

Explain the term CSS.

CSS is a web standard that describes style for XML/HTML documents. Cascading style sheet is used to control the layout of multiple pages at once. The CSS code is written in a separate file with extension .css. It is a mechanism to write and modify style definition in css file, and the changes will be reflected in all the web pages in which the css file is linked with link element. The link element should be placed in between <head> and </head> tags. Besides external CSS file, you can use css code at the embedded and inline style definition in web page.

Define the term "Rule" in respect of CSS.

The CSS rule is a way of writing a code that allows web designers to define styles definition easily and effectively for HTML pages. For example, <p style="font-weight: bold">some text</p>.

What is external Style Sheet? How to link?

External style sheet is a file containing only style definition code which can be linked with any number of HTML documents. This is the way of formatting the entire site. External style sheet must have extension .css i.e. style.css. The file is linked with HTML documents using the LINK element inside the HEAD element. The following is the way to use external style sheet in a page. <link rel="stylesheet" href="name_of_file.css" type="text/css" >.

```
<!DOCTYPE html>
<html><head>
<link rel="stylesheet" href="ignou.css" type="text/css" >
<style>p{ color:green;}</style>
</head><body>
<p style="color:red;">BCA:BCS053</p>
</body></html>
```

Inline Style: → Inline style sheet provide the finest level of control. This style sheet is written only for the single element. These style specifications appear within the opening tag of the element. The style specification format for Inline style sheet are as follows:

```
Style = "property_1: value_1; property_2: value_2; .....;property_n:
value_n;"
<!DOCTYPE html>
<html><head></head><body>
<table border="2"><tr><td>Normal TD Cell</td>
<td style="font-family: arial; border-style: solid; background-color: red"
> IGNOU:BCS053</td>
</tr></body></html>
```

Embedded Style Sheets: → This style is placed within the document. For this style sheet, CSS rule can appear inside the <head> element or between the <body> tag contained with a <style> element. The format for this embedded style is as follows: <style type="text/css"> rule list </style>

```
<html><head>
<style type="text/css">
body { background-color: #FF0000; color: #000000;}
h2, h3 {font-family: arial, sans-serif; color: #ffffff;}
p { color: #3300FF; font-size:20px;}
</style>
</head><body>
<header> HTML: Version 5.0
<hr><hgroup>
<h2>New features should be based on HTML, CSS and JavaScript</h2>
<h3>Reduce the need for external plugins (like Flash) </h3>
</hgroup></header>
<article>
<p>HTML5 should be device independent</p>
<p>The development process should be visible to the public</p>
</article>
<footer><hr> &copy; 2013 IGNOU</footer>
</body></html>
```

Define the CSS padding property.

The CSS padding properties define the space between the element border and the element content. In CSS, it is used to specify different padding to all four sides of an element. For example :

```
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
padding-left:50px;
```

The term „**box model**“ is used in CSS-based layout and design. Every element in HTML can be considered as a box, so the box model applies to all html and xhtml elements. Each box has four edges: the margin edge, border edge, padding edge, and content edge. Each edge may be broken down into a top, right, bottom, and left edge.

What is XML?

XML stands for eXtensible Markup Language. Very much like HTML, XML is also a markup language. The essence of XML is in its name:

Extensible :- It lets you define your own tags. Tags are displayed in the same order as they are defined in your xml document.

Markup :-The most important feature of any xml document is its tags, also called as elements. These are very much similar to the ones which are being used at html.

Language :-XML is a language similar to HTML, though more flexible giving you option to create your own custom tags. Basically we can call it as meta-language: it allows us to define or create other languages. For example, we can create RSS, MathML languages and tools such as XSLT.

Need for XML → As we know that HTML is designed to display elements in a web browser. It becomes cumbersome if you want to display the documents in some other language or upgrade the document for dynamic data display. XML is just suited for this purpose. It can be used in a variety of contexts. XML is used to store data in plain text format with strong support via Unicode, for the various languages of the world. This makes it easy to carry data irrespective of any platform. XML is also used as the base language for the communication protocols such as XMPP. It is also very simple and easy to learn and use. XML provides users the flexibility to define their own tags. XML as a document, does not do anything. It is the software which uses this document to connect with the Database and does the processing of data. XML is recommended by W3C as well, in February 1998 for the purpose of storing and transporting data. XML is so flexible and powerful that it is used in creating new Internet Languages such as XHTML, WML, SMIL etc.

