

Web Development Process, Front End Tools

Syllabus Topics

Introduction to web technology, Internet and WWW, Web site planning and design issues,

HTML: structure of html document, HTML elements: headings, paragraphs, line break, colors and fonts, links, frames, lists, tables, images and forms, Difference between HTML and HTML5.

CSS : Introduction to Style Sheet, Inserting CSS in an HTML page, CSS selectors.

XML : Introduction to XML, XML key component, Transforming XML into XSLT.

DTD : Schema, elements, attributes, Introduction to JSON.

Syllabus Topic : Introduction to Web Technology, Internet and WWW, Web Site Planning and Design Issues

1.1 Introduction to Web Technology

1.1.1 Internet and WWW

(A) Internet

Q. Explain the term Internet. (2 Marks)

- Now a days, we observe that internet is everywhere. There may be hardly any mobile used by young generation which does not have internet service. Number of TV shows and magazines are devoted to internet.
- Internet becomes a basic need of people. If for some time internet gets disconnected, people really get very disturbed and nervous just like some big problem get arise. Up till now you have used internet for various purposes, but now in this subject we are going to learn how to use this internet to become a web developer.
- **Definition :** The Internet is nothing but a global system which consists of inter-connected networks of computers that uses the Internet protocol suite named TCP/IP (Transmission Control Protocol/Internet Protocol) to link the devices throughout the world. It is a network of networks which contains various types of networks like private, public, government, business, and academic from local to global scope.
- The Internet is linked by a large array of wireless, electronic and optical technologies of networking.

- The Internet provides a broad range of information resources and services like the World Wide Web (WWW) applications, telephony, email , sharing of files and connected hypertext documents.
- In 1960, the United States Federal Government form a research commission to create robust, fault-tolerant communication through computer networks
- The primary form of network called ARPANET was formed in the year of 1980 which provides interconnection between the regional academic and military networks.
- In 1980, the National Science Foundation Network provides funds to develop new networking technologies as well as the integration of various networks.
- In 1990 the commercial networks and enterprises get linked which facilitate the transition to modern Internet and creates an effective growth in network technology as different generations of personal, mobile and institutional computers were connected in the network.
- In 1990, the commercial networks and different enterprises are connected which leads to the beginning of transition to advanced modern Internet. This effectively connects the different computers like personal, institutional and mobile in the network.
- Internet reshaped and even redefined the previous traditional communication media like newspaper, radio, television, etc.
- The new Internet technology is adapted by book and newspaper. These communication media are even reshaped in new concepts like blogging, web feeds and online news aggregators. The new forms of personal



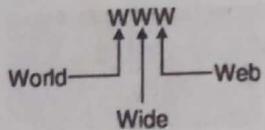
- interactions get accelerated through social networking, instant messaging and internet forums.
- Selling and buying of goods through Internet (online shopping) gives big opportunities to both small businesses as well as major retailers. It also gives a big comfort to consumers.

☞ Different uses of internet like

- (1) Social networking – chat
- (2) Email
- (3) Sharing of information
- (4) Getting live updates – news around the world
- (5) Virtual classrooms
- (6) Online Jobs
- (7) Remote access

(B) WWW

Q. Explain the term WWW. (2 Marks)



☞ Definition : WWW stands for World Wide Web. The World Wide Web sometimes only pronounced simply as Web, is a way of retrieving information over the Internet. It is model of information-sharing which is built on top of the Internet.

- HTTP protocol is used by the web to transmit data over the internet. Web service which allows applications to communicate with each other uses HTTP. These applications communicate to exchange business logic and use the Web to share information.
- A website is made of number of web pages. These web pages which are also known as web documents are interconnected with each other with the help of hyperlinks which we will learn in HTML. Such web pages can be accessed using various web browsers like Internet Explorer, Firefox, Google chrome etc. The web documents can contain simple text, graphics, audio and videos.

1.1.2 Web Technology

(A) Website

Q. Explain the term Website. (2 Marks)

- ☞ Definition :** A website is made of number of web pages. Simply a website is a file which is accessible anywhere in the world through internet.
- Websites are usually created for commercial purpose. As well there are various extensions of websites depending upon their prime purpose. Some of these are as follows:

- o .com – Commercial
- o .org – Organization
- o .net – Network
- o .biz – Business

- o .edu – Educational
- Also there are some extensions depending upon the countries:
 - o .in – India
 - o .us – America
 - o .uk – United Kingdom
 - o .nz – New Zealand
- Now a day, Web development can be definitely considered as a happening career option. So if you think to become a web developer, then first you should understand the entire technology of web.
- In any website there are basically two important aspects – Web Designing and web development (scripting).

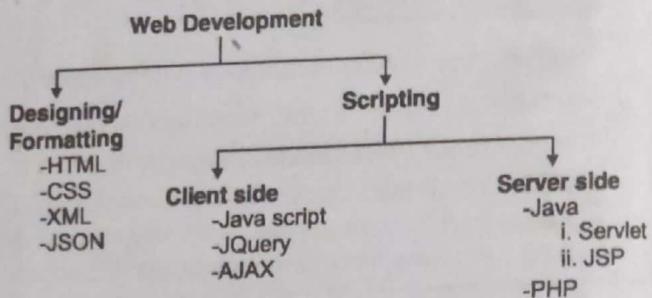


Fig. 1.1.1 : Web development

☞ Web Designing

- As the basic intention of website is always commercial, it is necessary that the website should be user friendly and attractive which is completely depending upon the design of website.
- For designing purpose various Scripting languages like HTML, CSS and Software or tools like Photoshop, CorelDraw, Flash etc are used.

☞ Web Development

- While developing software we write programs using programming languages. Such programs are also written in web development. These programs are known as scripts and the languages in which we write such scripts are known as Scripting Languages.
- Basically there are two types of scripts. To understand this we will take an example of email account form. To create an email account, we have to fill the email account form in which we have to give the username, password and some personal details.
- This email account form which contains different controls like textbox, checkbox, option buttons, submit buttons etc is designed using languages like HTML, CSS etc.
- Customized <tags> creation and data transfer is done by the languages like XML, JSON etc.
- You may remember that while filling such form it may be possible that we left some fields empty or may give some incorrect data. In such case the website gives error messages and forces us to fill or correct the data.

It means that the website has written the code (script) to check the data given by us. It is known as validation checking. This script directly interacts with the client (website users) hence called as client side script. The language in which such script is written is known as Client Side Scripting Language. There are number of Client Side Scripting Languages like **JavaScript, JQuery, AJAX etc.**

Now consider that you have given all the data correctly on email account form and submitted it. Now where this data goes? This data is stored in the databases like **Oracle, SQL-Server or MySQL** on the server. Now to store such data in the database, the website has to write the script to interact with database which is always there on the server. Hence such script is known as Server Side Script. The languages in which this script is written are known as Server Side Scripting languages. These languages are **Java(Servlet and JSP), PHP** etc.

Syllabus Topic : Website Planning and Design Issues

1.2 Website Planning and Design Issues

1.2.1 Website Planning → (Aug. 2014 - In Sem)

Q. Describe phases of web site development in brief.

SPPU - Aug. 2014 - In Sem, 6 Marks

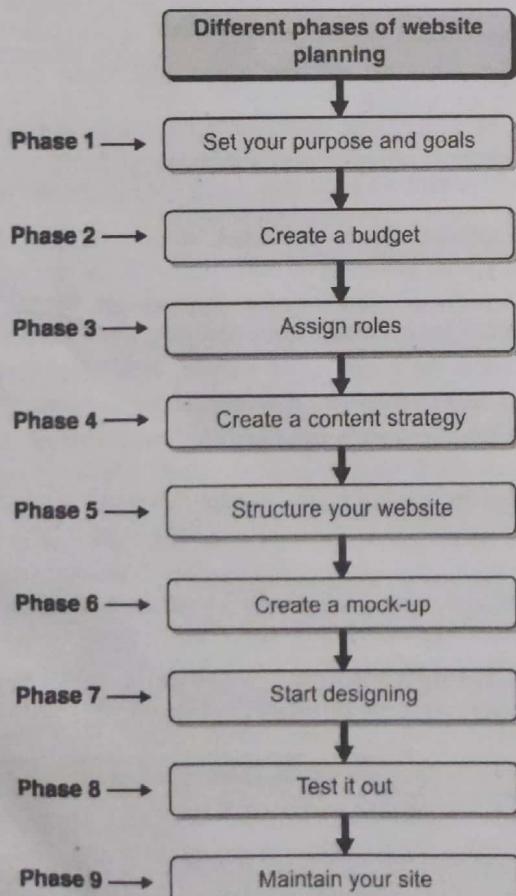


Fig. 1.2.1 : Different phases of website planning

- “If you fail to plan, you plan to fail”, this is an old adage which seem to be perfectly true. If we build any website without proper planning, then it will be same as of constructing a building without any blueprint. The lack of planning may lead to the things ends with wrong place, expected features may get missed and miscommunication between the web developer and clients may occur.
- The proper planning of website in time will definitely give a perfect direction as well as avoids missed deadlines and backtracking.
- The **different phases** of website planning :

Phase 1: Set your purpose and goals.

- The purpose of website is the most important factor. There are various purposes like :
 - o Gain publicity for the business
 - o Sell the inventory
 - o Get support behind a particular cause
- It is necessary to identify purpose of website and the target audience. The goals must be refined. Some basic questions we have to revise like:
 - o How many visitors we expect in a month for our website?
 - o How many visitors should sign in for our news later?
 - o How much sale we expect?
- We have to set specific, measurable goals for our website which should match with the marketing goals. There are some analytical tools available in market which can monitor the performance of our website over time.

Phase 2 : Create a budget.

- Whatever the extent of our business, it may be mid-sized organization or a start up, it is very important to set a proper budget of our website expenses. The expenses may include the costs for web design, scripting and web hosting. Some more expenses may get added later. For this purpose we have to conduct a proper market survey as well as have some discussions with professionals.
- Sometimes avoiding expenses for some important things to decrease the cost may lead to big headaches or creation of non effective website. Hence everything should be analyzed properly.
- The selection of team members should be based on knowledge, experience, devotion and foresight.

Phase 3 : Assign roles.

- The different stakeholders having different roles in the development process are :
 - o Owner of the business
 - o Marketing Manager



- Web and graphic designer
- HTML/CSS professional
- Web developer
- Content writer and/or editor
- Database administrator
- We have to make sure that every person in the team is well aware of his/her role and expectations from him/her. They should also aware with the deadlines and new developments.



Phase 4 : Create a content strategy.

- Content is nothing but any text on website which gives some information to visitors.
- The content of a website may include:
 - Blog posts
 - Pictures
 - Video
 - Slideshows
 - Embedded social media feeds like the Twitter stream or updates of Facebook page
 - Documents
- The content strategy helps to present the content in time. Consider we want to publish one blog post every week then we have to hire a professional web writer having experience of content writing.



Phase 5 : Structure your website.

- It is important to decide the exact features of all the web pages. In general all the websites have "About" and "Contact Us" pages, but we have to decide the features of web pages in such a way that they should meet our business needs.



Phase 6 : Create a mock-up.

- The mock-up which is also known as a wireframe is outline of our website. As it is the initial design, can be considered as first draft. Usually it is created in Photoshop or Fireworks. There is no need of putting in depth details in the mock-up. Placeholder text is used to fill the pages. This is just an idea for everyone that how the website will look like.
- If initially the design software like Photoshop is not available, then it can also be simply designed by using pen and paper.



Phase 7 : Start designing.

Designing is the major aspect in web development. The website with good design will certainly attract the visitors. The navigation (moving around the different web pages) of website should be user friendly. While designing the website, we have to take care of some important issues :

- The navigation of website should be easy to find and understand. Mostly the users expect it vertical and centered at the top of web page.
- The font of the text should be easy to read rather than stylish. The text color and background color should be contrast.
- The size of web pages should fit in the screen. Responsive design is best option which makes the website fit for all screen sizes or resolutions.
- The website should be light-weight so that it can be loaded quickly.
- The logo and tag line of company should be prominent on the page.
- Consistency of style and colors should be maintained across the website.
- The important information and features must be located at prominent locations.



Phase 8 : Test it out.

- Testing is necessary for finding out errors and retrieving missing details that we might have forgotten initially. Test the compatibility of website for various browsers like Chrome, Firefox, Internet Explorer, and mobile compatible browsers like Safari and Opera Mini. Test the website on mobile phones and tablets.
- The appearance of the website should be consistent.
- Make sure all the links work properly, images are in proper size.
- Make sure that all the forms and input fields are working properly.



Phase 9 : Maintain your site.

- As the website is uploaded, it does not mean that everything is finished. Now the work of maintenance get starts as the website is ongoing entity which continuously represents the company.
- Monitor the behavior of website with end users with the help of analytics software.
- We have to keep an eye on important aspects like the number of unique visitors and which pages are frequently visited on the website.
- The maintenance work also includes the posting new contents, monitoring security etc. Getting feedback from end users is also very important as it helps us to improve the website.

1.2.2 Design Issues

→ (Aug. 2015 - In sem)

Q. List and explain different design issues in web design.

SPPU - Aug. 2015 - In sem, 4 Marks

There are number of major technical issues which should be considered while developing a website.

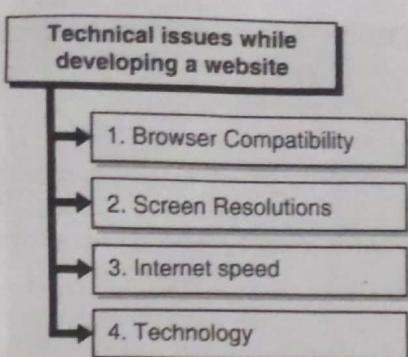


Fig. C1.1 : Technical issues while developing a website

→ 1. Browser Compatibility

- The webpage should be properly displayed on different browsers like Chrome, Firefox, Internet Explorer, and mobile compatible browsers like Safari and Opera Mini.
- Hence while building the website, the web pages should be checked on all possible browsers and operating systems for the compatibility. It is important to test the pages on latest as well as older versions of browsers, because not all of the site visitors may have latest versions.

→ 2. Screen Resolution

- Usually the common resolution of screen is 1024 x 768 pixels, but now there is growing trend towards higher resolutions. If the website is designed for higher resolution, it may be possible that some low-resolution screens may not be able to display all of the contents.
- Also it is very important to take care of mobile users. It is necessary to check the compatibility of our website for the mobiles, means how it look and feel on the mobile screens.

→ 3. Internet speed

- It is not possible that all the users have high speed internet. The download speed may get affected by the webpage design.
- The study suggests that: Near about 50 percent of web users expect that the webpage should be loaded in 2 seconds or even less. At least 40 percent visitors abandon a website which requires more than 3 seconds to load.
- Use of more images or heavy media like animations or videos slow down the downloading process of web pages. This can result in a fact that customers leaving the site. The speed affects the ranking of our website in search engine. It indicates that we should try to keep the file and image sizes as minimum as possible. The total size of a webpage is expected not more than 40 to 60 kilobytes.

→ 4. Technology

Some web technologies may also prevent end users from visiting our website or affect indexing of the website in search engines. These technologies include:

- HTML frames

- JavaScript
- Flash
- AJAX

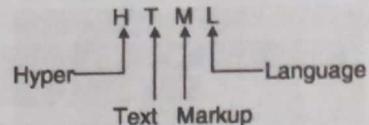
These technologies have some potential risks to the usability and accessibility of our website.

Syllabus Topic : HTML

1.3 HTML

Q. What is HTML ?

(2 Marks)



- **HTML** stands for **Hyper Text Markup Language**. Hypertext is nothing but the way in which the different web pages are linked with each other. Such links are called as Hypertext.
- As its name indicates, HTML is a Markup Language. That means we can use HTML to simply "mark-up" a text document with tags that instructs the Web browser like Chrome how to structure it to display.
- Basically the HTML is developed for the purpose of defining the structure of web document. It includes paragraphs, headings and lists.
- Now a day HTML is widely used for the purpose of formatting web pages by using various tags.

1.3.1 Structure of HTML Document

→ (Oct. 2016 - In Sem)

Q. Explain the basic structure of HTML document.

SPPU - Oct. 2016 - In sem, 6 Marks

- All the elements are included in the main opening and closing <html> tags.

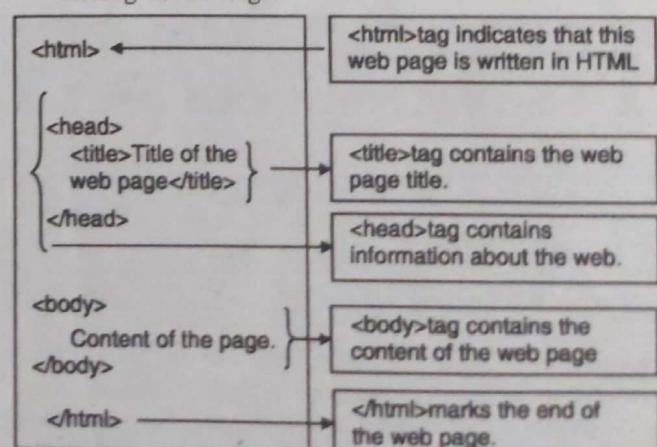


Fig. 1.3.1 : Structure of HTML Documents

- An HTML document contains two main parts:
 1. **Head** : The head element contains title and metadata of a web document. This section is used to declare variables and functions in scripting



- languages. These variables and functions are then accessible throughout the page.
2. **Body :** The body element contains the information which we like to display on a web page. All text to be displayed and control creation is done in this section.

Syllabus Topic : HTML Elements

1.4 HTML Elements / Tags

→ (Dec. 2014, May 2015, Dec. 2015, Oct. 2016 - In Sem)

Q. Give the list with definition of HTML components.

SPPU - Dec. 2014, May 2015, 2.5 Marks,

Oct. 2016 - In Sem, 5 Marks

Q. List various tags in HTML with simple example for a web page.

SPPU - Dec. 2015, 5 Marks

- ☞ **Definition :** HTML tags are standard keywords in the web page which define how the web browser should format and display the content.
- Usually most of the tags have two parts, an opening and a closing part. For example, <html> is the opening tag while </html> is the closing tag.
 - We have to remember that the closing tag has the similar text as of the opening tag, but has an extra forward-slash (/) character.
 - There are some tags which do not need closing tag. All the HTML files must have the necessary tags for it to be valid so that the web browser can understand and display it properly.
 - **An HTML element** is generally defined with starting tag. If the element has some content then it ends with a closing tag.
 - For example, <p> is starting tag of a paragraph and </p> is closing tag of the same paragraph.
<p>Phoenix InfoTech</p> is a paragraph element.
 - Some elements have attributes to which values are assigned to format the webpage.

<tag_name attribute=value>content</tag_name>

☞ **Example**

<body bgcolor="skyblue">--</body>

1.4.1 Headings

Q. Explain heading element in HTML. (2 Marks)

- ☞ **Use :** In general a document starts with heading. The heading elements are used to give title or subtitles to be displayed on the webpage. The different headings may require different sizes as per their importance. HTML has six levels of headings. Six elements namely <h1>, <h2>, <h3>, <h4>, <h5>, and <h6> are used to give headings. When any of the heading tag is used, one line before and one line after that heading is added by the browser.

- <h1> is the biggest heading tag while <h6> is the smallest heading tag. Hence h1 should be used for the most important heading text and h6 should be used for least important heading text.

Program 1.4.1

Write a program to display all heading tags from <h1> to <h6>.

Ans. : Program to display various headings

```
<!DOCTYPE html>
<html>
<head>
<title>www.PhoenixGlobe.com</title>
</head>
<body>
<h1> Heading 1 </h1>
<h2> Heading 2 </h2>
<h3> Heading 3 </h3>
<h4> Heading 4 </h4>
<h5> Heading 5 </h5>
<h6> Heading 6 </h6>
</body>
</html>
```

Output

The screenshot shows a web browser window with the URL "file:///c:/p1.html". The page displays a vertical stack of heading elements:

- Heading 1**
- Heading 2**
- Heading 3**
- Heading 4**
- Heading 5**
- Heading 6**

As we can observe that the contents of heading tags are by default bold and sizes are depending upon the particular heading tag.

1.4.2 Paragraph

Q. Explain paragraph element in HTML. (2 Marks)

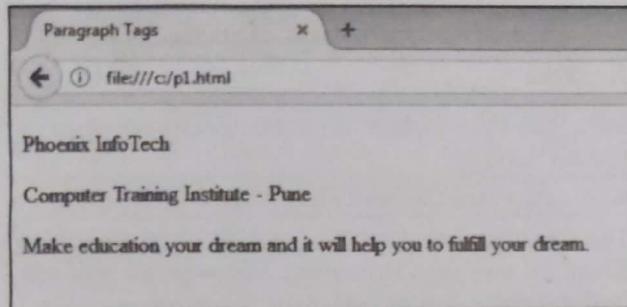
- ☞ **Use :** The <p> tag is used to define paragraph in HTML. This tag structures our text in different paragraphs. The paragraph of text is enclosed in opening <p> and closing </p> tags.

Program 1.4.2

Write a simple program to define the paragraph tag.

**Ans. : Program with the use of paragraph tag**

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Tags</title>
</head>
<body>
<p>Phoenix InfoTech</p>
<p>Computer Training Institute - Pune</p>
<p>Make education your dream and it will help you to fulfill your dream.</p>
</body>
</html>
```

Output**1.4.3 Line Breaks**

Q. Explain line break element in HTML. (2 Marks)

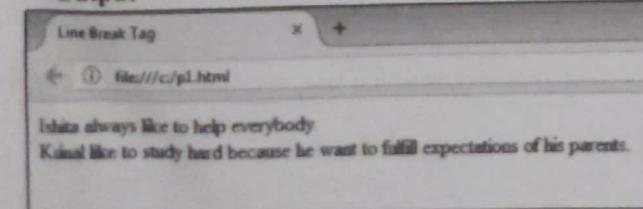
- ☞ **Use :** The
 tag is used to give the line break. It works just like '\n' of C programming language. This tag is called as empty element as it does not need any closing tag.

Program 1.4.3

Write a simple code for line break example.

Ans. : Program showing the line break example

```
<!DOCTYPE html>
<html>
<head>
<title>Line Break Tag</title>
</head>
<body>
Ishita always like to help everybody.
<br>
Kunal like to study hard because he want to fulfill expectations of his parents.
</body>
</html>
```

Output**1.4.4 Colors and Fonts****1.4.4(A) Colors**

Q. Explain setting colors in HTML. (4 Marks)

- The look and feel of a website is dependent upon the colors used in a webpage for text and backgrounds. The colors can be specified at page level with the help of <body> tag or can be set for individual tags.

Attributes of <body> tag

1. bgcolor
2. text
3. alink
4. link
5. vlink

Fig. C1.2 : Attributes of <body> tag

The <body> tag has number of attributes which are used to set colors to different entities.

- 1. **bgcolor**
Used to set color to the background of the page.
- 2. **text**
Used to set color to the body text.
- 3. **alink**
Used to set color to the active links or selected links.
- 4. **link**
Used to set color to the linked text.
- 5. **vlink**
Used to set color to the visited links – that is, for linked text that you have already clicked on.

☞ Color Coding Methods in HTML

- There are different methods to set colors in the web page as follows :
 - (i) **Color names** – We can directly specify color name like red, green, blue etc.
 - (ii) **Hex codes** – It is the six-digit code which represents the expected color.
 - (iii) **Color decimal or percentage values** – This is also called as RGB color scheme. Here we have to specify the amount of colors red, green and blue. The mixture of this gives expected color.
- Now we will see these color methods in detail.

☞ HTML Colors - Color Names

- In this method the color names are directly assigned to set color for text or background. The W3C (World Wide Web Consortium) has listed 16 standard color names for HTML. But maximum of browsers supports more than 200 color names.



- The W3C Standard 16 Colors are as follows: Black, Yellow, Red, Maroon, Gray, Lime, Green, Olive, Silver, Aqua, Blue, Navy, white, Fuchsia, purple, Teal.

☞ HTML Colors - Hex Codes

- This is the six digit representation of a color. The initial two digits (RR) represent the red value, the next two digits represent green value (GG), and the last two digits represent the blue value (BB).
- There are various graphics software like MS Paint, Adobe Photoshop or Paintshop Pro where we can get the hexadecimal value.
- Each hexadecimal code is preceded with pound or hash sign (#).

☞ HTML Colors - RGB Values

- The RGB method takes three values as arguments for Red, Green and Blue colors. These values are in integer format which range from 0 to 255 depending upon the required intensity of color.
- Not all the browsers support this method, hence not suggestive.

Program 1.4.4

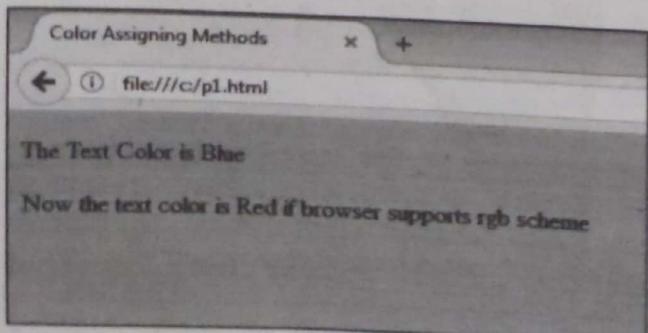
Write a simple code to display the text in various color format.

Ans. :

Code to display the text in various color format

```
<!DOCTYPE html>
<html>
  <head>
    <title>Color Assigning Methods</title>
  </head>
  <body text = "blue" bgcolor = "#00FF00">
    <p>The Text Color is Blue</p>
    <font color = "rgb(255,0,0)">Now the text color is Red if browser supports rgb scheme</font>
  </body>
</html>
```

Output



1.4.4(B) Font

Q. Explain setting fonts in HTML.

(4 Marks)

- User friendliness is the basic need of any website. This can be achieved by making it readable with the help of `` tag. The text in a webpage can be formatted by setting the ` ` tag and various font attributes.

- The `` tag has following attributes :

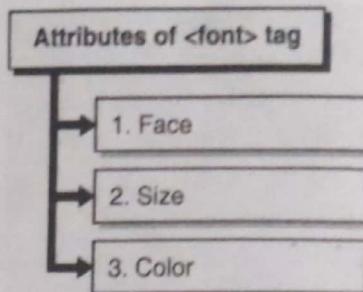


Fig. C1.3 : Attributes of `` tag

→ 1. Face

- ☞ Use : The Face attribute is used to specify the name of font for the text.

```
<font face="Arial"> Arial Font </font>
```

- The specified font must be installed on the machine where the web page is running; otherwise the text will be displayed in default font of the web browser.
- To the face attribute, we can give multiple font names separated by a comma.

```
<font face="Sans serif, Comic Sans MS, Lucida Console"> Multiple fonts</font>
```

- The first specified font is taken into consideration by the browser. But if it is not available on the machine then second font will be applied and if second font is not available then third font will be applied. Even this is also not available then the default font is applied to the text.

→ 2. Size

- ☞ Use : The size of a font can be set using the size attribute. The allowed range of values for font size is from 1(smallest) to 7(largest). 3 is the default font size.

```
<font size=5>Size is 5</font>
```

- It is also possible to set the relative font size. That means how many sizes greater or smaller than the current font size.

```
<font size="-2">Less by 2</font>
```

```
<font size="+2">Greater by 2</font>
```

→ 3. Color

- ☞ Use : Used to set the color of font. This attribute can accept value as standard name of a color or color code.

Program 1.4.5

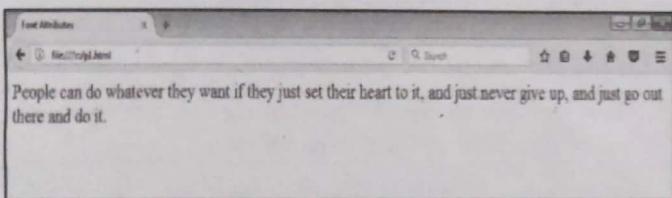
Write a program to display font size, font face and color.



Ans. : Program to display the font size, color and font face

```
<!DOCTYPE html>
<html>
<head>
<title>Font Attributes</title>
</head>
<body>
<font face = "Times New Roman,Sans serif" size = "5"
color="red">
People can do whatever they want if they just set their heart to
it, and just never give up, and just go out there and do it.
</font>
</body>
</html>
```

Output



1.4.5 Links - Hyperlinks

→ (Aug. 2014 - In Sem, Oct. 2016 - In Sem)

Q. Explain following HTML Tags with suitable example

<a> . SPPU - Aug. 2014 - In sem, 2.5 Marks

Q. What are the Hyperlinks and Anchors in the HTML?

Explain with suitable example(program).

SPPU - Oct. 2016 - In sem, 4 Marks

In a website there are multiple web pages.

☞ Use

- **Hyperlinks** are used to navigate in the website. That means to move from one webpage to another. A webpage can have various hyperlinks which can take us directly to another pages and even specific parts of the current page. For hyper linking words, phrases or even images can be used.
- The **anchor tag <a>** is used to specify the hyperlink. The text written in opening **<a>** and closing **** tag is known as hypertext. When this hypertext is clicked, the target webpage get opened. The hypertext has default formatting. It is in blue color and has underline.

Program 1.4.6 SPPU - Aug. 2014, Oct. 2016 - In sem

Write a program to display hyperlink example.

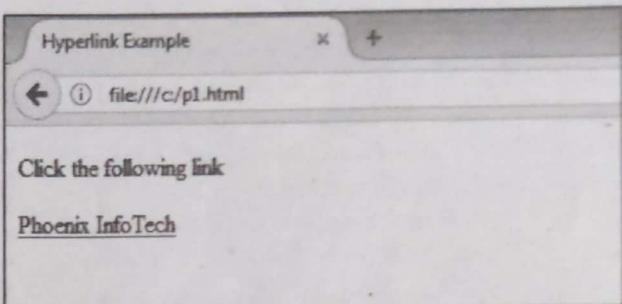
Ans. : Program to display the hyperlink example

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
```

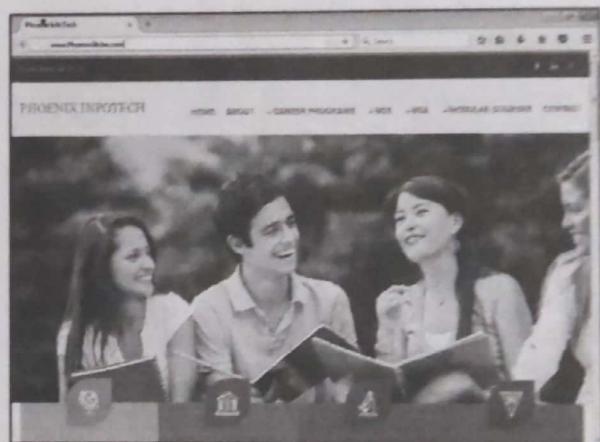
```
<body>
<p>Click the following link</p>
<a href = "http://www.PhoenixGlobe.com">Phoenix
InfoTech</a>
</body>
</html>
```

The **href** attribute is used to specify the address of webpage which we want to display.

Output



When this link is clicked, the home page of website will display.



1.4.5(A) The Target Attribute

☞ Use

The target attribute is used to specify location where the linked webpage should be opened. Different values for target attribute are as follows :

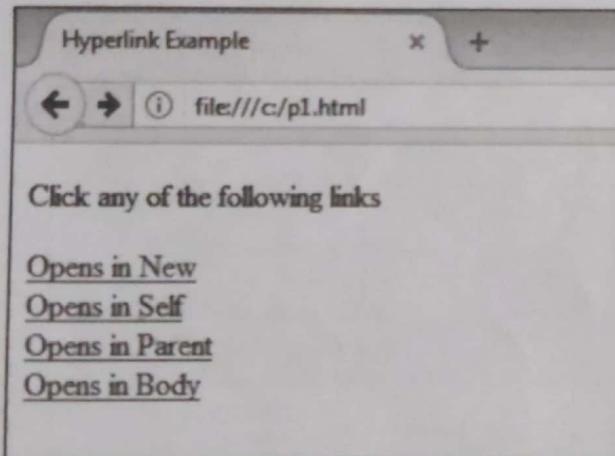
Sr. No	Option	Description
1	_blank	The linked webpage is opened in new window or tab.
2	_self	The linked webpage is opened in the same frame.
3	_parent	The linked webpage is opened in the parent frame.
4	_top	The linked webpage is opened in the full body of the window.
5	targetframe	The linked webpage is opened in a named targetframe.

**Program 1.4.7**

Write a program to display the hyperlinks using target attribute.

