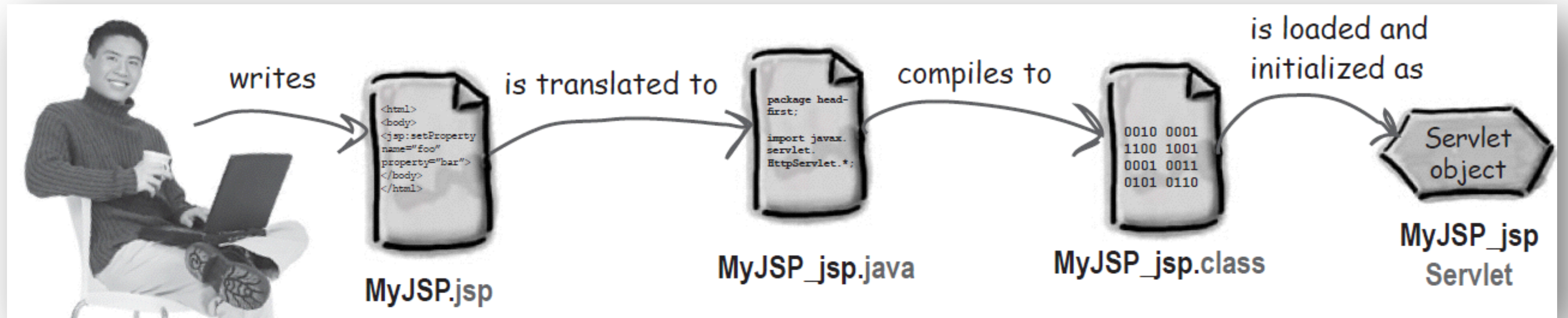


# Chapter 6

Being a JSP

# A JSP is a servlet



# How many times I've been accessed

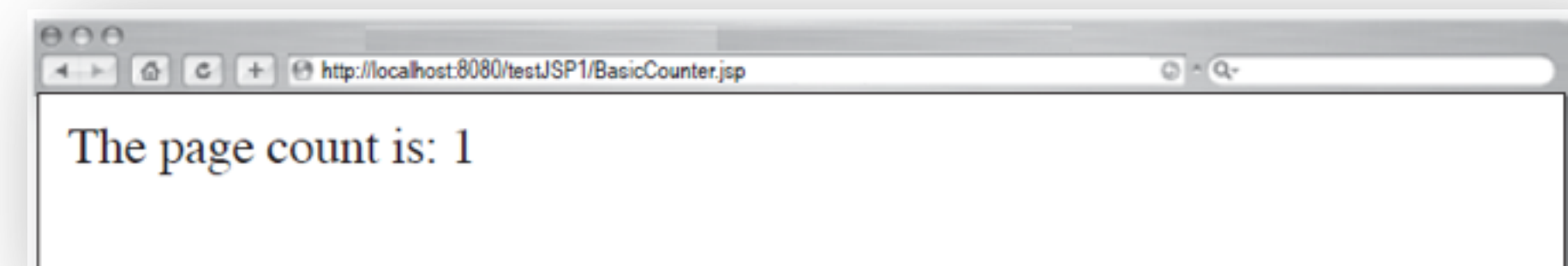
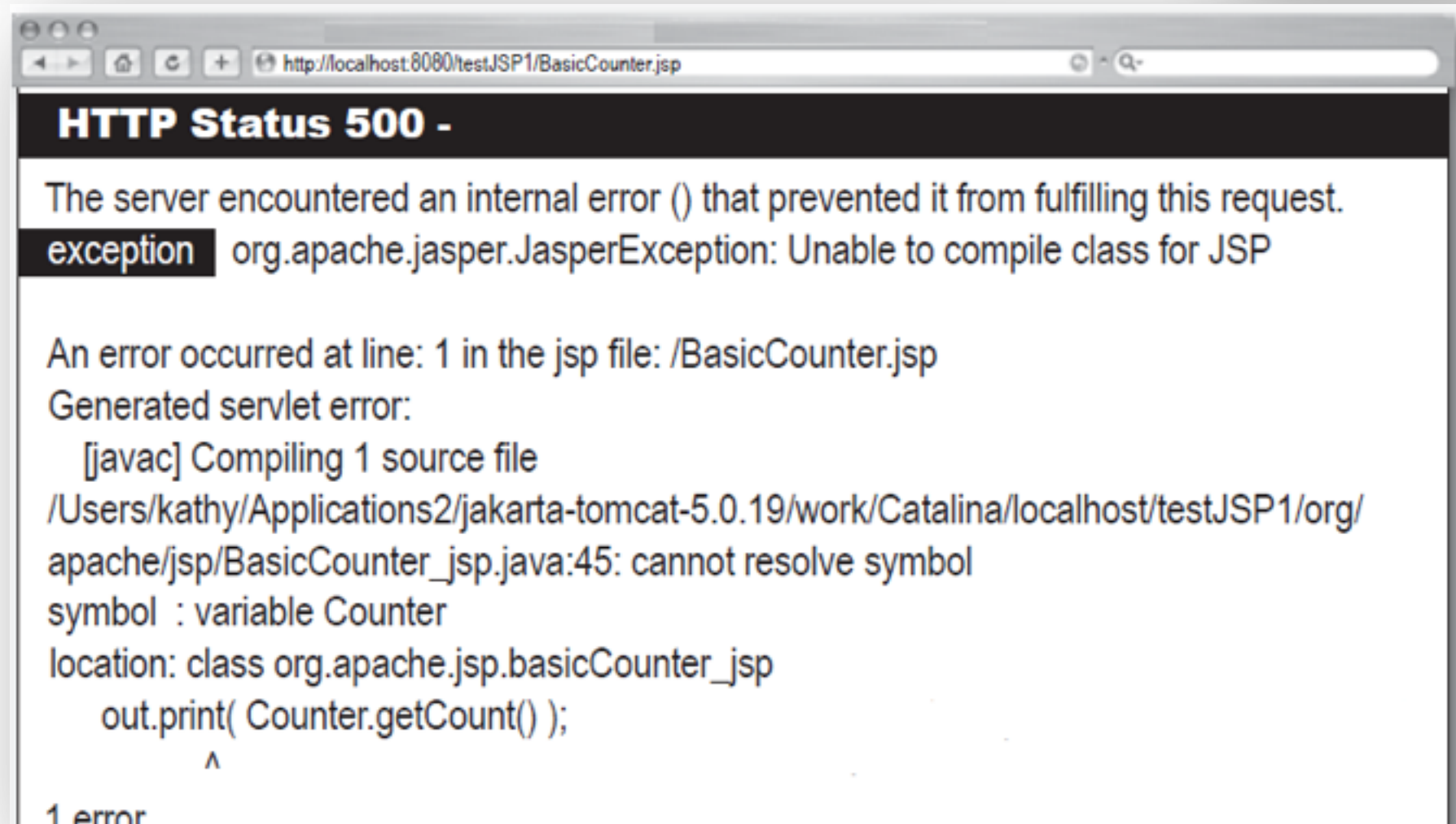
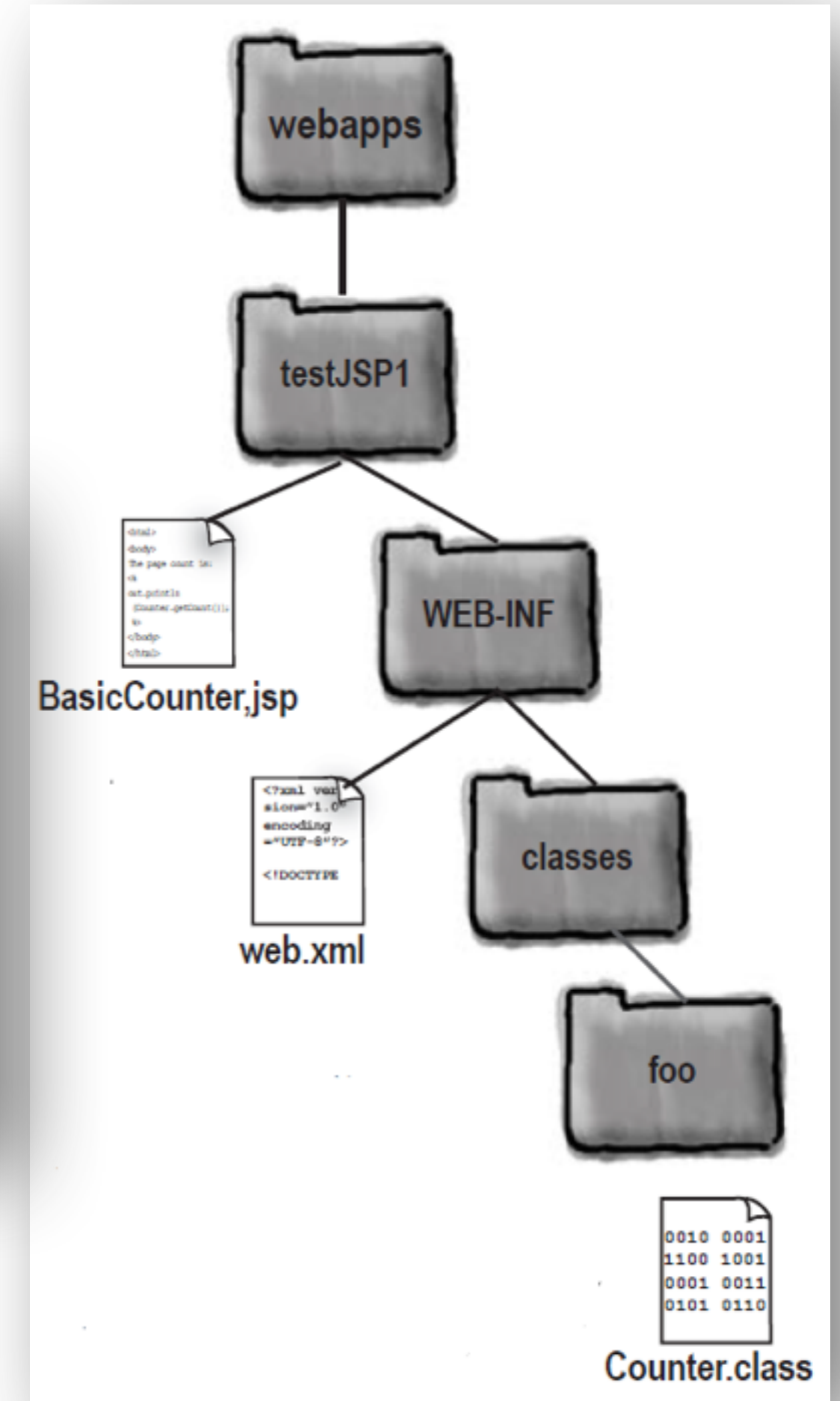
BasicCounter.jsp

```
<html>
<body>
The page count is:
<%
    out.println(Counter.getCount());
%>
</body>
</html>
```

Counter.java

```
package foo;

public class Counter {
    private static int count;
    public static synchronized int getCount() {
        count++;
        return count;
    }
}
```





# Fully qualified class or Import

```
<% out.println(foo.Counter.getCount()); %>
```

## To import a *single* package:

```
<%@ page import="foo.*" %>

<html>
<body>
The page count is:
<%
    out.println(Counter.getCount());
%>
</body>
</html>
```

## To import *multiple* packages:

```
<%@ page import="foo.*,java.util.*" %>
```

# Questions

- Where each part of the JSP goes into the servlet code?
- May I ServletContext and ServletConfig?
- Type and syntax of the elements?
- Lifecycle?