XML Document Structure

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<test>
<text1>My First XML</text1>
<text2>Let's have coffee after dinner!</text2>
</test>
```

1)The First line, it is the XML declaration. This line defines the XML version and the encoding used.XML version is 1.0 here and the encoding is ISO-8859-1.

2)The next line is the root element of the document. It describes that this is the test document.

3)Next 2 lines describe the 2 child elements of the root.

4)The last line describes the end of the root element.

Comments can be written in XML in the same way as in HTML: <!--This is a comment-->

What is a valid XML document?

If a document is structurally correct then it can be called as valid XML documents. A valid document conforms to the predefined rules of a specific type of document: Please note that a valid XML document is implicitly well-formed, but well-formed document may not be valid.

XML Syntax Rules:

XML Tags are case sensitive

All XML elements shall have closing tags

XML Elements shall be properly nested

XML Attribute values must be in quotes.

Some Special characters have entity reference. It means that few special characters are already used as HTML pre-defined tags. So If we need to use those characters, we shall write them in a different way as below:

```
&lt;    <    less than
&gt;    >    greater than
&amp;  &    ampersand etc....
```

XML Namespaces :-> XML allows you to define your own documents thus providing flexibility and scope. But it's a strong possibility that, when combining XML content from different sources, there could be clashes between codes in which the same element name serve very different purposes. For example, if you are using a bookstore, your use of <title> tag in xml may be for the used for tracking the book's title. A insurance agent may use the <title> in a different way – to track the income tax records of its clients. XML namespaces provide a method to avoid element and attribute name conflicts in any XML document. XML namespaces attempt to keep different semantic usages of the same XML elements separate and unambiguous. An XML namespace is declared using the reserved XML pseudo-attribute xmlns or xmlns:prefix, the value of which must be a valid namespace name. It has the syntax as follows:

xmlns:prefix="namespaceURI"

A namespace is named as Uniform Resource identifier (URI), identifying an internet resource uniquely, though doesn't point to an actual location of the resource. Uniform Resource Locator (URL) is the most common form of URI used. Another one is Universal Resource Name (URN) which is not so widely used. Any namespace declaration that's placed in a document's root element becomes available to all the elements in that document. Please note that namespaces have scope. Namespaces affect the element in which they are declared, as well as all the child elements of that document.

```
<root xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">
<h:table> <h:tr> <h:td>Apples</h:td> <h:td>Bananas</h:td> </h:tr>
</h:table> <f:table> <f:name>African Coffee Table</f:name>
<f:width>80</f:width> <f:length>120</f:length> </f:table> </root>
```

What is Document Type Definitions(DTD) .Explain?

A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of elements and attributes. If DTD is supplied with the XML file, then the XML parser will compare the content of the document with the rules that are set in the DTD. If the document does not conform to the rules specified by the DTD, the parser raises an error and indicates where the processing failed.

If the DTD is declared **inside the XML file**, it should be wrapped in a DOCTYPE definition with the following syntax:

```
<!DOCTYPE root-element [element-declarations]>
```

OR

If the DTD is declared in an **external file**, it should be wrapped in a DOCTYPE definition with the following syntax:

```
<!DOCTYPE root-element SYSTEM "filename">
```

Benefits of using DTD:

1. DTDs allow the declaration of standard public entity sets for publishing characters.
2. DTD support is ubiquitous due to its inclusion in XML 1.0
3. DTDs are terse compared to element-based schema and thus present more information in a single screen.
4. DTDs define a document type rather than those defined by the namespaces. It helps in grouping all document constraint in a single collection.

Limitations of DTDs:

1. They have no explicit support for namespaces. All DTD declarations are global, so you can't define two different elements with the same name, even if they appear in different contexts.
2. DTDs have support only for rudimentary datatypes.
3. DTDs lack readability.
4. DTDs use regular expression syntax to describe schema. This is less accessible to programmers than an element-based syntax may be.

The **Document Object Model (DOM)** is language and platform independent interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document. The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

The HTML DOM defines the **objects** and **properties** of all HTML elements, and the **methods** to access them. In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

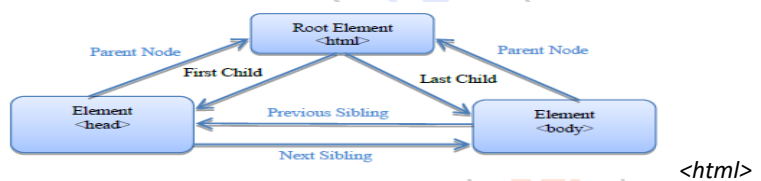
What is DOM and what are the objectives of DOM?

ANS:--The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. One important objective for the Document Object Model is to provide a standard programming interface that can be used in a wide variety of environments and applications.

Relationship between parent node, child, sibling

The nodes in the node tree have a hierarchical relationship to each other. The node directly above a node is the parent node of that node. The nodes at the same level and same parent are sibling nodes. The nodes that are one level directly below another node are the children of that node. So, to summarize

- i) In a node tree, the top node is called the root
- ii) Every node has exactly one parent, except the root (which has no parent)
- iii) A node can have any number of children
- iv) Siblings are nodes with the same parent



