**Web: Quiz 1 (Chapters 1 -4)**

**Chapter 1: An introduction to web programming with Java**

****

1. Name the software component that runs on the client of a typical web application

* Browser (chrome or firefox)

1. Name the two software components that run on the server of a typical web application.

* Web Server (Apache HTTP) : Software enables it to send web pages to web browsers
* Database Server (MySQL)

1. Distinguish between HTML and HTTP

* HTML : Hypertext Markup Language : the language that the web browser converts into web pages of a web app
* HTTP : Hypertext Transfer Protocol : the protocol that web browsers & web sources use to communicate

1. Distinguish between static web pages and dynamic web pages

* Static : an HTML document that’s stored in a file & does not change in response to user input, filename w/ extension of htm or html
* Dynamic :HTML doc that’s generated by a web application , changes according to parameters that are sent to the web app by the browser

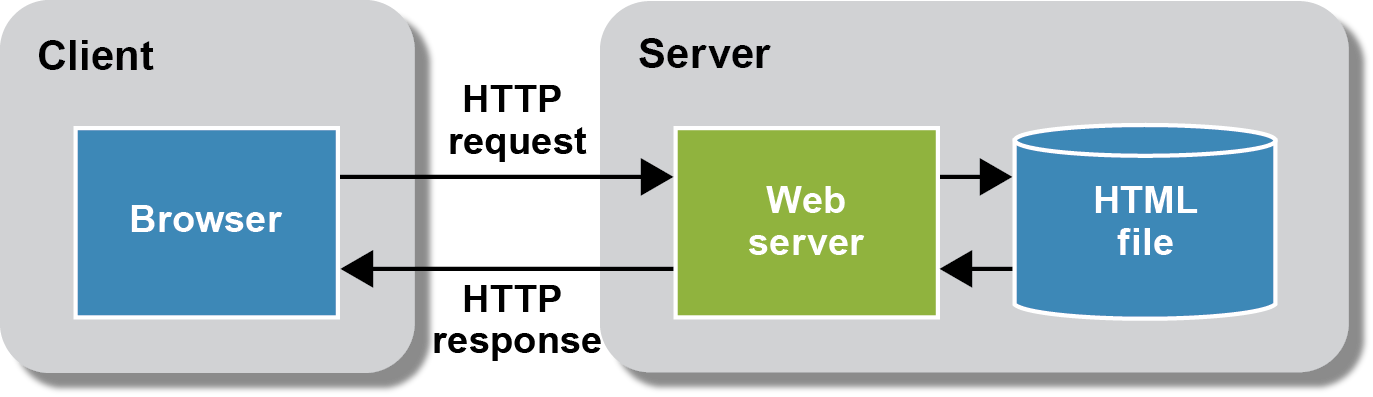
Static

1. Web browser requests a page from a web server by sending the server a message(HTTP request)

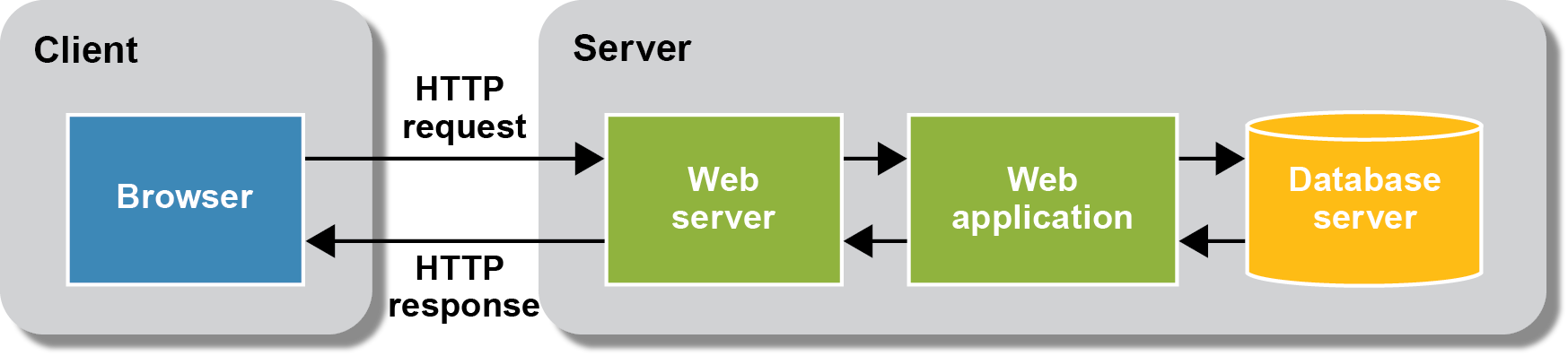
For a static page, the request includes the name of the HTML file requested