Ans. : Program of hyperlinks using target attribute

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click any of the following links</p>
<a href = "http://www.PhoenixGlobe.com"
target = "_blank">Opens in New</a> <br>
<a href = "http://www.PhoenixGlobe.com"
target = "_self">Opens in Self</a> <br>
<a href = "http://www.PhoenixGlobe.com"
target = "_parent">Opens in Parent</a> <br>
<a href = "http://www.PhoenixGlobe.com"
target = "_top">Opens in Body</a>
</body>
</html>
```

Output**Use of Base Path**

When web pages related to same website are linked, then it is not necessary to specify the complete path of the webpage. This path is known as URL. In document header we can use `<base>` tag to set base path for all the links. The browser will concatenate the given relative path with this base path and a complete URL is generated.

Program 1.4.8

Write a simple code for `<base>` tag.

Ans. : Simple code for `<base>` tag

In this program we will use `<base>` tag to specify the base URL and afterwards we can use the relative path for all the links rather than giving full URL for every link.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
```

```
<base href = "http://www.PhoenixGlobe.com/">
</head>
<body>
<p>Click following link</p>
<a href = "/html/student_corner.html"
target = "_blank">Student Corner</a>
</body>
</html>
```

Linking to a Page Section

It is also possible to link specific portion of the webpage with the help of name attribute of anchor `<a>` tag. There are two steps to implement it.

Step -1

Initially we have to create a link where we want to go within the webpage and give name to it using `<a>` tag as follows:

```
<h1>Link Section <a name = "top"></a></h1>
```

Step - 2

Now we have to create hyperlink to link the webpage and place where we want to reach –

```
<a href = "/html/html_text_links.htm#top">Go to the Top</a>
```

This will generate following link, where we can click to reach to the top of the webpage.

Go to the Top

1.4.6 Frames

→ (May 2015, Dec. 2015)

Q. Explain how frames are constructed in HTML document. Explain with program.

SPPU - May 2015, 5 Marks, Dec. 2015, 4 Marks

Q. How to create a frame in HTML? Explain with example (Program).

SPPU - May 2017, 5 Marks

Usually we can display only one webpage at a time in the browser window.

Use : If we wish to display multiple pages at the same time on the browse window, then we can use the frames to divide the browser window into different sections and in each section we can display a separate webpage. A group of frames in the browser window is called as a frameset.

- While dividing the window, it is considered as a table and organized in rows and columns.
- In the frameset application, basic tag of html like `<body>` is not used. Instead of it, `<frameset>` tag is used. This tag defines how to divide the window into different sections. There are two main attributes of frameset tag.

1. Rows – Defines the horizontal frames.

2. Cols – Defines the vertical frames

- The `<frame>` is sub-tag of `<frameset>` tag. The `<frame>` tag has attribute `src` (source) to which we have to specify the URL of webpage which we want to display in particular frame.



Attributes of the <frameset> tag

Following are the important attributes of tag <frameset>:

Sr. No	Attribute	Description
1.	cols	<p>Specifies the number of columns contained in the frameset and also the size of all the columns. Width of each column can be specified in four ways:</p> <ul style="list-style-type: none"> (1) Absolute values can be given in pixels. For example, to generate three vertical frames, we can set cols = "200, 400, 100". (2) A percentage amount of the current browser window. For example, to generate three vertical frames, use cols = "20%, 70%, 10%". (3) Using the special wildcard symbol. For example, to generate three vertical frames, use cols = "20%, *, 20%". Here the wildcard takes remaining percentage of the window. (4) As relative widths of current browser window. For example, to generate three vertical frames, use cols = "3*, 2*, 1*". This is substitute of percentages. The relative widths of the browser window can be used. This divides the window as: first column occupies half of the window, the second occupies one third, and the third occupies one sixth.
2.	rows	This attribute is same as of the cols attribute and accepts the same values, but it is used to mention the rows in the frameset. For example, to generate two horizontal frames, use rows = "20%, 80%". The height of each row can be specified in the same way as we have seen for columns.
3.	border	The width of the border of each frame can be specified with border attribute. For example, border = "3". A value of zero indicates frame without border.
4.	frameborder	This attribute mentions whether to display a three-dimensional border between frames or not. It takes value either 1 (true) or 0 (false). For example frameborder = "1" specifies frame with three-dimensional border.

Sr. No	Attribute	Description
5.	Framespacing	The amount of space between different frames in a frameset can be specified with the help of this attribute. It accepts any integer value. For example framespacing = "5" indicates that there should be 5 pixels spacing between the frames.

The <frame> Tag Attributes

Following are the important attributes of the sub-tag <frame> :

Sr. No	Attribute	Attribute & Description
1.	Src	<p>This attribute specifies the URL of web page that should be loaded in the frame.</p> <p>Example : src = "/html/index.htm".</p>
2.	Name	This attribute specifies the name to a frame. This is usually important when we want to establish links in one frame that load web pages into another frame. In this case the second frame must have a name to identify itself as the target for the link.
3.	Frameborder	This attribute is used to set whether or not the borders of the frame should be displayed; it overwrites the value specified in the frameborder attribute of the <frameset> tag if it is specified. It takes value either 1 (true) or 0 (false).
4.	Marginwidth	This attribute specifies the width of the space between the left and right borders of the frame and the content of the frame. The value is specified in pixels. Example : marginwidth = "5".
5.	Marginheight	This attribute specifies the height of the space between the top and bottom borders of the frame and its contents. The value is specified in pixels. For example marginheight = "5".
6.	Noresize	The borders of the frames can be resized by default. If we want to restrict user from resizing the frame then we can set : noresize = "noresize".
7.	Scrolling	This attribute specifies the behavior of the scrollbars which appear on the frame. It accepts value either "yes", "no" or "auto". Example : scrolling = "yes" indicates that the frame should have scroll bars.



Sr. No	Attribute	Attribute & Description
8.	longdesc	This attribute specifies link to another page which contains a long description about the contents of the frame. Example : longdesc = "details.htm"

☞ Browser Support for Frames

- If any user has old browser or such a browser which does not have support for frames then the `<noframes>` element is shown on the browser.
- In such case we have to place the `<body>` element inside the `<noframes>` element since the `<frameset>` element should be replaced by the `<body>` element.

Program 1.4.9 SPPU - May 2015, Dec. 2015, May 2017

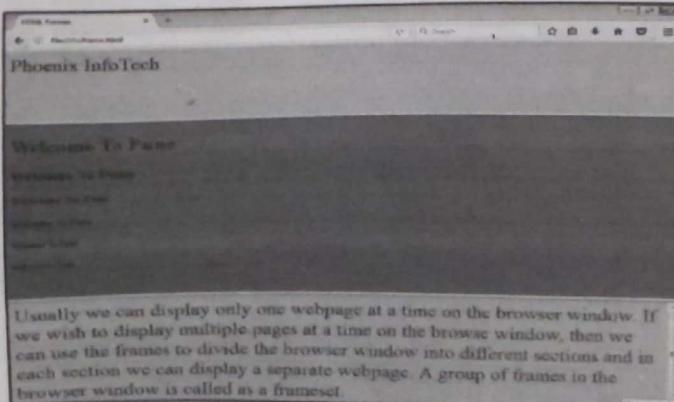
Write a code to create horizontal frames.

Ans. : Code to create the horizontal frames

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Frames</title>
</head>
<frameset rows = "20%,50%,30%">
    <frame name = "top" src = "First.html" />
    <frame name = "main" src = "Second.html" />
    <frame name = "bottom" src = "Third.html" />
<noframes>
    <body>Your browser does not support frames.</body>
</noframes>
</frameset>
</html>
```

Note : Here it is considered that pages First.html, Second.html and Third.html are already created.

Output



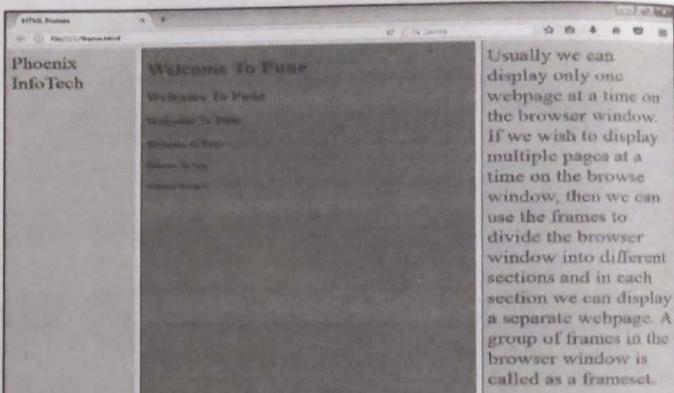
Program 1.4.10 SPPU - Aug. 2014 - In sem, 2.5 Marks

Write a code to create the vertical frames.

Ans. : Code to create the vertical frames

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Frames</title>
</head>
<frameset cols = "20%,50%,30%" border=10
framespacing=200>
    <frame name = "left" src = "First.html" />
    <frame name = "center" src = "Second.html" />
    <frame name = "right" src = "Third.html"
scrolling="no"/>
<noframes>
    <body>Your browser does not support
frames.</body>
</noframes>
</frameset>
</html>
```

Output



☞ Frame's name and target attributes

One of the best uses of frame is to place navigation bars in one frame and then load the link web pages into a separate frame.

Program 1.4.11

Write a code to divide the browser window into two frames.

Ans. : Code to divide the browser window into two frames

```
<!DOCTYPE html>
<html>
<head>
    <title>HTML Target Frames</title>
</head>
<frameset cols = "200, *">
    <frame src = "menu.html" name = "menu_page" />
    <frame src = "main.html" name = "main_page" />
</frameset>
</html>
```

Here the browser window is divided into two frames. The first frame is small and will contain the only the navigation menu options of menu.html page. The second

column occupies near about 80% of size and contain the main part of the page and it is implemented by main.htm file. In the links of the menu we will specify the target frame as main_page. Whenever any link in the menu is clicked, the related page will be opened in main page.

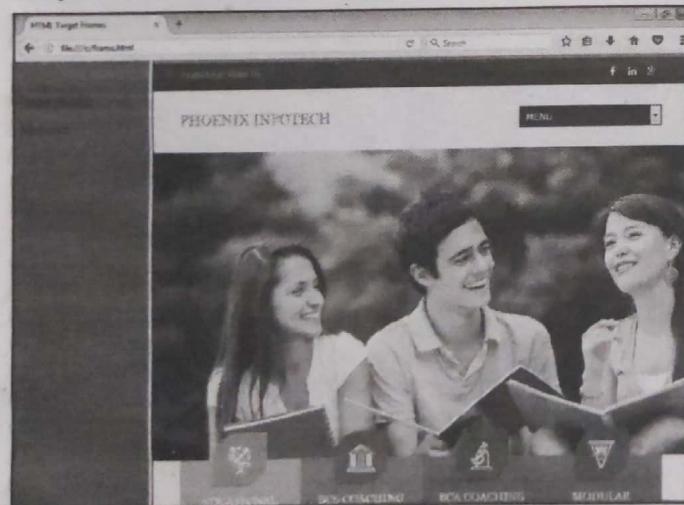
Following is the content of menu.html file

```
<!DOCTYPE html>
<html>
<body bgcolor = "#4a7d49">
<a href = "http://www.TechMaxBooks.com"
target = "main_page">Tech-Max</a>
<br />
<br />
<a href = "http://www.PhoenixGlobe.com"
target = "main_page">Phoenix InfoTech</a>
<br />
<br />
<a href = "http://www.edudecision.com"
target = "main_page">Edudecision</a>
</body>
</html>
```

Following is the content of main.html file –

```
<!DOCTYPE html>
<html>
<body bgcolor = "skyblue">
<h3>Welcome To Our Web</h3>
<p>Click any link and page will be displayed here.</p>
</body>
</html>
```

Output



1.4.6(A) iframe

Q. Explain the concept of iframe with suitable example (2 Marks)

iframe stands for inline frame. It is defined using tag <iframe>.

☞ **Use :** The <iframe> tag defines a region with rectangular shape inside the main page where the browser should display another webpage with scrollbars and borders.

- The URL of webpage which occupies the inline frame is specified by the src attribute.

Program 1.4.12

Write a code to define <iframe> tag.

Ans. : Simple code to define <iframe> tag

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Iframes</title>
</head>
<body>
<iframe src = "menu.html" width = "400"
height = "250">
</iframe>
<p>Document content go here...</p>
</body>
</html>
```

Output



1.4.7 Lists

Q. Explain the Lists in HTML.

(4 Marks)

☞ **Use :** As the name suggests, these elements are used to give list of items. Lists are used everywhere on the websites. There are number of things like articles, website navigation menus, and product features on e-commerce websites which makes frequent use of lists.

Types of HTML list

HTML provides three ways to give the lists :

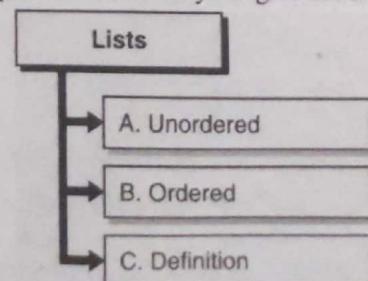


Fig. C1.4 : Types of HTML list



- A. **** – An unordered list. Plain bullets are used for the list items.
- B. **** – An ordered list. Different schemes of numbers are used for the list items.
- C. **<dl>** – A definition list. This is used to give list of definitions just like dictionaries.

→ A. HTML Unordered Lists

❖ Use :

In this list, plain bullets are used for the list items. Generally this option is used when there is no any standard sequence or order of the list items. The **** tag is used to give this list. Every element in the list is marked by a bullet. Different types of bullets are available.

The **** which is the sub-tag of **** is used for individual list elements.

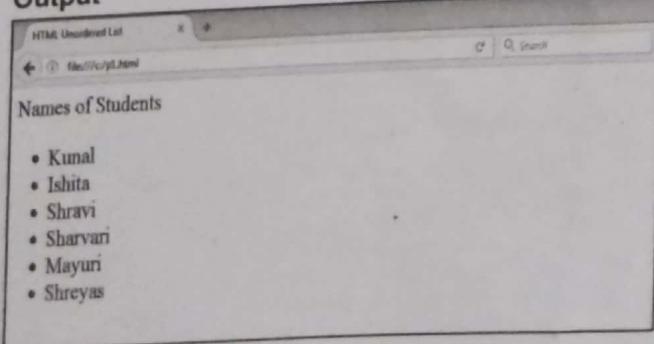
Program 1.4.13

Write a simple code demonstrating unordered list.

Ans. : Code to display an unordered list

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
<font size=5>
Names of Students
<ul>
  <li>Kunal</li>
  <li>Ishita</li>
  <li>Shravi</li>
  <li>Sharvari</li>
  <li>Mayuri</li>
  <li>Shreyas</li>
</ul>
</font>
</body>
</html>
```

Output



- The **** tag has attribute **TYPE** which is used to specify the type of bullet for the list items. The default type is disc which is displayed in above output.

- There are following options for the attribute **TYPE** of **** tag.
 - (i) Square
 - (ii) Circle
 - (iii) Disc
- See the following example which illustrates the use of attribute **TYPE**.

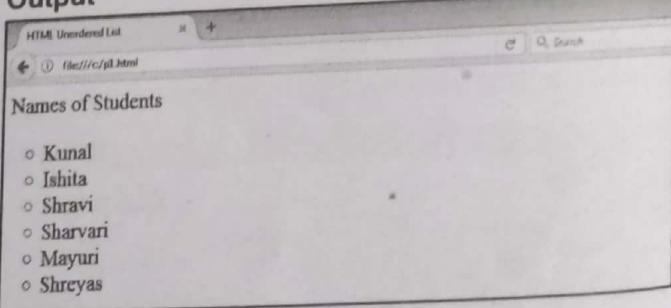
Program 1.4.14

Write a code illustrating the use of attribute **TYPE**.

Ans. : Code illustrating the use of **TYPE** attribute

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Unordered List</title>
  </head>
  <body>
<font size=5>
Names of Students
<ul type="circle">
  <li>Kunal</li>
  <li>Ishita</li>
  <li>Shravi</li>
  <li>Sharvari</li>
  <li>Mayuri</li>
  <li>Shreyas</li>
</ul>
</font>
</body>
</html>
```

Output



→ B. HTML Ordered Lists

❖ Use

Ordered lists are generally used when we want to specify numbers instead of bullets. The **** tag is used for such list. The number starts from 1 and also incremented by one for each successive list element in the ordered list.

Program 1.4.15

Write a program to display the ordered list.

Ans. : Program to display the ordered list

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML ordered List</title>
  </head>
```



```

<body>
<font size=5>
Names of Teachers
<ol>
<li>Savita Madam</li>
<li>Snehal Madam</li>
<li>Shwetangi Madam</li>
<li>Sapana Madam</li>
<li>Shalaka Madam</li>
<li>Sonal Madam</li>
</ul>
</font>
</body>
</html>

```

Output

```

Names of Teachers
1. Savita Madam
2. Snehal Madam
3. Shwetangi Madam
4. Sapana Madam
5. Shalaka Madam
6. Sonal Madam

```

- The `` tag has attribute `TYPE` which is used to specify the type of number for the list items. The default type is 1 which is displayed in above output.
- There are following options for the attribute `TYPE` of `` tag.
 - 1 - Number
 - I - Upper Roman
 - i - Lower Roman
 - A - Upper Alpha
 - a - Lower Alpha
 See the Program 1.4.16 which illustrates the use of attribute `TYPE`.

The start Attribute

This attribute of `` tag is used to specify the starting point of numbering.

Program 1.4.16

Write a program to illustrate the use of `TYPE` and `start` attribute in ordered list.

Ans. : Program to illustrate the use of TYPE and start attribute in ordered list

```

<!DOCTYPE html>
<html>
<head>
<title>HTML ordered List</title>
</head>
<body>
<font size=5>

```

Names of Teachers

```

<ol type="I" start="3">
<li>Savita Madam</li>
<li>Snehal Madam</li>
<li>Shwetangi Madam</li>
<li>Sapana Madam</li>
<li>Shalaka Madam</li>
<li>Sonal Madam</li>
</ul>
</font>
</body>
</html>

```

Output

```

Names of Teachers
III. Savita Madam
IV. Snehal Madam
V. Shwetangi Madam
VI. Sapana Madam
VII. Shalaka Madam
VIII. Sonal Madam

```

→ C. HTML Definition Lists

- ☞ **Use :** Definition Lists are used to give the list of definitions just like the dictionary or encyclopedia. In general this is considered as the ideal way to represent a glossary, list of terms, or some kind of name/value pair list.

Following tags are used in definition list.

- (i) `<dl>` – Defines the beginning of the list
- (ii) `<dt>` – Specify a term
- (iii) `<dd>` – Specify term definition
- (iv) `</dl>` – Defines the end of the list

Program 1.4.17

Write a simple code for definition list with the help of various tags in definition list.

Ans. : Code for definition list using various tags in `<dl>`

```

<!DOCTYPE html>
<html>
<head>
<title>HTML Definition List</title>
</head>
<body>
<font size=5>
<dl>
<dt><b>HTML</b></dt>
<dd>HTML is a scripting language used for formatting the web contents.</dd>

```



```

<dt><b>JavaScript</b></dt>
<dd>JavaScript is a client side scripting language used
for interaction with client and
validate the data.</dd>
<dt><b>PHP</b></dt>
<dd>PHP is a server side scripting language used to handle
the database on the server.</dd>
</font>
</dl>
</body>
</html>

```

Output

HTML
HTML is a scripting language used for formatting the web contents.
JavaScript
JavaScript is a client side scripting language used for interaction with client and validate the data.
PHP
PHP is a server side scripting language used to handle the database on the server.

1.4.8 Tables

→ (Aug. 2014 - In Sem)

Q. How to create Tables on the web pages using HTML?

SPPU - Aug. 2014 - In sem, 4 Marks

Use : The <table> tag is used to present the data in tabular format, means in rows and columns. <table> is the main tag which is used to create the table.

Following Table 1.4.1 shows the list of table related tags.

Table 1.4.1

Tag	Description
<table>	It defines a table.
<tr>	Table row - It defines a row in a table.
<th>	Table heading - It defines column headings in the table.
<td>	Table Data - It defines a cell in a table.
<caption>	It defines the caption of the table.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used in combination with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

1.4.8(A) Difference between <tr> and <td>

→ (May 2015 - In Sem)

Q. What is the difference between <tr> and <td>?

SPPU - May 2015-In Sem, 2 Marks

Sr. No.	Parameters	<tr>	<td>
1.	Longform	<tr> stands for table row	<td> stands for table data
2.	Use	Used to create row in a table	Used to create cell in a row
3.	Hierarchy	<tr> is parent tag of <td>	<td> is sub tag of <tr>

Attributes of <table> tag

The <table> tag has following attributes

Attribute	Value	Description
align	Left center right	Specifies the alignment of a table with respect to surrounding text
bgcolor	rgb(x,x,x) #xxxxxx colorname	Specifies the color for a table background
border	Any number	Specifies size of border
cellpadding	Pixels	Specifies the space between the cell wall and the cell content
cellspacing	Pixels	Specifies the space between cells
width	Pixels %	Specifies the width of a table
height	Pixels %	Specifies the height of a table
Bordercolor	rgb(x,x,x) #xxxxxx colorname	Sets color of the table border

Program 1.4.18 SPPU - May 2015 - In sem, 2 Marks

Write HTML code which includes table.

Ans.: HTML code for table

```

<!DOCTYPE html>
<html>
  <head>
    <title>Table</title>
  </head>
  <body>
    <table border=1 bordercolor="red" width=40%
           height = 40% cellspacing=2 cellpadding=10
           bgcolor="gray" align="center">
      <tr>
        <th>Rollno</th>
        <th>Name</th>
        <th>Marks</th>
    
```

```

<th>Project Completed</th>
</tr>
<tr>
<td>1</td>
<td>Kunal</td>
<td>90</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Ishita</td>
<td>90</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Shreya</td>
<td>92</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Shravi</td>
<td>80</td>
<td>Yes</td>
</tr>
</table>
</body>
</html>

```

Output

Rollno	Name	Marks	Project Completed
1	Kunal	90	Yes
2	Ishita	90	Yes
3	Shreya	92	No
4	Shravi	80	Yes

1.4.8(B) Colspan and Rowspan Attributes

Q. Explain Colspan and Rowspan attributes of table tag with suitable examples (programs). (4 Marks)

- A table is divided into rows and every row is divided into cells. Sometimes we require the table cells span across (merged) more than one column or row. Here we can use Colspan and Rowspan attributes.

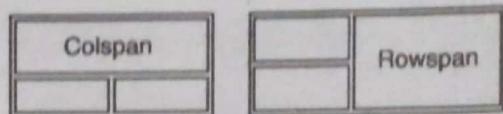


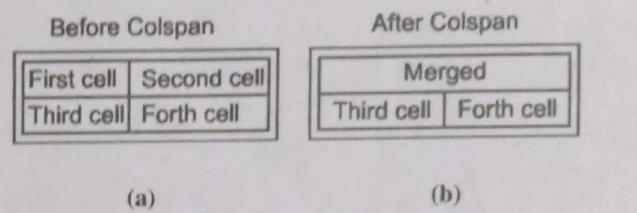
Fig. 1.4.1 : Colspan and Rowspan Attributes

1. Colspan

☞ Use : The colspan attribute is used to define the number of columns a cell should span (or merge) horizontally. That means, we like to merge two or more Cells in a row into a single Cell.

```
<td colspan=2 >
```

This given code will merge two Cells into one Cell horizontally.



(a) (b)

Fig. 1.4.2 : Colspan

In the Fig. 1.4.2 there are two tables. In the first Fig. 1.4.2(a), 2 rows and 2 columns in each row are present. In the second Fig. 1.4.2(b), 2 rows and 1 column in first row and 2 columns in second row are present. Now In the second Fig. 1.4.2(b) first two Cells are merged horizontally with the help of Colspan attribute.

☞ Code for Fig. 1.4.2(b)

```

<html>
<body>
<table border=1 >
<tr>
<td colspan=2 >
    Merged
</td>
</tr>
<tr>
<td>
    Third Cell
</td>
<td>
    Forth Cell
</td>
</tr>
</table>
</body>
</html>

```

Colspan (Column Span) merges the Cells horizontally that means from left to right.

The default value for the Colspan is 1.

Program 1.4.19

Write a program demonstrating the use of colspan attribute in table.

Ans. : Program using colspan attribute

```

<!DOCTYPE html>
<html>
  <head>
    <title>Table </title>
  </head>
  <body>
    <table border=2 width="40%">
      <caption>Marks Data</caption>
      <tr>
        <th colspan="2">Ishita</th>
        <th colspan="2">Kunal</th>
        <th colspan="2">Rahul</th>
      </tr>
      <tr>
        <th>Java</th>
        <th>Dot Net</th>
        <th>Java</th>
        <th>Dot Net</th>
        <th>Java</th>
        <th>Dot Net</th>
      </tr>
      <tr>
        <td>82</td>
        <td>85</td>
        <td>78</td>
        <td>82</td>
        <td>77</td>
        <td>81</td>
      </tr>
    </table>
  </body>
</html>

```

Output

Table

file:///c:/p1.html

Marks Data					
Ishita		Kunal		Rahul	
Java	Dot Net	Java	Dot Net	Java	Dot Net
82	85	78	82	77	81

2. Rowspan

Use : The rowspan attribute is used to define the number of rows a cell should span in vertical manner. That means , two or more Cells in the same column are merged as a single Cell vertically.

<td rowspan=2>

This give code will merge two Cells into one Cell vertically.

Before Rowspan		After Rowspan	
First cell	Second cell	First cell Third cell	Merged
Third cell	Forth cell		
(a)		(b)	

Fig. 1.4.3 : Rowspan

In the above Fig. 1.4.3 we can see there are two tables. In the first Fig. 1.4.3(a) there are 2 rows and each row has 2 columns. In the second Fig. 1.4.3(b) there are 2 rows in the first column and only 1 row in the second column. The Rowspan attribute is used vertically in the second column.

Code for Fig. 1.4.3(b)

```
<html>
<body>
    <table border=1>
        <tr>
            <td>
                First Cell
            </td>
            <td rowspan=2>
                Merged
            </td>
        </tr>
        <tr>
            <td valign=middle>
                Third Cell
            </td>
        </tr>
    </table>
</body>
</html>
```

Rowspan merged the Cells in vertical manner that means from top to bottom.

Program 1.4.20

Write a program demonstrating the use of rowspan attribute in table.

Ans.

Program using rowspan attribute

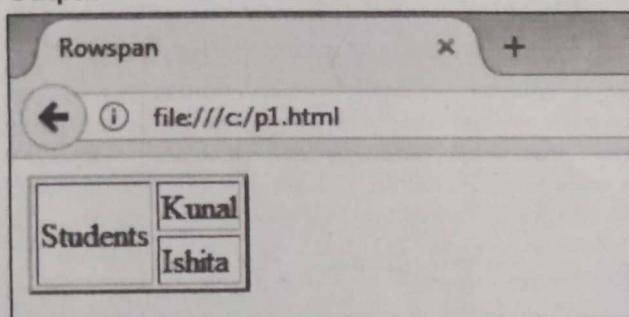
```
<html>
<head>
<title>
Rowspan
</title>
</head>
<body>
<table border=2>
<tr>
    <td rowspan="2">
        Students
```



```

</td>
</td>
Kunal
</td>
</tr>
<tr>
<td>
Ishita
</td>
</tr>
</table>
</body>
</html>

```

Output**1.4.9 Images and Forms****1.4.9(A) Images → (Aug. 2016 - In Sem)**

Q. Explain how the images can be inserted in HTML document with program.

SPPU - Aug. 2016 - In sem, 5 Marks

- Images are important in the designing of our webpage. Images are used to describe some complex concepts in simple pictorial format on the web page.
- In HTML there are two options to **insert image** in our webpage.

(1) Using <body> tag

The body tag has background attribute to set image as a background to the webpage.

Program 1.4.21 SPPU - Aug. 2016 - In sem

Write a simple code for setting the image at the background.

Ans. : Code for setting the image as background

```

<html>
<head>
<title>
Images
</title>
</head>
<body background = "Desert.jpg">
</body>
</html>

```

Output**(2) Using tag**

- The background image occupies the entire background of webpage. We cannot set size or location for it. **** tag is used to insert images at desired location and with desired size.
- The **** tag is considered as an empty tag, which indicates that, it can have only list of attributes. There is no need of closing tag.

Attribute	Value	Description
<td>Top bottom middle left right</td> <td>Specifies the alignment of an image with respect to the surrounding elements</td>	Top bottom middle left right	Specifies the alignment of an image with respect to the surrounding elements
alt	text	Defines an alternate text for the image
border	pixels	Defines the width of the border around an image
height	pixels	Defines the height of an image
hspace	pixels	Defines the whitespace on both of the left and right side of an image
src	URL	Defines the URL of an image
usemap	#mapname	Defines an image as a client-side image-map
vspace	pixels	Defines the whitespace on top and bottom of an image
width	pixels	Specifies the width of an image

Program 1.4.22

Write a program to display the image.

Ans. : Program to display the image

```

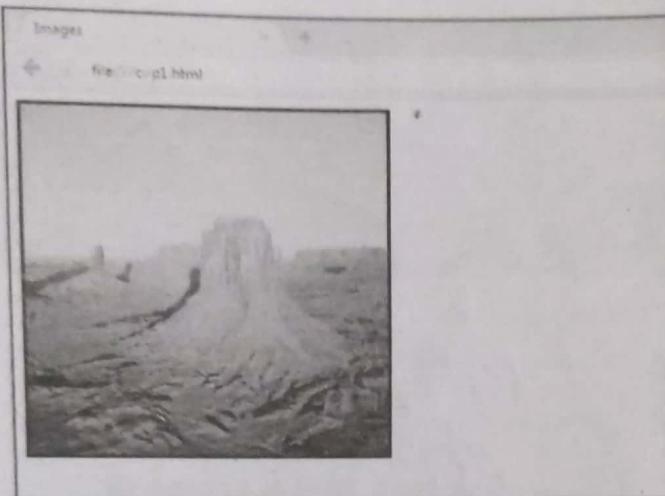
<html>
<head>
<title>
Images
</title>

```



```
</head>
<body>

</body>
</html>
```

Output**1.4.9(B) Image Maps → (May 2015, Dec. 2015)**

Q. Explain Image Map with example (program).

SPPU - May 2015, Dec. 2015, 5 Marks

- HTML provides the `<map>` tag to define a client-side image-map. It is an image with clickable areas.
- The `<map>` element has name attribute which is associated with the attribute use map and links the image with the map.
- The `<map>` element has sub-tag `<area>`, which defines the clickable areas in the image map.

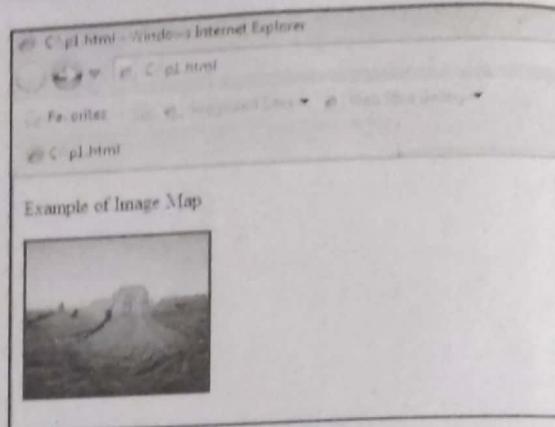
Program 1.4.23 SPPU - May 2015, Dec. 2015

Write a program to demonstrate use of `<map>` tag.

Ans. : Program to demonstrate use of `<map>` tag

```
<!DOCTYPE html>
<html>
<body>
<p>Example of Image Map</p>

<map name="smap">
  <area shape="rect" coords="0,0,82,126" alt="Jellyfish"
href="Jellyfish.htm">
  <area shape="circle" coords="90,58,3" alt="Koala"
href="Koala.htm">
  <area shape="circle" coords="124,58,8" alt="Penguins"
href="Penguins.htm">
</map>
</body>
</html>
```

**1.4.9(C) Forms**

Q. Explain Form element in HTML.

(8 Marks)

- The HTML form is a part of a document which may have general content, special elements known as *controls* like textfields, checkboxes, submit button, radio buttons etc. Users usually "complete" a form by modifying its elements. This modification may include entering text, selecting controls like checkbox etc. before submitting the form to the next page for processing. This data is posted to an application like CGI, ASP Script or PHP script etc. This application will then perform essential processing on the accepted data based on defined business logics of the application.