# Elements

- Scriptlet: `<% %>`
- Directive: `<%@ %>`
- Expression: `<%= %>`
- Declaration: `<%! %>`

# Directives, Scriptlets & Expressions

## **Scriptlet code:**

```
<%@ page import="foo.*" %>
<html>
<body>
The page count is:
<% out.println(Counter.getCount()); %>
</body>
</html>
```

## **Expression code:**

```
<%@ page import="foo.*" %>
<html>
<body>
The page count is now:
<%= Counter.getCount() %>
</body>
</html>
```

```
out.print(Counter.getCount());
```

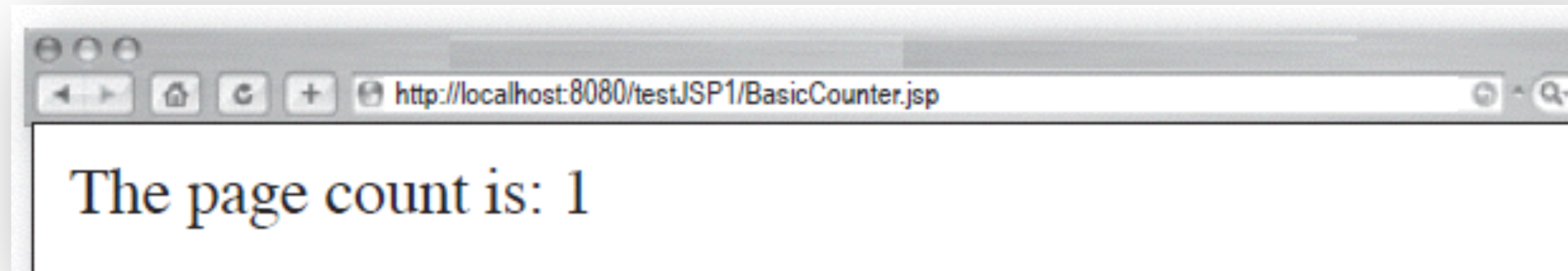
# Beware!

Expressions become arguments in `out.print()`



# But JSPs are servlets

```
<html>
<body>
<% int count=0; %>
The page count is now:
<%= ++count %>
</body>
</html>
```



```
public class basicCounter_jsp extends SomeSpecialHttpServlet {

    public void _jspService(HttpServletRequest request,
        HttpServletResponse response) throws java.io.IOException,
        ServletException {

        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.write("<html><body>");
        int count=0;
        out.write("The page count is now:");
        out.print( ++count );
        out.write("</body></html>");

    }
}
```

`<html><body>` →  
`<% int count=0; %>` → **int count=0;**  
`The page count is now:` →  
`<%= ++count %>` → **out.print( ++count );**  
`</body></html>` →

# Declaration!

```
<%! int count=0; %>
```

# Importing a variable

```
<html><body>
<%! int count=0; %>
The page count is now:
<%= ++count %>
</body></html>
```

```
public class basicCounter_jsp extends HttpServlet {
    int count=0;
    public void _jspService(HttpServletRequest request,
        HttpServletResponse response)throws java.io.IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.write("<html><body>");
        out.write("The page count is now:");
        out.print( ++count );
        out.write("</body></html>");
    }
}
```



# JSP Declatations

```
public class basicCounter_jsp extends SomeSpecialHttpServlet {  
  
    public void _jspService(HttpServletRequest request,  
        HttpServletResponse response) throws java.io.IOException,  
        ServletException {  
  
        PrintWriter out = response.getWriter();  
        response.setContentType("text/html");  
        out.write("<html><body>");  
        <% int count=0; %> int count=0;  
        The page count is now: out.write("The page count is now:");  
        <%= ++count %> out.print( ++count );  
        </body></html> out.write("</body></html>");  
    }  
}
```

Local Variable

```
public class basicCounter_jsp extends SomeSpecialHttpServlet {  
  
    int count=0;  
  
    public void _jspService(HttpServletRequest request,  
        HttpServletResponse response) throws java.io.IOException,  
        ServletException {  
  
        PrintWriter out = response.getWriter();  
        response.setContentType("text/html");  
        out.write("<html><body>");  
        out.write("The page count is now:");  
        out.print( ++count );  
        out.write("</body></html>");  
    }  
}
```

Instance Variable

# Importing methods & variables

```
public class basicCounter_jsp extends HttpServlet {
```

```
<html>
<body>
<%! int doubleCount() {
    count = count*2;
    return count;
}
%>
<%! int count=1; %>
The page count is now:
<%= doubleCount() %>
</body>
</html>
```

```
int doubleCount() {
    count = count*2;
    return count;
}
```

```
int count=1;
```

```
public void _jspService(HttpServletRequest request,
    HttpServletResponse response) throws java.io.IOException {
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    out.write("<html><body>");
    out.write("The page count is now:");
    out.print( doubleCount() );
    out.write("</body></html>");
}
```

```
}
}
```

*The method you want to use is just the way you typed it in your JSP*

*It's done, so no problem with forward-referencing. Declaring the variable first (as you used it in a method).*



# JSP lifecycle

- Look at the directives
- Create an HttpServlet subclass
- Writes
  - Import statements
  - Declaration statements
- Builds a service method `_jspService()`
- Combines all