1. Web server replies to HTTP request by sending a message known as an HTTP response back to the browser

For a static page that HTTP response includes the HTML doc that’s stored in HTML file



Dynamic

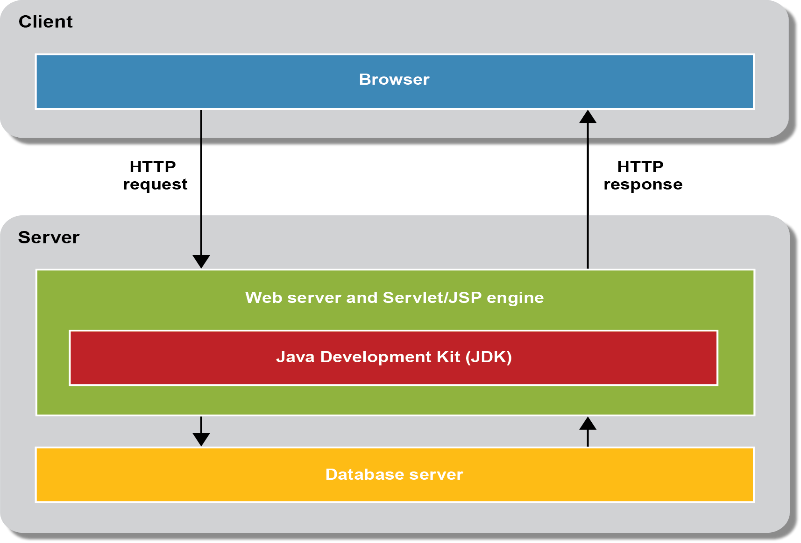


1. Name three approaches for developing Java web applications

* Servlet/JSP
* Is a lower level API that does less work for the programmer
* Provides a high degree of control over the HTML/CSS?JavaScript that’s returned to the browser
* JSF
* Higher level API does more work for programmer
* Makes it more difficult to control the HTML/CSS?JavaScript
* Spring Framework
* Higher level API that does more work
* Provides a high degree of control over the HTML/CSS/JavaScript



1. Describe the software components that are required for developing sevlet and JSP applications.

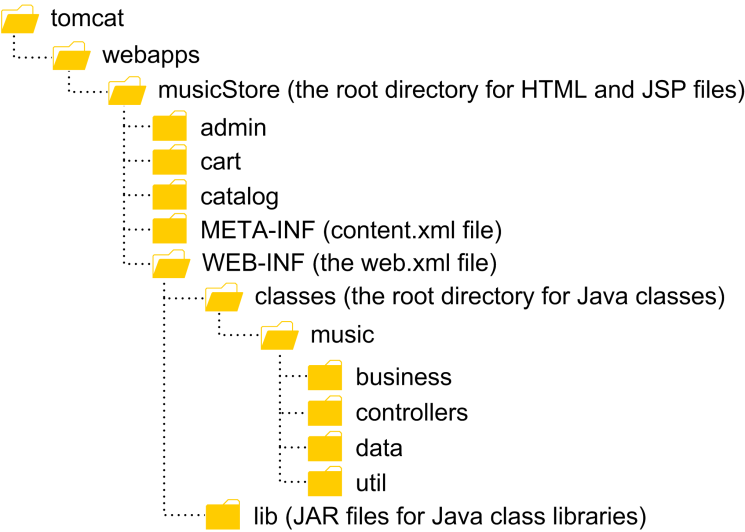
****

* Web server & servlet/JSP engine (container) => Tomcat => processes HTTP requests & returns HTTP response
* JDK : (part of Java SE) required by servlet/JSP (Servlet/JSP accesses JDK) JDK contains Java compiler & core classes & JRE (runs compiled classes)

1. List and describe the three layers of a typical Java web application

* Presentation : HTML pages & JSP
* Business Rules : servlets ( servlets may call other Java classes including Java Bean class)
* Data access : consists of classes that read & write data that’s stored on the server’s disk drive (pg 17)

