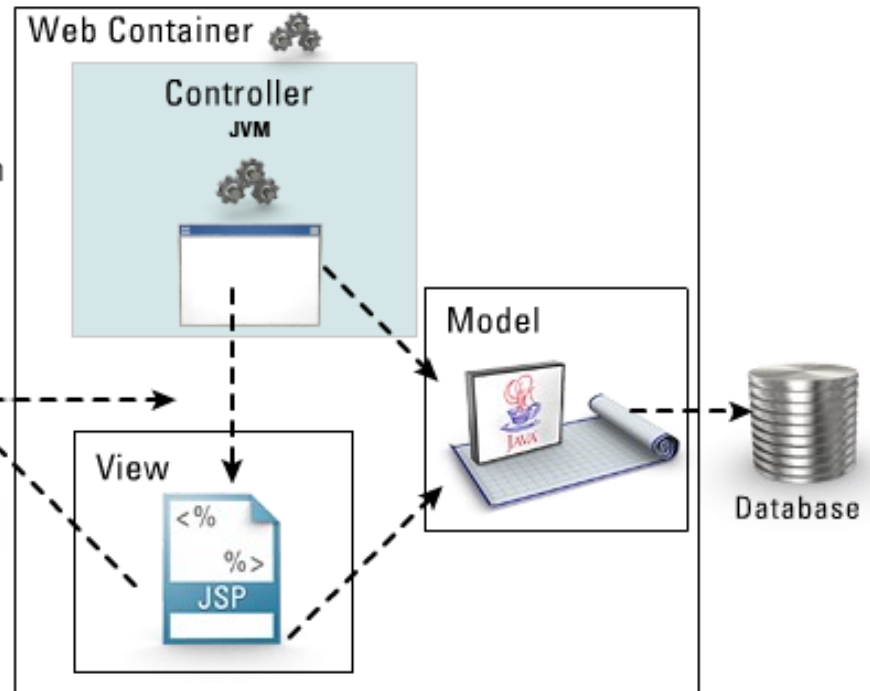

Model 2 Framework

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

אפשר גם JavaScript וגם Servlets?

שימושי JavaScript ■

אימות טפסים ■

Dynamic GUI ■

האתגר: כאשר המידע נמצא בשרת ■



The screenshot shows the Google Suggest BETA interface. The search bar contains the text "halfblood prince". Below the search bar, a list of suggestions is displayed, each with the number of results. The suggestions are:

Suggestion	Results
halfblood prince	717,000 results
halfbloodprince	802 results
halfblood chronicles	24,400 results
halfblock autochanger	2,610 results
halfblood	14,200 results
halfblock	3,790 results
halfbloodprince.info	1 result
halfblock paintball	906 results
halfblind	1,940 results
halfblack angelfish	216 results