# Me Container, you JSP

- I will
- give you the class
    - I might not show the generated Java
  - look at your directives
  - create an HttpServlet subclass
    - Tomcat extends: `org.apache.jasper.runtime.HttpJspBase`
  - write any imports or declarations statements in the class file,
    - imports just below the package statement
    - declarations below the class declaration and before the service method
  - build the service method `_jspService()`
    - called by the servlet superclass' overridden `service()` method, and
    - receives `HttpServletRequest` & `HttpServletResponse`.
  - declare & initialize all implicit objects
  - combine HTML, scriptlets & expressions into the service method

You should

- not worry on how I do the above
- Know how your elements work inside the generated servlet
  - and what `jspInit()`, `jspDestroy` & `_jspService()` are about

## Tomcat 5 generated class

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class BasicCounter_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    int count=0;
    private static java.util.Vector _jspx_dependants;

    public java.util.List getDependants() {
        return _jspx_dependants;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;

        try {
            _jspxFactory = JspFactory.getDefaultFactory();
            response.setContentType("text/html");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
            _jspx_out = out;
            out.write("\r<html>\r<body>\r");
            out.write("\rThe page count is now: \r");
            out.print( ++count );
            out.write("\r</body>\r</html>\r");
        } catch (Throwable t) {
            if (!(t instanceof SkipPageException)){
                out = _jspx_out;
                if (out != null && out.getBufferSize() != 0)
                    out.clearBuffer();
                if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
            }
        } finally {
            if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
        }
    }
}
```