1. In general terms, describe the use of these directories as defined by the J2EE specification:



* document root :
* (root) : contains the HTML & JSP files
* \WEB-INF : Contains the web.xml file and is not directly accessible from the web
* \WEB-INF\classes : contains the servlets & other Java classes for app. Each subdir corresponds with the package for the Java class
* \WEB-INF\lib : contains any JAR files that contain Java class libraries that are used by the web app
* \META-INF : contains the context.xml file that configures the web application context

1. Name 2 **IDEs** that can be used for developing Java web applications

* NetBeans
* Eclipse
* IntelliJ IDEA

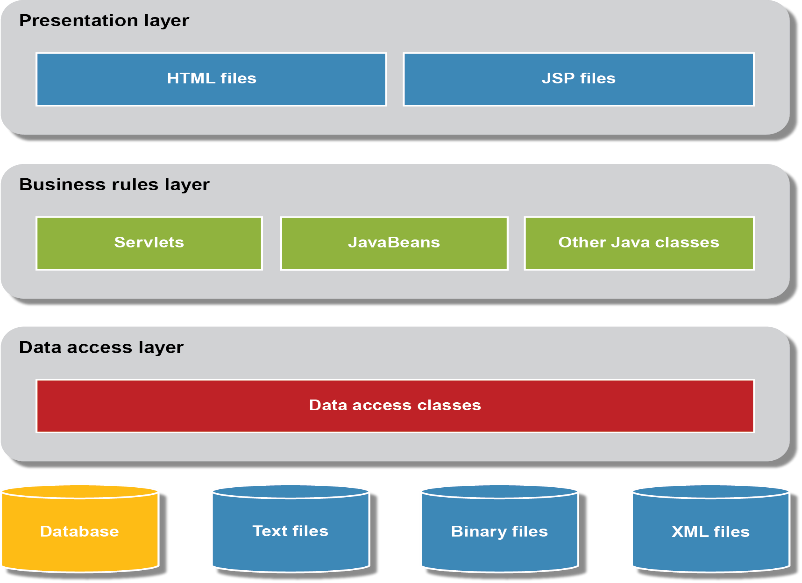
1. Name 2 **web servers** that can be used for developing Java web applications

* Tomcat : servlet/JSP engine that includes a webserver, free, open source, popular for Java web apps
* Glass Fish :complete Java EE app Server, free, open source, provides more features but also more system resources

1. Name a popular database server that can be used for developing Java web applications

* MySQL : relational DB server that implements most of the SQL standard, free, open source, most popular for Java web app
* PostgreSQL
* Oracle

1. Architecture for a servlet/JSP application



1. IDE : Integrated Development Environment : tool that provides all of the functionality that you need for developing web apps
2. Web application : set of web pages that are generated in response to user requests (search engines, online stores, auctions, nes sites,discussion groups, games)
3. JSF : Java Server Faces : provides a high levelAPI that replaces both Servlets & JSPs
4. Servlets : store the Java code that does the server side processing
5. JPS : Java Persistence API : an API for working with databases
6. Java Bean : Special class used to temporarily store& process data, typically used to define a business object such as a user or invoice object
7. Web host : internet service provider (ISP) that provides web hosting that supports servets & JSP, web host gives you an IP address, can transfer files to it via ftp (FileZilla) , get a domain name elsewhere to use
8. Tomcat :

* Coyote is its webserver
* Catalina is its servlet/JSP Engine

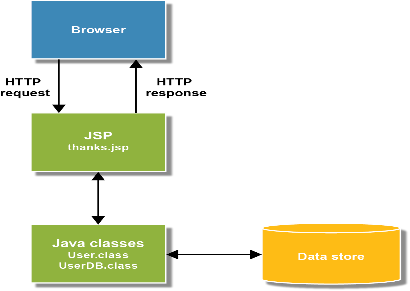
**Chapter 2: How to structure a web application with the MVC pattern**