Use

- HTML Forms are generally necessary when we want to accept some data from the website visitors. For example e-mail account form, credit card form where we may like to accept information like name, email address, phone number, gender etc.
- The HTML `<form>` tag creates an HTML form

Syntax

```
<form action = "Script URL" method = "GET/POST">
  form elements like input, button etc.
</form>
```

Tag attributes

Following is the list of form tag attributes

Sr. No	Attribute	Description
1.	Action	Url of next page(script) to which we want to submit the data
2.	Method	Method which will upload the data. The common methods used are GET and POST.
3.	target	Sets the name of the target window or frame where we want to display the result. It takes values like _blank, _self, _parent etc.
4.	enctype	Specify the way to encode the data before sending it to server



☞ HTML Form Controls

HTML provides various controls which can be used to collect the data using HTML form

Q. Explain Different Form elements / Form Controls.
(4 Marks)

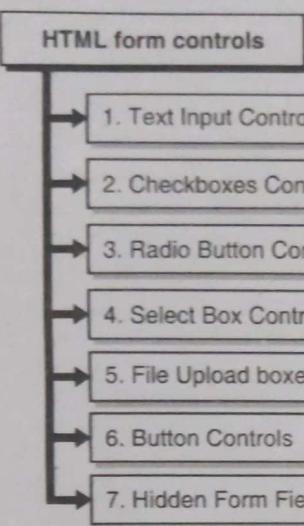


Fig. C1.5 : HTML form controls

→ 1. Text Input Controls

Three types of text input controls are used on forms –

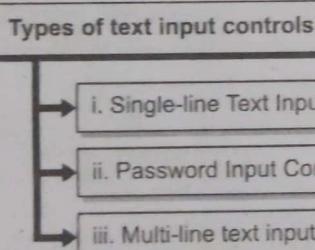


Fig. C1.6 : Types of text input controls

→ (i) Single-line Text Input Control

This control is used for one line of user input. It is created using HTML <input> tag.

<input type="text" name="txtUser"/>

→ (ii) Password Input Controls

This is too a single-line text input control except that it masks the character when user enters it. It is also created using the same <input> tag.

<input type="password" name="txtPass"/>

☞ Attributes of input tag

Sr. No	Attribute	Description
1.	Type	Specifies the type of control.
2.	Name	Used to assign a name to the control which is sent to the server to be recognized and retrieve the value.
3.	Value	This is used to set default value to the control.

Sr. No	Attribute	Description
4.	size	Sets the width of the control in terms of characters.
5.	maxlength	Specifies the maximum number of characters which user can give in the text box.

→ (iii) Multi-line text input controls

This is used to accept detailed information from user. It is created using HTML <textarea> tag.

<textarea rows=5 cols=30/>

☞ Attributes of <textarea> tag.

| Sr. No | Attribute | Description |
|--------|-----------|---|
| 1. | name | Used to assign a name to the control which is sent to the server to be recognized and retrieve the value. |
| 2. | Rows | Specifies the number of rows of text area. |
| 3. | Cols | Specifies the number of columns of text area. |

→ 2. Checkbox Control

This control is used to give list of items where multi-selection is allowed. It is also created using <input> tag but the value for type attribute is set as **checkbox**.

<input type="checkbox" name="chkPhoenix"/>

→ 3. Radio Button Control

This control is used to give list of items where single-selection is allowed. It is also created using <input> tag but where the value for type attribute is set as **radio**.

<input type="radio" name="optEngg"/>

→ 4. Select Box Control

A select box is also known as drop down box. It provides option to list number of options in the form of drop down list. Here the user can select one or multiple options.

<select name="sn">
<option>--</option>

</select>

☞ Attributes of <select> tag

| Sr. No | Attribute | Description |
|--------|-----------|---|
| 1. | name | Used to assign a name to the control which is sent to the server to be recognized and retrieve the value. |
| 2. | size | This can be used to specify a scrolling list box. |
| 3. | multiple | If this option is set, it creates listbox and user can select multiple elements. If not then combobox is created where single selection is allowed. |



Attributes of <option> tag

| Sr. No | Attribute | Description |
|--------|-----------|---|
| 1. | value | The value that will be used to check whether the option is selected or not. |
| 2. | Selected | It sets the option by default selected |
| 3. | Label | Sets label to options |

→ 5. File Upload Box

This is used to give option to user for file upload. This is created using <input> tag. The type attribute is set to **file**.

Attributes of file upload box

| Sr. No | Attribute | Description |
|--------|-----------|---|
| 1. | accept | Specifies the types of files that the server accepts. |

→ 6. Button Controls

HTML provides different types of buttons. The <input> tag is used to create buttons.

Types of Buttons

| Sr. No | Type | Description |
|--------|--------|--|
| 1. | Submit | This button submits a form. |
| 2. | Reset | This button resets (clears) form controls. |
| 3. | Button | This button calls a client-side script (function). |

→ 7. Hidden Form Fields

This element is used to hide the data while sending it to the server. This control as name suggests does not display on the form.

Program 1.4.24

Write a program to display the form containing all controls.

Ans. :

Program displaying the form containing all records

```
<html>
<head>
<title>
Form
</title>
</head>
<body>
<font size=6>
<center>
Email Account Form
</center>
</font>
<font size=5>
Enter Username <input type="text" name ="txtUser" maxlength=30/><br><br>
Enter Password <input type="password" name ="txtPass"/><br><br>
Address <textarea name="add" rows=5 cols=30></textarea><br><br>
```

Fields of interests
<input type="checkbox" name ="chk1"/>Entertainment
<input type="checkbox" name ="chk2"/>Sports
<input type="checkbox" name ="chk3"/>News

Gender
<input type="radio" name ="optGender"/>Male
<input type="radio" name ="optGender"/>Female

Select Languages
<select name="lang" multiple>
<option>Marathi</option>
<option>Hindi</option>
<option>English</option>
</select>

Select Country
<select name="cntry">
<option>India</option>
<option>USA</option>
<option>Japan</option>
<option>UK</option>
</select>

Upload Photo
<input type = "file" name = "fileupload" accept = "image/*" />

<input type="button" name = "vld" value="Validate">
<input type="reset" name = "rst" value="Reset">
<input type="submit" name = "sub" value="Submit">
<input type = "hidden" name = "pagenm" value = "11" />

</body>
</html>

Output

The screenshot shows a web browser window titled 'Form' with the URL 'file:///c/pl.html'. The page contains the following elements:

- Email Account Form**
- Enter Username** (text input field)
- Enter Password** (password input field)
- Address** (text area)
- Fields of interests** (checkboxes for Entertainment, Sports, News)
- Gender** (radio buttons for Male, Female)
- Select Languages** (multiple select dropdown with Marathi, Hindi, English options)
- Select Country** (dropdown menu showing India)
- Upload Photo** (input field with 'Browse...' button and 'No file selected.' message)
- Buttons**: Validate, Reset, Submit

**Syllabus Topic : Difference between HTML and HTML5****1.5 Difference between HTML and HTML5**

→ (Aug. 2014 - In Sem)

Q. Differentiate between HTML and HTML5**SPPU - Aug. 2014 - In sem, 5 Marks**

| Sr. No | Parameter | HTML | HTML5 |
|--------|-----------------------|---|---|
| 1. | Audio and Video | Audio and Video tags are not provided in HTML4 | Audio and Videos are provided in HTML5 e.g. <audio> and <video>. |
| 2. | Vector Graphics | Vector Graphics is only possible by use of technologies like VML, Silverlight, Flash etc. | Vector graphics is by default supported by HTML5 e.g. SVG and canvas |
| 3. | Tracing User Location | It is difficult to trace true GeoLocation of end user browsing the website particularly when it comes to mobile devices. | JS GeoLocation API in HTML5 is used to identify location of user browsing the website (with permission of user) |
| 4. | Cookies | HTML uses cookies. | It supports local storage instead of cookies. |
| 5. | Shapes | The different shapes such as circle, rectangle and triangle are not possible. | It is easy to draw different shapes such as circle, rectangle, and triangle. |
| 6. | Browser Support | Supported by old browsers | Supported by all new browsers. |
| 7. | Syntax | Doctype declaration in HTML is very long
<!DOCTYPE HTML PUBLIC "-//PhoenixGlobe//DTD HTML 4.01//EN" "http://www.PhoenixGlobe.com/TR/html4/strict.dtd"> | DOCTYPE declaration in HTML5 is very simple
"!DOCTYPE html" |
| 8. | Character Encoding | Character encoding in HTML is very long
<!DOCTYPE HTML PUBLIC "-//PhoenixGlobe//DTD HTML 4.0 Transitional//EN"> | Character encoding is very simple <meta charset="UTF-8"> |

Syllabus Topic : CSS**1.6 CSS****1.6.1 Introduction to Style Sheet**

→ (Aug. 2014 - In Sem, Dec. 2014, Oct. 2016)

Q. What do you mean by CSS?**SPPU - Aug. 2014 - In sem, 6 Marks****Q. Write a short note on: CSS.****Oct. 2016 - In sem, 5 Marks**

- **CSS - Cascading Style Sheet** is a simple design scripting language intended to simplify the process of making the web pages attractive with high level formatting.

→ **Use :** CSS is used to manage the look and feel of the web pages. CSS is used to control the color of the text, the style of fonts, the way columns are sized and laid out, different background images or colors used, the spacing between paragraphs, layout designs, variations in the display for various devices, screen sizes or resolutions and variety of other effects.

- The CSS is very easy to learn and understand but it has strong control over the presentation of web page. In general, CSS is integrated with the markup languages like HTML or XHTML.

- In simple words, CSS is nothing but declaration of style sheets which can be repeatedly used. It resembles to the function concept of C language. That means define once and use anytime anywhere repeatedly.

1.6.2 Advantages of CSS**Q. State the advantages of CSS.****(4 Marks)****Advantages of CSS**

- 1. Saves Time
- 2. Pages Load Faster
- 3. Easy Maintenance
- 4. Superior Styles to HTML
- 5. Multiple Device Compatibility
- 6. Offline Browsing
- 7. Platform Independence

Fig. C1.6 : Advantages of CSS

→ **1. CSS Saves Time**

Once written, CSS can be used in multiple HTML documents repeatedly. It helps to define style for different HTML elements and apply it to multiple web pages as per requirement.

→ **2. Pages Load Faster**

When CSS is used, there is no need to write HTML tag attributes repeatedly. Just write once and can be applied to all the occurrences of that tag in the webpage. As code is less, it speeds up the download process.

→ **3. Easy Maintenance**

Just change in the style sheet affects all the elements in different web pages.

→ **4. Superior Styles to HTML**

CSS has a wider range of attributes which helps to make the look of webpage far better than HTML.

→ **5. Multiple Device Compatibility**

Style sheet contents are compatible for multiple types of devices. With use of same HTML document, we can present various versions of a website.

→ **6. Offline Browsing**

CSS helps to store the web applications on local machines by using an offline cache. This helps to view offline websites. The cache provides fast loading of the website and improved overall performance.

→ **7. Platform Independence**

The CSS supports reliable platform independence and also supports all the latest browsers.

1.6.3 Inserting CSS in HTML Page

→ (Aug. 2015 - In sem, May 2017)

Q. What are the ways of creating style sheets? (5 Marks)

Q. What are types of CSS? Explain with example (program)

SPPU - May 2017, 5 Marks

Q. What is :

- (i) Inline CSS
- (ii) Internal CSS
- (iii) External CSS

SPPU - Aug. 2015 - In sem, 6 Marks

There are three ways of inserting a style sheet in the webpage:

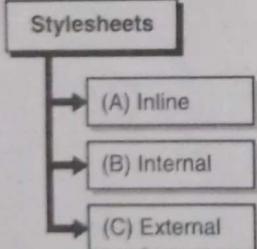


Fig. C1.7 : Stylesheets

→ **(A) Inline Style Sheet**

- An inline style is generally used to apply a unique style for an individual element.
- The Inline style is specific for the individual tag. The HTML "style" attribute is used by the inline style to style a particular tag. This is not suggestive, as each CSS change has to be made in each tag which has the inline style applied to it. The Inline style generally useful for an individual CSS change which we do not want to use repeatedly in the site.

Program 1.6.1 SPPU - May 2017

Write a simple code for inline style sheet.

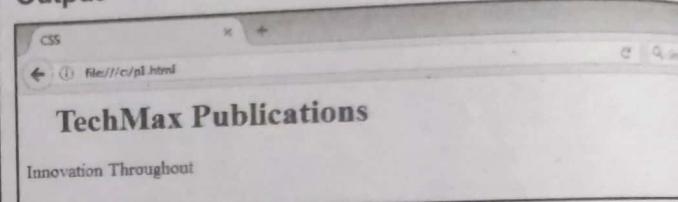
Ans. : Code for inline style sheet

```
<html>
<head>
```

```
<title>
Inline CSS
</title>
</head>
<body>
<h1 style="color:blue;margin-left:30px;">TechMax
Publications</h1>
<p style="color:red;font-size:18px">Innovation
Throughout</p>
</body>
</html>
```

- **<h1>** - This inline style will set color as blue and leave 30 px margin from left side
- **<p>** - This inline style will set the color of the paragraph as red and set size of font to 18 pixels.

Output



→ **(B) Internal Style Sheet**

- In internal style sheet, the CSS code is usually written in the head section of the webpage. This simplifies the applications of styles like classes or id's in contrast to repeated use of the code.
- While creating an internal style sheet in the web page, we have to use the **<style></style>** HTML tags in the Head section of the webpage. The entire code of the Internal CSS style sheet is included between the **<head></head>** section of the websites.
- Following is an example of how an Internal style sheet looks like.

Program 1.6.2 SPPU - May 2017

Write a simple code for internal style sheet.

Ans. : Code for internal stylesheet

```
<html>
<head>
<title>
Internal CSS
</title>
<style type="text/css">
body {background-color: linen}
h1 {color:blue;margin-left:30px}
p {color:red;font-size:18px}
</style>
</head>
<body>
<h1>TechMax Publications</h1>
<p>Innovation Throughout</p>
</body>
</html>
```

**Output**

Internal CSS

```
<html>
<head>
<title>TechMax Publications</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<link href="mystyle.css" rel="stylesheet"/>
</head>
<body>
<h1>TechMax Publications</h1>
<p>Innovation Throughout</p>
</body>
</html>
```

→ (C) External Style Sheet

- The External Style sheet is a file with extension .css which we link to website. This CSS file is used to declare the style sheets. Whenever any change is made in this css file, it gets reflected in all the web pages of the website. This simplifies our work and avoids making change in each and every page where the css is used.
- Now a day most of the websites use external style sheets. These are the styles which are written in a separate document or file and then linked to various web pages. External style sheets affects all the documents which are attached to, this means that if we have a 50-page website where all the pages use the same style sheet, just change in the style sheets will make visual change in all the web pages which are linked to it.
- This helps to make long term site management much easier.

Program 1.6.3 SPPU - May 2017

Write a simple code for external style sheet.

Ans. : Code for external style sheet

MyStyle.css

```
h1 {
    color: blue;
    font-family: verdana;
    font-size: 300%;
}
p {
    color: red;
    font-family: courier;
    font-size: 160%;
}
```

HTML File

```
<html>
<head>
<title>
External CSS
</title>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
<body>
<h1>TechMax Publications</h1>
<p>Innovation Throughout</p>
</body>
</html>
```

Output

External CSS

```
<html>
<head>
<title>TechMax Publications</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<link href="mystyle.css" rel="stylesheet"/>
</head>
<body>
<h1>TechMax Publications</h1>
<p>Innovation Throughout</p>
</body>
</html>
```

1.6.4 Selectors**☞ Use**

CSS **selectors** are used to *select the specific content which we want to style*. Selectors are the integral part of CSS rule set. CSS selectors choose the different HTML elements as per id, class, type, attribute etc.

☞ Types of selectors in CSS

Q. Explain types of selectors.

(4 Marks)

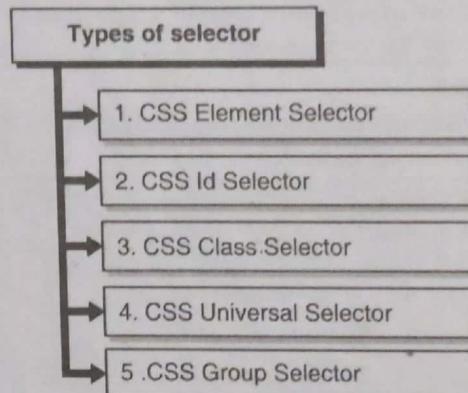


Fig. C1.8 : Types of selector

→ 1. CSS Element Selector

The HTML element by name is selected by the element selector

Program 1.6.4

Write a simple code to demonstrate the use of element selector.

Ans. : Code of element selector

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
    font-size: 20pt;
}
</style>
</head>
<body>
<p>Phoenix InfoTech</p>
<p id="para1">Computer Training Institute</p>

```



```
<p>Pune</p>
</body>
</html>
```

Output

→ 2. CSS Id Selector

- The id attribute of HTML element is used by the selector to select particular element. As the id is always unique within the page, it is better option to select single, unique element.
- It is expressed with the special character hash (#) which is followed by the id of the element.

Program 1.6.5

Write a code to demonstrate the use of Id selector.

Ans. : Code for Id selector

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    color: blue;
    font-size: 20pt
}
</style>
</head>
<body>
<p id="para1">www.PhoenixGlobe.com</p>
<p>The effect will not be applied here.</p>
</body>
</html>
```

Output

→ 3. CSS Class Selector

- The HTML elements with a specific class attribute is selected by the class selector

- It is used with a period character (.) which is followed by the class name.

Program 1.6.6

Write a code for class selector.

Ans. : Code for class selector

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: right;
    color: red;
    font-size: 20pt;
}
</style>
</head>
<body>
<h1 class="center">This heading is red and right-aligned.</h1>
<p class="center">This heading is red and right-aligned.</p>
</body>
</html>
```

Output

→ CSS Class Selector for specific element

We can apply effect to only specific element by using the element name with class selector.

Program 1.6.7

Write a code to apply the effect to only specific element using class selector.

Ans. : Code applying the effect to the specific element using class selector

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: blue;
    font-size: 20pt;
}
</style>
</head>
<body>
```



```
<h1 class="center"> This heading does not get the  
effect</h1>  
<p class="center">This paragraph is blue and center-  
aligned.</p>  
</body>  
</html>
```

Output

This heading does not get the effect
This paragraph is blue and center-aligned.

→ 4. CSS Universal Selector

The universal selector selects every element on the pages. It is used as a wildcard character.

Program 1.6.8

Write a code for universal selector.

Ans. : Code for universal selector

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
* {  
    color: green;  
    font-size: 20px;  
    text-align:center;  
}  
</style>  
</head>  
<body>  
<h2>Phoenix InfoTech</h2>  
<p>Computer Training Institute</p>  
<p id="para1">Pune</p>  
<p>Maharashtra</p>  
</body>  
</html>
```

Output

Phoenix InfoTech
Computer Training Institute
Pune
Maharashtra

→ 5. CSS Group Selector

The CSS group selector selects all the elements with the same style definitions. It helps to minimize the code. Comma is used as separator in selectors.

Program 1.6.9

Write a code for CSS group selector.

Ans. : Code for group selector

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1, h2, p {  
    text-align: center;  
    color: blue;  
    font-size: 20pt  
}  
</style>  
</head>  
<body>  
<h1>www.PhoenixGlobe.com</h1>  
<h2>This is heading</h2>  
<p>This is a paragraph.</p>  
</body>  
</html>
```

Output

www.PhoenixGlobe.com
This is heading
This is a paragraph.

Syllabus Topic : XML

1.7 XML

1.7.1 Introduction to XML

→ (May 2017)

Q. What is XML?

SPPU - May 2017, 4 Marks

- XML stands for eXtensible Markup Language.
- XML is basically designed to store and transport data.
- XML supports both human- and machine-readable data format.
- XML was designed to be self-descriptive.
- XML is a W3C Recommendation.
- XML is a simple text-based format which represents information in structured formats like documents, transactions, data, invoices, books etc. This language is derived from comparatively older standard format called **SGML (Standard Generalized Markup Language)**, to make it more suitable for Web use.

1.7.2 Need of XML

→ (May 2017)

Q. Why we need XML?

SPPU - May 2017, 4 Marks

There are several reasons for the need of XML as follows:

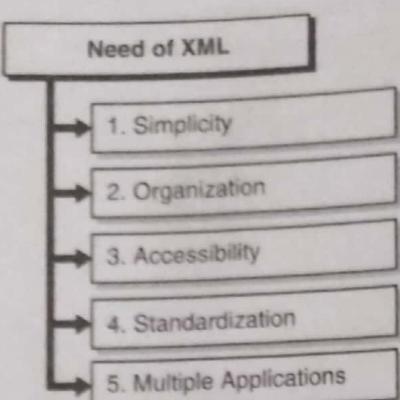


Fig. C1.9 : Need for XML

→ 1. Simplicity

XML can be easily understood. We can create our own tags and build the application. We are free to develop the system as per our requirements and with our own conventions. This makes the thing very simple for us.

→ 2. Organization

The design process can be segmented to build the platform. Data can be stored on one page while the formatting rules can be stored on another page. It is possible to create the data page to store the content first and later on we can work on design. XML allows us to create the website in stages and stay organized in the entire process.

→ 3. Accessibility

Data can be divided in XML. This makes the access of data easy and fast whenever there is need of making change in the data.

→ 4. Standardization

XML is an international standard. This means XML document can be viewed anywhere in the world.

→ 5. Multiple Applications

"Write once, use anywhere, any number of times" rule is applied to XML. For XML data we can create any number of display pages as we want. XML allows us to create various styles and formats for a single page as per requirement.

1.7.3 XML Key Components

Q. Explain various Key Components of XML. (8 Marks)

→ 1. Character

An XML document is a string of characters. Almost every legal Unicode character may appear in an XML document.

→ 2. Processor and Application

The processor analyzes the markup and passes structured information to an application. The specification places requirements on what an XML processor must do and not do, but the application is outside its scope. The processor (as the specification calls it) is often referred to formally as an XML parser.

XML Key Components

- 1. Character
- 2. Processor and Application
- 3. Markup and Content
- 4. Tag
- 5. Element
- 6. Attribute
- 7. XML Declaration

Fig. C1.10 : XML Key Components

→ 3. Markup and Content

The characters making up an XML document are divided into markup and content, which may be distinguished by the application of simple syntactic rules. Generally, strings that constitute markup either begin with the character < and end with a >, or they begin with the character & and end with a ;. Strings of characters that are not markup are content.

→ 4. Tag

A tag is a markup construct that begins with < and ends with >. Tags come in three flavors:

- Start-tag, such as <section>
- End-tag, such as </section>
- Empty-element tag, such as <line-break />

→ 5. Element

An element is a logical document component that either begins with a start-tag and ends with a matching end-tag or consists only of an empty-element tag. The characters between the start-tag and end-tag, if any, are the element's content, and may contain markup including other elements which are called as child elements.

An example is

<greeting>Hello, world!</greeting>.

Another is

<line-break />.

→ 6. Attribute

- An attribute is a markup construct consisting of a name-value pair that exists within a start-tag or empty element tag.

→ Example

,
where the names of the attributes are "src" and "alt",
and their values are " Rose.jpg" and " Rose".



- Another example is <step number="3">Connect A to B.</step>, where the name of the attribute is "number" and its value is "3".
- An XML attribute can only have a single value and each attribute can appear at most once in each element. In the common situation where a list of multiple values is desired, this must be done by encoding the list into a well-formed XML attribute with some format beyond what XML defines itself. Usually this is either a comma or semi-colon delimited list or, if the individual values are known not to contain spaces, a space-delimited list can be used.
- <div class="inner greeting-box">Welcome!</div>, where the attribute "class" has both the value "inner greeting-box" and also indicates the two CSS class names "inner" and "greeting-box".

→ 7. XML Declaration

XML documents may begin with an XML declaration that describes some information about themselves. An example is <?xml version="1.0" encoding="UTF-8"?>.

Program 1.7.1

Write a code to display a data with the help of XML.

Ans. :

```
<?xml version="1.0" encoding="UTF-8"?>
<contact-info>
  <name>Kunal</name>
  <company>Phoenix InfoTech</company>
  <phone>020 64700515</phone>
</contact-info>
```

Output

```
/c/p1.html
file:///c/p1.html
Kunal Phoenix InfoTech 020 64700515
```

OR

```
<?xml version = "1.0"? encoding="UTF-8"?>
<student-data>
<student>
  <rnno>101</rnno>
  <name>Kunal</name>
  <subject>HTML</subject>
</student>
<student>
  <rnno>103</rnno>
  <name>Ishita</name>
  <subject>CSS</subject>
</student>
<student>
  <rnno>102</rnno>
  <name>Shrey</name>
```

```
<subject>JavaScript</subject>
</student>
<student>
  <rnno>104</rnno>
  <name>Shravi</name>
  <subject>PHP</subject>
</student>
</student-data>
```

Output

```
/c/p1.html
file:///c/p1.html
101 Kunal HTML 103 Ishita CSS 102 Shrey JavaScript 104 Shravi PHP
```

1.7.4 Difference between HTML and XML

→ (May 2016)

Q. Differentiate between HTML and XML.

SPPU - May 2016, 8 Marks

Sr. No	Parameter	HTML	XML
1.	Long Form	HTML stands for HyperText Markup Language.	XML stands for eXtensible Markup Language.
2.	Purpose	HTML was designed to display data which concentrates on look of data.	XML was designed to store and transport the stored data.
3.	Markup Language	HTML is a markup language itself.	XML provides a framework for defining markup languages.
4.	Use	HTML is used for presentation purpose.	XML is not used for presentation purpose.
5.	Case Sensitivity	HTML is case insensitive.	XML is case sensitive.
6.	Tags	HTML uses only predefined tags.	XML allows user defined tags
7.	Restrictions	HTML is flexible related to syntax. E.g. no need to close the tags.	XML makes it compulsory for the user to close all the tags.
8.	White Space	HTML does not preserve white space.	XML preserves white space.
9.	Static or Dynamic	HTML is static.	XML is dynamic.

1.7.5 Transforming XML into XSLT

Q. How to transform XML into XSLT?

(8 Marks)

- XSLT (eXtensible Stylesheet Language Transformations) is the standard style sheet language for XML.
- XSLT is better than CSS. XSLT helps to add/remove elements and attributes to or from the output file. Also the elements can be rearranged and sort, execute tests and decide which elements to hide or display.



- XPath is used by XSLT to search information in an XML document.
- XSL stands for EXtensible Stylesheet Language. It is considered as same to XML as CSS is to HTML.

Need for XSL

- In HTML all tags used are predefined such as font, body, heading etc. and browser has knowledge of adding styles to them and display them to the end user with the help of CSS styles. This is not same in XML as it consists of user defined tags.
- Hence to understand and style an XML document, W3C (World Wide Web Consortium) has provided XSL which can work as XML based Stylesheet Language. An XSL document is used to specify the way by which a browser should render an XML document.

Following are the main parts of XSL -

- XSLT - Converts XML document into number of other types of documents.
- XPath - Helps to navigate through XML document.
- XSL-FO - Helps to format XML document.

What is XSLT ?

XSLT stands for Extensible Stylesheet Language Transformations. It is used to transform XML data from one format into another.

How XSLT Works ?

- In an XSLT stylesheet we can define the transformation rules which is to be applied on the target XML document. The format of XSLT stylesheet is XML.
- The XSLT stylesheet is taken by the XSLT Processor which applies the transformation rules on the target XML document and further creates a formatted document in the format of XML, HTML, or text.
- XSLT formatter then uses this formatted document to generate the actual output to display to the end user.

Advantages of XSLT

- Programming is independent. An xsl file (which is a XML document) is used to write the transformations.
- It is possible to update the output by just making changes in the transformations in xsl file. There is no need of changing any code. Hence it is easily possible for web designers to update the stylesheet and observe the change in the output quickly.

Program 1.7.2

Write a code to transform the XML document in XSLT.

Ans. : Code to transform the XML document in XSLT

Syntax

- Let's consider the following sample students.xml file, which is required to be transformed into a well-formatted HTML document.

students.xml

```
<?xml version = "1.0"?>
<class>
  <student rno = "101">
    <fname>Kunal</fname>
    <lname>B.</lname>
    <marks>85</marks>
  </student>
  <student rno = "102">
    <fname>Ishita</fname>
    <lname>B.</lname>
    <marks>90</marks>
  </student>
  <student rno = "103">
    <fname>Atharv</fname>
    <lname>P.</lname>
    <marks>75</marks>
  </student>
</class>
```

Now for above XML document we have to define an XSLT style sheet document to fulfill the following criteria

- The title of page should be Students.
- Page should contain table having student details.
- Columns should have give headers: Roll_No, First_Name, Last_Name, Marks

Step 1 : Create XSLT document

Now to fulfill above requirements, we will create an XSLT document.

students.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0"
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">

  <xsl:template match = "/">

    <html>
      <body>
        <h3>Students</h3>

        <table border = "2">
          <tr bgcolor = "lightgray">
            <th>Roll_No</th>
            <th>First_Name</th>
            <th>Last_Name</th>
            <th>Marks</th>
          </tr>
          <xsl:for-each select = "class/student">
            <tr>
              <td>
                <xsl:value-of select = "@rno"/> </td>
                <td><xsl:value-of select = "fname"/> </td>
```



```

<td><xsl:value-of select = "marks"/></td>

</tr>
</xsl:for-each>

</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Step 2 : Link the XSLT Document to the XML Document

Make changes in student.xml document by adding xmlstylesheet tag. Set href value to students.xsl

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "students.xsl"?>
<class>
-----
</class>

```

Step 3 : View the XML Document in a browser

Roll_No	First_Name	Last_Name	Marks
101	Kunal	B.	85
102	Ishita	B.	90
103	Atharv	P.	75

Syllabus Topic : DTD

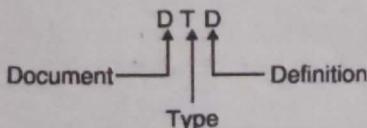
1.8 DTD

1.8.1 Introduction

→ (Dec. 2014)

Q. Explain the term Document Type Definition (DTD).

SPPU - Dec. 2014, 4 Marks



Definition : XML Document Type Declaration which is also known as DTD is a way which helps to describe specifically the XML language. DTDs examine the validity of structure and vocabulary of an XML document against the grammatical rules and regulations of the suitable XML language.

- An application can take help of a DTD to check that XML data is valid.
- Generally an XML document is defined as:
 - o **Well-formed** : An XML document is considered as well-formed if it follows all the XML rules like tags must be correctly nested, open-close tags are balanced, and empty tags ends with '>' **OR**
 - o **Valid** : Valid means, any XML document is not just only well-formed but also ensures availability of DTD that indicates which tags it uses, attributes of those tags, and which tags can be nested inside other tags, among other properties.
- In the Fig. 1.8.1 we can observe that a DTD is used to structure the XML document:

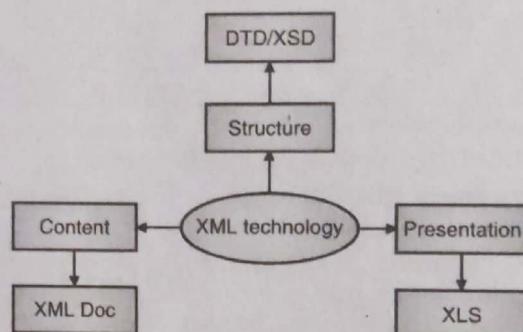


Fig. 1.8.1 : DTD structure

1.8.2 Features of DTD

Q. Write features of DTD.

(2 Marks)

DTD describes following important points:

- The elements which can appear in an XML document.
- The order of elements.
- Optional and mandatory elements.
- Attributes of elements and also whether they are optional or mandatory.
- Whether or not attributes have default values.

1.8.3 Advantages of using DTD

Q. State advantages of DTD.

(2 Marks)

1. **Documentation** : User defined formats are allowed for XML files. These formats help user/developer to understand the structure of the data.
2. **Validation** : It gives a system to ensure the validity of XML files by verifying whether the elements are in proper sequence, mandatory elements and attributes are exactly at their locations and placed in correct way etc.

1.8.4 Schema

Q. Write note on Schema of DTD.

(4 Marks)

Q. Write note on internal and external DTD.

(4 Marks)

On its declaration basis, there are two types of DTD in the XML document:

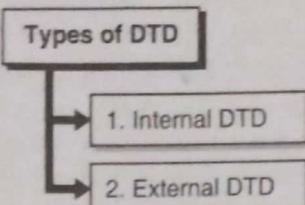


Fig. C1.11 : Types of DTD

1. **Internal DTD** : Declared inside the file itself
2. **External DTD** : Declared in a separate file

☞ Syntax

```

<!DOCTYPE element DTD identifier
[
declaration1
declaration2
.....
]>
  
```

In the above syntax

- **<!DOCTYPE** delimiter is beginning of DTD
- An **element** indicates the parser to parse the document from the given root element.
- **DTD identifier** is an identifier for the document type definition. This may be the path of a file on the system or URL of a file on the internet. When DTD points to external path, it is known as **external subset**.
- The **square brackets []** indicates optional list of entity declarations which is called as **internal subset**.

→ 1. Internal DTD

In this DTD elements are declared inside the XML files. In XML declaration the *standalone* attribute is set to yes to reference it as internal DTD.

☞ Syntax

```

<!DOCTYPE root-element [element-declarations]>
root-element - name of root element
element-declarations - is where the elements are declared.
  
```

☞ Example (Declaration)

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"
?>
<!DOCTYPE details [
<!ELEMENT details (emp_name,company,emp_phone)>
<!ELEMENT emp_name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT emp_phone (#PCDATA)>
]>
<Details>
< emp_name>Kunal</ emp_name>
<company>Phoenix InfoTech</company>
< emp_phone>020 64700515</ emp_phone>
</ Details >
  
```

Let us discuss the above code:

☞ Start Declaration

XML declaration is started with following statement

```

<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
  
```

☞ DTD

Immediately after the XML header, the *DTD* follows, usually referred as the DOCTYPE:

```

<!DOCTYPE details [
  
```

There is an exclamation mark (!) at the beginning of the element name in DOCTYPE declaration. The DOCTYPE instructs the parser that a DTD is linked with this XML document.