```
<html><body>
<%! int count=0; %>
The page count is now:
<%= ++count %>
</body></html>
```

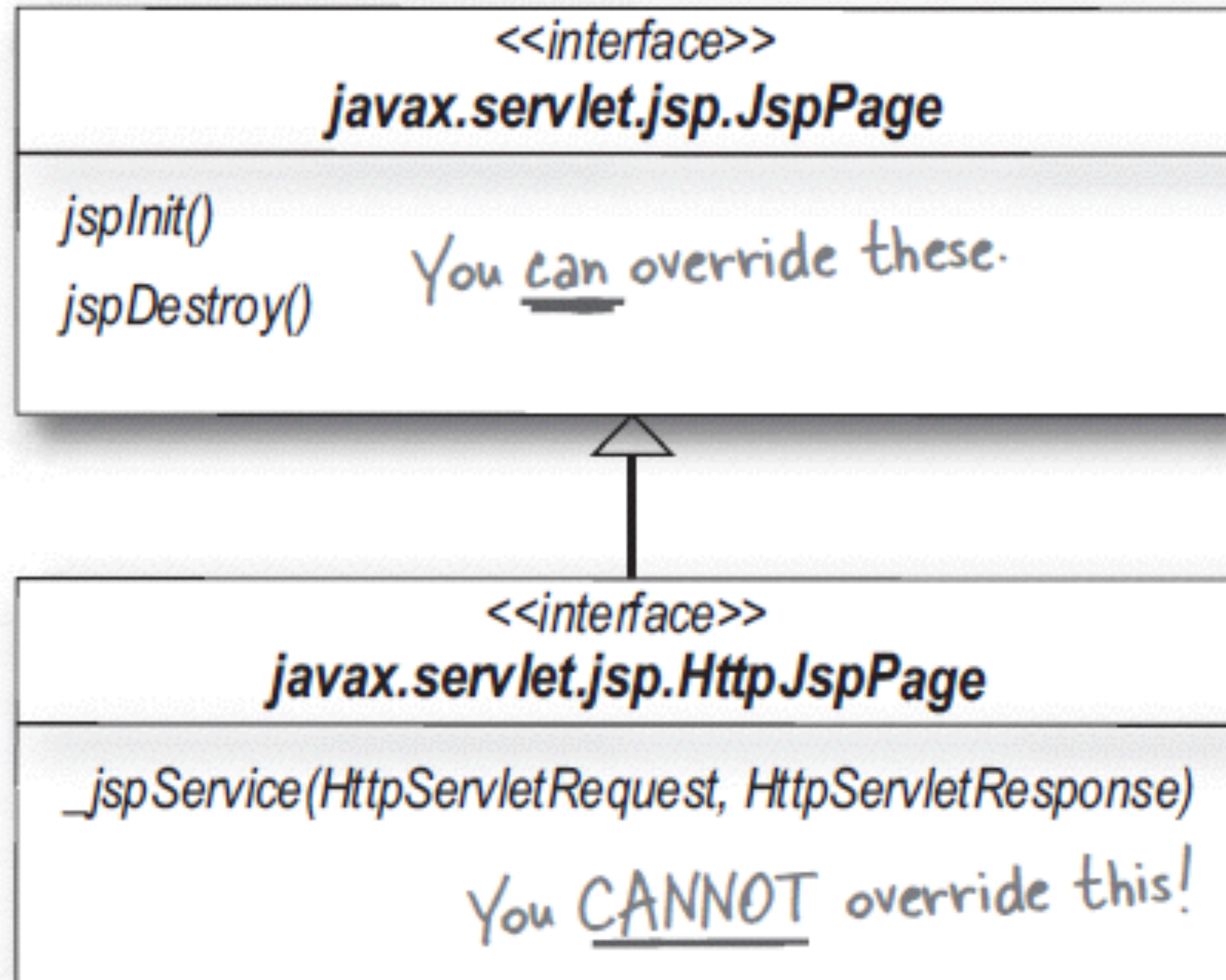
# Example

# API and implicit objects

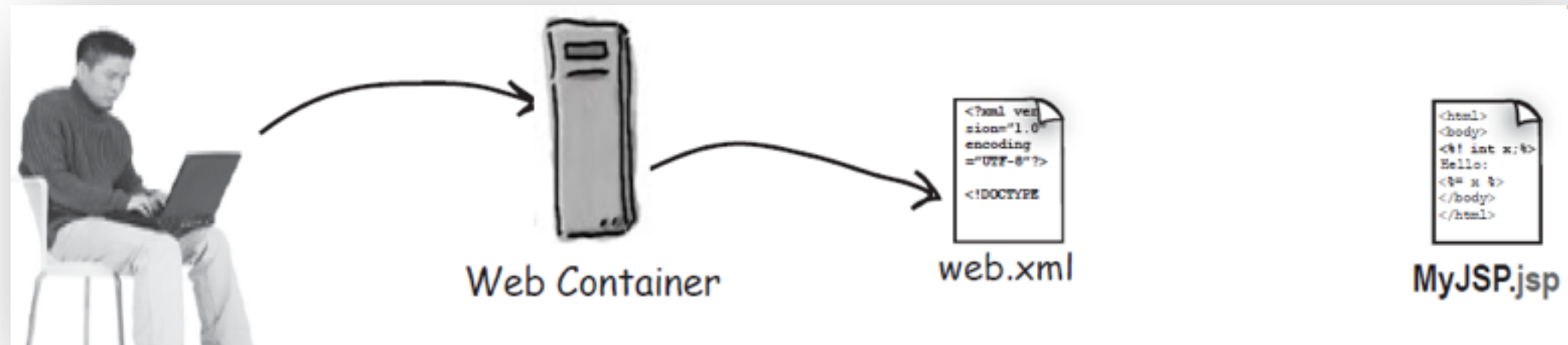
<code>JspWriter</code>	_____	<code>out</code>
<code>HttpServletRequest</code>	_____	<code>request</code>
<code>HttpServletResponse</code>	_____	<code>response</code>
<code>HttpSession</code>	_____	<code>session</code>
<code>ServletContext</code>	_____	<code>application</code>