```
<html>
<head> <title>Welcome to IGNOU</title> </head> <body> <h1>Courses
Offered</h1> <p>Computer Science</p> </body> </html>
```

What are the functionalities performed by onload() and onUpload()?

onload() and onUpload() are two functions that get activated or the event gets triggered when the user enters or leaves the page. The onload() event is used to verify and check the user's visit according to the browser's type and it loads a version of the event. It provides the web page information that allows easy access to the website and consists of the information regarding the event that is triggered. Onload() and onUpload() events use the cookies to hold down the values given by the users when it enters or leaves the page.

What is AJAX? List out the differences between AJAX and JavaScript.

Ajax is abbreviated as Asynchronous Javascript and XML. It is a new technique used to create better, faster and more interactive web systems or applications. Ajax uses asynchronous data transfer between the Browser and the web server. Here on sending request to the server, one needn't wait for the response. Other operations on the page can be carried out Hence, Asynchronous. On the other hand, JavaScript sends an HTTPRequest to the server and waits for the XML response e.g. populating State field. Using JavaScript we need to use the "Onchange" event where as using ajax, the request is just sent to populate the state list. Other operations can be carried out on the page. Ajax is a part of JavaScript programming. JavaScript is used to manage and control a web page once downloaded. Ajax does not need to wait for the whole page to download.

What are the advantages of AJAX?

Following are the advantages of Ajax:

- Bandwidth utilization – It saves memory when the data is fetched from the same page.
- More interactive
- Speedier retrieval of data

What are the limitations of Ajax?

Limitations of AJAX:

- Back functionality cannot work because the dynamic pages don't register themselves to the browser's history engine
- The page cannot be bookmarked if implemented using Ajax.
- If JavaScript is disabled, Ajax will not work.
- Because different components of the pages are loaded at different times it may create confusion for the user.

Describe the structure of WAP Protocol Stack

It consists of five layers, namely: application layer, session layer, transaction layer, security layer and datagram layer

Application Layer (Wireless Application Environment, WAE)

WAE provides an application environment intended for the development and execution of portable applications and services. WAE User Agents are designed to understand text in encoded WML (Wireless Markup Language) and compiled WMLScript. WAE consists of WML and WMLScript as the two main building blocks of the user agents located on the client side

Session Layer (Wireless Session Protocol, WSP)

WSP supplies methods for the organized exchange of content between client/server applications.

Transport Layer (Wireless Transport Protocol WTP)

WTP is responsible for providing reliable transmission of WSP data packets between the client and the server over a wireless medium.

Security Layer (Wireless Transaction Layer Security WTLS)

WTLS is an optional layer and is the solution to the security issue, provided by the WAP Forum. WTLS is based on SSL(Secure Socket Layer)and when present provides services that ensure privacy, server authentication, client authentication and data integrity.

Network Layer (Wireless Datagram Protocol WDP)

WDP is the bottom most layer of the WAP stack. It is modelled after User Datagram Protocol (UDP) and is a datagram oriented protocol. WDP shields the upper layers from the bearer services like SMS, CMD etc. provided by the network and hence allows applications a transparent transmission of data over different bearers. WDP must have bearer specific implementation since it is the only layer that has to interface various bearer networks.

Working of WAP Model

When it comes to actual use, WAP works like this: The user selects an option on their mobile device that has a URL with Wireless Markup language (WML) content assigned to it. The mobile device uses phone network to send the URL request to a WAP gateway, using the binary encoded WAP protocol. This WAP request is translated into a conventional HTTP request for the specified URL, and sent to the Internet by the WAP gateway. This HTTP request is now picked up by the appropriate Web server. The request is processed by the server. If the URL refers to a static WML file, the server delivers it. If a CGI script is requested, it is processed and the content is returned as usual. The HTTP header is added to the WML content and returned to the gateway by the Web server. The WML is compiled into binary form by WAP gateway. The WML response is sent back to the phone by WAP gateway. The WML is received by the mobile device via the WAP protocol. The WML is processed and the content is displayed on the screen by micro-browser.

How Events Work?

When events happen to an HTML element in a web page, it checks to see if any event handlers are attached to it. If the answer is yes, it calls them in respective order, while sending along references and further information for each event that occurred. The event handlers then act upon the event. DOM elements can be nested inside each other. And somehow, the event handler of the parent works even if we click on its child. The reason for this is *event bubbling*. There are two types of event order:

Event bubbling: It begins by checking the target of the event for any attached event handlers, and then bubbles up through each respective parent element until it reaches the HTML element. The main principle of bubbling states that after an event triggers on the deepest possible element, it then triggers on parents in nesting order.

Event capturing starts with the outer most element in the DOM and works inwards to the HTML element the event took place on and then out again. For example, a click in a web page would first check the HTML element for onclick event handlers, then the body element, and so on, until it reaches the target of the event.