☞ DTD Body

After DOCTYPE declaration, there is body of the DTD. Here we can declare elements, attributes, entities, and notations:

```

<!ELEMENT details
(emp_name,company,emp_phone)>
<!ELEMENT emp_name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT emp_phone (#PCDATA)>
  
```

Several elements are declared here. **<!ELEMENT name (#PCDATA)>** implies the element *name* to be of type "#PCDATA". Here the meaning of #PCDATA is parseable text data.

☞ End Declaration

At the end, DTD's declaration section is closed with a closing bracket and a closing angle bracket (**>**). This efficiently ends the definition, and then, the XML document follows immediately.

☞ Rules

- The document type declaration should be at the beginning of the document. It can only be preceded by the XML header.
- The element declarations should begin with an exclamation mark.
- The Name in the DTD and element type of the root element must be same.

→ 2. External DTD

- In this DTD the elements are declared out of the XML file. They are retrieved by mentioning the system attributes which may be either the legal .dtd file or any valid URL.
- When *standalone* attribute in the XML declaration is set to **no**, it indicates that declaration get information from the source outside the file.

☞ Syntax

```

<!DOCTYPE root-element SYSTEM "file-name">
  
```

Here the *file-name* is file with .dtd extension.

**Example**

→ (Dec. 2014)

Q. Write suitable example of DTD.**SPPU - Dec. 2014, 4 Marks**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"
?>
<!DOCTYPE Details "Details.dtd">
< Details>
<emp_name>Kunal </emp_name>
<company>Phoenix InfoTech </company>
<emp_phone>020 64700515 </emp_phone>
</Details>
```

The content of the DTD file **address.dtd** are as shown:

```
<!ELEMENT details (emp_name,company,emp_phone)>
<!ELEMENT emp_name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT emp_phone (#PCDATA)>
```

Types of External DTD

The external DTD can be referred by using **system identifiers** or **public identifiers**.

Types of External DTD

- 1. System Identifiers
- 2. Public Identifiers

Fig. C1.12 : Types of External DTD**1. System Identifiers**

This helps to specify the location of an external file which contains DTD declarations.

```
<!DOCTYPE name SYSTEM "Details.dtd" [...]>
```

2. Public Identifiers

A mechanism is provided by public identifier to locate DTD resources.

```
<!DOCTYPE name PUBLIC "-// XML Docs//DTD
Exm//EN">
```

It starts with keyword PUBLIC and then followed by a specialized identifier.

1.8.5 Elements**Q.** Explain the XML Elements with their types. (8 Marks)**Definition**

XML elements are the building blocks of an XML document. Elements are considered as a container to store elements, attributes, text, media objects or mixture of all.

Use

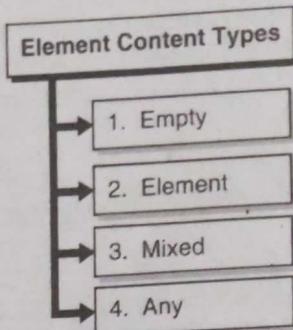
An ELEMENT declaration is used to declare DTD element. When DTD validates the XML file, parser at the beginning checks the root element and then the child elements.

Syntax

```
<!ELEMENT elementname (content)>
```

Element Content Types

Elements contents in the declaration of DTD are categorized as:

**Fig. C1.13 : Element Content Types****1. Empty content**

The element declaration do not have any content.

```
<?xml version="1.0"?>
<!DOCTYPE hr[
    <!ELEMENT Details EMPTY>
]>
<Details />
```

2. Element content

Here the content is included in the elements within parentheses.

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
<!DOCTYPE Details [
    <!ELEMENT Details
(emp_name,company,emp_phone)>
        <!ELEMENT emp_name (#PCDATA)>
        <!ELEMENT company (#PCDATA)>
        <!ELEMENT emp_phone (#PCDATA)>
]>
<Details>
    <emp_name>Kunal </emp_name>
    <company>Phoenix InfoTech </company>
    <emp_phone>020 64700515 </emp_phone>
</Details>
```

3. Mixed Element Content

In this we use the combination of (#PCDATA) and children elements. PCDATA means parsed character data, that means, text which is not markup. Here the text can appear by itself or it can be placed between the elements.

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes" ?>
<!DOCTYPE Details [
    <!ELEMENT Details (#PCDATA | name)*>
    <!ELEMENT emp_name (#PCDATA)>
]>
```



```
<Details>
The text is combined with child element
<emp_name>Kunal</emp_name>
</Details>
```

→ 4. Any Element Content

An element can be declared using the ANY keyword in our content. It is usually referred as mixed category element. When the type of content is not fixed, ANY is useful.

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<!DOCTYPE Details [
    <!ELEMENT Details ANY>
]>
< Details >
Some Text can be written here
</ Details >
```

1.8.6 Attributes

Q. Explain the attributes of XML Elements. (4 Marks)

- Attribute provides some more data about an element or more precisely it defines a property of an element. In XML the attribute is in the form of a name-value pair.
- There may be multiple unique attributes to an element.

☞ Syntax

Basic syntax of DTD attributes declaration is as follows:

```
<!ATTLIST element-name attribute-name attribute-type
attribute-value>
```

In the above syntax

- The DTD attributes start with **<!ATTLIST** keyword if the element contains the attribute.
- **element-name** specifies the name of the element to which the attribute applies.
- **attribute-name** specifies the name of the attribute which is included with the element-name.
- **attribute-type** defines the type of attributes.
- **attribute-value** takes a fixed value that the attributes must define.

☞ Example

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

☞ <!DOCTYPE Syntax

The syntax of DTD attributes declaration:

```
<!ATTLIST element-name attribute-name attribute-type
attribute-value>
```

In this syntax

- If there is attribute in element, then the DTD attributes start with **<!ATTLIST** keyword.
- **element-name** - name of the element to which the attribute applies.

- **attribute-name** - name of the attribute which is included in the element-name.
- **attribute-type** - type of attributes.
- **attribute-value** - fixed value

☞ Example

```
<?xml version = "1.0"?>
<!DOCTYPE Details [
    <!ELEMENT Details (emp_name )>
    <!ELEMENT emp_name ( #PCDATA )>
    <!ATTLIST emp_name id CDATA #REQUIRED>
]>
<Details>
    <emp_name id="123">Kunal</emp_name>
</ Details >
```

1.8.6(A) Rules of Attribute Declaration

Q. Explain the rules of Attribute Declaration. (2 Marks)

- It is necessary to declare all the attributes in Document Type Definition (DTD) using an Attribute-List Declaration which we want to use in XML document.
- Attributes may appear in either start or empty tags.
- ATTLIST keyword should be in upper case.
- Duplication of attribute names is not allowed in the attribute list for a given element.

1.8.6(B) Attribute Types

Q. List the attribute types. (2 Marks)

When attributes are declared, we can specify how the processor will handle the data of attribute value. The attribute types are categorized in three different categories:

1. String type
2. Tokenized types
3. Enumerated types

Syllabus Topic : Introduction to JSON

1.9 Introduction to JSON

- **JSON** stands for **JavaScript Object Notation** - is a format used to share the data. As its name suggests, JSON is basically derived from the client side scripting language – JavaScript, but it can be used by many languages like Python, Ruby, PHP, and Java.
- As stand alone, JSON uses the extension ".json". When it is defined in another file format like ".html", it may appear inside quotes as a JSON string, or it may be an object which is assigned to a variable. This format is simple to transmit between web server and client or browser.
- JSON is very readable and lightweight. It is a good alternative to XML and also requires very less Syntax and Structure.
- In JSON, the object is a key-value data format which is usually included in curly braces.

Simple format of JSON

```
{
  "first_name": "Phoenix",
  "last_name": "InfoTech",
  "location": "Pune"
}
```

JSON can also be written all in one line.

```
{"first_name": "Phoenix", "last_name": "InfoTech",
"location": "Pune"}
```

Data types in JSON

Q. List data types in JSON. (2 Marks)

1. strings
2. numbers
3. objects
4. arrays
5. booleans (true or false)
6. null

1.10 Exam Pack (University and Review Questions)

Syllabus Topic : Introduction to Web Technology, Internet and WWW, Web Site Planning and Design Issues

- Q. Explain the term Internet.
(Refer section 1.1.1-A) (2 Marks)
- Q. Explain the term WWW
(Refer section 1.1.1-B) (2 Marks)
- Q. Explain the term Website
(Refer section 1.1.2-A) (2 Marks)

Syllabus Topic : Website Planning and Design Issues

- Q. Describe phases of web site development in brief.
(Refer section 1.2.1) (6 Marks) (Aug. 2014 - In sem)
- Q. List and explain different design issues in web design.
(Refer section 1.2.2) (4 Marks) (Aug. 2015 - In sem)

Syllabus Topic : HTML

- Q. What is HTML ? (Refer section 1.3) (2 Marks)
- Q. Explain the basic structure of HTML document
(Ref. Sec 1.3.1) (6 Marks) (Oct. 2016 - In sem)

Syllabus Topic : HTML Elements

- Q. Give the list with definition of HTML components.
(5 Marks) (Dec. 2014, May 2015, Oct 2016 - In Sem)
- Q. List various tags in HTML with simple example for a web page. (5 Marks) (Dec. 2015)
- Q. Explain heading element in HTML.
(Refer section 1.4.1) (2 Marks)
- Q. Explain paragraph element in HTML
(Refer section 1.4.2) (2 Marks)
- Q. Explain line break element in HTML.
(Refer section 1.4.3) (2 Marks)
- Q. Explain setting colors in HTML.
(Refer section 1.4.4 (A)) (4 Marks)

- Q. Explain setting fonts in HTML.
(Refer section 1.4.4(B)) (4 Marks)
- Q. Explain following HTML Tags with suitable example.
<a>.
(Refer section 1.4.5 and program 1.4.6) (2.5 Marks) (Aug. 2014 - In sem)
- Q. What are the Hyperlinks and Anchors in the HTML?
Explain with suitable example.
(Refer section 1.4.5 and program 1.4.6) (4 Marks) (Oct. 2016 - In Sem)
- Q. Explain how frames are constructed in HTML document.
(Refer section 1.4.6, program 1.4.9 and 1.4.10) (5 Marks) (May 2015, Dec. 2015)
- Q. How to create a frame in HTML? Explain with example.
(Refer section 1.4.6, program 1.4.9 and 1.4.10) (May 2017) (5 Marks)
- Q. Explain the concept of iframe with suitable example
(Refer section 1.4.6(A)) (2 Marks)
- Q. Explain the Lists in HTML.
(Refer section 1.4.7) (4 Marks)
- Q. How to create Tables on the web pages using HTML?
(Refer section 1.4.8) (4 Marks) (Aug. 2014 - In sem)
- Q. What's the difference between <tr> and <td>?
(Refer section 1.4.8(A)) (2 Marks) (May 2015)
- Q. Write HTML code which include table.
(Refer program 1.4.18) (2 Marks) (May 2015 - In sem)
- Q. Explain Colspan and Rowspan attributes of table tag with suitable examples (programs).
(Refer section 1.4.8(B)) (4 Marks)
- Q. Explain how the images can be inserted in HTML document.
(Refer section 1.4.9 (A) and program 1.4.21) (5 Marks) (Aug. 2016 - In sem)
- Q. Explain Image Map with example.
(Refer section 1.4.9(B) and program 1.4.23) (5 Marks) (May 2015, Dec. 2015)
- Q. Explain Form element in HTML.
(Refer section 1.4.9 (C)) (8 Marks)
- Q. Explain Different Form elements / Form Controls
(Refer section 1.4.9 (C)) (4 Marks)
- Q. Syllabus Topic : Difference between HTML and HTML5
- Q. Differentiate between HTML and HTML5
(Refer section 1.5) (5 Marks) (Aug. 2014 - In sem)
- Q. Syllabus Topic : CSS
- Q. What do you mean by CSS?
(Refer section 1.6.1) (6 Marks) (Aug. 2014 - In sem)
- Q. Write a short note on: CSS Style Sheets.
(Refer section 1.6.1) (5 Marks) (Oct. 2016 - In sem)
- Q. State the advantages of CSS.
(Refer section 1.6.2) (4 Marks)
- Q. What are the ways of creating style sheets?
(Refer section 1.6.3) (5 Marks)
- Q. What are types of CSS? Explain with example(program)
(Refer section 1.6.3, programs 1.6.1, 1.6.2 and 1.6.3) (5 Marks) (May 2017)



Q. What is :

- i) Inline CSS
- ii) Internal CSS
- iii) External CSS (*Refer section 1.6.3*) (6 Marks)

(Aug. 2015 - In Sem)

Q. Explain types of selectors.

(*Refer section 1.6.4*) (4 Marks)

☞ Syllabus Topic : XML

Q. What is XML? (*Refer section 1.7.1*) (4 Marks)

(May 2017)

Q. Why we need XML? (*Refer section 1.7.2*) (4 Marks)

(May 2017)

Q. Explain various Key Components of XML

(*Refer section 1.7.3*) (8 Marks)

Q. Differentiate between HTML and XML.

(*Refer section 1.7.4*) (8 Marks)

(May 2016)

Q. How to transform XML into XSLT?

(*Refer section 1.7.5*) (8 Marks)

☞ Syllabus Topic : DTD

Q. Explain the term Document Type Definition (DTD).

(*Refer section 1.8.1*) (4 Marks)

(Dec. 2014)

Q. Write features of DTD. (*Refer section 1.8.2*) (2 Marks)

Q. State advantages of DTD.

(*Refer section 1.8.3*) (2 Marks)

Q. Write note on Schema of DTD.

(*Refer section 1.8.4*) (4 Marks)

Q. Write note on internal and external DTD

(*Refer section 1.8.4*) (4 Marks)

Q. Write suitable example of DTD.

(*Refer section 1.8.4*) (4 Marks)

(Dec. 2014)

Q. Explain the XML Elements with their types.

(*Refer section 1.8.5*) (8 Marks)

Q. Explain the attributes of XML Elements.

(*Refer section 1.8.6*) (4 Marks)

Q. Explain the rules of Attribute Declaration.

(*Refer section 1.8.6 (A)*) (2 Marks)

Q. List the attribute types.

(*Refer section 1.8.6 (B)*) (2 Marks)

☞ Syllabus Topic : Introduction to JSON

Q. List data types in JSON. (*Refer section 1.9*) (2 Marks)



Client Side Technologies

Syllabus Topics

JavaScript : Overview of JavaScript, Using JS in an HTML (Embedded, External), Data types, Control Structures, Arrays, Functions and Scopes, Objects in JS,

DOM : DOM levels, DOM Objects and their properties and methods, Manipulating DOM

jQuery : Introduction to jQuery, Loading jQuery, Selecting elements, changing styles, creating elements, appending elements, removing elements, handling events.

Syllabus Topic : JavaScript - Overview of JavaScript

2.1 Overview of JavaScript

Q. What is JavaScript ?

(4 Marks)

- JavaScript is an open source probably the most popular client side scripting language which is supported by all the browsers.
- ☛ **Use :** JavaScript is an extremely powerful **client-side scripting language**. This language is used mostly for increasing the interaction of an end user with the webpage. JavaScript helps to make our webpage more lively and interactive. JavaScript is widely used in mobile application development as well as in game development.
- JavaScript supports dynamic scripting. It is lightweight and mostly used as an integral part of web pages where there is need of client side script and dynamic pages. JavaScript supports object oriented concepts.
- JavaScript was developed by Mr. Brendan Eich in 1995 who was working in Netscape.
- JavaScript was initially called as LiveScript and later on the name is changed to JavaScript. The syntax of JavaScript is typically influenced by the C programming language.

☛ Client-side JavaScript

- This is the most common form of the language. The script (code) of JavaScript is embedded into HTML or referenced by an HTML document so as to be interpreted by the browser.

- It proves that the web page not necessarily is a static HTML, but it can include scripts that interact with the end user, controls the browser, and create HTML content dynamically.
- The client-side features of JavaScript have various benefits over the previous traditional CGI (Common Gateway Interface) server-side scripts. For example, JavaScript can be used to verify whether the user has given a valid e-mail id in a form field or not.
- When user submits the form, the JavaScript validates the data, if all the given data is valid then only it is submitted to the Web Server.
- JavaScript helps to handle events fired by user like button clicks, navigation through links, and other actions which are taken by user explicitly or implicitly.

2.1.1 Characteristics or Advantages of JavaScript ➔ (May 2017)

Q. Explain characteristics of JavaScript.

SPPU - May 2017, 4 Marks

- JavaScript is a **lightweight, interpreted** client side scripting language.
- Designed **for developing network-based** applications.
- JavaScript is **complementary to Java**.
- JavaScript is **complementary to and integrated with HTML**.
- It is **Open source** and **cross-platform**.



- The user input is validated before sending the page to the server. This **minimizes the server traffic**, which tends to fewer loads on the server.
- There is no need for the user to wait to see if something have been forgotten to enter.
- Interactive interfaces can be created which can give responses to end user actions like mouse or keyboard activities.
- JavaScript can include elements like drag-drop components and sliders to provide a feel of rich interface to the users.

2.1.2 JavaScript Development Tools

Q. Enlist JavaScript editing tools. (2 Marks)

☞ **Advantage :** One of main advantage of JavaScript is that there is no need of development tools which are expensive. A simple text editor like Notepad can be used to write the scripts. Inside the context of a web browser, JavaScript is an interpreted language; hence there is no need to buy a compiler.

☞ JavaScript Editing Tools

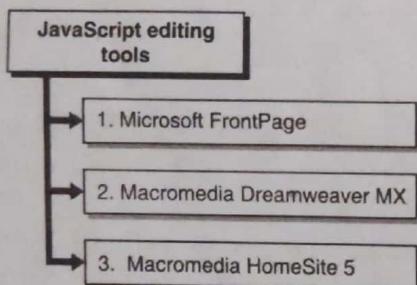


Fig. C2.1 : JavaScript editing tools

→ 1. Microsoft FrontPage

This is popular product of Microsoft. It provides various JavaScript tools to web developers for assistance to create interactive websites.

→ 2. Macromedia Dreamweaver MX

It is very popular HTML and JavaScript tool used for web development professionally. It provides various JavaScript components to handle databases, and supports new standards like XHTML and XML.

→ 3. Macromedia HomeSite 5

This is a product of Macromedia which is well-liked HTML and JavaScript editor. It helps to effectively manage personal websites.

2.1.3 Difference between Java and JavaScript → (May 2015)

Q. Write the difference between java and JavaScript.

SPPU - May 2015, 8 Marks

Sr. No	Parameter	Java	JavaScript
1.	Execution	Java creates application which can be executed on virtual machine or browser.	JavaScript creates application which can be executed on browser only.

Sr. No	Parameter	Java	JavaScript
2.	Features	Java code allows programmer full functionality.	JavaScript code contains limited number of commands and features.
3.	Naming	The first name of Java was OAK and was developed by James Gosling, Sun MicroSystems.	JavaScript was earlier known as LiveScript and was developed by Brendan Eich, Netscape.
4.	Type Safety	Java is high-level, compiled and strongly type language.	JavaScript is text based and weakly typed language.
5.	Variables	Variables are created using the data type names like int, char, double etc.	Variables are created using the var keyword.
6.	Extension	Java program has file extension ".Java" and after compilation it creates ".class" file.	JavaScript file has file extension ".js" or ".html".
7.	Objects	Objects of Java are class based.	Objects of JavaScript are prototype based.
8.	Scope	Java has block based scope.	JavaScript has function based scope and object based context.

Syllabus Topic : JavaScript - Using JS in an HTML (Embedded, External)

2.2 Using JS in an HTML (Embedded, External)

2.2.1 JavaScript Syntax

Q. Explain syntax of JavaScript.

(2 Marks)

- The code (script) of JavaScript is written in the script opening `<script>` and closing `</script>` HTML tags in a web page. Usually the `<script>` tag is allowed anywhere in the html page but the `<head>` section is normally recommended.
- The `<script>` tag basically notifies the browser that the code written is a script of JavaScript.

☞ Syntax

Syntax of JavaScript will appear as follows :

```

<script ...>
  JavaScript statements;
</script>
  
```

- The `<script>` tag has two important attributes which basically serves the same purpose.



- **Language and Type :** These attributes specify the name of scripting language used. Here its value will be "JavaScript" for attribute *Language* and "text/JavaScript" for attribute *Type*.
- JavaScript segment will look like :

```
<script language="JavaScript">  
    JavaScript statements  
</script>
```

Or

```
<script type="text/JavaScript">  
    JavaScript statements  
</script>
```

2.2.2 Embedded JavaScript

Program 2.2.1

Write an embedded JavaScript code displaying welcome message.

Soln. :

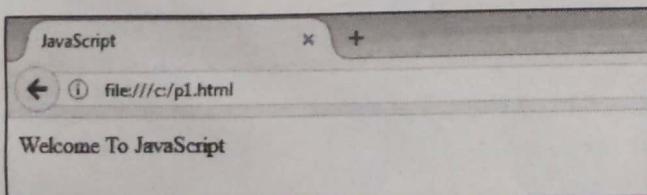
Embedded JavaScript code displaying welcome message

The JavaScript code can be embed into HTML file using the `<script>` tag

```
<!DOCTYPE html>  
<html>  
<head>  
<title>  
JavaScript  
</title>  
<script language="JavaScript">  
document.write("Welcome To JavaScript");  
</script>  
</head>  
<body>  
</body>  
</html>
```

document.write : This method writes a string into HTML document.

Output



Explanation

Outside the `<script>` element, the HTML tags can be used in their normal way, but if we want to use the HTML tags in `<script>` section, then `document.write()` method is used.

Program 2.2.2

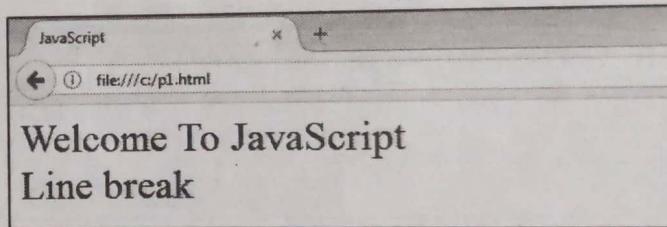
Write a program to demonstrate line break example.

Soln. :

Program to demonstrate the line break

```
<html>  
<head>  
<title>  
JavaScript  
</title>  
</head>  
  
<body>  
<font size=6 color="blue">  
  
<script type="text/JavaScript">  
document.write("Welcome To JavaScript");  
document.write("<br>Line break");  
</script>  
  
</font>  
</body>  
</html>
```

Output



2.2.3 External JavaScript

- An external JavaScript file can be created to embed it in many html pages.
- It supports the concept of **code reusability** as single JavaScript file can be embed into several html pages. An external JavaScript file is saved by the extension ".js".

Program 2.2.3

Write an external JavaScript code demonstrating welcome message.

Soln. :

External JavaScript code demonstrating welcome message

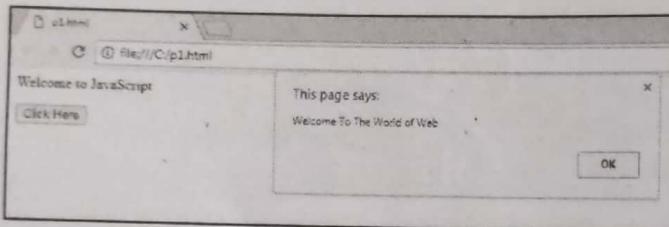
MyFile.js

```
function msg()  
{  
    alert("Welcome To The World of Web");  
}
```

**HTML page**

```
<html>
<head>
<script language="JavaScript" src="MyFile.js">
</script>
</head>

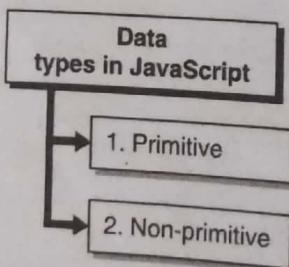
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="Click Here"
onclick="msg()"/>
</form>
</body>
</html>
```

Output**Syllabus Topic : Data Types in JavaScript****2.3 Data Types in JavaScript**

→ (May 2017)

Q. Explain various data types in JavaScript.**SPPU - May 2017, 4 Marks**

- Variable is a name given to memory location where we can store some value. The value depends upon the **data type** of variable.
- In JavaScript there are number of **data types** used to store different types of values. These data types are primarily categorized as :

**Fig. C2.2: Data types in JavaScript****→ 1. JavaScript primitive data types**

In JavaScript, there are five types of primitive data types as follows :

Sr. No.	Data Type	Description
1.	String	Represents sequence of characters e.g. "Ishita"
2.	Number	Represents numeric values e.g. 101
3.	Boolean	Represents Boolean value either true or false

Sr. No.	Data Type	Description
4.	Undefined	Represents undefined value
5.	Null	Represents null means no value at all

→ 2. JavaScript non-primitive data types

Sr. No.	Data Type	Description
1.	Object	Represents instance which helps to access members
2.	Array	Represents set of same values
3.	RegExp	Represents regular expression

- JavaScript is considered as a **dynamic type language** that is there is no need to specify type of the variable. This type is dynamically decided by the JavaScript engine. While declaring a variable, "var" keyword is used on place of data type. Var means variant, that is the variable can store any type of value like numbers, strings, dates etc.

Examples

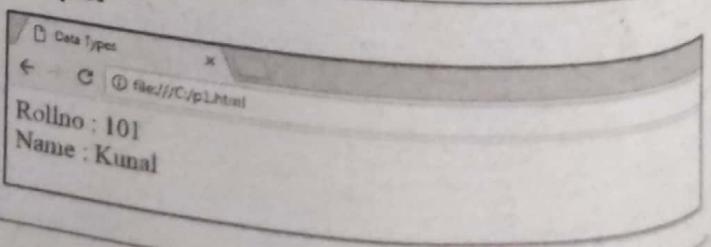
1. var rno = 101; //holding number
2. var sname="Kunal"; //holding string

Program 2.3.1

Write a program to display roll no. and name of student using "var" keyword.

Soln. :**Program to display roll no. and name of student using "var" keyword**

```
<html>
<head>
<title>
Data Types
</title>
</head>
<body>
<script language="JavaScript">
document.write("<font size=5 color=blue>");
var rno = 101;
var sname = "Kunal";
document.write("Rollno :" + rno);
document.write("<br>Name :" + sname);
document.write("</font>");
</script>
</body>
</html>
```

Output



2.4 prompt() and alert()

Q. Explain the functions prompt() and alert() with suitable example (program). **(4 Marks)**

- **prompt()** : This method is used to display an input box which can accept value from user.
- **alert()** : This method is used to display a message box.

Program 2.4.1

Write a program to accept two numbers from user and display their summation.

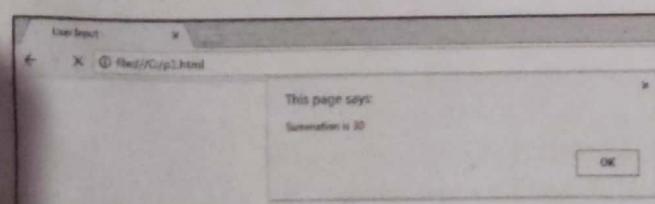
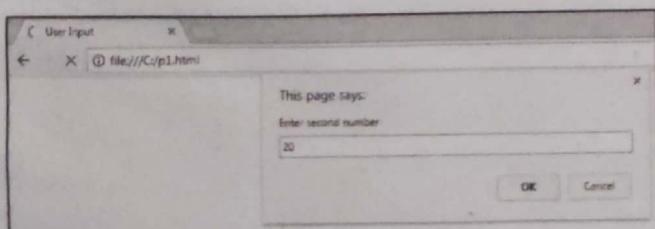
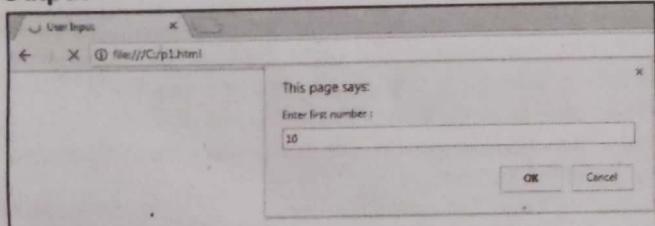
Soln. :

Program to accept two numbers from user and displaying their summation

```
<html>
<head>
<title>
User Input
</title>
</head>
<body>
<script language="JavaScript">
var n1,n2,sum;
n1 = parseInt(prompt("Enter first number : "));
n2 = parseInt(prompt("Enter second number : "));
sum = n1 + n2;
alert("Summation is "+sum);
</script>
</body>
</html>
```

The parseInt() method converts the string type of data in integer format.

Output



Syllabus Topic : Control Structures in JavaScript

2.5 Control Structures

Q. Explain control structures in JavaScript. **(8 Marks)**

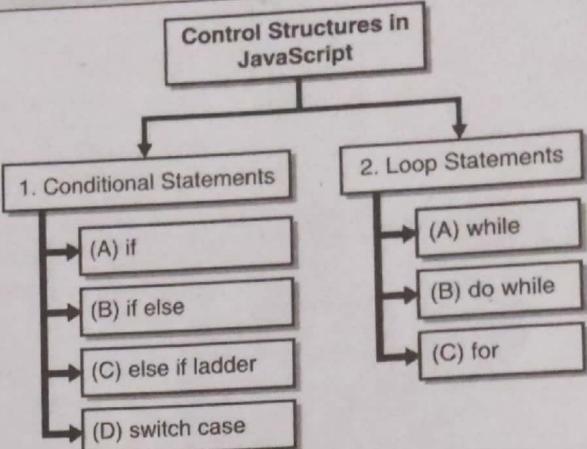


Fig. C2.3 : Control Structures in JavaScript

→ 2.5.1 Conditional Statements

JavaScript provides following types of conditional statements :

- **if** – used to specify a block of statements to be executed if the given condition is true
- **else** – used to specify a block of statements to be executed if the given condition is false
- **else if** – used to specify another condition to test if the previous condition is false
- **switch** – used to specify many alternative blocks of code to be executed

→ 2.5.1(A) The if Statement

☞ Use : The if statement used to specify a block of statements to be executed, if the given condition is true.

☞ Syntax

```
if (condition)
{
  statements;
}
```

Program 2.5.1

Write a program to accept a number and check whether it is greater than 100 or not.

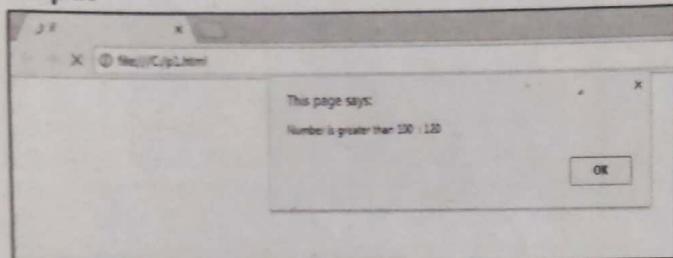
Soln. :

Program to accept a number to check whether it is greater than 100 or not

```
<html>
<head>
<title>
if
</title>
</head>
<body>
```



```
<script language="JavaScript">
var n;
n = parseInt(prompt("Enter a number : "));
if(n>100)
{
    alert("Number is greater than 100 : "+n);
}
</script>
</body>
</html>
```

Output**→ 2.5.1(B) The else Statement**

- ☞ **Use :** The **else** statement is used to specify a block of statements to be executed, if the given condition is false

☞ Syntax

```
if (condition)
{
    statements;
}
else
{
    statements;
}
```

Program 2.5.2

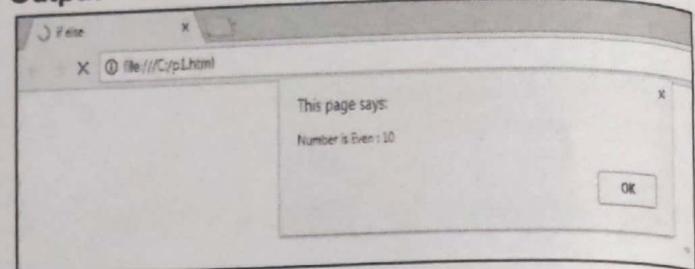
Write a program to accept a number from user and check whether it is even or odd.