<code>ServletConfig</code>	_____	<code>config</code>
<code>JspException</code>	_____	<code>exception</code>
<code>PageContext</code>	_____	<code>pageContext</code>
<code>Object</code>	_____	<code>page</code>



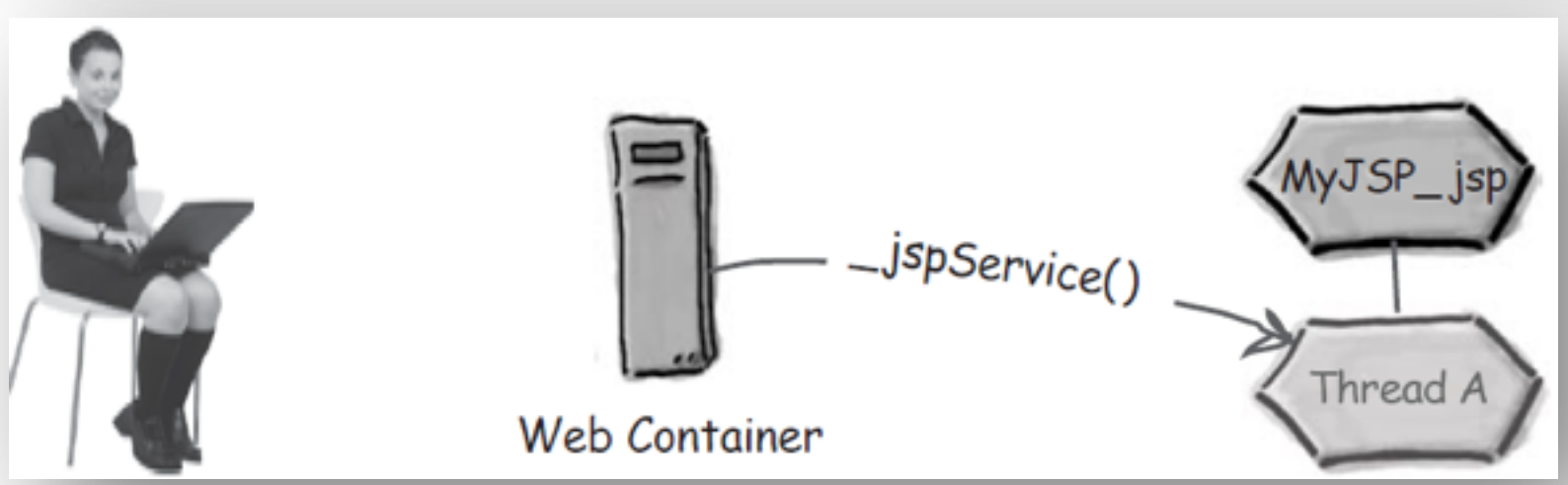
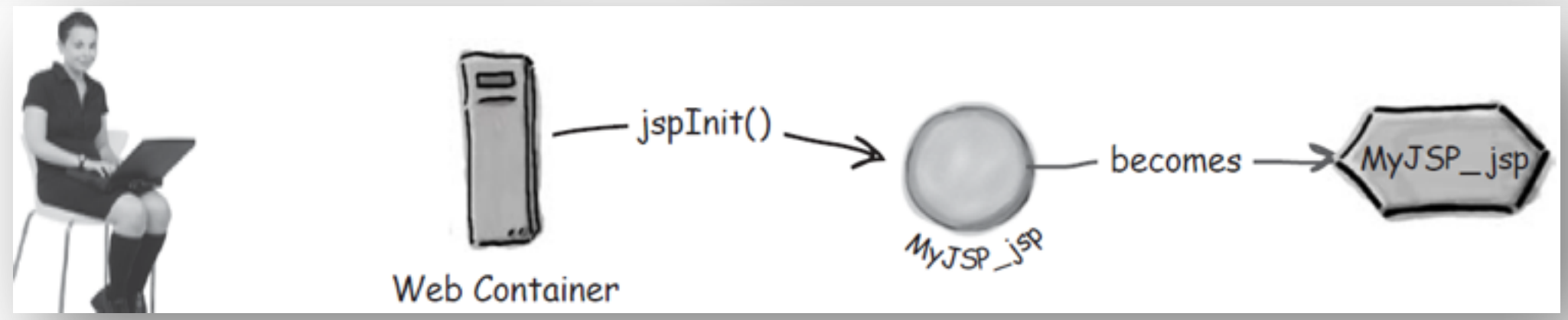
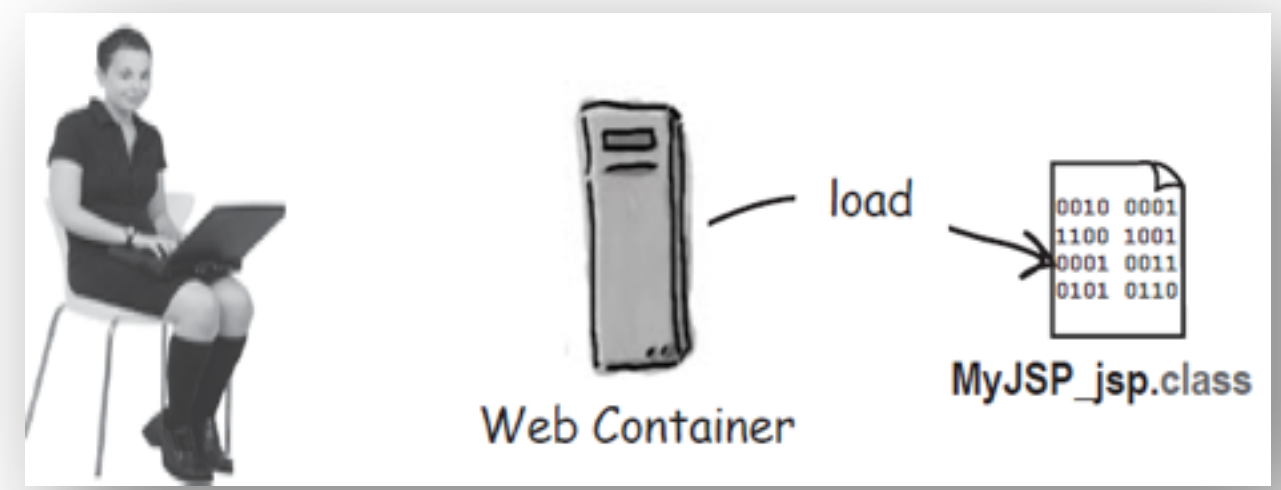
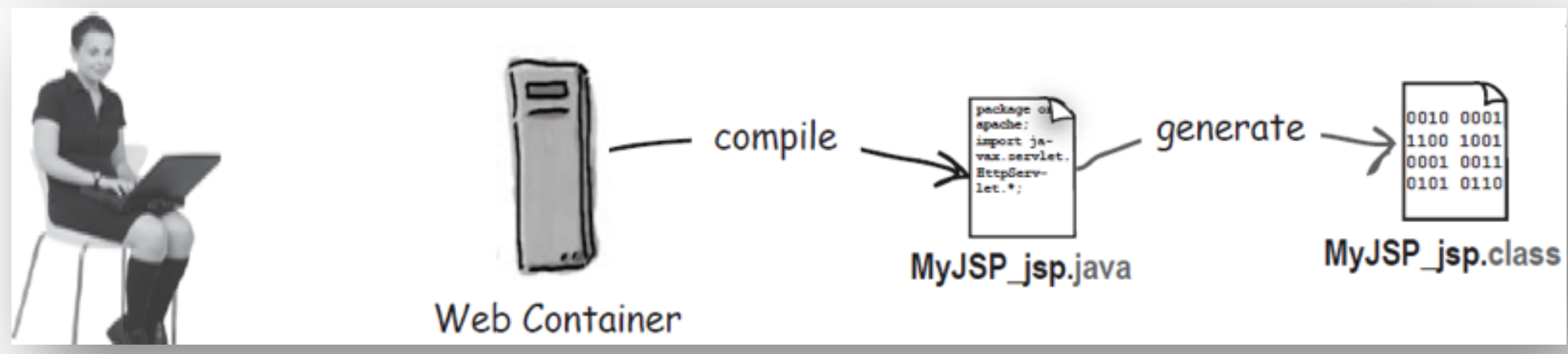
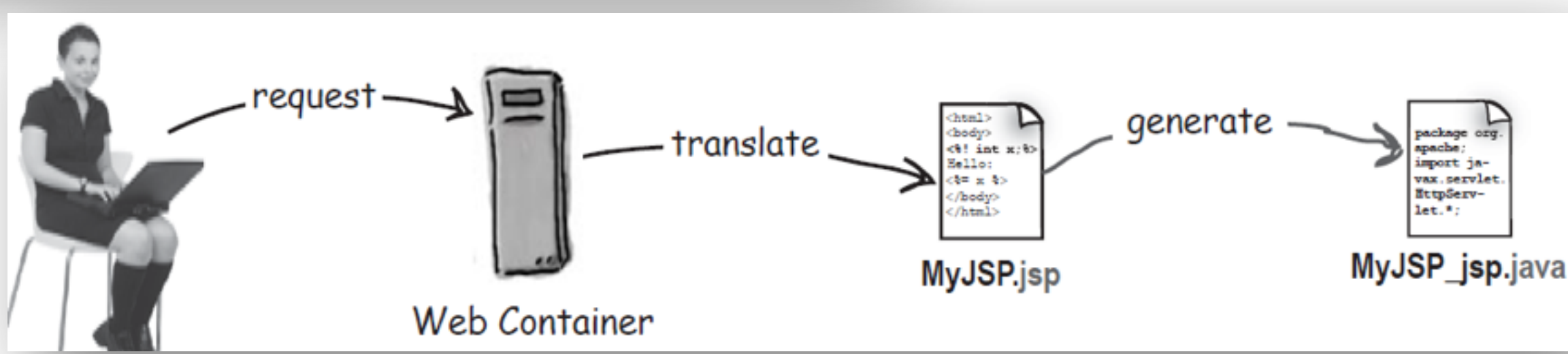
# API and the generated servlet







# Lifecycle of a JSP



# Initializing your servlet JSP