```
<card id="card1" title="First card" ontimer="#card2"> <timer value="10"/> <p>
Welcome to IGNOU...<br/> This is First Card <do type = "accept" label="Next">
<go href="#card2"/> </do> </p> </card> <card id="card2" title="Second card"
ontimer="#card1"> <timer value = "20"/> <p> Let us learn to make a Wireless
World <br/>
```

```
This is the Second Card <do type = "prev" label="Back">
<prev/></do> </p> </card>
```

Write a function to execute a script that changes the color of an image when moving the mouse over / out of the image?

```
<html> <head> <script> function bgChange(bg) {
document.body.style.background=bg; } </script> </head> <body>
<b>Mouse over the squares and the background color will change!</b>
<table width="200" height="100"> <tr> <td
onmouseover="bgChange('red') " onmouseout="bgChange('transparent') "
bgcolor="red"> </td> <td onmouseover="bgChange('blue') "
onmouseout="bgChange('transparent') " bgcolor="blue"> </td> <td
onmouseover="bgChange('green') "
onmouseout="bgChange('transparent') " bgcolor="green"> </td> </tr>
</table> </body> </html>
```

WML TASK:→ A WML task is an element that specifies an action to be performed by the browser, rather than something to be displayed. For example, the action of changing to a new card is represented by a <go> task element, and the action of returning to the previous card visited is represented by a <prev> task element. Task elements encapsulate all the information required to perform the action. WML provides four elements to handle four WML tasks called go task, pre task, refresh task and noop tasks.

go task:→ The go element declares a go task, indicating navigation to an URL. A go executes a push operation on the history stack. As the name suggests, the <go> task represents the action of going to a new card.

```
<card title="GO Element">
<p><anchor>Semester 1 : <go href=semester1.wml"/></anchor> </p>
</card>
```

The <prev> Task: The <prev> task represents the action of returning to the previously visited card on the history stack. When this action is performed, the top entry is removed from the history stack, and that card is displayed again, after any <setvar> variable assignments in the <prev> task have taken effect. If no previous URL exists, specifying <prev> has no effect.

```
<card title="Prev Element">
<p> <anchor> Previous Page :<prev/> </anchor> </p> </card>
```

The <refresh> Task: The <refresh> task performs the variable assignments specified by its <setvar> elements and then redisplay the current card with the new values. It is most often used to perform some sort of "reset" action on the card. The <go> and <prev> tasks perform the same action just before displaying the new card.