**(from power point presentation Murach’s Java servlets & JSP)**

* Pattern : a standard approach to solve a problem
* When you perform a request that modifies data on the server you usually use the POST method. (pg 36)
* If you are not modifying data you use a method attribute of “get” to use the GET method for the HTTP request
* Servlet is a special Java class running on the server and does the processing for dynamic pages of a web app
* Servlet extends HttpServletClass
* Override doPost method , executed when URL receives an HTTP request that uses the Post method
* doPost method accepts both a request and a response object from the web server
* also overrides doGet (420
* web.xml file : describes how the web app should be configures when it is deployed on a server, (deployment descriptor). If it exiss, it is always in WEB-INF (page 42
* servlet 3.0 specification use @WebServlet to map the servlet to URL pattern
* prior to 3.0 must use web,xml to map a servlet to a URL pattern
* servlet provides a response object that you can use to return an HTTP response to the browser. Typycically you return HTTP response for an HTML page by forwarding the request and response objects to a Jsp page
* User class is a java bean or bean because it follows 3 rules:
  + Zero arg constructor (Example sets 3 instance variable to null (stores empty strings))
  + Get & set methods for all of the properties that need to be accessed by JSPs . To access a Boolean value, you code is and set methods
  + Implement Serializable or Externalizable interface
    - Serializable is a tagging interface in the java.io package that indicates that the class contains get set & is methods that another class can use to read & write an objects instance variables to and from a persistent data source
* Invisible java beans : define business objects of an app : don’t define visible components
* EJB : Enterprise Java bean : more complex & difficult to code
* Java bean is also a POJO plain old java object
* Main benefit of coding business classes as java beans : you can use special JSP tags for working with beans (EL tags)
  + - Embedding java code in a JSP is not recommended)
    - Recommended to restructure the app to move the Java code to a servlet that forwards the request and response objects to the JSP (servlet does the processing & JSP provides the HTML for the user interface. With this approach the JSP doesn’t require embedded Java code, only special JSP tags … the web designer can develop the JSPs with minimal interaction with the Java Programmer, and the Java Programmer can develop the servlets and not the HTML
* JSP first requested, JSP engine (part of servlet/Jsp engine) converts JSP code into a servlet and compiles the servlet. Then the JSP engine loads that servlet into the servlet engine which ruuns it. Subsequent requests just run it.

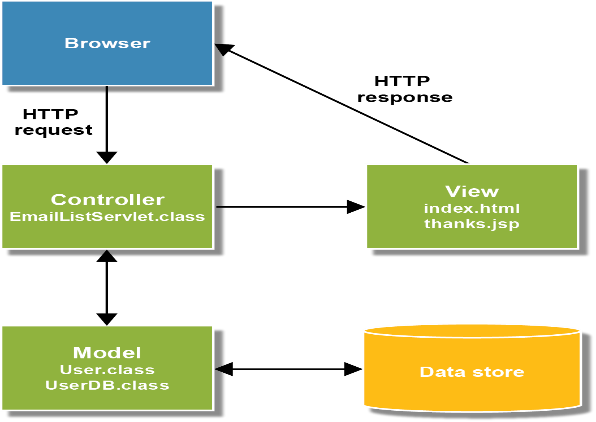
1. Describe the Model 1 pattern (not recommended)

* Uses JSPs to handle all of the processing and presentation for the application
* Request & response
* JSP interacts with Java classes & objects that represent the data of business objects in the app & provide methods for business processing
* Stores data in an object
* Then maps to a data store to save data
* (persistent data storage)
* Sometimes adequate for apps that have limited processing requirements



1. Describe the Model 2 (MVC) pattern (recommended) page 32

* Separates the code into a model, a view, and a controller
* Model (Business layer) consists of business objects like the User object (Java Beans)
* View (presentation layer)consists of HTML pages and JSPs
* Controller (manages flow) consists of servlets
* Servlet reads parameters of request, updates model & saves to data store, forwards model to a JSP for presentation



* Data access layer consists of classes like the UserDB class that read and write business objects like the User object to and from the data store
* Construct each layer so it’s as independent as possible

1. Explain how the MVC pattern can improve application development

* Improve app deployment when processing requirements are substantial
* Easier to modify

1. Distinguish between the HTML and CSS for a web page

* HTML file contains tags that define the content of the web page
* CSS (Cascading Style Sheet) file contains the formatting for the web pages

1. Distinguish between the code for a servlet and a JSP

* Servlets contain Java code for a web application
* When a servlet controls the flow of the application, it is known as a controller
* JSP (Java Server Page) consists of special Java Tags such as Expression Laguage(EL) tags that are embedded within HTML code. An EL tag begins with a dollar sign ($)