```
<web-app ...>
  <servlet>
    <servlet-name>MyTestInit</servlet-name>
    <jsp-file>/TestInit.jsp</jsp-file>
    <init-param>
      <param-name>email</param-name>
      <param-value>ikickedbutt@wickedlysmart.com</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyTestInit</servlet-name>
    <url-pattern>/TestInif.jsp</url-pattern>
  </servlet-mapping>
</web-app>
```

Overwrite using declaration



```
<%!
  public void jspInit() {

    ServletConfig sConfig = getServletConfig();
    String emailAddr = sConfig.getInitParameter("email");

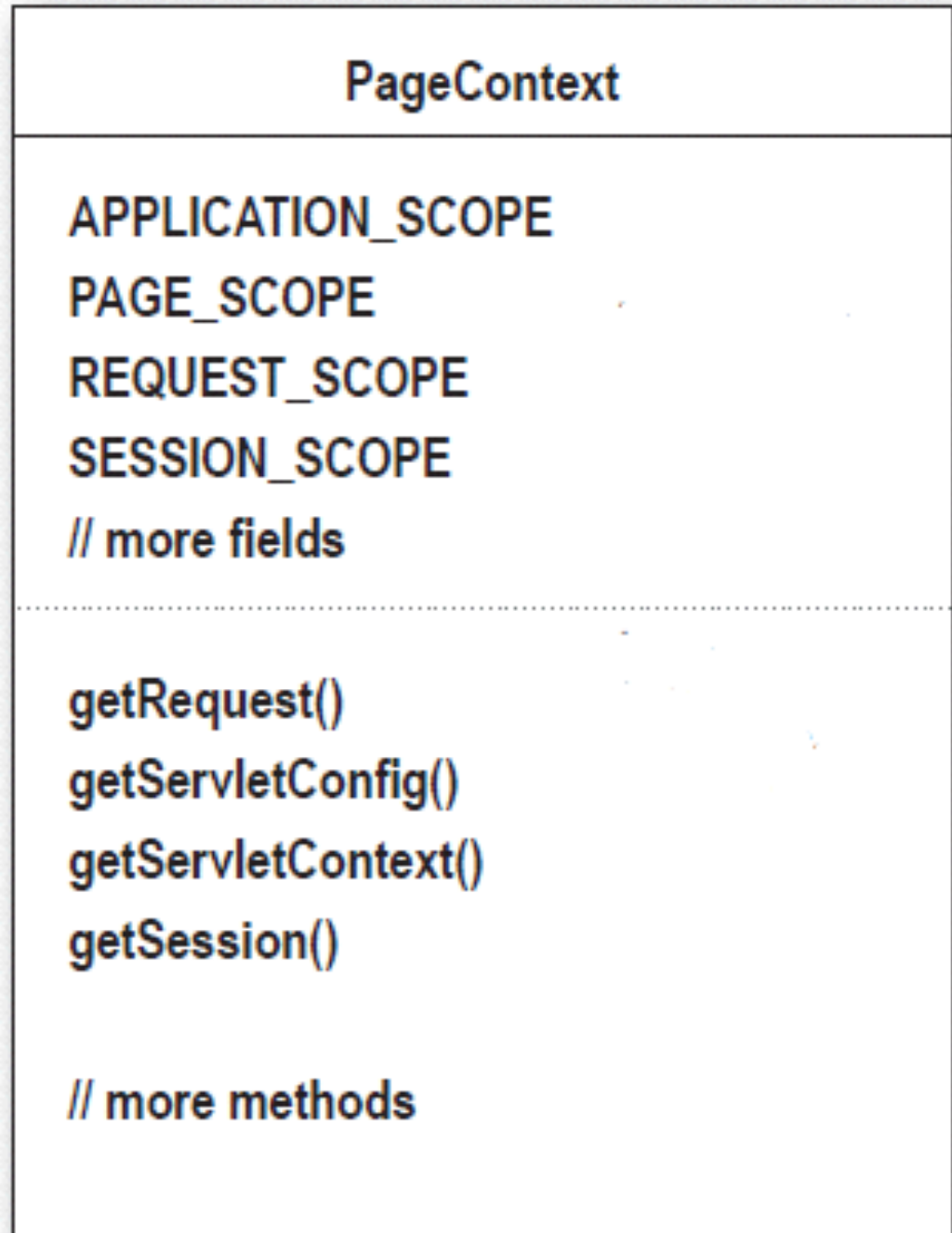
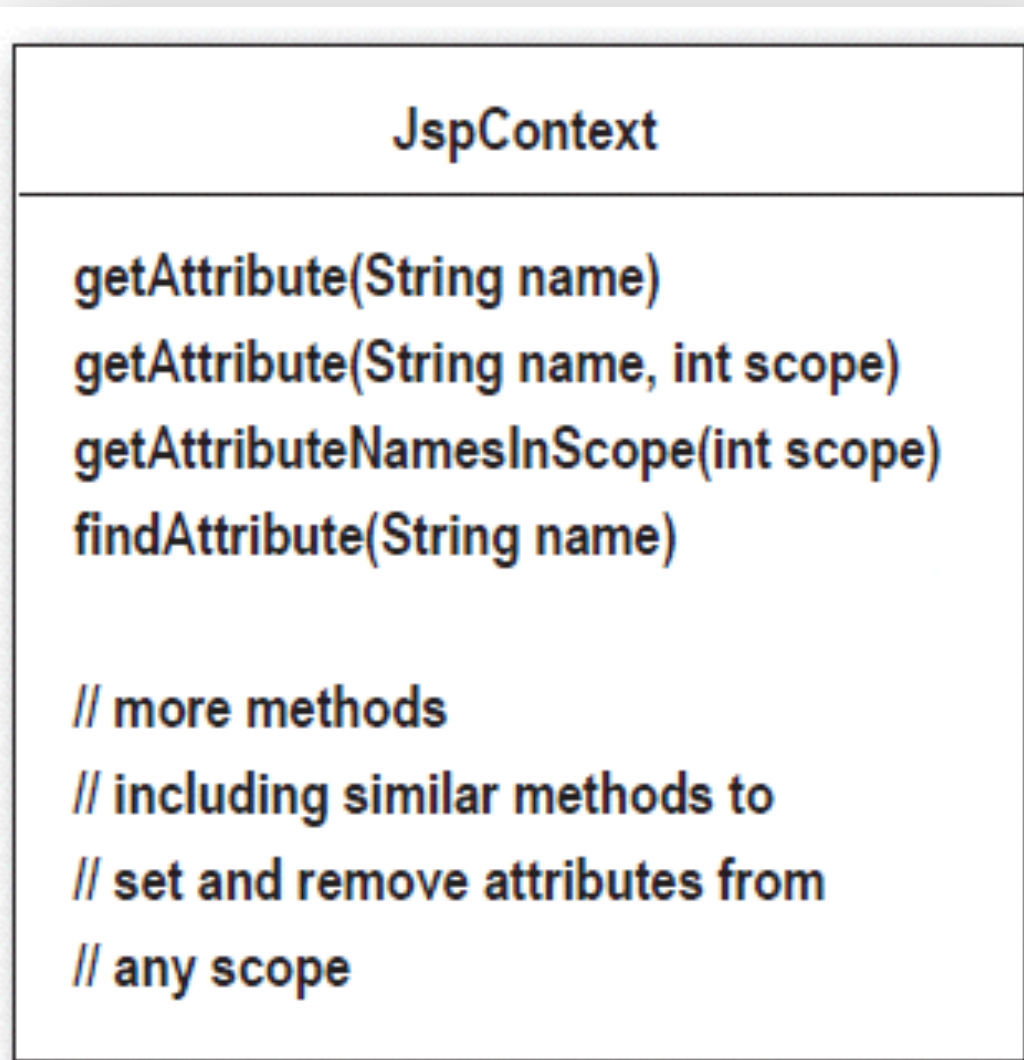
    ServletContext ctx = getServletContext();