```
<card title="Referesh Element">
<p> <anchor> Refresh this page: <go href="test.wml"/> <refresh>
<setvar name="x" value="100"/> </refresh> </anchor> </p> </card>
```

The <noop> Task: The purpose of the <noop> task is to do nothing (no operation). This attribute is useful for overriding deck-level do elements. The only real use for this task is in connection with templates

```
<card title="Noop Element">
<p> <do type="prev" label="Back"> <noop/> </do> </p> </card>
```

WML – Events:→ Events can be used to make WAP applications dynamic. Event in ordinary language as the name suggests, is simply the happening of something notable. In programming event is identical in meaning, but with one major difference. When something happens in a computer system, the system itself has to (1) detect that something has happened and (2) know what to do about it. WML language also supports events and we can specify an action to be taken whenever an event occurs. This action could be in terms of WMLScript or simply in terms of WML. WML supports following four event types:

- **onenterbackward:** This event occurs when the user hits a card by normal backward navigational means. That is, user presses the Back key on a later card and arrives back at this card in the history stack.
 - **onenterforward:** This event occurs when the user hits a card by normal forward navigational means.
 - **onpick:** This is more like an attribute but it is being used like an event. This event occurs when an item of a selection list is selected or deselected.
 - **ontimer:** This event is used to trigger an event after a given time period.
- WML <timer> Element:** A timer is declared inside a WML card with the <timer> element. It must follow the <onevent> elements if they are present. (If there are no <onevent> elements, the <timer> must be the first element inside the <card>.) No more than one <timer> may be present in a card.

MVC Architecture: → An important architecture that is used by Java for web application development. This architecture is known as Model-View-Controller (MVC). This architecture can also be used for the development of a web application. The MVC architecture has three functional component of a web application that communicates through a user as well as among them. The following are the basic components of MVC architecture:

Model: In the context of MVC a Model defines the data model and its access controls. The main role of this component is to represent data and perform updates on it as per the defined rules. The main responsibility of this component is to accept user requests and the data entered by the user and perform the necessary data related function.

View: The function of the view will be to accept the data from the Model and convert it to form that can be seen by a user in a user friendly way. Thus, view is responsible for displaying web pages for the user. Any change in the model should also change the view. This can be achieved using two processes - 1) Push processes, 2) Pull process

Controller: A user may be allowed to interact with the web pages created by the View component. This interaction may be in the form of selection of options of Menu, pull down lists, check boxes etc., filling up data in text boxes etc. It is the controller which accepts this data from the user and initiates suitable actions that should be carried out by the models.

The layers and their basic functions 3-tier architecture are: (short note)

Presentation Layer/Tier: The presentation tier of the three tier architecture interfaces with the client. The presentation tier is responsible for displaying the information to the client as well as extracting information from the client. Technically, for a web based application the presentation tier resides on the web server on request the presentation layer displays interactive web pages in the browser of client computers.

Application Logic Layer/ Tier: This is also called the middle ware. This layer is primarily responsible to provide business rules, sharable components of a web application, access control etc. This layer shields the data layer from direct use of the clients. This layer provides an interface between the presentation and data layer.

Data Layer/Tier: The data for a web application may be hosted on a database management system or a file system. The data layer controls the integrity of data residing on some data storage system. The application logic sends queries to data layer which sends back the query results to the application logic layer.

1 Explain the term JSP page.

JSP page contains JSP technology specific elements as well as static HTML or XML tags. A JSP page has extension .jsp which tells the web server that the JSP engine will process elements on this page.

What are the advantages of JSP over Servlet?

Both Servlet and JSP are server side technology but the development process of JSP page is easier than Servlet. The content and display logic is separated in a JSP page and you can insert directly java code inside the JSP page while in servlet, you can write java code with plain html using plenty of out.println() statement for front end display of the page which is very tedious work.

Another advantage of JSP page over the Servlet is that the JSP page can use custom tags while in Servlet, it is not possible. The power of JSP is to provide a framework for Web application development.

Describe the _jspService() method.

The _jspService() method is defined in javax.servlet.jsp.HttpJspPage interface and it is invoked every time a new request comes to a JSP page. This method takes HttpServletRequest and HttpServletResponse objects as an arguments. It returns no value. A page author cannot override this method. It is defined automatically by the processor.

Write the basic steps for processing JSP request.

When a client requests a JSP page, the browser sends a request to web server which is forwarded to the JSP engine. If it is a new request from the client, then it translated and compiled the JSP engine into a servlet. Now, servlet is ready for servicing the client request and generates response which returns back to the browser via web server. For the second time same request from the client including browser and web server request to JSP engine, the JSP engine determines the request that the JSP-Servlet is up to date, it immediately passes control to the respective JSP-Servlet.

Explain the life cycle of JSP.

When the JSP is first accessed, it is translated into corresponding servlet (i.e. java class) and compiled, then JSP page services request as a servlet. The translation of JSP page is done by the JSP engine of the underlying web container/servlet container (eg. Tomcat). The life cycle of JSP page is controlled by three methods i.e. jspInit(), _jspService() and jspDestroy().

jspInit() - The jspInit() method is same as the init() method of servlet and it is called only once during life cycle of a JSP/Servlet. It is used to initialize objects and variables that are used throughout the life cycle of JSP. This method is defined in JspPage interface. This method is invoked when the JSP page is initialized. Once the JSP page is initialized, the getServletConfig() method will return the desired value. It has no parameters, return no value and thrown no exceptions. The signature of the method is as follows:

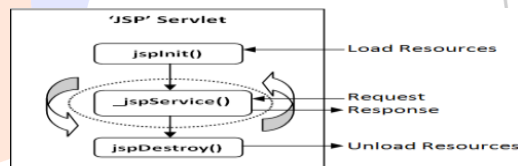
```
public void jspInit() { // Initialization code }
```

_jspService() - _jspService() is the method which is called every time the JSP page is requested to serve a request. This method is defined in the javax.servlet.jsp.HttpJspPage interface. This method takes HttpServletRequest and HttpServletResponse objects as an arguments. The _jspService() method corresponds to the body of the JSP page. It is defined automatically by the processor and should never be redefined by the JSP author. It returns no value. The underscore ('_') signifies that you can not override this method. The signature of the method is as follows:

```
public void _jspService(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws javax.servlet.ServletException, java.io.IOException
    { // services handling code }
```

jspDestroy() - The jspDestroy() is invoked when the JSP page is to be terminated. It is synonymous with the destroy() method of a servlet. It has no parameters, return no value and thrown no exceptions. Override jspDestroy() when you need to perform any cleanup, such as releasing database connections or closing open files. The signature of the method is as follows:

```
public void
jspDestroy(){ //
cleanup code }
```



What is difference between <JSP:include> and include directive?

This include directive is used to insert text and code at the translation time. You can not pass the parameter in this directive. The syntax is: <%@ include file="relativeURL" %>

The include action is used to include static as well as dynamic content in the current JSP page at the request or run time. You can pass the parameter using param action. The syntax is: <jsp:include page="relativeURL"/> or <jsp:include page="relativeURL" /> <jsp:param ..></jsp:include>

What is declaration scripting element? Explain with example.

The declaration tag is used to declare variables and methods which are placed inside the declaration part of the generated servlet. In JSP page, the declaration tag is start with <%! and end with %>. The code inside this tag must end with semicolon. For example: <html><body><p>Example of declaration tag </p><%! private int i = 4; %><%! private int square(int i) { i = i * i; return i; %><%= square(i) %> </body></html>

Tag Library Descriptor (TLD) file - A tag library descriptor file is an xml document. It defines a tag library and its tags. The file extension of this file is .tld. It contains one <taglib> root element and tlibversion, jspversion, shortname, tag are sub elements of the taglib element. The tag is the most important element in TLD file because it specifies the name of the tag and class name of the tag. You can define more than one tag element in the same TLD file.

Deployment Descriptor file (web.xml) - The deployment descriptor is an xml file that specifies the configuration details of the tag. The most important element for custom tag in web.xml file is <taglib-location>. Using the web.xml, the JSP container can find the name and location of the TLD file.

Write a JSP program for odd numbers.

```
<%@ page language="java"%><html><head><title>Odd number example
written in JSP</title></head> <body><p>Odd number are:</p> <% for(int
```



```
i=0;i<=100;i++) { if((i%2)!=0) { out.println(i); out.println(""); }
}%></body></html>
```

JSP Compilation Time Error :-> The first type of error comes at translation or compilation time when JSP page is translated from JSP source file to Java Servlet class file. These errors are usually the result of compilation failures due to some syntax error or spelling mistakes. These errors are known as translation time errors and are reported to the client browser with some error status code 500. The compilation time error is handled by JSP engine. e.g