1. Explain why you typically use both servlets and JSPs in a Java web application
   * Main benefit of coding business classes as java beans : you can use special JSP tags for working with beans (EL tags)
     + Embedding java code in a JSP is not recommended)
     + Recommended to restructure the app to move the Java code to a servlet that forwards the request and response objects to the JSP (servlet does the processing & JSP provides the HTML for the user interface. With this approach the JSP doesn’t require embedded Java code, only special JSP tags … the web designer can develop the JSPs with minimal interaction with the Java Programmer, and the Java Programmer can develop the servlets and not the HTML
2. Describe the purpose of the deployment descriptor in a web application

* The web xml file, or deployment descriptor (DD), describes how the web application should be configured when it’s deployed

1. Describe the purpose of a JavaBean within a web application

* A Java Bean, or bean, is a Java class that:

1. Provides a zero-argument constructor
2. Provides get and set methods for all its instance variables
3. Implements the Serializable or Externalizable interface

**Chapter 3: How to work with NetBeans and Tomcat**

1. Describe a NetBeans project

* To create a project

1. Select File -> New Project

* Work with files

1. Open a file ( double-click)
2. Rename a file (right click and select rename)
3. Delete (right click & delete)

* Open an existing Project

1. Select File-> Open project

* Remove a project (right-click & close)
* Run project ( F6 or select Run Project from toolbar or Run menu)
* Run specific project (right-click select run file ) displays in default browser
* Change default browser: Tools-> Options, General, browser
* View test run, use tabs in output window
* Window-> Output
* Clear output: right-click in the window & clear
* When you run an app:
  1. Compiles all files that need to be compiles
  2. Deploys the files for the project to the specified server
  3. Starts the default web browser
  4. Displays the first page of the application in that browser
* To add HTML

1. Right-click on the Web Pages folder or subfolder
2. Select New-> HTML
3. Specify the name and location of the file

* To add JSP

1. Right-click on the Web Pages folder or one of its subfolders and
2. select New-> JSP
3. Specify the name and Location of the file

* To add Java classes to a project

1. Right-click on one of the project folders and
2. select New-> Java Class
3. Specifythe name and location of the file

* To specify a package for the class

1. Use the Package combo box to
2. select an existing package or to enter the name for a new package

* To add a servlet,

1. right-click on the package, and
2. select New\_>Servlet
3. Specify a name and URL for the servlet
4. Double-click on it to open it
5. Then use tab s across the top of the page to modify the web.xml file
6. Use the Source tab to manually edit the file in the XML editor

* To Validate an XML file against its schema,

1. right-cick on the file and
2. select Validate XML.
3. The results of the validation will be displayed in an XML .
4. Check window within the Output window

* Add existing files to a project

1. Copy files form windows explorer or mac finder
2. Paste them into appropriate folder in projects window or files window

* Deploy an application to a remote server

1. Copy the WAR file for the application
2. from the dist folder
3. to the appropriate folder for web apps on the remote server

* View a server

1. Expand Servers group
2. Right-click on a server to start, stop or refresh

* View web applications that are running on Tomcat server

1. Expand Web Applicatins folder for the server
2. Right-click on an application to start, stop, or undeploy

* Add a library file to libraries for a project

1. Right-click on the Libraries folder
2. Select the Add Library command
3. Use the resulting dialog box to selct library
4. May add 1 or more JAR files to the project

* Add a JAR file to a project

1. Right-click on the Libraries folder,
2. Select the Add JAR/Folder command and use the resulting dialog box to select the JAR file

* Register a database connection

1. Right-click on the project in the Project window and select the Resolve Data Source Problem command
2. Select the data source and click the Add Connection button
3. If necessary, change any of the entries for the connection
4. Click OK to register the db connection

* Folders used by Netbeans

1. Build/web : all folders & files for the web app after it has been built
2. Dist : The war file for the application
3. Nbproject : The configuration files and build scripts
4. Src : The source code for the Java files and servlets
5. Web : The HTML, JSP, XML files for the app
6. Describe the difference between the

* Projects window,
* File window &
* Services window

1. Web xml file is known as the deployment descriptor(DD)
2. Code completion

* after class or object name and period (select/Tab or Enter)
* Code completion: Ctrl+Spacebar
* On HTML or JSP tags, cc provides a list of possible entries

**Chapter 4: A crash course in HTML5 & CSS**

1. Describe the use of HTML’s elements:

* head : header for a page
* title : name that the browser shows in its title bar or tab
* body :
* link: identifies file (css)

1. Describe the use of HTML’s elements

* h1 :block : level 1 heading bold 200% font size
* h2 : block : level 2 heading bold 150% font size
* p : block : paragraph 100%
* img : block : image
* form : block : form can be submitted toweb server for processing
* a : inline : link to another page
* input : inline : a control on a from like text box or button
* label :inline : identifies control on a form
* br : newline