    ctx.setAttribute("mail", emailAddr);
  }
%>
```



# Attributes in a JSP

	<b>In a servlet</b>	<b>In a JSP (using implicit objects)</b>
<b>Application</b>	<code>getServletContext().setAttribute("foo", barObj);</code>	<code>application.setAttribute("foo", barObj);</code>
<b>Request</b>	<code>request.setAttribute("foo", barObj);</code>	<code>request.setAttribute("foo", barObj);</code>
<b>Session</b>	<code>request.getSession().setAttribute("foo", barObj);</code>	<code>session.setAttribute("foo", barObj);</code>
<b>Page</b>	Does not apply!	<code>pageContext.setAttribute("foo", barObj);</code>



**Setting a page-scoped attribute**

```

<% Float one = new Float(42.5); %>
<% pageContext.setAttribute("foo", one); %>
  
```

**Getting a page-scoped attribute**

```

<%= pageContext.getAttribute("foo") %>
  
```

**Using the pageContext to set a session-scoped attribute**

```

<% Float two = new Float(22.4); %>
<% pageContext.setAttribute("foo", two, PageContext.SESSION_SCOPE); %>
  
```

**Using the pageContext to get a session-scoped attribute**

```

<%= pageContext.getAttribute("foo", PageContext.SESSION_SCOPE) %>
(Which is identical to: <%= session.getAttribute("foo") %> )
  
```

**Using the pageContext to get an application-scoped attribute**

```

Email is:
<%= pageContext.getAttribute("mail", PageContext.APPLICATION_SCOPE) %>
  
```

Within a JSP, the code above is identical to:

```

Email is:
<%= application.getAttribute("mail") %>
  
```

**Using the pageContext to find an attribute when you don't know the scope**

```

<%= pageContext.findAttribute("foo") %>
  
```

Using  
PageContext



# Other (page) directives & attributes

```
<%@ page import="foo.*" session="false" %>
```

```
<%@ taglib tagdir="/WEB-INF/tags/cool" prefix="cool" %>
```

```
<%@ include file="wickedHeader.html" %>
```

**import**

**isThreadSafe**

**contentType**

**isELIgnored**

**isErrorPage**

**language**

**errorPage**

**extends**

**session**

**buffer**

**autoFlush**

**info**

**pageEncoding**



# EL (Expression Language)

- Web page designers shouldn't have to know Java.
- Java code in a JSP is hard to change and maintain.

## **This EL expression:**

```
Please contact: ${applicationScope.mail}
```

## **Is the same as this Java expression:**

```
Please contact: <%= application.getAttribute("mail") %>
```

# We have a choice

```
<web-app ...>
...
<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <scripting-invalid>
      true
    </scripting-invalid>
  </jsp-property-group>
</jsp-config>
...
</web-app>
```

```
<web-app ...>
...
<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <el-ignored>
      true
    </el-ignored>
  </jsp-property-group>
</jsp-config>
...
</web-app>
```

# One last word: Actions (details later)

## **Standard Action:**

```
<jsp:include page="wickedFooter.jsp" />
```

## **Other Action:**

```
<c:set var="rate" value="32" />
```