```
<% out.println("Hello Students in JSP World"); >
```

JSP Request Time Error :-> The second type of JSP error which is called as request time error occurs during run time or request time. These run time errors results in a form of exception. These run time exception can be caught and handled by the calling JSP. These exceptions occur in the body of the JSP Page.

URL Rewriting :-> URL rewriting is used in place of the cookies when the browser cookies are functionally turned off. It is used to maintain the session. In this approach, the token is embedded in each URL. In each dynamically generated page, the server embeds an extra query parameter or extra path information. When client submits request using such URL, the token is retransmitted to the server.

Session Object :-> This is easiest to use and powerful for session management. Session and Cookies go hand in hand, the only difference between the cookies and session is that the cookie is stored in the client side and session is stored on server side which is another method to handle a session. Any web application can use the concept of sessions and Session object is used to store data for a particular client when that client is connected to server.

Explain MIME?

Multipurpose Internet Mail Extensions (MIME) defines mechanisms for sending other kinds of information in email like images, sounds, movies, and computer programs. MIME is an Internet standard that extends the format of email to support such as text in character sets other than ASCII, non-text attachments, message bodies with multiple parts and header information in non-ASCII character sets

Write the process of sending an email from client program to destination server.

When you send an email from source to destination server the following step are follows:

Email client Program sends the message to Email server.

Email server contact to the recipient's email server provided in the email address.

Email server checks that user name is valid or not.

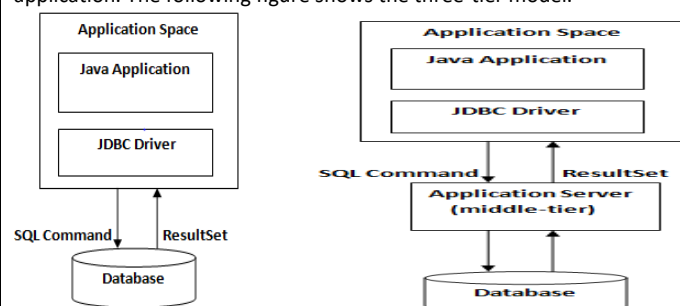
If it is valid send email to the address's email server.

When recipient log on his mail account, gets his Email.

Need for JDBC :-> Today, the most widely used interface to access database is ODBC. ODBC stands for Open Database Connectivity. It is a standard in relational database connectivity published by Microsoft (though originally developed jointly by Microsoft and Sybase). ODBC cannot be use directly with Java program because it uses a C interface and it also makes use of Pointers which have been removed from Java. The JDBC comes after the ODBC but JDBC is a Java API and contains Java interface for working with SQL

Two- and Three-Tier Database Access Models:- The JDBC provides support for two- and three-tier database access models. In **two-tier model**, java application is directly connect to the database. This is done through the use of JDBC driver. The JDBC driver sends commands directly to the database and result of these commands are sent back directly to application. The following figure shows the two-tier model.

three-tier JDBC model, Java application can not connect directly to the database. A middle-tier comes between the Java application and database. When you use three-tier model, JDBC driver of Java application sends commands to middle-tier, which in turn sends commands to the database. The result of these commands is sent back from the database to middle-tier and middle-tier sent back to the application. The following figure shows the three-tier model.



PCDATA here means **parsed character data**. PCDATA is text that WILL be parsed by a parser. The text will be examined by the parser for entities and markup. Tags inside the text will be treated as markup and entities will be expanded.

However, parsed character data should not contain any &, <, or > characters; these need to be represented by the & < and > entities, respectively. **CDATA** means character data. Think of character data as the text found between the start tag and the end tag of an XML element. CDATA is text that will NOT be parsed by a parser. Tags inside the text will NOT be treated as markup and entities will not be expanded.

Use the following XML documents and prepare a DTD or an XML schema

```
<?xml version="1.0"?>
<test>
<text1>My First extensible XML</text1>
<text2>Let's have coffee after dinner!</text2>
<text3>After that, shall we go for a long drive?</text3>
</test>
```

DTD File is as follows: **test.dtd**

```
<!ELEMENT test (text1,text2,text3)>
```

```
<!ELEMENT text1 (#PCDATA)>
```

```
<!ELEMENT text2 (#PCDATA)>
```

```
<!ELEMENT text3 (#PCDATA)>
```

Now XML Schema file is as follows: **test.xsd**

```
<?xml version="1.0"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
targetNamespace="http://www.w3schools.com"
```