1. Describe the use of HTML5’s elements

* header : the header for a page
* section : generic section
* article : composition
* nav : a section of a page that controls links to other pages or placeholders
* aside : sidebar related to content that is near it
* figure : image table or component treated like an image
* footer : footer of page

1. Describe HTML tags

* div : blocke element: provides a containter for other elements:divider
* span :inline element that lets you identify text that can be formatted with CSS

1. Describe the use of HTML’s tags page 104

* table : defines a table
* tr : table row
* th : header cell within a row
* td “ data cell within a row

1. Explain why it’s a best practice to use an external style sheet

**Best practice**

* to use external style sheets because that leads to better separation of concerns
* specifically you separate the content from the formatting
* makes it easy to use the same style for multiple pages
* (embedded & inline you have to copy the style to each page)
* The last rule overrides the earlier rules

1. Name and describe three types of CSS selectors

**Element, class and id selectors page 108**

* Type (element) selector: You code a selector for all elements of a specific type by naming the **element**.

This is referred to as a **type or element selector**

* You code a selector for an element with a class attribute by coding a

**period followed by the** **class name**. Then, the rule set applies to all elements with that class name. This is known as a **class selector.**

* You code an id selector for an element with an id attribute by coding a pound sign

**# followed by the id value**. This is known as an **id selector.**

1. Name and describe the components of a CSS rule set

**Rule sets, selectors, rules and comments**

* A **CSS rule set consists of a selector and one or more rules within braces**

1. A CSS selector consists of the identifiers that are coded at the beginning of the rule set. If more than on selector is coded for a rule set, the selectors are separated by commas
2. **A CSS rule consists of:**

* Property
* Colon
* Value
* Semicolon (optional but a best practice
* Make code easier to read with spaces, identation & blank lines within a rule set
* CSS comments begin with /\* end with \*/ single or multiple lines

1. Describe how to use HTML to display

* Text boxes page 116
* Checkbox : <input type=”checkbox” name=”addemail” checked>
* Check boxes : allow user to supply true/false (you can select/deselect any combo) page 120
* Radio buttons L select an option (to create a group, use the same name for all radio buttons… if you don’t group them, more than one can be on at the same time
* Buttons page 118

**Use select element for combo and list boxes**

Must have 2 or more option elements

The size attribute determines if it is combo or list

Default size is 1 = combo

* Combo boxes : page 122 (select one option from a drop down list)
* List boxes : page 122 (select one or more options from list)

1. Explain how to use an HTML form to send data to a servlet
2. Page 114 form : contains one or more controls (text boxes, buttons, check box & list boxes)

Submit button calls a JSP or a servlet when the user clicks it

Usually specify servlet name:

<form action=”contactList” method post>

But can also specify a JSP by specifying url for file name

action=”contact\_list.jsp”

**Notes:**

HTMLprovides provides the structure and contrnt of a web page

CSS provides the formatting

DOCTYPE : says page uses HTML5 <!DOCTYPE html>

<html> includes all of the elements for the page

<head>

<title> browser displays this text in the title bar or tab

<link rel= “stylesheet” href=”styles/main.css” type=”text/css/> identifies the file that stores the external style sheet

<meta charset=”UTF-8”>

</head>

How to code HTML elements (page 92)

Each element is coded with a tag

Tag starts with < and ends with >

**Most HTML elements have 3 parts**

* Start tag : starts the element includes the Element name <h1 plus on or more optional attributes
* Content : text or other data
* End tag : marks the end of the element </h1>

**Self-closing tags :** elements that don’t have content and end tags

Most attributes are coded with an attribute name, an equals sign and a value in quotes

* href provides URL to go to when clicked
* title attribute provides text that’s displayed for the link

Boolean attributes can be coded with just the name of the attribute

* input element “checked” attribute is “on because it is coded, so check box is checked <input type=”checkbox” name=”maillist” checked>

Comments : <!- -This is a comment - ->

Blank space : &nbsp; ( non-breaking space)

Character entities begin with & and end with ;

Block elements : displayed on their own lines

Inline elements : flow to the right of the preceding elements and don’t start a new line

(use a br element after an inline element to start a new line)