Soln. :

Program to check number is even or odd

```
<html>
<head>
<title>
if else
</title>
</head>
<body>
<script language="JavaScript">
var n;
n = parseInt(prompt("Enter a number : "));
if(n%2==0)
{
    alert("Number is Even : "+n);
}
else
{
```

```
    alert("Number is Odd: "+n);
}
</script>
</body>
</html>
```

Output**→ 2.5.1(C) else if Ladder**

- ☞ **Use :** It is used to specify another condition to test if the previous condition is false

☞ Syntax

```
if (condition)
{
    statements;
}
else if(condition)
{
    statements;
}
-----
else
{
    statements;
}
```

Program 2.5.3

Write a program which accepts roll no., name and marks of 3 subjects from student. Calculate the total and average of marks and print the grade (avg >=80; then grade - A, >=60 then grade - B, >=40 then grade - C else Fail...)

(Note: The student should get the grade only if he/she is pass in all the subjects.)

Soln. :

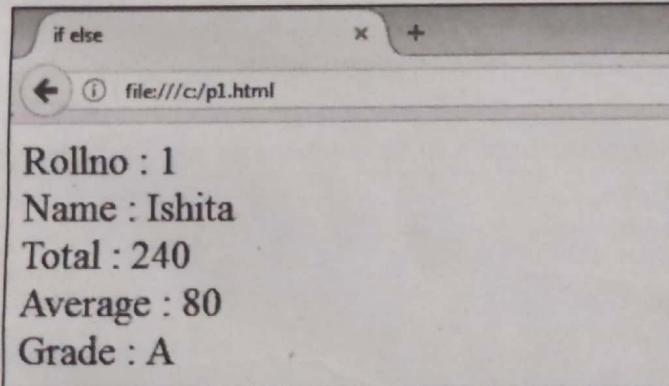
Program to calculate the average and display the grades of students

```
<html>
<head>
<title>
if else
</title>
</head>
<body>
<font size=6 color="blue">
<script language="JavaScript">
var rno,sname,Web,DAA,SYS,total,avg;
rno = prompt("Enter roll number : ");
```

```

sname = prompt("Enter name : ");
Web = parseInt(prompt("Enter marks of Web : "));
DAA = parseInt(prompt("Enter marks of DAA : "));
SYS = parseInt(prompt("Enter marks of SysPro and OS : "));
total = Web + DAA + SYS;
avg = total / 3;
document.write("Rollno : "+rno);
document.write("<br>Name : "+sname);
document.write("<br>Total : "+total);
document.write("<br>Average : "+avg);
if(Web>=40 && DAA>=40 && SYS>=40)
{
    if(avg>=80)
        document.write("<br>Grade : A");
    else if(avg>=60)
        document.write("<br>Grade : B");
    else if(avg>=40)
        document.write("<br>Grade : C");
}
else
{
    document.write("<br>Fail...");
}
</script>
</font>
</body>
</html>

```

Output

```

case constant_expression:
statements;
break;
-----
default:
statements;
}

```

Working

- The switch expression is evaluated only once.
- The value of expression is compared with the values of each constant expression.
- If there is a match, then the related statements are executed.

Program 2.5.4

Write a program to print the numbers between 1 to 5.

Soln. : Program to print the numbers between 1 to 5

```

<html>
<head>
<title>
switch case
</title>
</head>
<body>
<font size=5 color="blue">
<script language="JavaScript">
var n;
n = parseInt(prompt("Enter a number between 1 to 5 : "));
switch(n)
{

```

```

    case 1:
        document.write("One");
        break;
    case 2:
        document.write("Two");
        break;
    case 3:
        document.write("Three");
        break;
    case 4:
        document.write("Four");
        break;
    case 5:
        document.write("Five");
        break;
    default:

```

```

        document.write("Not in the range");
}
</script>
</font>
</body>
</html>

```

→ 2.5.1(D) Switch Case Statement

☛ **Use :** Switch statement is used to select one of many blocks of code to be executed.

☛ Syntax

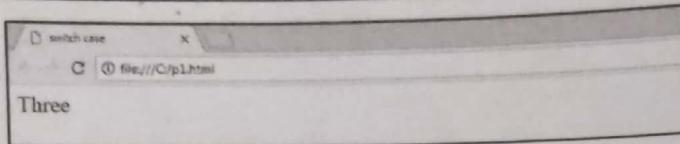
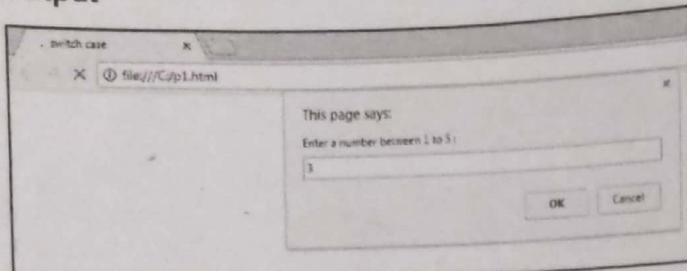
```

switch(expression)
{
    case constant_expression:
        statements;
        break;
}

```



Output



Program 2.5.5

Write a program to display the name of weekday using switch case.

Hint : The `getDay()` method returns the number of weekday in the range from 0 and 6.

(For example Sunday=0, Monday=1, Tuesday=2 ..)

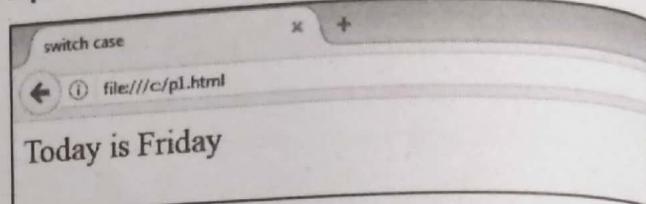
Soln. :

Program to display the name of day using switch case statement

```
<html>
<head>
<title>
switch case
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
switch (new Date().getDay())
{
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
        break;
    case 2:
        day = "Tuesday";
        break;
    case 3:
        day = "Wednesday";
        break;
    case 4:
        day = "Thursday";
        break;
    case 5:
        day = "Friday";
        break;
    case 6:
        day = "Saturday";
        break;
}
document.write("Today is "+day);
</script>
</font>
</body>
</html>
```

```
day = "Saturday";
}
document.write("Today is "+day);
</script>
</font>
</body>
</html>
```

Output



→ 2.5.2 Loop Statements

It contains while, do while and for loop.

→ 2.5.2(A) while loop

☞ Syntax

```
While (condition)
{
    Statements;
}
```

☞ Working

The while is an entry controlled loop. That means if the given condition is not satisfied then the loop statements will never get execute.

Program 2.5.6

Write a program to print 1 to 10 numbers using while loop.

Soln. :

Program to print 1 to 10 numbers using while loop

```
<html>
<head>
<title>
while loop
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var i = 1;
while(i<=10)
{
    document.write(i + "<br>");
    i = i + 1;
}
</script>
</font>
</body>
</html>
```

**Output**

```
1
2
3
4
5
6
7
8
9
10
```

Program 2.5.7

Write a program to accept a number from user and print factorial of it.

Soln. :

Program to print factorial of a number

```
<html>
<head>
<title>
while loop
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var n,n1,f;
f = 1;
n = parseInt(prompt("Enter a number :"))
n1 = n;
while(n>0)
{
f = f * n;
n--;
}
document.write("Factorial of " + n1 + " is " + f);
</script>
</font>
</body>
</html>
```

Output

```
Factorial of 5 is 120
```

→ 2.5.2(B) do while loop**☞ Syntax**

```
do
{
    Statements;
} while (condition);
```

☞ Working

The do while is an exit controlled loop. That means the first execution of loop statement is done without checking any condition. From second execution the condition get checked. Hence even if the condition is not satisfying then also the loop statements get executed once.

Program 2.5.8

Write a program to accept a number from user and check whether it is Armstrong number or not.

(Hint : Armstrong number means the summation of cubes of all the digits of the number should be exactly to the number)
E.g. $153 = (1*1*1) + (5*5*5) + (3*3*3)$

Soln. :

Program to check the Armstrong number

```
<html>
<head>
<title>
while loop
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var n,n1,r,sum;
sum = 0;
n = parseInt(prompt("Enter a number :"));
n1 = n;
do
{
r = n % 10;
sum = sum + (r*r*r);
n = Math.floor(n / 10);
}while(n>0);
if(n1==sum)
document.write("Number is Armstrong - "+n1);
else
document.write("Number is not Armstrong - "+n1);
</script>
</font>
</body>
</html>
```

Note : **Math.floor()** This method returns the integer less than the number.

**Output**

while loop
file:///c/p1.html
Number is Armstrong - 153

→ 2.5.2(C) for loop**Syntax**

```
for (initialization; test condition; iteration statement)
{
    Statements;
}
```

Explanation

In for loop the initialization, condition and increment or decrement of loop variable is done in a single statement. For loop helps to minimize the code.

Program 2.5.9

Write a program to print 1 to 10 numbers using for loop.

Soln. : Program to print 1 to 10 numbers using for loop

```
<html>
<head>
<title>
for loop
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var i;
for(i=1;i<=10;i++)
{
    document.write(i + "<br>");
}
</script>
</font>
</body>
</html>
```

Output

for loop
file:///c/p1.html
1
2
3
4
5
6
7
8
9
10

Nested for loop**Program 2.5.10**

Write a program to print the following pattern.

```
*
**
***
*****
*****
*****
```

Soln. :**Program to display the given ascending * pattern**

```
<html>
<head>
<title>
for loop
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var i,j;
for(i=1;i<=5;i++)
{
for(j=1;j<=i;j++)
{
document.write("* ");
}
document.write("<br>");
}
</script>
</font>
</body>
</html>
```

Output

for loop
file:///c/p1.html
*
**

Syllabus Topic : Arrays**2.6 Arrays**

→ (May 2015)

Q. How to create arrays in JavaScript?

SPPU - May 2015, 8 Marks



Definition

An array is a group of elements of same data type. All the elements in array have index numbers which starts from zero. These index numbers are used to access the specific array element.

Syntax

```
var students = new Array( "Kunal", "Ishita", "Shravi" ,  
"Shrey");
```

- The array is created using new keyword. The maximum size allowed for an array is 4,294,967,295.
- You can create array by simply assigning values as follows –

```
var students = ["Kunal", "Ishita", "Shreya", "Shravi"];
```

- The index numbers are used to access the array elements

students[0] is the first student – Kunal

students[1] is the second student – Ishita

2.6.1 Different Ways to Create an Array

1. Empty array without elements

```
var empty = [];
```

2. Array with 2 string elements

```
var days = ["Sunday", "Monday"];
```

3. Array with different types of elements

```
var mixed = [true, 100, "Hello"];
```

4. Two dimensional array with object literals

```
var arr = [[1,{x:10, y:20}], [2, {x:30, y:40}]];
```

5. The 3rd element is undefined

```
var colors = ["Red", "Blue", undefined];
```

6. No value in the 1st position, it is undefined

```
var hobbies = [, "Sports"];
```

Program 2.6.1

Write a simple program of array displaying student name.

Soln. :

Program of array displaying student name

```
<html>  
<head>  
<title>  
Arrays  
</title>  
</head>  
<body>  
<font size=5>  
<script language="JavaScript">  
var students = new Array( "Kunal", "Ishita", "Shravi" ,  
"Shrey");  
document.write("First student is : "+students[0]);  
document.write("<br>Second student is : "+students[1]);  
</script>
```

```
</font>  
</body>  
</html>
```

Output

2.6.2 Array Methods

Method	Description
Length	Returns length of an array (It is a property of array)
concat()	Concatenates (merge) multiple arrays.
every()	If all the elements in the array satisfy the given condition of testing function, it returns true.
filter()	Returns array elements which satisfy the filter criteria of given function.
forEach()	Calls a specific function for all the elements in the array.
indexOf()	Return the index of first occurrence of given element. Returns -1 if element not found.
join()	Joins all the elements of an array and converts into a string.
lastIndexOf()	Return the index of last occurrence of given element. It returns 1, if the specified element not found.
pop()	Removes and returns the last element of an array.
push()	Adds one or more elements at the end of an array.
reverse()	Reverses the sequence of the elements of an array.
shift()	Removes and returns the last element of an array.
slice()	Extracts a portion of an array and returns it.
sort()	Sorts the elements of an array
splice()	Adds and/or removes elements from an array.
toString()	Returns a string representation of the given array
unshift()	Adds one or more elements at the beginning of an array and returns it.

Program 2.6.2

Write a program to display the length of array and index of array element.

**Soln. :****Program to display the length of array and index of array element**

```
<html>
<head>
<title>
Arrays
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var students = ["Kunal", "Ishita", "Shravi", "Kunal", "Shrey"];
document.write("Array length : "+students.length);
document.write("<br>First index of Kunal : "+students.indexOf("Kunal"));
document.write("<br>Last index of Kunal : "+students.lastIndexOf("Kunal"));
</script>
</font>
</body>
</html>
```

Output

```
Array length : 5
First index of Kunal : 0
Last index of Kunal : 3
```

2.6.3 Iterating Through an Array

Use : Iterating or traversing of an array means visiting each element at least once. For this purpose we can use the loops.

Program 2.6.3

Write a simple program of array iteration.

Soln. : Program of array iteration

```
<html>
<head>
<title>
Array Iteration
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var cars = [];
cars[0] = "Ford";
cars[1] = "BMW";
```

```
cars[2] = "Enova";
cars[3] = "Honda City";
for (var i = 0; i < cars.length; i++)
{
    document.write(cars[i] + "<br>");
}
</script>
</font>
</body>
</html>
```

Output

```
Ford
BMW
Enova
Honda City
```

Program 2.6.4

Write a program to perform various methods like adding a new element, sorting, reversing, removal of last element etc.

Soln. :**Program to perform various methods on array**

```
<html>
<head>
<title>
Array Iteration
</title>
</head>
<body>
<font size=4>
<script language="JavaScript">
var cars = [];
cars[0] = "Ford";
cars[1] = "BMW";
cars[2] = "Enova";
cars[3] = "Honda City";
for (var i = 0; i < cars.length; i++)
{
    document.write(cars[i] + "<br>");
}
cars.pop();
document.write("<br>After removal of last element<br>");
for (var i = 0; i < cars.length; i++)
{
    document.write(cars[i] + "<br>");
}
cars.push("i10");
document.write("<br>After adding new element<br>");
for (var i = 0; i < cars.length; i++)
```

```
{
document.write(cars[i] + "<br>");
}
cars.sort();
document.write("<br>After Sorting<br>");
for (var i = 0; i < cars.length; i++)
{
document.write(cars[i] + "<br>");
}
cars.reverse();
document.write("<br>After Reversing<br>");
for (var i = 0; i < cars.length; i++)
{
document.write(cars[i] + "<br>");
}
</script>
</font>
</body>
</html>
```

Output

File:///c/p1.html

Ford
BMW
Enova
Honda City

After removal of last element

Ford
BMW
Enova

After adding new element

Ford
BMW
Enova
i10

After Sorting

BMW
Enova
Ford
i10

After Reversing

i10
Ford
Enova
BMW

2.6.4 Deleting Element from an Array

Q. Explain how to delete an element from array in JavaScript. (2 Marks)

- Use : The delete operator is used to remove an element from an array. Deleting an element from an array does not affect the length property and the array becomes sparse. Also the elements which are at the right of the deleted element do not get shifted to left to fill in the gap.

Example

```
var days = ["Sunday", "Monday", "Tuesday", "Wednesday"];
delete days[1]; // delete the element "Monday"
```

2.6.5 Array Method : Splice()

Q. Explain array method splice() with suitable examples in JavaScript. (2 Marks)

splice() method

- Use : The splice() method is used to insert new, delete existing, and replace existing elements by new elements in the array.
- It moves the elements to higher or lower positions as per the requirement to avoid any gap.
- The first argument of splice() indicates the starting position and second argument indicates the number of elements to delete.

Examples

```
var letters = ["a", "b", "c", "d", "e", "f", "g"];
alert(letters.splice(5, 2)); // f, g (deleted elements)
alert(letters); // a, b, c, d, e
alert(letters.splice(2, 1)); // c (the deleted element)
alert(letters); // a, b, d, e
```

- The third argument in the splice method is used to replace one or more elements with others.

```
var letters = ["a", "b", "c", "d"];
alert(letters.splice(1, 2, "e", "f", "g")); // b, c (deleted ones)
alert(letters); // a, e, f, g, d
```

- In this example, the splice starts at position 1 and removes two elements b and c. Then it fills the gap with the three elements e, f and g.

Syllabus Topic : Functions and Scopes in JavaScript**2.7 Functions and Scopes****2.7.1 Functions**

Q. Explain user defined function in JavaScript with suitable example. (4 Marks)

- Definition : A function is a block of statements related to such a task which we want to execute repeatedly in our program.
- Function helps to avoid repetition of code and write modular codes. A program can be divided into small and manageable modules called as functions.



- Just like the other programming languages such as C, C++ and Java, the function concept with all its features is supported by JavaScript.
- In previous sections we have seen the built in functions like alert() and write()
- We can create our own functions known as user defined functions.

2.7.1(A) Function Definition

Before using a function, we have to define it. Usually in JavaScript, a function is defined by using the **function** keyword, followed by name of function, a parameters list [optional], and a statement block inside curly braces.

Syntax

```
<script type="text/javascript">
    function function_name([parameter-list])
    {
        Statements;
    }
</script>
```

Program 2.7.1

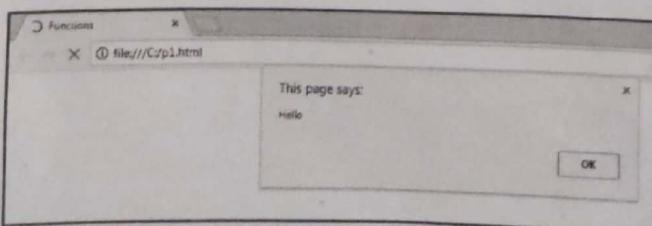
Write a simple function calling program.

Soln. :

Program to display hello message using function call

```
<html>
<head>
<title>
Functions
</title>
</head>
<body>
<script language="JavaScript">
function callMe() → Function Definition
{
    alert("Hello");
}
callMe(); → Function Call
</script>
</body>
</html>
```

Output



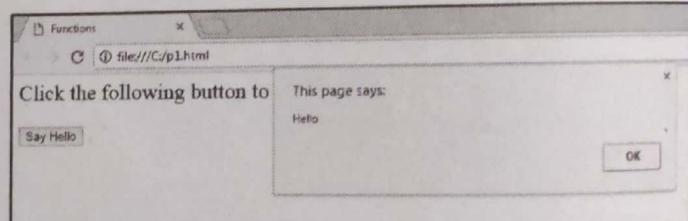
Program 2.7.2

Write a program to call a function on click event of button.

Soln. :
Calling a function on click event of a button

```
<html>
<head>
<title>
Functions
</title>
<script language="JavaScript">
function callMe()
{
    alert("Hello");
}
</script>
</head>
<body>
<font size=5>
<p>Click the following button to call the function</p>
<p><input type="button" onClick="callMe();" value="Say Hello"></p>
</font>
</body>
</html>
```

Output



2.7.1(B) Parameterized Function

While defining a function, we can declare variables in the header statement of the function. These variables are known as parameters or formal arguments. When this function gets called, we can pass values for these variables. These values are known as arguments or actual arguments.

Program 2.7.3

Write a program to display the summation of two values using parameterized function.

Soln. :

Program to display the summation using parameterized function

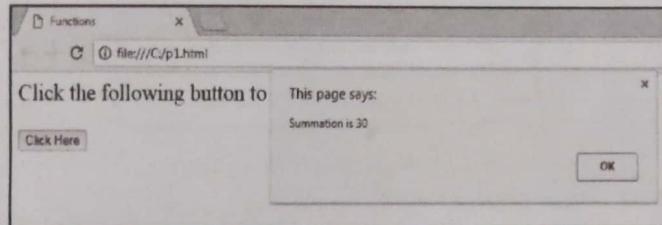
```
<html>
<head>
<title>
Functions
</title>
<script language="JavaScript">
function add(a,b)
```



```
{
var sum = a + b;
alert("Summation is "+sum);
}
</script>
</head>
<body>
<font size=5>
<p>Click the following button to call the function</p>

<form>
    <input type="button" onClick="add(10,20);"
value="Click Here">
</form>
</font>
</body>
</html>
```

Output



2.7.1(C) The return Statement

- In JavaScript function we can have **return** statement which is optional. This helps to return a value from a function. This statement is written at the last in a function.
- The returned value goes to the location where the function is called.

Program 2.7.4

Write a program which will accept two numbers as arguments and returns their summation.

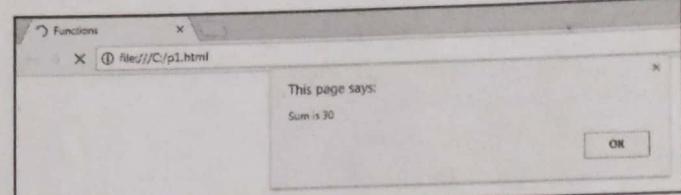
Soln. :

Program to display the summation accepting two numbers as arguments

```
<html>
<head>
<title>
Functions
</title>
<script language="JavaScript">
function add(a,b)
{
var sum = a + b;
return(sum);
}
</script>
</head>
```

```
<body>
<script language="JavaScript">
var r = add(10,20);
alert("Sum is "+r);
</script>
</body>
</html>
```

Output



2.7.2 Scopes

Q. Explain Scope in JavaScript.

(4 Marks)

In JavaScript, scope is usually set to the variables, objects, and functions. The scope decides the accessibility area for these elements. In other words, scope decides the visibility of variables and other elements in areas of the code.

☞ Types of scopes

JavaScript supports two types of scopes :

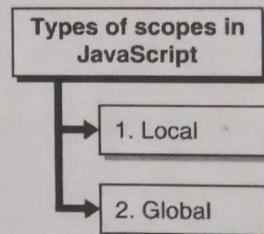


Fig. C2.4 : Types of scope

→ 1. Local Scope

- Variables declared inside a JavaScript function are **LOCAL** to the function.
- Local variables have the local scope that means they are accessible inside the function only.

☞ Example

```
function student()
{
var sname = "Kunal";
// Code here can use the sname
}
```

- As local variables are only accessible inside the functions, variables with similar names can be used in different functions.
- The local variables are created at the time when a function starts, and destroyed when the function is completed.



→ 2. Global Scope

- When variables are created outside a function, they become **GLOBAL**.
- These variables have **global scope**, that means all the scripts and functions inside the web page can access these elements.

Example

```
var sname = "Kunal";
// Code here can use the sname
function student()
{
    // Code here can use the sname
}
// Code here can use the sname
```

Automatic Global

If value is assigned to a variable which has not been declared, it will be automatically considered as a **GLOBAL** variable.

```
// Code here can use the sname
function student()
{
    sname = "Kunal";
    // Code here can use the sname
}
// Code here can use the sname
```

→ The Lifetime of JavaScript Variables

- The lifetime of a JavaScript variable is considered to be starts when it is declared.
- When a function is completed, the local variables get automatically destroyed.
- In a web browser where we access the websites, global variables get destroyed when the browser window is closed.

Syllabus Topic : Objects in JS

2.8 Objects in JS

- **Definition :** An object is nothing but an entity having its own state and behavior (properties and methods).
- **For example :** A flower is an object having properties like color, fragrance etc. Other examples of objects are car, pen, bike, chair, glass, keyboard, monitor etc.
- JavaScript is an object-oriented language. Everything in JavaScript is considered as an object.

Examples of objects

- Following are some of the examples of objects in JavaScript.
 - Booleans (when defined with the new keyword)
 - Numbers (when defined with the new keyword)
 - Strings (when defined with the new keyword)
 - Dates

(v) Regular expressions

(vi) Arrays

(vii) Functions

- We can create our own user defined objects in JavaScript.

- JavaScript is basically a template based scripting language not class based. Hence, we directly create the object without class.

2.8.1 Creating Objects in JavaScript

→ (May 2016)

Q. Explain how object is created and modified in JavaScript.

SPPU - May 2016, 8 Marks

Ways of creating objects in JavaScript

- 1. By object literal
- 2. By creating instance of Object directly
- 3. By using an object constructor

Fig. C2.5 : Ways of creating objects

→ 1. JavaScript Object by Object Literal

→ Syntax

Following is the syntax of creating object using object literal :

```
object = {property1:value1,property2:value2,...,propertyN:valueN}
```

As we can observe, the property and value is separated by the separator : (colon).

Program 2.8.1

Write a sample program of creating object using object literal.

Soln. :

Program of creating object using object literal

```
<html>
<head>
<title>
Objects
</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
employee = {id:101,name:"Pranav Dilip Dev",salary:40000}
document.write(employee.id + "-" + employee.name + ","
+ employee.salary);
</script>
</font>
</body>
</html>
```

**Output**

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>101-Pranav Dilip Dev-40000</font>
</body>
</html>
```

→ 2. By creating instance of Object**Syntax**

Following is the syntax of creating instance of object :

```
var objectname=new Object();
```

Here, **new keyword** is used to create object.

Program 2.8.2

Write a simple program of creating the instance of object.

Soln. :**Program of creating the instance of object**

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
var employee=new Object();
employee.id=101;
employee.name="Pranav Dilip Dev";
employee.salary=40000;
document.write(employee.id+" - "+employee.name+" - "
+employee.salary);
</script>
</font>
</body>
</html>
```

Output

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>101-Pranav Dilip Dev-40000</font>
</body>
</html>
```

→ 3. By using an Object constructor

- Here, we have to create parameterized function. "**this**" keyword is used to assign each argument value in the current object.
- The "**this**" keyword refers to the current object.

Program 2.8.3

Write a program with the use of this keyword.

Soln. :**Program using "this" keyword**

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
function student(id,sname,marks)
{
    this.id=id;
    this.sname=sname;
    this.marks=marks;
}
s = new student(11,"Ishita",92);
document.write(s.id+" - "+s.sname+" - "+s.marks);
</script>
</font>
</body>
</html>
```

Output

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>11 - Ishita - 92</font>
</body>
</html>
```

2.8.2 Defining Method in JavaScript Object**Q. How to define method in JavaScript Object ?**

(4 Marks)

It is possible to define method in JavaScript object. For this purpose we have to add property in the method with same name as of the method name.

Program 2.8.4

Write a program to define the method in JavaScript.

Soln. :**Program to define the method in JavaScript**

```
<html>
<head>
<title>Objects</title>
</head>
<body>
<font size=5>
<script language="JavaScript">
function employee(id,sname,salary)
{
```



```

this.id=id;
this.sname=sname;
this.salary=salary;
this.changeSalary=changeSalary;
function changeSalary(otherSalary)
{
    this.salary=otherSalary;
}
}

e = new employee(101,"Kalpesh",50000);
document.write(e.id + " " + e.sname + " " + e.salary);
e.changeSalary(55000);
document.write("<br>" + e.id + " " + e.sname + " " + e.salary);
</script>
</font>
</body>
</html>

```

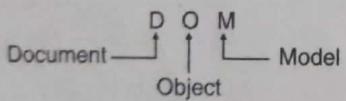
Output

101 Kalpesh 50000
101 Kalpesh 55000

Syllabus Topic : DOM – DOM Levels, DOM Objects and their Properties and Methods, Manipulating DOM

2.9 DOM

Q. Explain DOM in JavaScript. **(8 Marks)**



- **DOM stands for Document Object Model.**
- The entire html document is represented by the **document object**.
- The html document becomes document object when it is loaded in the browser. The **root element** represents the html document. The document object has properties and methods. The document object helps to add content dynamically in the web page.
- Any element of HTML page can be accessed by using the document object.
- According to W3C(World Wide Web Consortium) - "The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

Hierarchy of objects in web document

Document Object Model (DOM) is the method by which the content of document is accessed and

modified. In a web document, the organization of objects is implemented in a hierarchical structure.

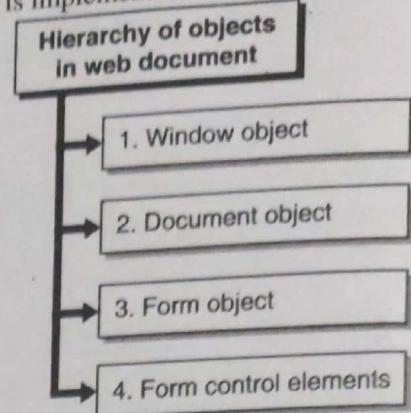


Fig. C2.6 : Hierarchy of objects in web document

→ 1. Window object

It resides at top of the hierarchy. It is the topmost element of the object hierarchy.

→ 2. Document object

All the HTML documents which get loaded into browser are considered as document objects. The contents of the page are stored in the document object.

→ 3. Form object

Everything which is contained in the opening <form> and closing</form> tag sets the form object.

→ 4. Form control elements

The form object has all the elements which are defined for the objects like text fields, buttons, checkboxes, select box, radio buttons etc.

2.9.1 DOM Levels

The DOM provides all the **features** to JavaScript to create dynamic HTML :

- Changes can be made in all HTML elements.
- Changes can be made in attributes of HTML elements.
- Changes can be made in all CSS styles in the page.
- Existing HTML elements and attributes can be deleted.
- New HTML elements and attributes can be added.
- Response can be given to HTML events.
- New HTML events can be created in HTML page.

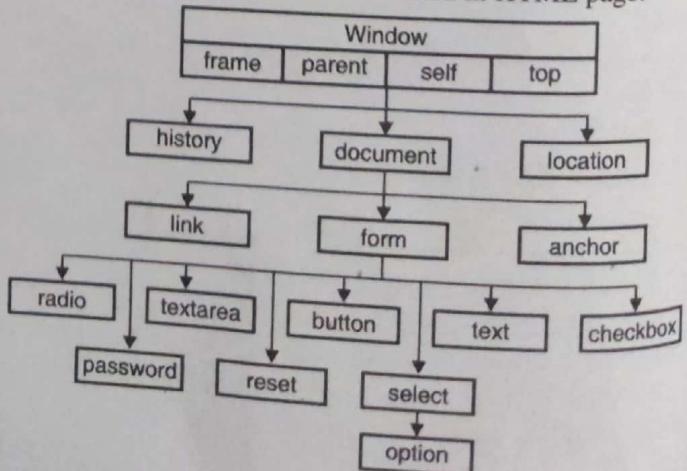


Fig. 2.9.1 : DOM levels



2.9.2 Properties and Methods of Document Object

The contents of document can be accessed and modified with the help of methods.

Sr. No.	Method / Properties	Description
1.	Value	Returns value of specified text field
2.	write("string")	Writes the specified string in the document.
3.	writeln("string")	Writes the specified string in the document with newline character at the end.
4.	getElementById()	Returns the element having the specified id value.
5.	getElementsByName()	Returns all the elements having the specified name value.
6.	getElementsByTagName()	Returns all the elements having the specified tag name.
7.	getElementsByClassName()	Returns all the elements having the specified class name.

1. Accessing field value by document object

Program 2.9.1

Write a program to accept value from user with the help of text field and access it using the document.form1.name.value.

Soln. :

Program of document object which accesses value from text field

```
<html>
<head>
<title>
DOM
</title>
<script language="JavaScript">
function callMe()
{
var sname=document.form1.txtname.value;
alert("Welcome: "+sname);
}
</script>
</head>
<body>
<font size=5>
```