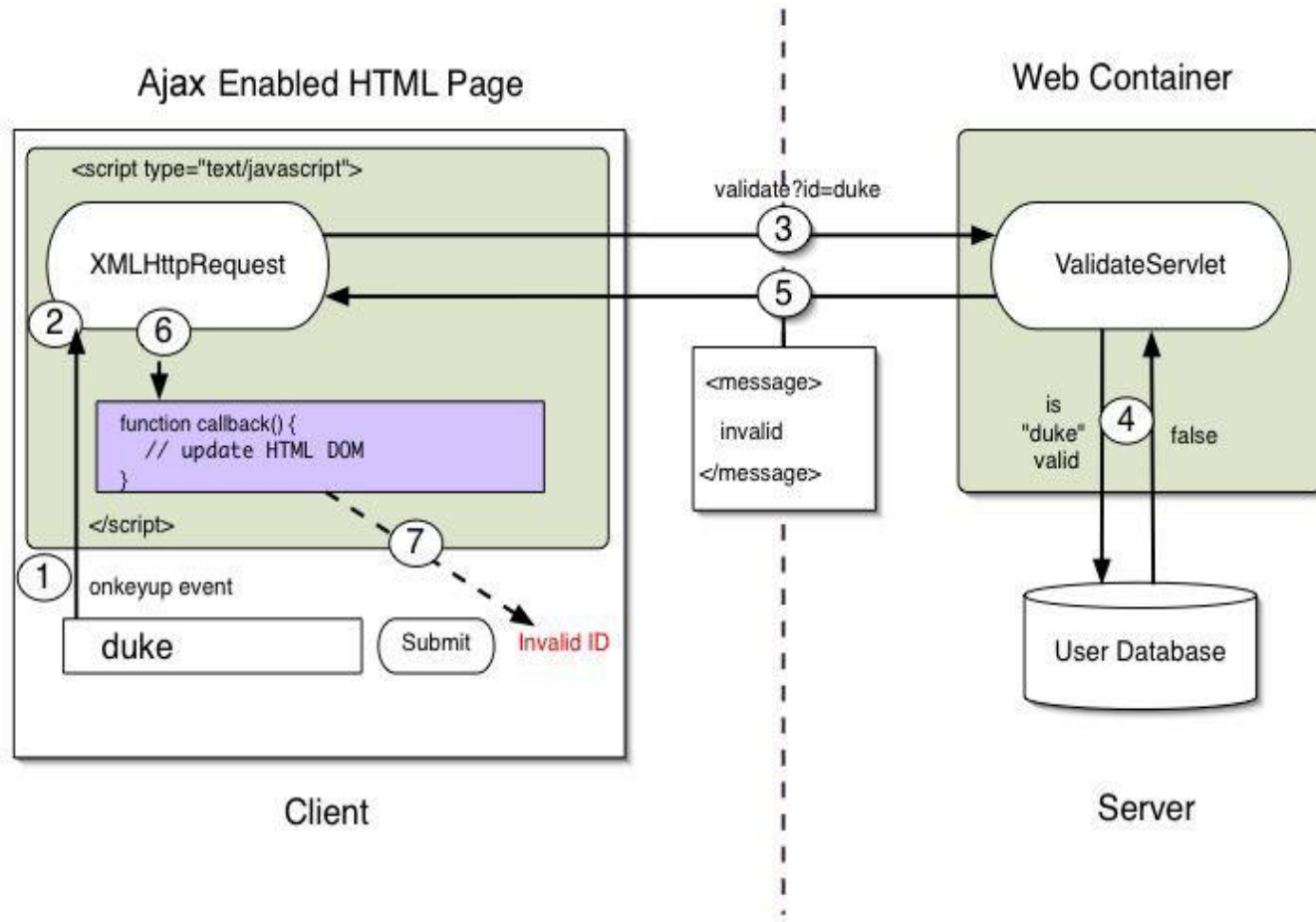
Navigation links: [Web](#), [Images](#), [Groups](#), [News](#), [Froogle](#), [Local](#), [more »](#)

Advanced Search, Preferences, Language Tools

learn more

As you type

XMLHttpRequest



Asynchronous JavaScript and XML

■ הודעות ה HTTP הסטנדרטיות עובדות במודל של request-response

■ בעזרת טכנולוגיית Ajax ניתן להוסיף "תת הודעות" בפורמט XML אשר אינן דורשות את רענון העמוד

■ בשונה מהשרת הדינאמי (שמחולל תשובה בצורה דינאמית) ומהלקוח הדינאמי (קוד JavaScript שרץ בדפדפן) Ajax מוסיף עוד מימד של דינאמיות באינטראקציה בין שרת ולקוח