**Block elements**

* h1 : level 1 heading with content in bold at 200% of the base font size
* h2 : level-2 heading with content in bold at 150% of base font size
* p : paragraph at 100% of base font size
* img : image
* form : form that can be submitted to the web server for processing

**Inline elements**

* a : link that goes to another page or location on the current page when clicked
* input : a control on a form like a text box or button
* label : label identifies a control on a form

Line Break

* br : starts a newline

When you use HTML5, you can use syntax of HTML or XHTML

**Benefits of HTML5:**

* SEO
* Simplified HTML & CSS ( you don’t need to code id attributes that are used by CSS… you apply CSS to the elements)

Older browsers wont recognize HTML5 semantics & can’t use easy CSS formatting

Workaround: code a script element in the head section that provides a JavaScript shiv and use CSS to identify block elements

**HTML**

* <br>
* <img src=”logo.gif” alt=”Murach Logo”>

**XHTML**

* <br />
* <img src=”logo.gif” alt=”Murach Logo”/>

**<img> : Differences Between HTML 4.01 and HTML5**

* The <img> tag defines an image in an HTML page.
* The <img> tag has two required attributes: src and alt.

**Note:** Images are not technically inserted into an HTML page, images are linked to HTML pages. The <img> tag creates a holding space for the referenced image.

**Tip:** To link an image to another document, simply nest the <img> tag inside <a> tags.

The following attributes: align, border, hspace, and vspace are not supported in HTML5.

* In HTML the <img> tag has no end tag.
* In XHTML the <img> tag must be properly closed.

**<br>**

**Tip:** The <br> tag is useful for writing addresses or poems.

**Note:** Use the <br> tag to enter line breaks, not to separate paragraphs.

* In HTML, the <br> tag has no end tag.
* In XHTML, the <br> tag must be properly closed, like this: <br />.

**HTML5 semantic elements (page 95)**

* header : The header for a page
* section : generic section of a document that doesn’t indicate the type of content
* article : composition like an article in the paper
* nav : section of a page that contains links to other pages or place holders
* aside section of a page like a sidebar that is related to the content that’s near it
* figure : An image, table, or other component that’s treated as a figure
* footer : the footer for a page

Older Practices

<div> : block element that provides a container for other elements

<span> : An inline element that lets you identify text that can be formatted with CSS

HTML5:

Instead of div use semantic elements (structure of a page is more apparent)

Instead of span use elements that identify the contents such as : cite code q

HTML5 Browser ratings (perfect score 500)

Website: <http://www.html5test.com>

Class: Test apps on Chrome & 1 other browser like firefox or ie (ie is least compatible)

Production: test on all 5 browsers including older versions that are still in use

**Links:**

* a : anchor element: defines a link to another URL. When the user clicks on the text that’s displayed by the tag, the browser requests the page that is identified by the href attribute of the tag
* href : specifies the URL for the link
* Relative URL : specified in href attribute, relative to the current directory, current HTML page or relative to the web server’s directory for web apps
* Absolute URL : complete URL with name of host or IP address of host

**Images:**

* Img : Specifies how to place a PNG, GIF or JPEG image
* Attributes
* src : Specifies the relative or absolute URLL for the GIF or JPEG file
* alt : Specifies the text that’s displayed when the image can’t be displayed
* height : specifies the height of the image in pixels
* width : specifies the width of an image in pixels

Three types of Image formats

* Portable Network Graphics (PNG)
* Graphic Interchange Format (GIF)
* Joint Photographic Experts Group (JPEG or JPG)

Typical uses for images

* JPEG or JPG for photographs and scans
* PNG and GIF files for other types of images such as logos

**Table consists of rows and columns**

**Cell : the intersection of a row & a column can hold data**

**Table elements:**

* table : defines a table
* tr : defines a row
* th : defines a header cell within a row
* td : defines a data cell within a row

**td element attributes:**

* colspan : Specifies the number of the columns that the cell spans
* rowspan Specifies the number of rows that the cell spans

**CSS:**

**3 ways to provide styles**

* External style sheet

<link rel=”stylesheet” href=”styles/main.css”>

* Embedded styles

<style>

body{

Font-family: Arial, Helvetica, sans-serif; font-size: 100%;

}

h1 {

font-size: 250%;

}

* Inline styles : Use the style attribute of an element

<span style=”color: red; font-size: 14pt;”>Warning!</span>

**The sequence in which styles are applied**

* Styles from an external style sheet
* Embedded styles
* Inline styles

**Two external style sheets**

* From the first to the last

**Best practice**

* to use external style sheets because that leads to better separation of concerns
* specifically you separate the content from the formatting
* makes it easy to use the same style for multiple pages
* (embedded & inline you have to copy the style to each page)
* The last rule overrides the earlier rules

**Element, class and id selectors**

* You code a selector for all elements of a specific type by naming the element. This is referred to as a type or element selector
* You code a selector for an element with a class attribute by coding a period followed by the class name. Then, the rule set applies to all elements with that class name. This is known as a class selector.
* You code an id selector for an element with an id attribute by coding a pound sign(#) followed by the id value. This is known as an id selector.