```
<form name="form1">
Enter Your Name:<input type="text" name="txtname"/>
<input type="button" onClick="callMe();" value="Click Here"/>
</form>
</font>
</body>
</html>
```

Explanation

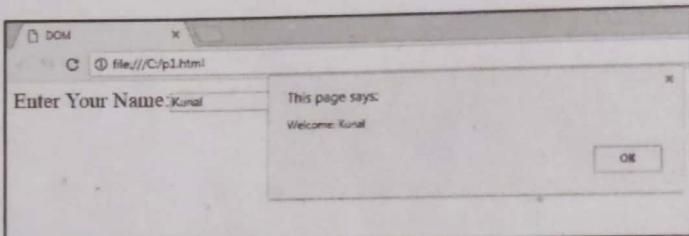
Here, **document** represents the html document. It is the root element.

form1 - name of the form.

name - attribute name of the input text.

value - property, that returns the value of the input text.

Output



2. write("String") method

This method is used to write output stream.

The write() method is used to writes HTML expressions or JavaScript code to a document.

Program 2.9.2

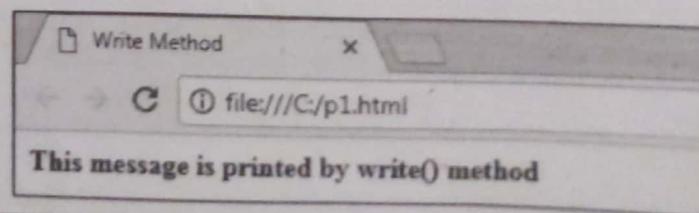
Write a program to display the message using write("string") method.

Soln. :

Displaying the message using write("string") method

```
<html>
<head>
<title>
Write Method
</title>
</head>
<body>
<script language="JavaScript">
document.write("<b>This message is printed by write() method</b>");
</script>
</body>
</html>
```

Output





3. writeln("String") method

This method is used to write output stream. writeln() add a new line after each statement.

Program 2.9.3

Write a program to demonstrate the use of writeln("String") method.

Soln. :

Program to demonstrate the use of writeln("String") method

```
<!DOCTYPE html>
<html>
<body>
<font size=5>
<p>Note : write() does not add a new line after each statement:</p>
<pre>
<script>
document.write("First Line");
document.write(.. Also on the First Line");
</script>
</pre>
<p>Note : writeln() add a new line after each statement:</p>
<pre>
<script>
document.writeln("First Line");
document.writeln("Second Line");
</script>
</pre>
</font>
</body>
</html>
```

Output

Note : write() does not add a new line after each statement:
First Line.. Also on the First Line
Note that writeln() add a new line after each statement:
First Line
Second Line

4. document.getElementById() method

This method returns the element of specified id.

Syntax

```
document.getElementById("id")
```

In the above script, we have used document.form1.name.value to access the value of text field. For the same purpose now we will use Document.getElementById() method. Here we have to define id for the input text field.

Program 2.9.4

Display the cube of number using Document.getElementById() method.

Soln. :

Program to display cube of number using Document.getElementById() method

```
<html>
<head>
<title>
DOM
</title>
<script language="JavaScript">
function cube()
{
var no = document.getElementById("num").value;
alert(no*no*no);
}
</script>
</head>
<body>
<font size=5>
<form>
Enter No:<input type="text" id="num" name="num"/><br/> <br/>
<input type="button" value="Display Cube" onClick="cube()"/>
</form>
</font>
</body>
</html>
```

Output

Enter No:15
This page says:
125

5. document.getElementsByName() method

This method returns all the elements of given name.

Syntax

```
document.getElementsByName("name")
```

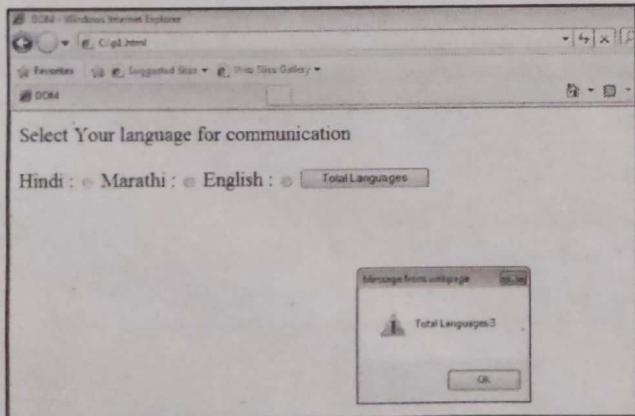
In this example, we will count total number of languages. Here, we are using getElementsByName() method to get all the languages.

Program 2.9.5

Write a program to display number of languages available using radio buttons. Display the count of languages on click of a button.

**Soln. :****Program to display number of languages and their count on click event of button**

```
<html>
<head>
<title>
DOM
</title>
<script type="text/javascript">
function languages()
{
var lan = document.getElementsByName("lang");
alert("Total Languages:" + lan.length);
}
</script>
</head>
<body>
<font size=5>
<form>
Select Your language for communication <br><br>
Hindi : <input type="radio" name="lang" value="Hindi">
Marathi : <input type="radio" name="lang" value="Marathi">
English : <input type="radio" name="lang" value="English">
<input type="button" onClick="languages()" value="Total Languages">
</form>
</font>
</body>
</html>
```

Output**6. document.getElementsByTagName() method**

All the elements of specified tag are returned by this method.

Syntax

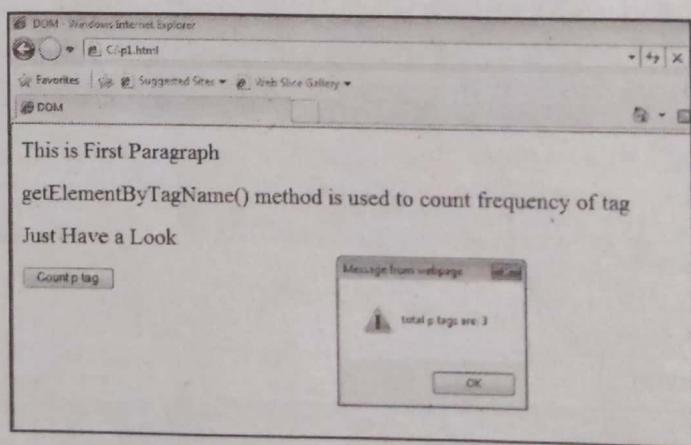
```
document.getElementsByTagName("tag_name")
```

Program 2.9.6

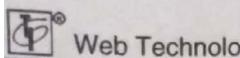
Write a program to count total number of tags used by document.getElementsByTagName() method.

Soln. :**Program to count total number of tags used by document.getElementsByTagName() method**

```
<html>
<head>
<title>
DOM
</title>
<script type="text/javascript">
function cntp()
{
var cnt = document.getElementsByTagName("p");
alert("total p tags are: " + cnt.length);
}
</script>
</head>
<body>
<font size=5>
<form>
<p>This is First Paragraph</p>
<p>getElementByTagName() method is used to count frequency of tag </p>
<p>Just Have a Look</p>
<button onclick="cntp();">Count p tag</button>
</form>
</font>
</body>
</html>
```

Output**7. document.getElementsByClassName()**

- This method returns a set of all the elements of the document with the given class name, as a NodeList object.
- The NodeList object issued to represent a set of nodes. The nodes are accessed using index numbers which starts from 0.

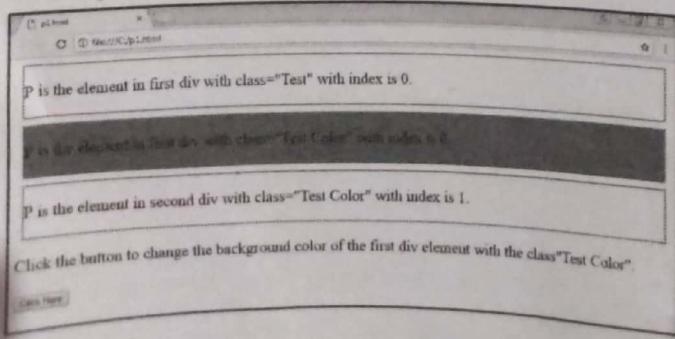
**Program 2.9.7**

Write a program to demonstrate the use of `getElementsByClassName()` method.

Soln. :

Program to demonstrate the use of `getElementsByClassName()` method

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 2px solid gray;
    margin: 10px;
}
</style>
</head>
<body>
<font size=5>
<div class="Test">
    <p>P is the element in first div with class="Test" with
    index is 0.</p>
</div>
<div class="Test Color">
    <p>P is the element in first div with class="Test Color"
    with index is 0.</p>
</div>
<div class="Test Color">
    <p>P is the element in second div with class="Test Color"
    with index is 1.</p>
</div>
<p>Click the button to change the background color of the
first div element with the class "Test Color".</p>
<button onclick="changeColor()">Click Here</button>
<script language="JavaScript">
function changeColor() {
    var x = document.getElementsByClassName("Test Color");
    x[0].style.backgroundColor = "gray";
}
</script>
</font>
</body>
</html>
```

Output**2.9.3 JavaScript - innerHTML**

Q. Explain `innerHTML` in JavaScript with the help of example (program). (2 Marks)

The dynamic html contents can be written in document with the help of **innerHTML** property. It is generally used in the html documents to generate dynamic contents like registration form, comment form, links etc.

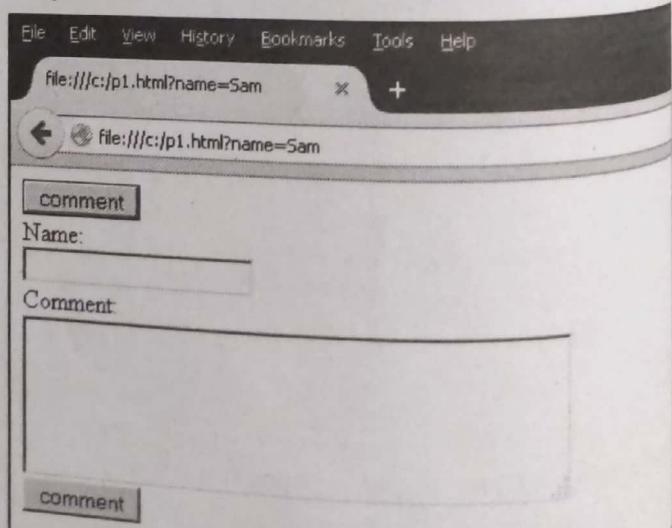
Program 2.9.8

Write a program to generate comment form using `innerHTML` property.

Soln. :

Program of comment form using `innerHTML` property

```
<html>
<body>
<script type="text/javascript" >
function disp() {
var data="Name:<br><input type='text'
name='name'><br>Comment:<br><textarea rows='5'
cols='40'></textarea><br><input type='submit'
value='comment'>";
document.getElementById('myloc').innerHTML=data;
}
</script>
<form name="myForm">
<input type="button" value="comment" onclick="disp()">
<div id="myloc"></div>
</form>
</body>
</html>
```

Output**2.9.4 Manipulating DOM**

Q. Write a note on DOM manipulation. (4 Marks)

Using JavaScript we can add or remove nodes (HTML elements) in the document.



1. Creating New HTML Elements (Nodes)

Initially we have to create the element (element node) which can be then appended to the existing element.

Program 2.9.9

Write a program to append the new element to an existing element node.

Soln. :

Program to append the new element to an existing element node

```
<html>
<head>
<title>
DOM Manipulation
</title>
<div id="div1">
<p id="p1">It's not that I'm so smart, it's just that I stay with
problems longer.
</p>
<p id="p2">Albert Einstein : Theoretical Physicist,
Philosopher, Nobel Prize Winner</p>
</div>
<script>
var prg = document.createElement("p");
var nd = document.createTextNode("A new Paragraph.");
prg.appendChild(nd);
var element = document.getElementById("div1");
element.appendChild(prg);
</script>
</body>
</html>
```

Explanation : Now see the code

```
var prg = document.createElement("p");
```

A new `<p>` element is created

```
var nd = document.createTextNode("A new paragraph.");
```

A text node is created and text is added to element `<p>`.

```
prg.appendChild(nd);
```

The text node is appended to the element `<p>`

```
var element = document.getElementById("div1");
```

This code finds an existing element:

```
element.appendChild(prg);
```

This code appends the new element to an existing element.

Output

```
DOM Manipulation
file:///c:/p1.html

It's not that I'm so smart, it's just that I stay with problems longer.
Albert Einstein : Theoretical Physicist, Philosopher, Nobel Prize Winner
A new Paragraph.
```

2. Creating new HTML Elements - insertBefore()

- In the previous program, we have seen the `appendChild()` method which appends the element at the end in parent.
- the `insertBefore()` method is used to add the new element at the beginning.

Program 2.9.10

Write a program to add new element at the beginning using `insertBefore()` method.

Soln. :

Program to add new element at the beginning using `insertBefore()` method

```
<html>
<head>
<title>
DOM Manipulation
</title>
<div id="div1">
<p id="p1">It's not that I'm so smart, it's just that I stay with
problems longer.
</p>
<p id="p2">Albert Einstein : Theoretical Physicist,
Philosopher, Nobel Prize Winner</p>
</div>
<script>
var prg = document.createElement("p");
var nd = document.createTextNode("A new Paragraph.");
prg.appendChild(nd);
var element = document.getElementById("div1");
var child = document.getElementById("p1");
element.insertBefore(prg, child);
</script>
</body>
</html>
```

Output

```
DOM Manipulation
file:///c:/p1.html

A new Paragraph.

It's not that I'm so smart, it's just that I stay with problems longer.
Albert Einstein : Theoretical Physicist, Philosopher, Nobel Prize Winner
```

3. Removing Existing HTML Elements

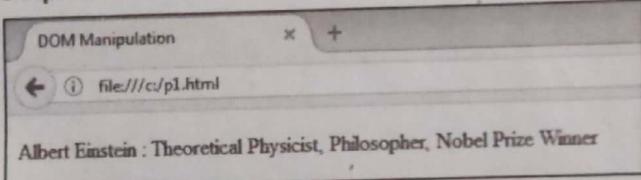
To remove an HTML element, we should have the information about the parent element. The `removeChild()` method is used to remove the HTML element.

Program 2.9.11

Write a program to remove an existing element using `removeChild()` method.

**Soln. :****Program to remove an existing element using
removeChild() method**

```
<html>
<head>
<title>
DOM Manipulation
</title>
<div id="div1">
<p id="p1">It's not that I'm so smart, it's just that I stay with
problems longer.
</p>
<p id="p2">Albert Einstein : Theoretical Physicist,
Philosopher, Nobel Prize Winner</p>
</div>
<script>
var parent = document.getElementById("div1");
var child = document.getElementById("p1");
parent.removeChild(child);
</script>
</body>
</html>
```

Output**4. Replacing HTML Elements**

It is also possible to replace an existing element by new one. The replaceChild() method is used for this purpose.

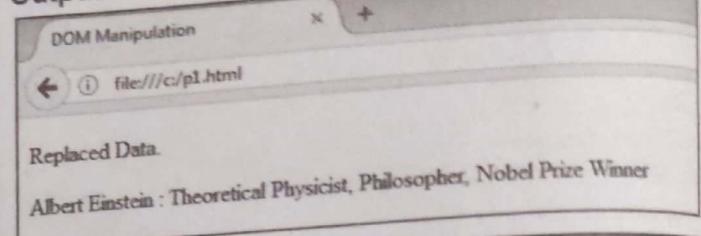
Program 2.9.12

Write a program to replace an existing element by new element using replaceChild() method.

Soln. :**Program to replace an existing element by new element using replaceChild() method**

```
<html>
<head>
<title>
DOM Manipulation
</title>
<div id="div1">
<p id="p1">It's not that I'm so smart, it's just that I stay with
problems longer.
</p>
<p id="p2">Albert Einstein : Theoretical Physicist,
Philosopher, Nobel Prize Winner</p>
</div>
<script>
var prg = document.createElement("p");
var nd = document.createTextNode("Replaced Data.");
prg.appendChild(nd);
var parent = document.getElementById("div1");
var child = document.getElementById("p1");
parent.replaceChild(prg, child);
</script>
</body>
</html>
```

```
prg.appendChild(nd);
var parent = document.getElementById("div1");
var child = document.getElementById("p1");
parent.replaceChild(prg, child);
</script>
</body>
</html>
```

Output

Syllabus Topic : jQuery - Introduction to jQuery, Loading jQuery, Selecting elements, Changing styles, Creating elements, Appending elements, Removing elements, Handling events

2.10 Introduction to jQuery

- jQuery is basically a JavaScript library which is cross-platform and used to support the client side scripting of HTML. jQuery is open source and free application available on internet.
- jQuery provides various functionalities like navigation in web site, selecting DOM elements, managing events, creating different types of animations etc. jQuery provides a modular approach which simplifies the creation of dynamic content.
- jQuery was initially released by John Resig at BarCamp NYC in January 2006. Now a day jQuery is maintained by Timmy Willison and his team.

2.10.1 Features of jQuery

Q. Write features of jQuery.

(4 Marks)

Features of jQuery

1. DOM manipulation
2. Event handling
3. AJAX Support
4. Animations
5. Lightweight
6. Cross Browser Support
7. Latest Technology

Fig. C2.7 : Features of jQuery

→ **1. DOM manipulation**

The DOM manipulation which includes selection of elements, traversing and modification in contents is easy with jQuery. jQuery make it easy with the help of cross-browser open source selector engine called Sizzle.

→ **2. Event handling**

Handling a wide variety of events like link click, mouse over etc. are effectively handled in the jQuery using event handlers.

→ **3. AJAX Support**

AJAX technology can be used with jQuery to develop responsive websites having a rich set of advanced features.

→ **4. Animations**

Number of built-in animation effects are provided by the jQuery.

→ **5. Lightweight**

The jQuery is a lightweight library which takes minimum time to load. It is about 19KB in size only.

→ **6. Cross Browser Support**

jQuery is supported by various latest browsers like Chrome, Mozilla, IE, Opera, Safari etc.

→ **7. Latest Technology**

The advanced technologies like CSS3 selectors and basic XPath syntax are supported by jQuery.

2.10.2 Loading jQuery

- **Local Installation :** The jQuery library can be loaded on our local machine. This library can be used in the HTML document.
- **CDN Based Version :** It is also possible to add jQuery library in the HTML document directly from Content Delivery Network (CDN).

A. Local Installation

- URL to download jQuery : <https://jquery.com/download/>
- Now save this downloaded **jQuery-3.2.1.min.js** file in a directory of your website, e.g. /jQuery.
- Now you can include *jQuery* library in your HTML file as follows.
- As the webpage and *jQuery* library are in the same directory, there is no need to specify whole path in "src" attribute.

Program 2.10.1

Write a simple program displaying hello world message using jQuery library

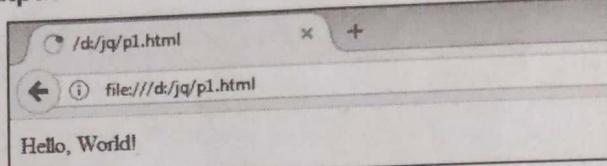
Soln. :

Program displaying hello world message using jQuery library

```
<html>
<head>
<title>jQuery</title>
```

```
<script type = "text/javascript" src = "jQuery-
3.2.1.min.js"></script>
<script type = "text/javascript">
$(document).ready(function()
{
    document.write("Hello, World!");
});
</script>
</head>
<body>
</body>
</html>
```

Output



B. CDN Based Version

- It is also possible to include jQuery library in the HTML document directly from Content Delivery Network (CDN). Google and Microsoft are the providers for the latest version.
- Here we have to set the value of *src* attribute as follows
src : <https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js>

2.10.3 Selecting Elements

Q. Explain different selectors in jQuery. (4 Marks)

- jQuery provides selectors to select and make changes in HTML elements.
- jQuery selectors use the name, id, classes, types, attributes, values of attributes etc to select HTML elements.
- In jQuery, the selectors start with the symbol \$ (dollar sign) and parentheses: \$()
- Now we will discuss different type of selectors available in jQuery.

☞ Types of selectors

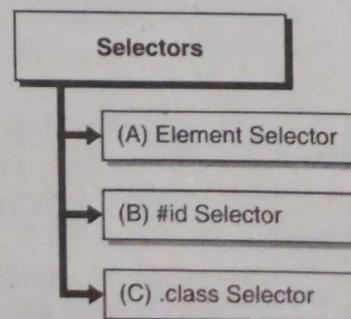


Fig. C2.8 : Different types of selectors

→ **(A) The Element Selector**

The jQuery element selector selects the HTML elements depending upon the element name.



<p> elements can be selected as follows :

`$(“p”)`

Program 2.10.2

Write a program to demonstrate the use of Element Selector.

Soln. :

Program to demonstrate the use of Element Selector

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery-3.2.1.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script>
</head>
<body>
<h1>jQuery</h1>
<p>Example of jQuery Selector</p>
<p>Try it</p>
<button>Click Here to hide the paragraphs</button>
</body>
</html>
```

Output

The screenshot shows a browser window with the URL `/d/jq/p1.html`. The page title is "jQuery". The content area contains the text "Example of jQuery Selector" and "Try it". Below this is a button with the text "Click Here to hide the paragraphs".

The screenshot shows the same browser window after interacting with the button. The button now displays the text "Click Here".

→ (B) The #id Selector

- The jQuery #id selector selects the HTML id attribute of the HTML tag to find the specific element. All the elements should have unique id.
- To search an element with a given id, hash character is written which is followed by the HTML element id.

`$("#id1")`

Program 2.10.3

Write a program to demonstrate the use of #id Selector.

Soln. :

Program to demonstrate the use of #id Selector

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery-3.2.1.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#id1").hide();
    });
});
</script>
</head>
<body>
<h1>jQuery</h1>
<p>Example of jQuery selector</p>
<p id="id1">Try it</p>
<button>Click Here</button>
</body>
</html>
```

Output

The screenshot shows a browser window with the URL `/d/jq/p1.html`. The page title is "jQuery". The content area contains the text "Example of jQuery selector" and "Try it". Below this is a button with the text "Click Here".

→ (C) The .class Selector

- The jQuery class selector finds elements with a specific class.
- To search an element with a specific class, period character is written which is followed by the name of the class.

`$(".myclass")`

Program 2.10.4

Write a program to demonstrate the use of .class Selector.

Soln. :

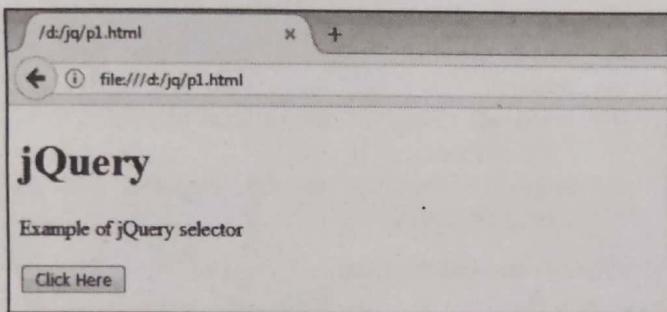
Program to demonstrate the use of .class Selector

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery-3.2.1.min.js"></script>
<script>
```

```

$(document).ready(function(){
    $("button").click(function(){
        $(".myclass").hide();
    });
});
</script>
</head>
<body>
<h1 class="myclass">jQuery</h1>
<p class="myclass">Example of jQuery selector</p>
<button>Click Here</button>
</body>
</html>

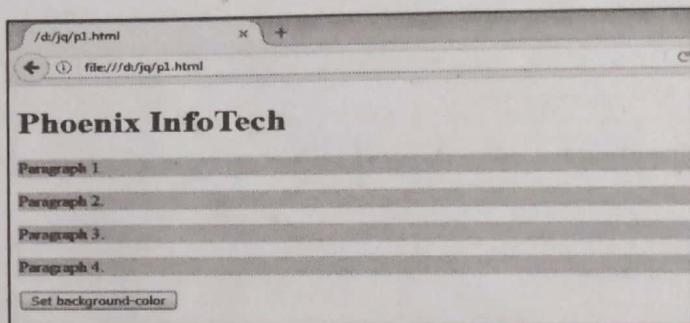
```

Output

```

    });
});
</script>
</head>
<body>
<h1>Phoenix InfoTech</h1>
<p style="background-color:red">Paragraph 1.</p>
<p style="background-color:green">Paragraph 2.</p>
<p style="background-color:blue">Paragraph 3.</p>
<p>Paragraph 4.</p>
<button>Set background-color</button>
</body>
</html>

```

Output**2.10.4 Changing Styles**

Q. How to change style in jQuery? Give example. **(4 Marks)**

jQuery css() Method is used to set or return one or more style properties for the selected elements.

Syntax

To set a specified CSS property, use the following syntax :

```
css("propertyname", "value");
```

Example

```
$(“p”).css(“background-color”, “skyblue”);
```

Program 2.10.5

Write a program to change style in jQuery.

Soln. :

Program to change style in jQuery

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").css("background-color", "skyblue");
    });
});

```

2.10.5 Creating and Appending Elements

Q. Write an example (program) to create and append HTML element in jQuery. **(4 Marks)**

jQuery provides **append()** method to add an element.

Program 2.10.6

Write a program to create and append HTML element in jQuery.

Soln. :

Creating and appending HTML element in jQuery

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btnA").click(function(){
        $("p").append("<b>This is New Appended text</b>.");
    });
    $("#btnB").click(function(){
        $("ol").append("<li>Nagpur</li>");
    });
});

```



```
</script>
</head>
<body>
<p>We will learn how to append element</p>
<ol>
<li>Pune</li>
<li>Mumbai</li>
<li>Goa</li>
</ol>
<button id="btnA">Append the text</button>
<button id="btnB">Append new city</button>
</body>
</html>
```

Output

We will learn how to append element This is New Appended text

1. Pune
2. Mumbai
3. Goa
4. Nagpur

Append the text **Append new city**

Program 2.10.7

Write a program to add multiple elements at a time in jQuery.

Soln. :

Program to add multiple elements at a time in jQuery

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
function appText()
{
    var t1 = "<p> Text with HTML</p>";
    var t2 = $("<p></p>").text("Text with jQuery");
    var t3 = document.createElement("p");
    t3.innerHTML = "Text with DOM";
    $("body").append(t1, t2, t3); // Append new elements
}
</script>
</head>
<body>
<p>Paragraph Text.</p>
<button onclick="appText()">Click Here</button>
</body>
</html>
```

Output

Paragraph Text.

Click Here

[Text with HTML](#)

[Text with jQuery](#)

[Text with DOM](#)

2.10.6 Removing Elements

Q. Write an example (program) to remove HTML element in jQuery. (4 Marks)

- Removing existing HTML elements is very easy in jQuery.
- To remove elements and its content, jQuery provides two methods :
 - (i) **remove()** : Removes the selected element with its child elements.
 - (ii) **empty()** : Removes the child elements from the selected element.

jQuery remove() Method

This method removes the selected element with its child elements.

Syntax

```
$("#div1").remove();
```

Program 2.10.8

Write a program to demonstrate the use of remove method.

Soln. :

Program to demonstrate the use of remove method

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#mydiv").remove();
    });
});
</script>
</head>
<body>
<div id="mydiv"
style="height:200px; width:400px; color:white; border:2px solid black; background-color:gray;">
Read it
</div>
<button>Delete</button>
</body>
</html>
```

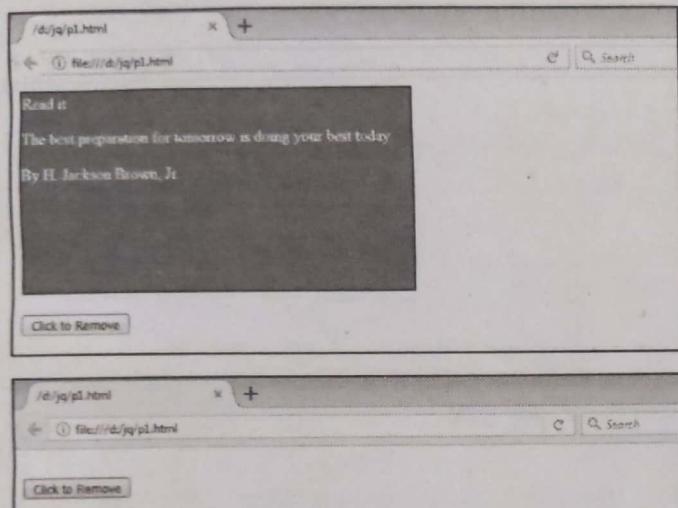
```

<p>The best preparation for tomorrow is doing your best today.</p>
<p>By H. Jackson Brown, Jr.</p>
</div>
<br>

<button>Click to Remove</button>

</body>
</html>

```

Output**2.10.7 Handling Events**

Q. Explain Event Handling in jQuery. (4 Marks)

Events are the actions which are taken by the end user while browsing the websites. We can execute specific scripts on fire of different events. jQuery provides various types of events.

- Mouse click
- Page load
- Mouse over
- Submitting an HTML form
- Key Events

When these events get fired we can call custom functions to execute scripts. Such custom functions are called as **Event Handlers**.

A. Binding event handlers

The events handlers can be established using bind() method on DOM elements in jQuery Event Model.

☞ Syntax

The syntax of the bind() method is as follows

```
selector.bind( eventType[, eventData], handler)
```

- **eventType** – It is the string which contains the JavaScript event types like click or submit
- **eventData** – This is a map of data that will be passed to the event handler. It is optional.
- **handler** – It is a function which will be executed when the event is triggered.

Program 2.10.9

Write a program to draw three sections using div. Display message on click event of these divs.

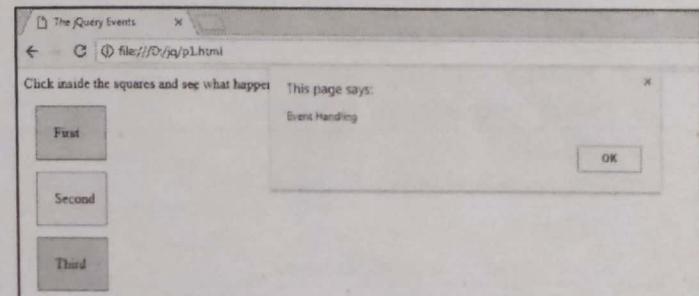
Soln. :

Program to displaying three sections on click event

```

<html>
<head>
    <title>The jQuery Events</title>
    <script type = "text/javascript"
        src =
        "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
    <script type = "text/javascript" language = "javascript">
        $(document).ready(function() {
            $('div').bind('click', function( event ) {
                alert('Event Handling');
            });
        });
    </script>
    <style>
        .div{ margin:13px;padding:20px; border:1px solid #666; width:40px; }
    </style>
</head>
<body>
    <p>Click inside the squares and see what happens:</p>
    <div class = "div" style = "background-color:pink;">First</div>
    <div class = "div" style = "background-color:yellow;">Second</div>
    <div class = "div" style = "background-color:skyblue;">Third</div>
</body>
</html>

```

Output**B. Removing event handlers**

Sometimes we may want to remove the event handler for a particular reason which we have already created. jQuery provides unbind() method to remove the event handler.

☞ Syntax

```
selector.unbind(eventType, handler)
```



or

selector.unbind(eventType)

C. Event Types

There are various types of events in jQuery. Some of them are as follows :

Sr. No.	Event Types & Description
1.	Blur : Fires when the element lost the focus.
2.	Change : Fires when the element changes.
3.	Click : Fires when element get clicked.
4.	Dblclick : Fires when element get double clicked.
5.	error : Fires when error occurs in loading or unloading etc.
6.	focus : Fires when the element gets focus on it.
7.	keydown : Fires when key is pressed.
8.	keypress : Fires when key is pressed and released.
9.	keyup : Fires when key is released.
10.	Load : Fires when document is loaded.
11.	mousedown : Fires when mouse button is pressed.
12.	mouseenter : Fires when mouse pointer moves on the element.
13.	mouseleave : Fires when mouse pointer leaves the element.
14.	mousemove : Fires when mouse pointer moves.
15.	mouseup : fires when mouse button is released.
16.	resize : Fires when window is resized.
17.	Scroll : Fires when window is scrolled.
18.	select : Fires when a text is selected.
19.	submit : Fires when form is submitted.
20.	unload Fires when documents are unloaded.

D. The Event Attributes : There are number of attributes for an event in jQuery. These attributes provides different types of information about the particular event object.

Program 2.10.10

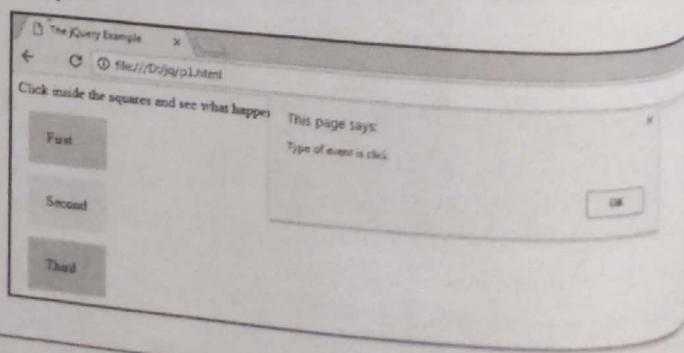
Write a program to draw three sections using div, displaying information about event type, X and Y co-ordinates, and section which get clicked.

Soln. :

Program to draw the three sections and displaying the information about event type, X, Y co-ordinates and section which get clicked