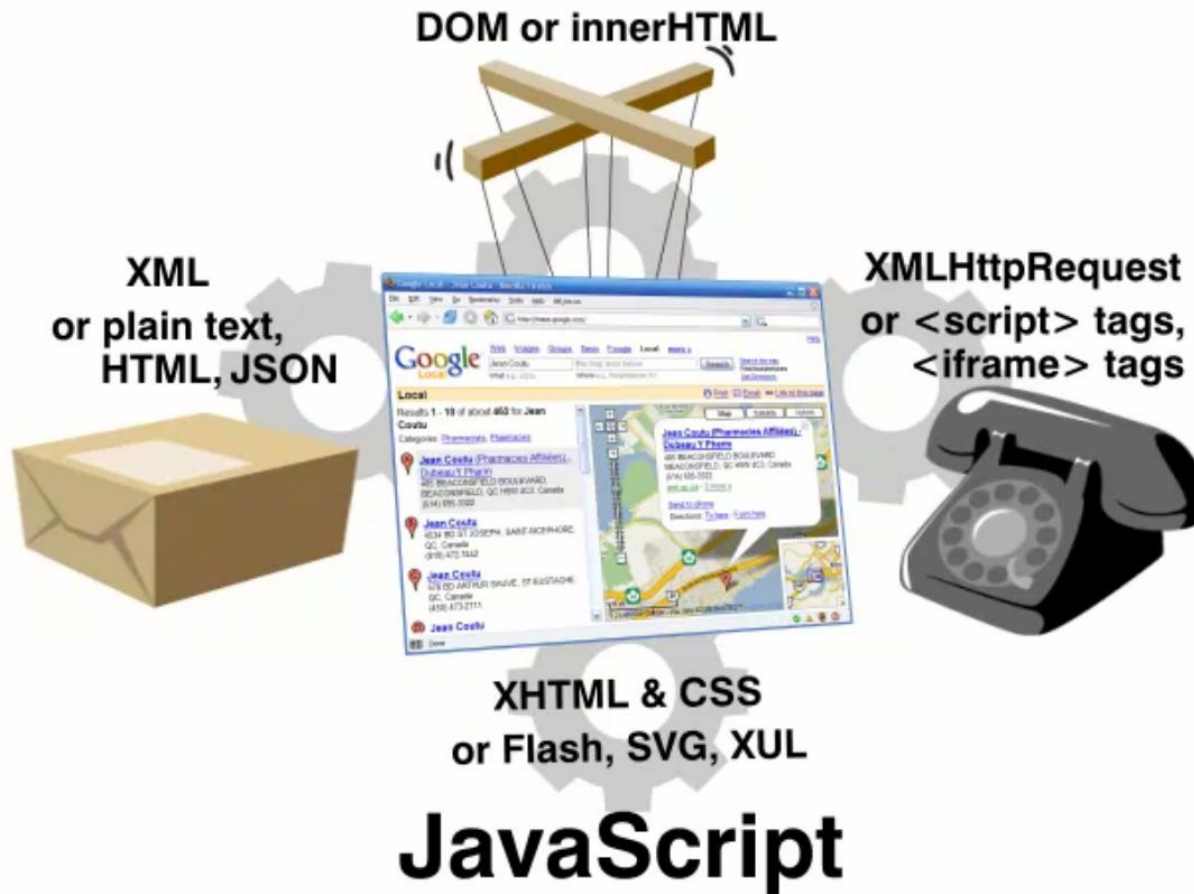
■ דוגמא טובה:

■ <http://www.nettravel.co.il/basic/flights/>

■ דוגמא לא טובה:

■ <http://msdn.microsoft.com/en-us/default.aspx>

Ajax Fundamentals



שוב פעם html ו-javascript?

■ לא חייבים

■ קיימות כמה טכנולוגיות שחוסכות כתיבה של
JavaScript

■ אחת מהן נקראת GWT

■ דוגמאות:

■ <http://code.google.com/webtoolkit/examples/>

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/>)

לסיכום

- ה- Web לא תוכנן כפלטפורמה להרצת יישומים
- בעזרת טכנולוגיות שונות (ביניהן Java) ניתן להוסיף לוגיקה (דינאמיות) הן לצד השרת והן לצד הלקוח (הדפדפן)
Applets, Servlets ■
- שפת האינטרנט היא HTML/JavaScript אולם אפשר להסתדר גם בלי
GWT ■