**Rule sets, selectors, rules and comments**

1. A CSS rule set consists of a selector and one or more rules within braces
2. A CSS selector consists of the identifiers that are coded at the beginning of the rule set. If more than on selector is coded for a rule set, the selectors are separated by commas
3. A CSS rule consists of:

* Property
* Colon
* Value
* Semicolon (optional but a best practice
* Make code easier to read with spaces, identation & blank lines within a rule set
* CSS comments begin with /\* end with \*/ single or multiple lines

Common properties for formatting tables

* border-collapse : A keyword that determines whether space exists between the borders of adjacent cells. Possible values are: “separate” and “collapse”. Default is separate
* padding : the space between the cell contents and the outer edge of the cell
* text-align : the horizontal alignment of text
* verticle-align : the vertical alignment of text
* with HTML5 you should use CSS not HTML to format tables

**Form**

* A form contains one or more controls such as text boxes, buttons, check boxes and list boxes

Elements for working with a simple form

* form : defines the start and end of the form
* input defines the input type

Attributes of the form element

* action : Specifies the URL of the servlet or JSP that’s called when the user clicks on the submit button.
* method : Specifies the HTTP method that the browser uses for the HTTP request. The default method is the GET method, but the POST method is also commonly used, especially when the request includes data that’s saved to the server
* name : control attribute : Specifies the name of the control. When writing Java code,use this attribute to refer to the control
* value : control attribute : the default value of the control. This varies depending on the type of control. For a text bos, this attribute sets the text that’s displayed in the box. For a button, this attribute sets the text that’s displayed on the button.
* type : text control : specifies the type of input control. Traditional values are “text”, “password” and “hidden”. HTML5 intorduced some new types like “email” for a text box that should receive an email entry.
* name : text control : Specifies the name of the control. This is the name that is used to refer to the data in the control from a servlet or JSP
* value : text control : Specifies the value of data in the control
* maxlength : text control :Specifies the maximum number of characters that can be entered into the text box
* required : text control : specifies that the user must enter a value for a text bos. If the user submits the form and the field is empty, the browser displays its default error message.

**Text Box**

* Use the type attribute to specify the type text box
* A value of “text” creates a standard text box
* A value of “email” created a special box for email addresses (HTML5)
* A value of “password” creates a password box that displays astericks instead of text
* A value of “hidden” creates a hidden field that stores a name and value that’s sent to the server but isn’t shown by the browser

**Button**

* The type attribute identifies the type of button to be used.
* A type attribute of “submit” creates a submit button that activates the action attribute of the form when it is clicked
* A type attribute of “reset” creates a reset button that resets all controls on the form to their default values when its clicked
* A type attribute of “button” creates a generic button that can be used to trigger JavaScript actions.

Attributes of these controls:

* type : Specifies the type of control. A value of “checkbox” creates a checkbox while a value of radio creaes a radio button
* checked : Selects the control. When several radio buttons share the same name, only one radio button can be selected at a time

Check boxes and radio buttons

* Use check boxes to allow the user to supply a tru/false value
* Use radio buttons to allow a user to select one option from a group of options. To create a group of buttons, use the same name for all of the radio buttons
* If you don’t group radio buttons,more than one can be on at the same time

Attributes of the select element

* size : The number of items to display in the control. If the value is 1, the control will be a combo box. If more than 1, the control will be a list box
* multiple : If coded, the suser can select more than one option. This is only valid with a list box.

Attributes of the option element:

* selected : Selects the option

Combo and list boxes

* A combo list box provides a drop-down list that lets the user select a single button
* A list box provided a list of options and lets the user select more than one option if the multiple attribute is coded
* To select more than one option from a list box, the user can hold down the Ctrl key on a windows system of the Command key on a Mac and then click the options