```
xmlns="http://www.w3schools.com" elementFormDefault="qualified">
```

```
<xs:element name="test">
```

```
<xs:complexType> <xs:sequence> <xs:element name="text1"
```

```
type="xs:string"/> <xs:element name="text2" type="xs:string"/> <xs:element
```

```
name="text3" type="xs:string"/> </xs:sequence> </xs:complexType>
```

```
</xs:element> </xs:schema>
```

WML <timer> Element: A timer is declared inside a WML card with the <timer> element. It must follow the <onevent> elements if they are present. (If there are no <onevent> elements, the <timer> must be the first element inside the <card>.) No more than one <timer> may be present in a card.

```
<<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="card1" title="First card"
```

```
ontimer="#card2">
```

```
<timer value="10"/>
```

```
<p>
```

```
Welcome to IGNOU...<br/>
```

```
This is First Card
```

```
<do type = "accept" label="Next">
```

```
<go href="#card2"/>
```

```
</do>
```

```
</p>
```

```
</card>
```

```
<card id="card2" title="Second card"
```

```
ontimer="#card1">
```

```
<timer value = "20"/>
```

```
<p>
```

```
Let us learn to make a Wireless World <br/>
```

```
This is the Second Card
```

```
<do type = "prev" label="Back">
```

```
<prev/>
```

```
</do>
```

```
</p>
```

```
</card>
```

```
</wml>
```

Write a program in JSP to store name and roll into a database(Oracle).

Index.html

```
<html>
<head></head>
<body>
<form method="post" action="insertCode.jsp">
Name : <input type="text" name="TxtName"><br>
Roll : <input type="text" name="TxtRoll"><br>
<input type="submit" value="Save" name="option">
</form> </body> </html>
```

insertCode.jsp

```
<%@ page contentType="text/html; charset=utf-8" language="java" import="java.sql.*" import="java.util.*" errorPage="" %>
<%
Connection conn = null;
Statement s = null;
Class.forName("oracle.jdbc.driver.OracleDriver");
conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "system", "manager");
String opt = request.getParameter("option");
String sname = request.getParameter("TxtName");
String sroll = request.getParameter("TxtRoll");
try{
    s = conn.createStatement();
    s.executeUpdate("insert into Student(NAME,ROLL) values('"+sname+"','"+sroll+"')");
    out.println("Data Save Successfully");
}
catch(Exception e)
{    out.println(e); }
%>
```

Program on Date function

```
<%@ page contentType="text/html; charset=utf-8" language="java"
import="java.util.*" errorPage="" %>
<html>
<head>
<title>Date Program</title>
</head>
<body>
    Date is : <%= new java.util.Date() %>
</body>
</html>
```

JSP comment

JSP comment marks the JSP container should ignore. A JSP comment is useful when you want to hide on comment. Syntax:

```
<% -- This is JSP Comment -- %>
```

Example

```
<%@ page contentType="text/html; charset=utf-8"
language="java" import="java.sql.*" errorPage="" %>
<html>
<head><title>Untitled Document</title></head>
<body>
Date is :
    <% -- This is JSP Comment -- %>
</body>
</html>
```

Print Hello world in JSP

```
<html>
<body>
    <% out.print("Hello World in JSP"); %>
</body>
</html>
```

Java Server page(JSP) is server side technology that enables web programmers to generate web page dynamically in response to client request. Among many server side technologies used to build dynamic web applications. It is basically high-level extension to the java servlets technology. It enable us to insert pure java code in an HTML document directly by making use of special JSP tags most of which start with <% and end with %>. These pages are executed under the supervision of an environment called web-container. The web container first compiles the page on the server and generate a servlets, which is loaded in the Java Runtime Environment(JRE) automatically.

Advantage of JSP:

- JSP is easy to developed server-side program.
- It separate front-end development with business logic.
- JSP uses all facilities of servlets since it is built on top of Java Servlets API so it can use J2EE technology like JDBC, JNDI, EJB, etc.
- JSP can be used with servlets to process business logic.

JSP Directives

JSP directive is used to control overall action. There are three type directives

1. <% @ Page ___ %> Defines page dependent attributes, such as scripting language, error page and buffering requirements.
2. <% @ include ___ %> Includes a file during the transition.
3. <% @ taglib ___ %> Declares a tag library containing custom actions used in page.