**Document Object Model**

 Document Object Collections

o anchors[]

o forms[]

o images[]

o links[]

 Document Object Properties

o cookie

o documentMode

o domain

o lastModified

o readyState

o referrer

o title

o URL

 Document Object Methods

o close()

o getElementById()

o getElementsByName()

o getElementsByTagName()

o open()

o write()

o writeln()

**Servlets**

* Java Object based on the Servlet API
* Runs in a server application to answer client requests; technically, servlets are not tied to a specific client-server protocol, but they are most

commonly used with HTTP and the term „servlet‟ is often used in the context of an “HTTP Servlet”

* Web-tier components in the Java EE architecture.
* Runs in, and is managed by, a web-tier container called the „Servlet Container‟
* Mapped to URLs to which clients send requests
* Typically asked with (among other things)
* Processing and/or storing data submitted vial HTML forms
* Generating dynamic content
* javax.servlet
  + Servlet, GenericServlet
  + ServletRequest, ServletResponse
  + ServletConfig, ServletContext
  + RequestDispatcher
* javax.servlet.http
  + HttpServlet
  + HttpServletRequest
  + HttpServletResponse
  + HttpSession
  + Cookie
* Servlet Processing
* Client sends a request to a web server URL that is mapped to a servlet. Web server passes on the request to the servlet container
* Servlet container checks if servlet is already loaded
* If it is not yet loaded, servlet container loads the servlet class and instantiates the servlet, and calls its init method.
* Servlet container invokes the servlet‟s service method, passing request and response objects as arguments
* Servlet processes the request using the response object to create the response, which is returned by the servlet container to the web server, which in turn sends the response to the client
* Subsequent request to the servlet will not require servlet re-instantiation, unless the servlet has been unloaded; before a servlet is unloaded, the servlet container invokes its destroy method.

Servlets or JSPs?

* The common practice is to leverage both technologies to implement the MVC design pattern

Model-View-Controller (MVC) Design Pattern

* Model
* Represents business objects (logic and state)
* View
* Presentation of the model in some appropriate way
* Controller
* Mediates application flow

A sample web MVC framework can use

* JavaBeans for the model
* JSPs for the View
* Servlets for the controller

JSP Standard Tag Library (JSTL)

* Set of custom JSP elements that provide various programmatic functionality via markup syntax
* Core Tag Library
* variable support, flow control, URL management
* SQL Tag Library
* Database connections, queries, updates
* Internationalization Tag Library
* Locate setting, message bundling, number formatting, date formatting
* XML
* Core XML processing, flow control, transformation
* JSTL Function
* String functions, collection lengths
* In addition to the JSTL, developers can also create their owntag libraries for commonly occurring tasks