```
<html>
<head>
<title>The jQuery Example</title>
<script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script type = "text/javascript" type = "text/javascript">
$(document).ready(function() {
    $('#div').bind('click', function( event ) {
        alert('Type of event is ' + event.type);
        alert('X co-ordinate : ' + event.pageX);
        alert('Yco-ordinate : ' + event.pageY);
        alert('The Section click is : ' +
event.target.innerHTML);
    });
});
</script>
<style>
.div{ margin:13px;padding:20px;color:white border:3px
solid #666; width:50px;}
</style>
</head>
<body>
<p>Click inside the squares and see what
happens:</p>
<div class = "div" style = "background-
color:pink;">First</div>
<div class = "div" style = "background-
color:yellow;">Second</div>
<div class = "div" style = "background-
color:skyblue;">Third</div>
</body>
</html>
```

Output





2.11 Exam Pack (University and Review Questions)

- ☞ Syllabus Topic : JavaScript - Overview of JavaScript
 - Q. What is JavaScript ? (Refer section 2.1) (4 Marks)
 - Q. Explain characteristics of JavaScript.
(Refer section 2.1.1) (4 Marks) (May 2017)
 - Q. Enlist JavaScript editing tools.
(Refer section 2.1.2) (2 Marks)
 - Q. Write the difference between java and JavaScript.
(Refer section 2.1.3) (8 Marks) (May 2015)
- ☞ Syllabus Topic : JavaScript - Using JS in an HTML (Embedded, External)
 - Q. Explain syntax of JavaScript.
(Refer section 2.2.1) (2 Marks)
 - Q. Write a simple embedded JavaScript code.
(Refer Program 2.2.1) (4 Marks)
 - Q. Write a simple code to demonstrate the use of External JavaScript. (Refer Program 2.2.3) (4 Marks)
- ☞ Syllabus Topic : Data Types in JavaScript
 - Q. Explain various data types in JavaScript.
(Refer section 2.3) (4 Marks) (May 2017)
 - Q. Explain the functions prompt() and alert() with suitable example (program).
(Refer section 2.4 and Program 2.4.1) (4 Marks)
- ☞ Syllabus Topic : Control Structures in JavaScript
 - Q. Explain control structures in JavaScript.
(Refer section 2.5) (8 Marks)
- ☞ Syllabus Topic : Arrays
 - Q. How to create arrays in JavaScript?
(Refer section 2.6) (8 Marks) (May 2015)
 - Q. Explain how to delete an element from array in JavaScript. (Refer section 2.6.4) (2 Marks)
 - Q. Explain array method splice() with suitable examples in JavaScript. (Refer section 2.6.5) (2 Marks)

- ☞ Syllabus Topic : Functions and Scopes in JavaScript
 - Q. Explain user defined function in JavaScript with suitable example. (Refer section 2.7.1) (4 Marks)
 - Q. Explain Scope in JavaScript.
(Refer section 2.7.2) (4 Marks)
- ☞ Syllabus Topic : Objects in JS
 - Q. Explain how object is created and modified in java script. (Refer section 2.8.1) (8 Marks) (May 2016)
 - Q. How to define method in JavaScript Object ?
(Refer section 2.8.2) (4 Marks)
- ☞ Syllabus Topic : DOM – DOM Levels, DOM Objects and their Properties and Methods, Manipulating DOM
 - Q. Explain DOM in JavaScript.
(Refer section 2.9) (8 Marks)
 - Q. Explain innerHTML in JavaScript with the help of example (program).
(Refer section 2.9.3 and Program 2.9.8) (2 Marks)
 - Q. Write a note on DOM manipulation.
(Refer section 2.9.4) (4 Marks)
- ☞ Syllabus Topic : jQuery - Introduction to jQuery, Loading jQuery, Selecting elements, Changing styles, Creating elements, Appending elements, Removing elements, Handling events
 - Q. Write features of jQuery.
(Refer section 2.10.1) (4 Marks)
 - Q. Explain different selectors in jQuery.
(Refer section 2.10.3) (4 Marks)
 - Q. How to change style in jQuery? Give example.
(Refer section 2.10.4) (4 Marks)
 - Q. Write an example (program) to create and append HTML element in jQuery.
(Refer section 2.10.5 and Program 2.10.6) (4 Marks)
 - Q. Write an example (program) to remove HTML element in jQuery. (Refer Program 2.10.8) (4 Marks)
 - Q. Explain Event Handling in jQuery.
(Refer section 2.10.7) (4 Marks)



Server Side Technologies - I

Syllabus Topics

Introduction to Server Side technology and TOMCAT,

Servlet : Introduction to Servlet, need and advantages, Servlet Lifecycle, Creating and testing of sample Servlet, session management.

JSP : Introduction to JSP, advantages of JSP over Servlet,

Elements of JSP page : directives, comments, scripting elements, actions and templates, JDBC Connectivity with JSP.

Syllabus Topic : Introduction to Server Side Technology and TOMCAT

3.1 Introduction to Server Side Technology and Tomcat

3.1.1 Server Side Technology

Q. Write short note on Sever Side Technology. (4 Marks)

- Up till now we have learned the client side technology and various client side scripting languages. Now to understand the server side technology we again remember the example of email account form which we have seen at the beginning of our subject.
- The email account form is designed using HTML, CSS and jQuery. The data provided by user is validated with the help of JavaScript. Now consider the data provided by user is complete and valid, hence it is submitted to the server side scripting language. It is responsibility of server side scripting language to interact with database and store the e-mail account data in the database provided by the user.

Comparison between client and server side technologies

Q. Compare Client and Server Side Technologies.

(4 Marks)

Now in short we will compare both the client and server side technologies :

1. Client side scripting

- Access data from the browser
- Validates the input provided by the user.
- Supports ActiveX controls, applets, etc. to enhance web pages.
- Manipulates browser documents.
- JavaScript is the most popular client-side script.

Client Side validation

Minimizes the total number of user requests that are necessary to pass to server.

Client Side Scripting Limitations

- Browser dependency.
- The code is viewable to end users through View Source command.

2. Server side scripting

All server-side frameworks share a common set of features:

- Accepts the data which is submitted by the client side scripting.
- Creates dynamic HTML as per the input provided by the user.
- Determine information regarding the client side browser
- Access and interacts with database systems.
- Exploit the HTTP protocol

Fig 3.1.1 shows the backend development and frameworks in server side software.

FRAMEWORKS are libraries of server-side programming languages that construct the back-end structure of a site

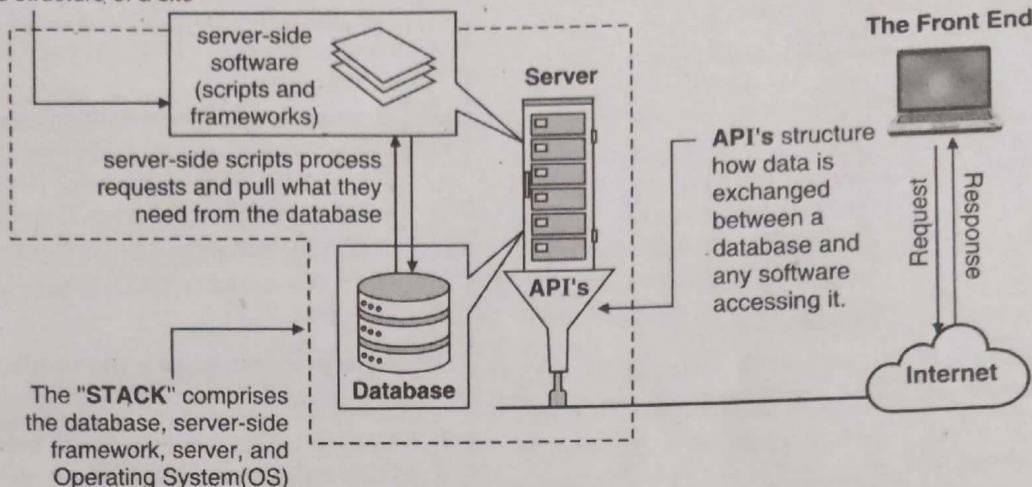


Fig. 3.1.1 : Backend development and frameworks in server side software

- Server-side scripting is the creation of dynamic Web pages served up by the Web server, as contrast to "static" web pages which are stored on the server and provided to the web browsers. In other words, some portion of the content which is sent in response to a HTTP request is tested by a script that executes on the server after the HTTP request has been received and creates content after the execution.
- 8. **Device mapping – dynamic creation** of different types of content like HTML, XML or WML depending upon the user agent (browser) which has sent the HTTP request.
- 9. **Communication** with other applications, libraries and APIs (Application Programming Interfaces) - e.g. sending e-mail, handling various message queues etc.

3.1.2 Detail Purposes and Major Uses of Server Side Scripting

Q. Write purposes and uses of Server Side Technology. **(4 Marks)**

1. **Insertion of frequently updating content into a web page**, for example – weather updates, live cricket match score, share market details etc. Also, special arbitrary logics can be implemented to decide specific content should be shown or not.
2. The **advanced authentication and authorization** which is not in the scope of client side script is done by the server side scripts.
3. **Template-driven webpage creation**. It may include repeated content such as header, footer and navigation menus around the "content area" of a web page.
4. **Personalization and customization of the web content** depending upon the authentication and authorization is done. Means interface can be dynamically created as per user requirements.
5. **Images** can be generated dynamically.
6. **HTTP headers** can be created.
7. **Managing POST form input** - accepting the input on the form provided by user and storing it to storage medias like file system, database or session. This also includes input error handling.

3.1.3 Tomcat

Q. What is Tomcat?

(2 Marks)

- Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, Java Server Pages (JSP), Java EL, and Web Socket, and provides a "pure Java" HTTP web server environment in which Java code can run.
- Tomcat is developed and maintained by an open community of developers under the support of the Apache Software Foundation, released under the Apache License 2.0.
- The Apache Tomcat is container for web which supports to run Servlet and Java Server Pages (JSP) depending upon web applications. Now a day number of modern Java web frameworks are based on servlets. Some of them are Java Server Faces, Struts, Spring etc.
- A default a HTTP connector on port 8080 is provided by the Apache Tomcat; hence Tomcat can work as a HTTP server.
- Apache Tomcat enhances various large-scale, complex web applications across a wide range of industries and organizations.



Components of Tomcat

Q. Explain Components of Tomcat.

(4 Marks)

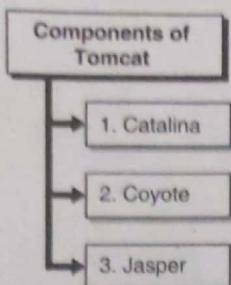


Fig. C3.1 : Tomcat components

Tomcat 4.x was released with Catalina (a Servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

→ 1. Catalina

- **Catalina is Tomcat's Servlet container.** Catalina implements Sun Microsystems's specifications for Servlet and Java Server Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles assigned to those users.
- Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

→ 2. Coyote

- **Coyote is a Connector component for Tomcat** that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as HTTP documents.
- Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client. Another Coyote Connector, Coyote JK, listens similarly but instead forwards its requests to another web server, such as Apache, using the JK protocol. This usually offers better performance.

→ 3. Jasper

- **Jasper is Tomcat's JSP Engine.** Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.
- As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems's JSP 2.0 specification.

From Jasper to Jasper 2, important features were added

- **JSP Tag library pooling** - Each tag markup in JSP file is handled by a tag handler class. Tag handler class

objects can be pooled and reused in the whole JSP Servlet.

Background JSP compilation - While recompiling modified JSP code, the older version is still available for server requests. The older JSP Servlet is deleted once the new JSP Servlet has finished being recompiled.

Recompile JSP when included page changes : Pages can be inserted and included into a JSP at runtime. The JSP will not only be recompiled with JSP file changes but also with included page changes.

JDT Java compiler - Jasper 2 can use the Eclipse JDT (Java Development Tools) Java compiler instead of Ant and javac.

Some new components were added with the release of Tomcat 7

1. **Cluster** : This component has been added to manage large applications. It is used for load balancing that can be achieved through many techniques. Clustering support currently requires the JDK version 1.5 or higher.
2. **High availability** : A high-availability feature has been added to facilitate the scheduling of system upgrades (e.g. new releases, change requests) without affecting the live environment. This is done by dispatching live traffic requests to a temporary server on a different port while the main server is upgraded on the main port. It is very useful in handling user requests on high-traffic web applications.

3. Web application

- It has also added user- as well as system-based web applications enhancement to add support for deployment across the variety of environments. It also tries to manage sessions as well as applications across the network.
- Tomcat is building additional components. A number of additional components may be used with Apache Tomcat. These components may be built by users or they can be downloaded from internet.

Syllabus Topic : Servlet - Introduction to Servlet, Need and Advantages, Servlet Lifecycle, Creating and Testing of Sample Servlet, Session Management

3.2 Introduction to Servlet → (Dec. 2014)

Q. What is a Servlet?

SPPU - Dec. 2014, 2 Marks

- **Servlet** is a server side java application which is used to create dynamic web pages.
- A Java Servlet extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers.
- The Servlets executes on a web or application server. It works as an interface between a client request which

- comes from a Web browser or other HTTP client and server side applications or databases.
- **Use :** Servlets are used to accept input from end users through the form, presents the records from server database or any other storage media and create dynamic web pages.
- Before Servlet, there was a language known as CGI (Common Gateway Interface) which works as a server side language to create dynamic content. Servlets serve the similar purpose as of the Common Gateway Interface but it offers several advantages over CGI.
- Performance of Servlet is significantly better than the CGI.

3.2.1 Need of Servlet

Q. Explain the need of Servlet. (4 Marks)

- Servlet is considered as an integral part in J2EE web applications. The server side component of a Servlet provides a great mechanism for development of server side web applications. It plays significant role in the explosion of Internet with the help of its reusability, performance and scalability.
- We can create server side programs or applications fast and efficiently by using servlets.
- The major benefit of using servlets over CGI is, the CGI scripts run outside the web server because of which every time a new process should be started before running CGI programs.
- CGI scripts cannot handle multiple requests at a time. When execution gets completed, the CGI script returns the result in the web server and exit. Servlets can handle multiple requests at a time. Dynamic content is generated by the Servlets which is easily written and fast executed within web servers
- The access to any J2SE and J2EE APIs is available to Servlet and it can take advantage of powerful JDK of Java.
- Servlets are the server side scripts which are component based, platform independent which creates the web based applications without the limitations of performance like CGI programs.

3.2.2 Advantages of Servlet → (Dec. 2014)

Q. What are the advantages of Servlet?

SPPU - Dec. 2014, 4 Marks

→ 1. Portability

Servlets are written entirely in java. As Servlets are part of java, they are portable on the basis of operating systems and server implementations.

Servlets follows the Java Rule “**Writing Once, Run Anywhere**” (**WORA**), because it is possible to develop a Servlet on Windows machine with the help of tomcat server or any other server and later on it can be deployed on any other operating system such as UNIX. As Servlets are tremendously portable they can be

executed in any platform. Hence servlets are platform independent.

→ 2. Powerful

Servlet can handle number of complex tasks which were difficult for CGI. For example Servlet can directly interact with web server, servlets can share data or information among each other, Servlets can easily make database connection pools which can be implemented easily.

Servlet support the mechanism of session tracking with the help of which it is possible to maintain and track status and information from request to request.

→ 3. Efficiency

The invocation (calling) of Servlet is highly efficient than the CGI scripts. When the Servlet is loaded in the server, it remains persistent in the memory of server as a single object instance.

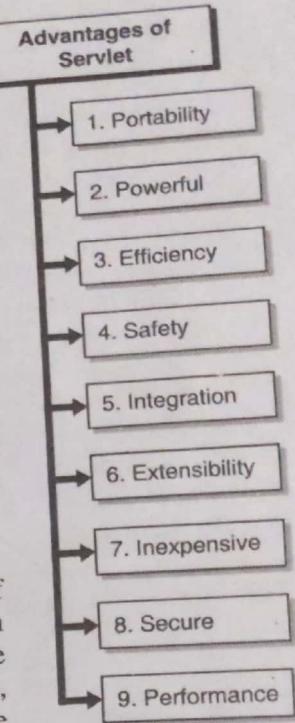


Fig. C3.2 : Advantages of Servlet

The initialization code of Servlet is executed only the first time when it is loaded by the web server. Afterword for every request only the service() method of Servlet gets called.

→ 4. Safety

As servlets are part of java, they inherit the strong type safety of JDK. The automatic garbage collection mechanism and lack of pointers in the Java environment protect the servlets from the problems of memory management. With the exception handling mechanism, it is easy to handle errors in Servlet.

→ 5. Integration

Servlets are strongly integrated with the server. Server can be used by the Servlet to perform various operations like translating the file paths, checking authorization, MIME type mapping and perform logging etc.

→ 6. Extensibility

The Servlet API is designed purposely in such a manner that it can be easily extensible. Java is robust, well-designed, object oriented language which supports the extended features. As Servlets are created in Java, they can take various such advantages of java and can be extended from existing class to give the ideal solutions as per client requirements. Hence Servlets are extensible.

→ 7. Inexpensive

Servlet are executed with the help of web servers which are available without any cost for personal as well as commercial purpose.



→ 8. Secure

Servlets are server side components which run through the web servers. Servlets can inherit the security of the web server. Servlets can also have security of Java Security Manager.

→ 9. Performance

- As compare to CGI, Servlets are relatively very faster because in CGI for every script, a new process is generated which takes a lot much of time for execution. While in Servlets, only new thread is created for new request. A single Servlet can handle multiple requests simultaneously.
- The initialization process which takes lot much times is performed only the first time when Servlet is loaded and remains in memory till times out or server shut downs. Later on handling new requests is only the matter of calling service() method for Servlet. Hence the performance of Servlet to give response to client request is higher.

3.2.3 Disadvantages of Servlet → (Dec. 2014)

Q. What are the disadvantages of servlets?

SPPU - Dec. 2014, 4 Marks

1. The process of Servlet designing is **complicated** and **slows down** the application.
2. The complex business logics written make it difficult to understand the Servlet.
3. To run the Servlets, there is **requirement of Java Runtime Environment** on the server.
4. It is very difficult for **enhancement** and **bug fixing** in Servlet applications
5. The Servlet technology has requirement of **comparatively more steps to develop**. Also Servlet need **too long time** for development.

3.3 Architecture of Servlet → (Dec. 2014)

Q. How do servlets work? SPPU - Dec. 2014, 4 Marks

Q. Explain the architecture of Servlet. (4 Marks)

- A Servlet is a class, which is implemented from the inbuilt interface javax.servlet.Servlet. However in our Servlet application, rather than directly implementing the javax.servlet.Servlet interface, we can inherit our class from such a class which has implemented this interface
- There are two main classes provided by java which are extended from javax.servlet.Servlet.

⇒ Classes of Java

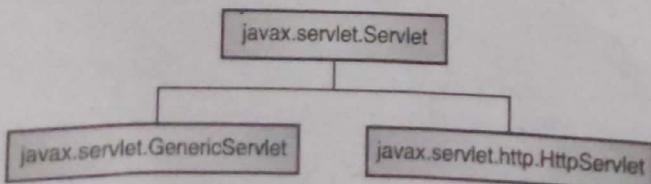


Fig. 3.3.1 : Classes of Java

- The javax.servlet package has a rich set of classes and interfaces for writing servlets.

⇒ Architecture of package

- The architecture of the package is described below:

→ (a) Servlet Interface

To create a Servlet, we have to implement the Servlet interface. This interface can be implemented by two ways directly or indirectly by extending either of the class **GenericServlet** or **HttpServlet**.

→ (b) GenericServlet

The GenericServlet class is implemented from Servlet interface. To create Servlet, if we extend the GenericServlet class then it is compulsory for us to implement the service method which is declared as an abstract method by java.

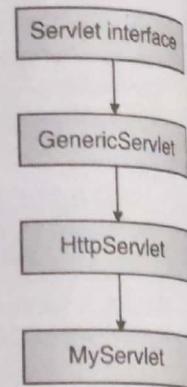


Fig. 3.3.2 : Architecture of package

→ (c) HttpServlet

The HttpServlet class is extended from GenericServlet class. To create Servlet, if we extend the HttpServlet class then there is no need to implement the service method as HttpServlet has already implemented it.

→ (d) MyServlet

This is our user defined class used to create the Servlet.

- (i) Client Interaction (ii) Servlet Execution

(i) Client Interaction

- When a Servlet accepts a call from a client, it receives two objects:

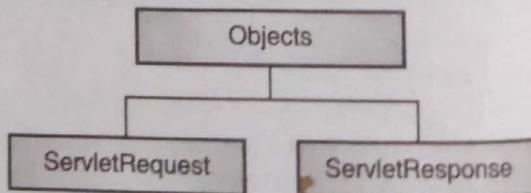


Fig. 3.3.3 : Objects

1. **A ServletRequest** - It encapsulates the communication from the client to the server.
2. **A ServletResponse** - It encapsulates the communication from the server back to the client.
- The ServletRequest interface can access information like the names of the parameters passed in by the client, the protocol (scheme) which is set by the client, and the remote host name which sends the request and the server which received it.
- It also provides the input stream **ServletInputStream** which is used by the Servlets to get data from clients. The ServletResponse interface provides the Servlet methods for giving response to the client.



- It permits the Servlet to set the length and MIME type of the content which is to be replied. It also provides an output stream : `ServletOutputStream`, and a writer which is used by the Servlet to send the reply to client.

(ii) Servlet Execution

- Fig 3.3.4 shows the execution of Servlet when client (browser) makes a request to the web server.

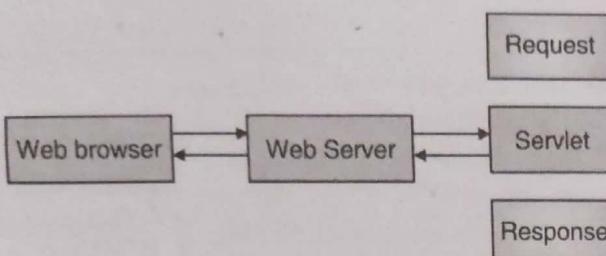


Fig. 3.3.4 : Servlet Execution

3.4 Life Cycle of Servlet

→ (May 2016, Dec. 2016)

Q. Explain the lifecycle of Servlet.

SPPU - May 2016, Dec. 2016, 6 Marks

- The life cycle of Servlet is considered as a series of steps during which a Servlet comes across through its life span, beginning from loading process till last step of destroying.
- The life cycle of Servlet is a simple object oriented design.
- Before going to life cycle of Servlet, we will discuss some technologies to understand the things better.
 - o **Web Server :** This server is also called as HTTP Server. The main functionality of web server is to handle the requests means to accept the client requests and generates the response.
 - o **Web Container :** This is also called as Servlet Container or Servlet Engine. It is an integral part of web server which communicates with the Servlets. The life cycle of Servlet is managed by the Web Container.

Life Cycle of Servlet

Steps of life cycle of servlet

The life cycle of Servlet undergoes five main steps as follows :

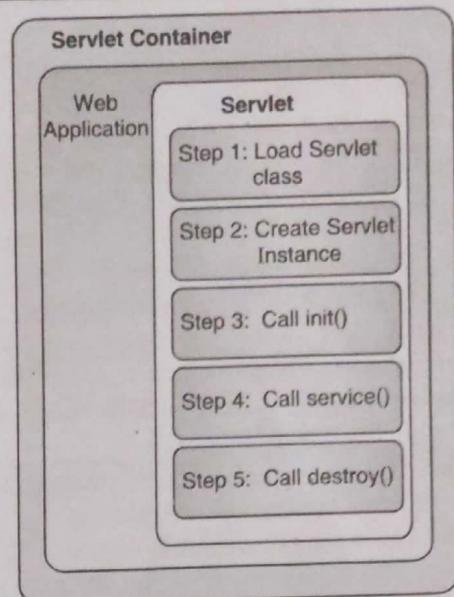
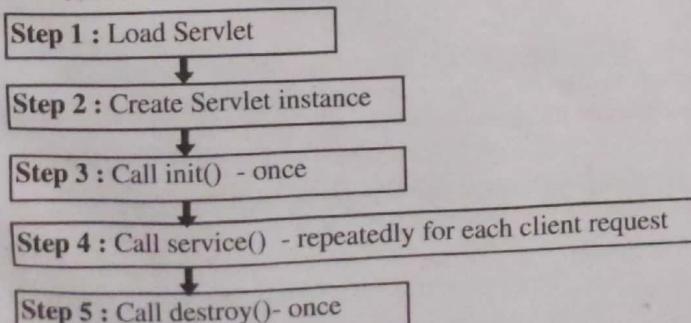


Fig. 3.4.1 : Life cycle of Servlet

Step 1 : Load Servlet

When the service of web server like Apache Tomcat is started, the Servlet is deployed and loaded by Servlet Container. Before invoking a Servlet, the Servlet Container should load its class definition first.



Step 2 : Creating instance of Servlet

As soon as the Servlet class is loaded, an instance of Servlet class is created by the Servlet Container. For every Servlet, only a single instance is created which handles all the client requests.



Step 3 : Call init() method

The `init()` method is considered as the beginning of Servlet's life cycle. As soon as the Servlet class is instantiated, the server calls the `init()` method for instantiated Servlet. This method helps to initialize the Servlet. The `init()` method is called only once in the life cycle of Servlet. Here the server creates and initializes all the necessary resources which are required for Servlet to handle the client requests.

Signature of init() method

The signature of `init()` method is as follows:

`public void init(ServletConfig config) throws ServletException;`

- In the `init()` method, the `ServletConfig` object is created as a parameter. This object can be saved to reference later.
- The `init()` method throws the `ServletException` if because of some reason the Servlet cannot initialize the resources which are necessary to handle the client requests.



Step 4 : Call service() method



- Whenever the web server receives a client request for Servlet, it initiates a new thread which invokes service() method. That means it gets called multiple times in the Servlet's life cycle. All the client requests are handled by the service() method. It is not possible to call the service() method before the execution of init() method.
- When the Servlet is extended from the GenericServlet class then the request is handled by the service() method itself, but if the Servlet is extended from HttpServlet then service() method receives the request and sends it to the correct handler method which depends upon the type of request.
- **Example :** If the request is Get Request then the service() method sends the request to the doGet() method by invoking the doGet() method with request parameters. In the same manner, for the requests such as Post, Head, Put etc. request is sent by the service() method to the related method handlers like doPost(), doHead(), doPut() etc.

Signature of service() method

The signature of service() method is as follows:

```
public void service(ServletRequest Request, ServletResponse Response) throws ServletException, IOException;
```

- The service method has two parameters **ServletRequest** and **ServletResponse**. The Servlet request object holds the information about the service request and ServletResponse object stores the data which we want to send back to client.



Step 5 : Call destroy() method

- This method indicates the end of Servlet's life. When Servlet container shuts down which generally happens when web server is stopped, it unloads all the running servlets from the memory and calls destroy() method for all the initialized servlets.
- This is where all the resources created by the init() method are cleaned up. The non-Java resources like file handle or database connections if opened in the application can be closed here.

Signature of destroy() method

The signature of destroy() method is as follows:

```
Public void destroy();
```

3.5 Creating and Testing Sample Servlet

3.5.1 Simple Servlet – Example/Program → (May 2016, Dec. 2016)

Program 3.5.1 SPPU - May 2016, Dec. 2016, 4 Marks

Write an example (program) of Servlet.

Soln. :
Simple Servlet example printing welcome message

```
import java.io.*;
import javax.servlet.*;
public class s1 extends
GenericServlet
```

Packages are imported

```
{
```

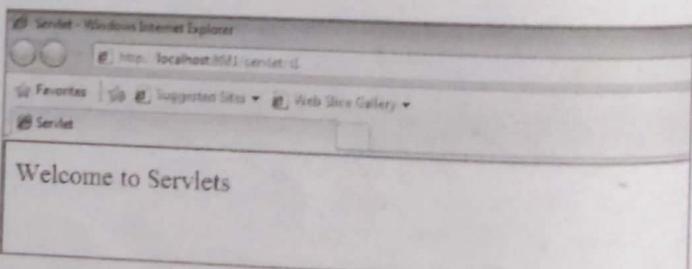
Inherited from GenericServlet
hence definition of Service() is
must

```
public void service(ServletRequest request,
ServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<font color=red size=5>");
out.println("Welcome to Servlets");
out.println("</font>");
out.println("</body>");
out.println("</html>");
out.close();
}
```

Object to use
HTML tags
and
print message

Here the Servlet class is inherited from GenericServlet, hence it is necessary for us to define the service method.

Output



3.5.2 Client Server Application – Example/Program

→ (May 2017)

Program 3.5.2 SPPU - May 2017, 8 Marks

How can we read client data using Servlet?

OR

Create HTML page which should accept name from user. Send it to Servlet. The Servlet should display welcome message with the given name.

Soln. :

Displaying welcome message accepting the name
from user

HTML Script

```
<html>
<head>
<title>Servlet</title>
```

```

</head>
<body>
<font color="blue" size=5>
<form
method="post" ACTION="http://localhost:8081/servlet/s2">
Enter your name <input type="text"
name="n"><br><br>
<input type="submit" value="Submit">
</form>
</font>
</body>
</html>

```

Explanation of HTML Script (s2.html file)

- The form method takes two arguments.
- First is **method** to which we have to assign the name of method to send the data like get, post etc depending upon the requirement.
 - Second argument is **action** to which we provide the URL of Servlet to which we want to send the data.

Servlet Code

```

import java.io.*;
import javax.servlet.*;
public class s2 extends GenericServlet
{
    public void service(ServletRequest request,
ServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>Servlet</title>
</head>");
        out.println("<body bgcolor=skyblue>");
        out.println("<font color=red size=5>");
        String nm = request.getParameter("n");
        out.println("Hello " + nm + " Welcome to Servlet");
        out.println("</font></body></html>");
        out.close();
    }
}

```

"n" is name of text field created in HTML

Explanation of Servlet code (s2.class file)

`response.setContentType("text/html");` -

The `setContentType()` is method of `ServletResponse`. It is used to set the type of content which we will return to the browser on client side.

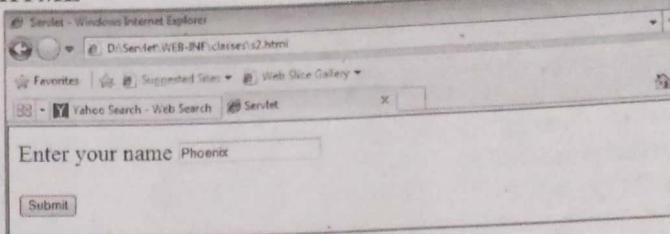
`String nm = request.getParameter("n");` -

The `getParameter()` is method of `ServletRequest`. It takes argument as name of HTML form control and returns data provided by user in it or the status of form

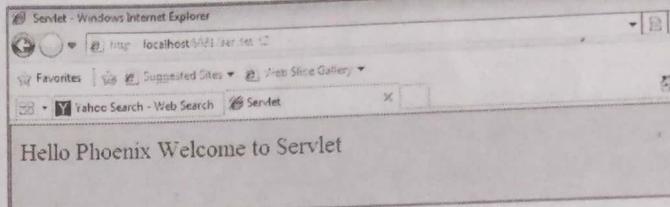
control like checked or unchecked. Here the "n" is the name of text field in HTML form.

Output

HTML



Servlet



3.5.3 doGet() and doPost() Methods

- Use :** Both of these methods are identical means both are used to transfer data between client side HTML application and server side Servlet.
- The important difference between them is `doGet()` handles requests which are sent using `get()` method by the HTML application while `doPost()` handles requests which are sent using `post()` method.

3.5.3(A) Difference between GET and POST

→ (May 2016, May 2017)

Q. What is the difference between doGet() and doPost() in Servlet? **SPPU - May 2016, May 2017, 8 Marks**

Sr. No.	Parameter	GET	POST
1.	Data	The data to be passed is appended to the URL of resource requested.	The data is passed to the server in the form of HTTP body.
2.	Hide	Does not hide the data to be passed.	Hides the data to be passed.
3.	Display	Displays data on the address bar of browser.	Does not display data on the address bar of browser.
4.	Security	Security is not maintained, hence not secure for private important data.	Security is maintained, hence secure for private important data.
5.	Data Limit	Limit on the amount of data to be transferred. (2KB with most servers)	No any limitation on the amount of data to be transferred.

3.5.4 Multi-Valued Parameter – Example/Program

Sometimes, it may be possible that the input parameter have multiple values. Here we have to use the

getParameterValues() method to accept the input provided by the user.

Program 3.5.3

Create HTML page which shows list of products. The user will select the desired products and submit the data to Servlet. The Servlet should display the total bill generated.

Soln. :

Example/program of shopping cart using multi-valued parameter

HTML Script

```
<html>
<head>
<title>Shopping Cart</title></head>
<body>
<font color="blue" size=6>
<form method ="get"
ACTION="http://localhost:8081/servlet/s4">

<input type="checkbox" name="product"
value=30000>Computer : rs 30000
<br>
<input type="checkbox" name="product"

value=50000>Laptop : rs 50000
<br>
<input type="checkbox" name="product"
value=20000>Mobile : rs 20000
<br>
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form></font></body>
</html>
```

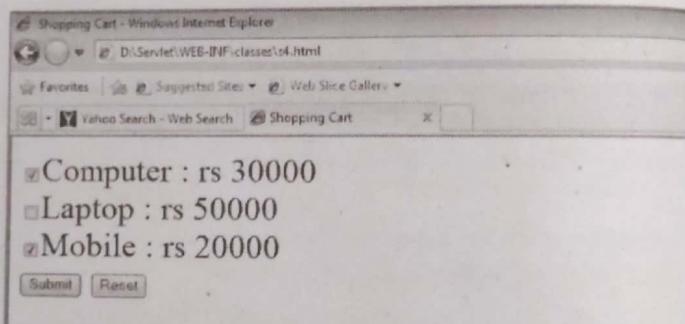
All checkboxes should have same name

```
double bill = 0;
String[ ] amounts = request.getParameterValues("product");
if(amounts != null)
{
    for(int i=0;i<amounts.length;i++)
    {
        bill = bill + Double.parseDouble(amounts[i]);
    }
}
out.println("<font size=5>");
out.println("The final bill Bill is :" + bill);
out.println("</font>");
out.println("</body>");
out.println("</html>");

out.close();
}
```

Output

HTML



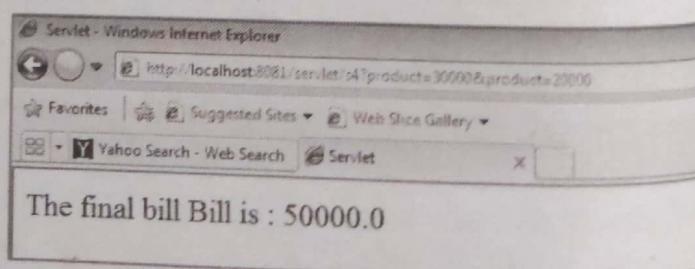
Servlet Code

```
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class s4 extends HttpServlet {

    protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Servlet</title>");
    out.println("</head>");
    out.println("<body>");
```

Servlet



3.6 Session Management / Session Tracking

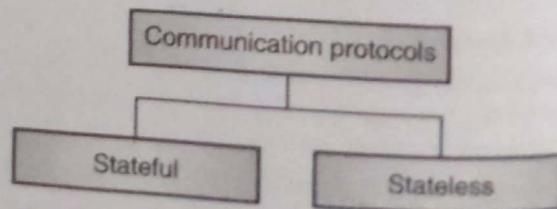


Fig. 3.6.1 : Types of Communication Protocols



The communication protocols used in internet are generally divided into **two types: stateful and stateless**.

There is difference in the nature of connection which is defined between a client and a server.

☞ Stateful Protocols

- **Telnet and FTP (File Transfer Protocol) are the examples of stateful protocols.** The client connects to the server. A series of transactions is performed by the client and then it disconnects.
- All these requests are associated by the server and it knows that they all came from the same user. That means the server associates a state with connection.
- Server knows everything about the user like who the user is, and what are his/her transactions or operations.
- **Example :** A file download request is sent by the FTP client, the FTP server can associate the request with previous request of the same user and work out which directory the user was looking at, and henceforth which exact file is requested.

☞ Stateless Protocols

- **HTTP is an example of stateless protocol.** This protocol is mainly concerned with requests and responses which are basically a series of simple and isolated transactions.
- This is suitable for simple web browsing where each request is going to be converted into a file (an HTML document, GIF etc) which is to be sent back to the client.
- There is no need for server to have knowledge of whether a series of requests come from the same, or from different client, or whether all these requests are related with each other or distinct.
- Stateless protocols are best suited for the web applications, but a big question is that; how to associate all the related requests together – how to maintain user state on the server.
- The process of maintaining state between series of requests to a Web Application is known as session tracking. This usually means passing data created from one request onward, so that it can be associated with the data created from subsequent requests.
- There are several approaches considered when the Web Application uses servlets to handle those requests.

3.6.1 Sessions

→ (Dec. 2014)

Q. What is session?

SPPU - Dec. 2014, 2 Marks

- ☞ **Definition :** A session can be defined as a series of transactions which are related with each other between a single client and the web servers, which executes over a period of time.
- The session could be series of transactions that a user makes while making updations in a stock portfolio, or the series of requests that are made to check an email account with the help of browser-based email service.

- The session may have more than one request to the same Servlet or of the requests to different resources on the same web site.
- HTTP is a stateless protocol; hence the web server does not by default have knowledge about which session a specified request belongs to. It can be obtained as follows.
- We can get the client to identify itself every time a request is made and store and retrieve data related to that specific client in some storage media on the server. It is possible to send the data to the client, and make the client to send it back with request whenever a request is made.

3.6.2 Ways for Tracking the Session and Maintaining State → (Dec. 2014)

Q. List the session tracking techniques.

SPPU - Dec. 2014, 2 Marks

There are different ways to track the session and maintain state of a client.

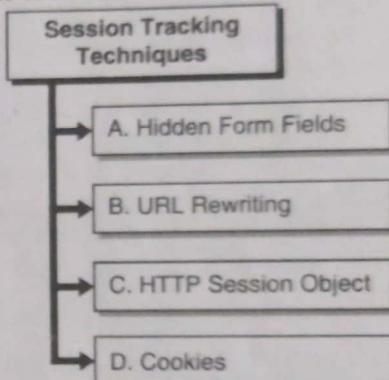


Fig. C3.3 : Session tracking techniques

→ A. Hidden Form Fields

Q. Explain Hidden Form Fields with example. List their advantages and disadvantages. (4 Marks)

- Hidden fields are the input fields which are basically not displayed on the webpage but their values are sent to the Servlet just like any other input fields. It is possible to track the session data by storing it in hidden form fields and later on retrieving it with the help of HttpServletRequest object.

☞ Example

```
<Input type="Hidden" name="pass" VALUE="SessionData">
```

- Now when the form is submitted, the specified name and value of the created hidden form fields are included in the data of GET or POST method.
- In case of hyperlinks, as the form is not submitted like submit button, the Hidden Form Field is not suggestive way to track the session.

☞ Advantage of Hidden Form Field

It will work perfectly every time without considering whether cookie is disabled or not.

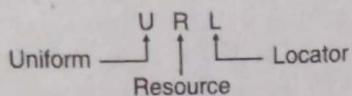


Disadvantages of Hidden Form Field

- (i) The maintenance of Hidden Form Field is done at server side.
- (ii) There is need of Extra form submission for each page.
- (iii) Sending only text type of data is allowed.

→ B. URL Rewriting

Q. Explain "URL Rewriting" with example. List its advantages and disadvantages. (4 Marks)



- At the end of each URL (Uniform Resource Locator), we can add some extra text related to session. The session identifier is associated with session data by the server.

Example

- <http://PhoenixGlobe.com/index.htm;sessionid = 13579>, the session identifier is attached as sessionid == 13579 which is later on can be accessed at server side to identify the client.
- This is good option for session tracking. This is generally helpful when browsers do not support cookies or cookies are disabled. The problem with URL re-writing is that dynamically we have to generate each URL to assign a session ID. This is applicable for simple HTML static pages also.

Advantages of URL Rewriting

- (i) It will work perfectly every time without considering whether cookie is disabled or not.
- (ii) No need of Extra form submission for each page.

Disadvantages of URL Rewriting

- (i) It will work only with links.
- (ii) Sending only text type of data is allowed

→ C. HTTP Session Object

- Servlet provides an interface HttpSession for session handling and store information about a particular user. This interface is used by the Servlet Container to create session between client and HTTP server. The session remains in the memory for a specified time interval across several request through different web pages by the user.

→ Getting the HttpSession object

There are two ways provided by the HttpServletRequest interface to get the HttpSession object.

1. public HttpSession getSession()
2. public HttpSession getSession(boolean create)

1. **public HttpSession getSession()** : Returns the current session which is associated with the specified request. If the session is not available for the request, then it creates the session and then returns it.

2. **public HttpSession getSession(boolean create)** Returns the current session which is associated with the specified request. If the session is not available for the request and the value of **create** attribute is true, then it creates the session and then returns it.

→ Commonly used methods of HttpSession interface

Q. What are Commonly used methods of HttpSession interface ? (4 Marks)

Methods of HttpSession interface

1. public String getId()
2. public long getCreationTime()
3. public long getLastAccessedTime()
4. public void invalidate()

Fig. C3.4: Methods of HttpSession interface

→ 1. public String getId()

Returns a string which has the unique identifier value.

→ 2. public long getCreationTime()

Returns the creation time of session measured in milliseconds.

→ 3. public long getLastAccessedTime()

Returns the time when the client sent the last request associated with the particular session, measured in milliseconds.

→ 4. public void invalidate()

Invalidates the particular session then unbinds all the objects bound to it.

Program 3.6.1

Create HTML page which should accept the user name from user in one session scope in one Servlet and retrieve this value (name) from same session scope in second Servlet.

Soln.: Program using HttpSession

- In this program, we will set the attribute in the session scope in one Servlet and retrieve this value from the same session scope in different Servlet.
- HttpSession interface provides method setAttribute() to set the attribute in the session scope , and to get attribute getAttribute() method is used.

Session.html

```
<html><head>
<title>Session Tracking</title>
</head>
<body>
<font color="blue" size=5>
<form method="get"
ACTION="http://localhost:8081/servlet/ServletOne">
```



```
Enter the user Name:<input type="text"
name="user_name"/>
<br/> <br/>
<input type="submit" value="Submit"/>
</form>
</font>
</body>
</html>
```

ServletOne.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletOne extends HttpServlet {
    public void doGet(HttpServletRequest request,
HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.print("<font size=6>");
            String nm = request.getParameter("user_name");
            out.print("Welcome To our Web "+nm);
            HttpSession session = request.getSession();
            session.setAttribute("user",nm);
            out.print("<br><br><a href='http://localhost:8081/servlet/ServletSecond'>Click To Proceed</a>");
            out.print("</font>");

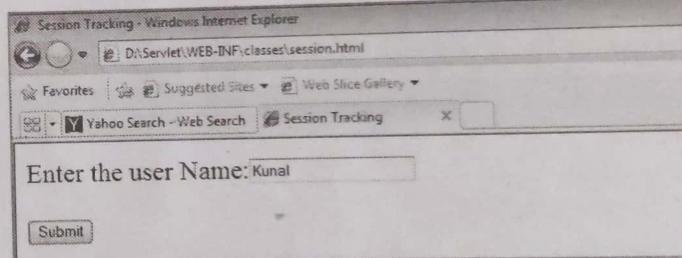
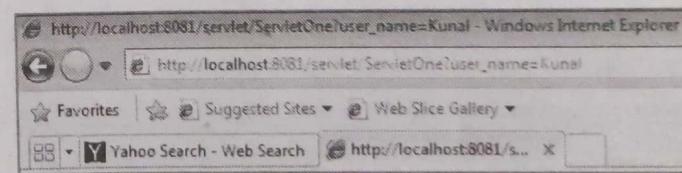
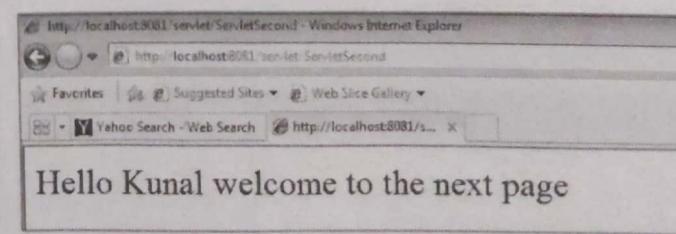
            out.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

ServletSecond.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletSecond extends HttpServlet {
    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    {
```

```
try
{
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession(false);
    String nm = (String)session.getAttribute("user");
    out.print("<font size=6>");
    out.print("Hello " + nm + " welcome to the next page");
    out.print("</font>");

    out.close();
}
catch(Exception e)
{
    System.out.println(e);
}
}
```

Output**Session.html****ServletOne.java****Welcome To our Web Kunal****Click To Proceed****ServletSecond.java**



→ D. Cookies

→ (Dec. 2014)

Q. How cookies are used to track a session?

SPPU - Dec. 2014, 4 Marks

- Cookies are the small piece of information which is sent by the web server to the client. When next request is made by client, it adds the cookie to the request header. This can be utilized to keep **track of the session**.
- Advantage :** Cookies are considered as the simplest technique used in session management for storing client state.
- Storage :** Cookies are stored on client's computer. They have a specific lifespan and are automatically destroyed by the client browser when the lifespan completes.
- We can maintain a session with cookies very efficiently but if the client disables the cookies, then it won't work.
- The Cookie class of the Servlet API is used to create cookies.
- The addCookie() method is used to add the cookies to response object. This method transfers the cookie information over the HTTP response stream.

Program 3.6.2

Write a program to set an expiry time/data for cookie.

Soln. : Setting cookie program**HTML**

```
<html><head>
<title>
Setting Cookies
</title>
</head>
<body>
<form method = "post"
action = "http://localhost:8081/servlet/s5">
<input type = "text" name = "nm" size = 25><br><br>
<input type = "submit" value = "Set Cookies">
</form>
</body>
</html>
```

Servlet

```
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;

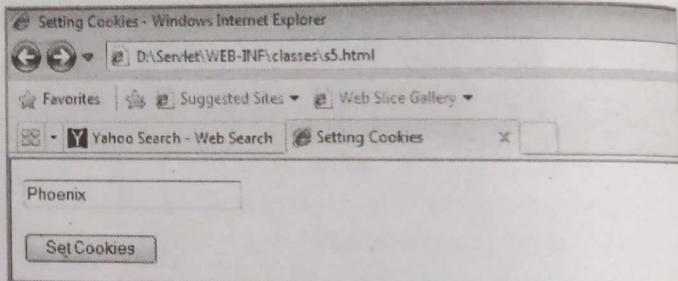
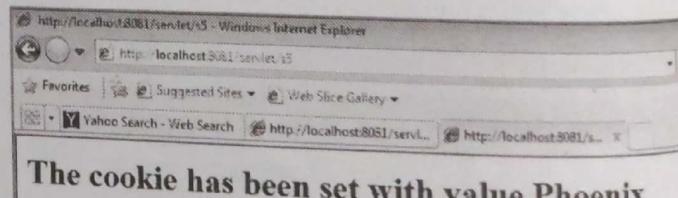
public class s5 extends HttpServlet
{
protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
```

```
{ response.setContentType("text/html");
PrintWriter out = response.getWriter();
String data = request.getParameter("nm");
Cookie ck = new Cookie("cookienm", data);
ck.setMaxAge(2*24*60*60);
```

Cookie is created by passing name and value

Setting expiry time for cookie

```
response.addCookie(ck);
out.println("<b><font size=6>The cookie been set with value "+data);
}
```

}
Output**HTML****Servlet****Program 3.6.3**

Write a program of retrieving cookies.

Soln. :**Retrieving Cookie**

Use : getCookies() method is used to access the cookies that are added to response object.

HTML

```
<html>
<head>
<title>Retrieving Cookies</title>
</head>
<body>
<form method = "post"
action = "http://localhost:8081/servlet/s6">
<input type = "submit" value = "Display Cookies">
```

```
</form>
</body>
</html>
```

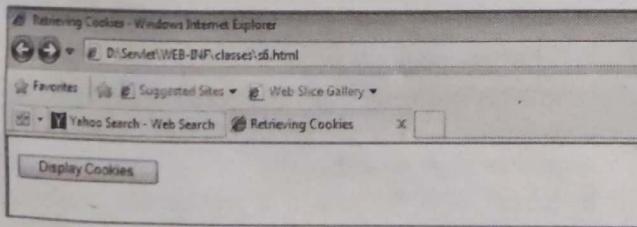
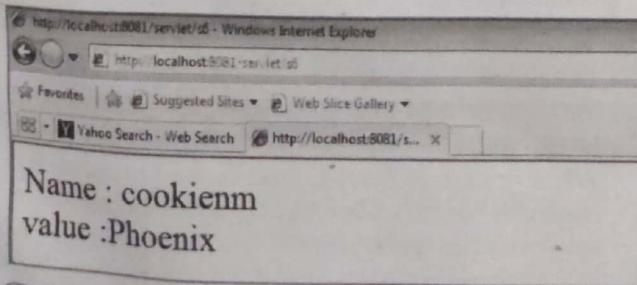
Servlet

```
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class s6 extends HttpServlet
{
    public void doPost(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException
    {
        Cookie ck[] = request.getCookies();

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        for(int i=0;i<ck.length;i++)
        {
            String nm=ck[i].getName();
            String val= ck[i].getValue();
            out.println("<font size=6>Name : "+nm+"<br>value : "+val);
        }
    }
}
```

Output**HTML****Servlet****Example/Program on Session****Program 3.6.4**

Write a program/example to count visit counter / hit counter of a website using session.

Soln. :
Program to count visit counter / hit counter of a website using session

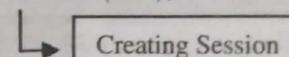
Servlet

```
import java.io.*;
import java.net.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class s7 extends HttpServlet
{
    protected void doGet(HttpServletRequest req,
    HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Visit Counter</title>");
        out.println("</head>");
        out.println("<body>");
        Integer count;
        HttpSession session = req.getSession(true);

        String heading;
        count =(Integer)session.getAttribute("accessCount");
        if(count==null)
        {
            count = new Integer(1);
            heading = "Welcome To our Website";
        }
        else
        {
            count = new Integer((count.intValue()) + 1);
            heading = "Welcom Again..";
        }
        session.setAttribute("accessCount",count);
        out.println("<font size=6>");
        out.println("<br>" + heading);
        out.println("<br>SessionID : "+session.getId());
        out.println("<br>Last accessed : "+new Date(session.getLastAccessedTime()));
        out.println("<br>Count : "+count);
        out.println("</body>");
        out.println("</html>");
        out.close();
    }

    protected void doPost(HttpServletRequest req,
    HttpServletResponse res)
```





```
throws ServletException, IOException
{
    doGet(req, res);
}
```

Output

Welcome To our Website
SessionID : BCAF872446483E0699EAAB49A971C8C4
Last accessed : Thu Aug 10 11:45:55 GMT 2017
Count : 1

Advantages and disadvantages of cookies

Q. Write advantages and disadvantages of Cookies. (2 Marks)

Advantages

- (i) It is the simplest technique used to maintain the state.
- (ii) Cookies are maintained at client side.

Disadvantage

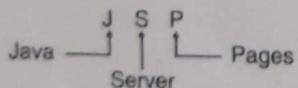
- (i) It will not work if the user disables the cookie from the browser.
- (ii) Cookie object can sent only text type of information.

Syllabus Topic : JSP – Introduction to JSP, Advantages of JSP over Servlet

3.7 JSP**3.7.1 Introduction to JSP ➔ (Dec. 2015)**

Q. What is JSP?

SPPU - Dec. 2015, 2 Marks



- **JSP stands for Java Server Pages.** JSP is a server side Java technology used to create dynamic contents just like Servlet. JSP is an extension to Servlet and provides more functionality as compared to Servlet.
- **Advantage :** A JSP page is combination of HTML tags and JSP tags. The JSP pages are comparatively easy to maintain than those of Servlet because it is possible to separate designing and development in JSP. JSP provides some important extra features like Expression Language, Custom Tags etc.
- JSP can perform various operations like accepting input from user through the forms of web pages, present records from any data source on the server, and create dynamic web pages.
- The JSP tags provide various functionalities like accessing data from the database, transfer control

between different web pages, using components of JavaBeans, registration of preferences regarding users, transferring various type of information within different requests as well as pages etc.

3.7.2 Advantages of JSP over Servlet

Q. Write advantages JSP over Servlet.

(4 Marks)

JSP has number of advantages over Servlet as follows:

Advantages of JSP over Servlet

- 1. Extension to Servlet
- 2. Easy to maintain
- 3. Fast Development
- 4. Less code than Servlet
- 5. Exception Handling
- 6. Readability
- 7. Less complexity

Fig. C3.5 : Advantages of JSP over Servlet

➔ **1. Extension to Servlet**

JSP technology is considered as the extension to previous Servlet technology. JSP provides all the features of Servlet by adding some new features like implicit objects, custom tags, predefined tags, expression language. All these new features ease the development of dynamic web pages.

➔ **2. Easy to maintain**

- In JSP the business logic and presentation logics are implemented independently and hence separated. It helps to manage the JSP easily.
- In Servlet both business and presentation logics are mixed which make management of Servlet complicated.

➔ **3. Fast Development**

- In case of JSP, only the first time JSP is compiled. There is no need of recompilation and deployment of project for each call.
- While in case of Servlet, there is need to update and recompile the page when any changes are made in look and feel of the application.

➔ **4. Less code than Servlet**

- JSP provide number of built in elements like implicit objects, action tags etc. which helps to reduce the code.
- Such elements are not available in Servlet.



→ **5. Exception Handling**

- JSP takes care of Exception handling.
- Servlet does not take care of Exception handling. It is the responsibility of programmers.

→ **6. Readability**

- JSP increases readability of code by using tags.
- In Servlet, readability of code is less.

→ **7. Less Complexity**

- JSP is easy to learn and apply.
- Servlet is comparatively complex to learn and apply.

3.7.3 Advantages of JSP over Other CGI

→ (Dec. 2015)

Q. Enlist advantages of JSP over CGI?

SPPU - Dec. 2015, 8 Marks

- JSP page can use dynamic Elements in HTML documents rather than having separate files. Hence performance is better than CGI.
- Before being processed by the Servlet, JSP is compiled. While the CGI requires the server to load an interpreter and the target script every time whenever the page is requested.
- JSP has the access of very powerful Enterprise Java API like EJB, JAXP JDBC, JNDI etc.
- The servlets can handle the business logic and JSP pages can work in combination with servlets.
- JSP is part of Java; hence it has access to entire JDK which helps to create simple to robust applications.

3.7.4 Advantages of JSP over Other Technologies

Q. Write advantages of JSP over different technologies.
(4 Marks)

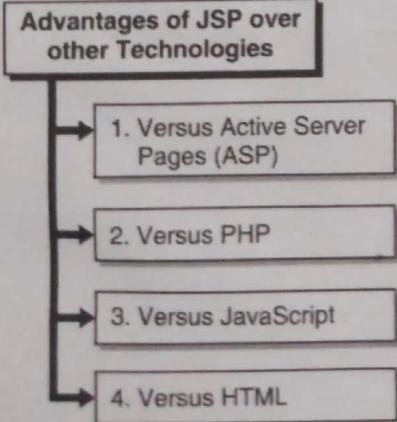


Fig. C3.6 : Advantages of JSP over different technologies

→ **(1) Versus Active Server Pages (ASP)**

- ASP is a product of Microsoft. Just like JSP it is a tag base scripting language used to create dynamic web pages. ASP code is not portable while as JSP is a part of Java, it is portable means the JSP script can be executed on any operating system.

- Also JSP can work with various web servers while ASP is stick with web servers like IIS (Internet Information Server) or PWS (Personal Web Server).

→ **2. Versus PHP**

PHP is an HTML embedded scripting language used to create dynamic pages and it is also open source such as Java. PHP is also a good scripting language but JSP has an inbuilt support of entire class library of Java. Java is more powerful, flexible, reliable and portable as compared to PHP.

→ **3. Versus JavaScript**

JavaScript is basically a client side scripting language. It can create client side applications. It is not possible to create server side applications in JavaScript like JSP. JavaScript also cannot access database of the server.

→ **4. Versus HTML**

There is no any comparison between JSP and HTML. HTML is basically used to create static web pages. Also HTML cannot access server side resources like database. JSP can have both static and non-static content. JSP contains HTML as static part.

3.7.5 Lifecycle of JSP

→ (Dec. 2015)

Q. What do you mean by JSP processing? How JSP pages are handled? Explain with architecture

SPPU - Dec. 2015, 8 Marks

There are four phases in the life cycle of JSP which are used to process the JSP page. These are shown in Fig. C3.7.

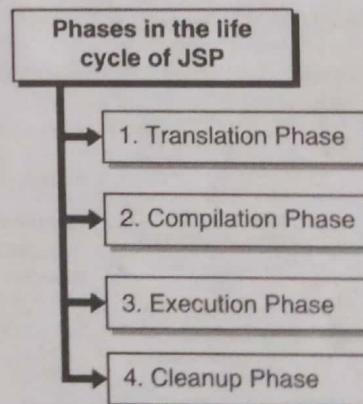


Fig. C3.7: Phases in life cycle of JSP

→ **1. Translation Phase**

At first time when there is a request for JSP page, the JSP page is translated into Servlet.

→ **2. Compilation Phase**

- Here the JSP engine checks whether the JSP page request is for the first time. If yes then the generated Servlet is compiled.
- The compilation is done only if the page is requested for the first time or any changes have been made in it.

→ **3. Execution Phase**

In this phase the JSP Servlet methods are executed.

**(a) public void jspInit()**

- This method is called when the JSP is initialized before serving any requests. This method is same as of the init() method of Servlet. If there is any need of JSP-specific initialization process then the jspInit() method can be overridden.
- In general, this method can be called only once in the life cycle of JSP. This method is used to load some non-Java resources like database connections, file handles etc.

(b) public void jspService(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

- This method is basically corresponds to the body of the JSP page. This is the main method which handles the request response paradigm. This method is implemented by the JSP Container.
- The request object holds the information about the service request and response object stores the data which we want to send back to client.
- This method is called repeatedly for each client request.

→ 4. Cleanup Phase

This phase indicates the end of JSP life cycle. In this phase the JSP is destroyed and removed from the container.

(c) jspDestroy()

This method get called when the container is about to destroy the JSP. This method is similar to Servlet's destroy() method. In this method we can perform the cleanup operation by releasing the non Java resources like database connection, file handle which we have initialized in the init() method.

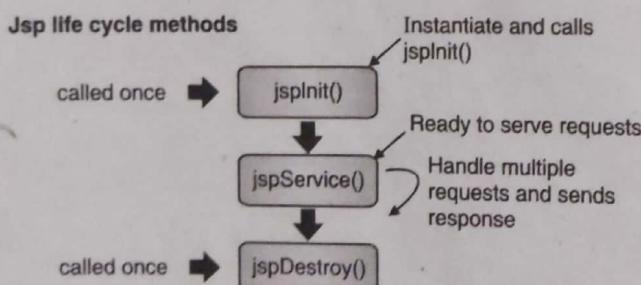


Fig. 3.7.1 : JSP life cycle method

Syllabus Topic : Elements of JSP
Page : Directive, Comments, Scripting Elements, Actions and Templates

3.7.6 Elements of JSP Page

In JSP there are three important types of elements available.

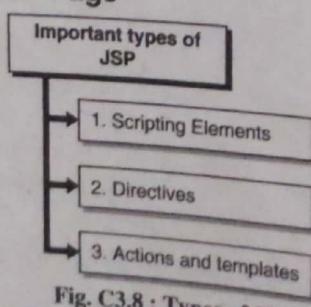


Fig. C3.8 : Types of JSP

3.7.6(A) Scripting Elements

Scripting elements are used to include scripting code (java code) within the JSP. These elements help to perform various operations like declaring variables and methods, include implementation code and evaluate an expression.

→ Types of scripting elements

Q. Explain different types of scripting elements in JSP.

(4 Marks)

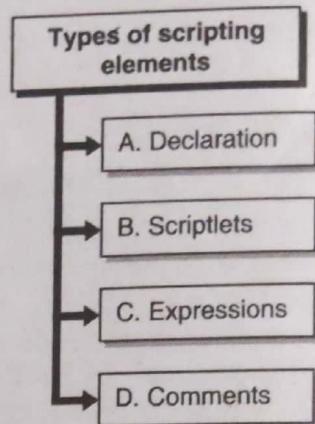


Fig. C3.9 : Types of scripting elements

→ (A) Declaration

The declaration section is used to declare class-wide variables and methods. This is block of Java code in JSP. When the JSP page is initialized all the declarations in the file get initialized. These members have class scope means they are accessible within the JSP anywhere to other declarations, expressions or code.

The declaration section is enclosed in `<%!` and `%>`.

`<%!` Declaration statements of variables and methods `%>`

→ Example

```

<html>
<head>
<title>
Scriptlet
</title>
<%!
private int n = 1;
%>
</head>
<body>
</body>
</html>
  
```

→ (B) Scriptlets

The scriptlet is a block of java statements. This block is executed when the client request is processed by the JSP. The scriptlet generates output in output stream to



send to client. In a JSP there may be multiple scriptlets as per the requirement.

A scriptlet is instance of service() method of Servlet. When the JSP is converted into Servlet by the JSP engine, the code of scriptlet is put into service() method. The scriptlet is processed for every client request.

The scriptlet section is enclosed in **<% and %>**.

```
<% Java statements; %>
```

Example

```
<html>
<head>
<title>
Scriptlet
</title>
<%!
private int n = 1;
%>
</head>
<body>
<font size=5>
<%
while(n<=10)
{
out.println("<br>Welcome to JSP");
n++;
}
%>
</body>
%>
</html>
```

→ (C) Expressions

- The expression outputs a value in the response stream which is to be sent to client. Expression is considered as shorthand notation of scriptlet. After evaluation of the expression, the result is converted into string format and displayed to the client.
- The expression is enclosed in **<%= and %>**

Example

```
<html>
<head>
<title>
Scriptlet
</title>
<%!
private int n = 1;
%>
</head>
<body>
<font size=5>
<%
```

```
while(n<=10)
{
    out.println("<br>Welcome to JSP");
    n++;
}
%>
<br> Now the value of n is <%=n%>
</font>
</body>
</html>
```

→ (D) Comments

- **Use :** Comments are used to give informative text in the JSP. The comments are ignored by the JSP container.

Syntax

- Following is the syntax of the JSP comments –

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

Example : Following example shows the JSP Comments –

```
<html>
<head><title>Example of comment</title></head>

<body>
<h2>Welcome to the world of Web</h2>
<%-- This is comment which is invisible in the page
source --%>
</body>
</html>
```

3.7.6(B) Directives

→ (Dec. 2015)

Q Explain various JSP directives.

SPPU - Dec. 2015, 4 Marks

- The directive in JSP serves as messages and directions to the JSP container. It instructs the container the way to handle some important aspects of the JSP processing.
- The entire structure of the Servlet class is affected by the directives.

☞ **Syntax :** The standard form of JSP directive is as

```
<%@ directive attribute = "value" %>
```

- The directive usually has multiple attributes which can be specified as key-value pair which should be separated by the commas.

☞ Types of directive tag

Sr. No.	Directive	Description
1.	<%@ page ... %>	This directive is used to define page dependent attributes like language, buffer, content type etc.



Sr. No.	Directive	Description
2.	<%@ include ... %>	Used to include external files during the translation phase.
3.	<%@ taglib ... %>	Used to define a tag library to be used in JSP

1. The Page Directive

The Page Directive gives instructions to the JSP container. These instructions affect the entire current JSP page. As per convention, the Page Directive should be declared at the top of the JSP, but can be declared anywhere in the JSP page.

☞ Syntax

```
<%@ page attribute = "value" %>
```

☞ Different attributes of Page Directive

Q. Explain different attributes of page directive in JSP.
(4 Marks)

Sr. No.	Attribute	Purpose
1.	Buffer	Buffer is temporary memory. This attribute is used to specify a buffering model for the output stream to the client.
2.	autoFlush	If set to "true" then the output buffer to the client is flushed automatically.
3.	contentType	Defines the character encoding for the JSP and MIME type for the response of JSP page.
4.	errorPage	Defines the URL of another JSP page. When unchecked runtime exception occurs then the control is transferred to this error page.
5.	isErrorPage	Specifies if this JSP page is a URL mentioned by other JSP page's errorPage attribute.
6.	extends	Specifies a parent class through which the generated Servlet must be extended.
7.	Import	Specifies a comma separated list of packages or classes for use in the JSP.
8.	info	Defines an information string which can be accessed with the servlet's getServletInfo() method.

Sr. No.	Attribute	Purpose
9.	isThreadSafe	Defines the model of thread for the generated Servlet. If set to true, multiple request can be handled simultaneously
10.	Language	Defines the scripting language used in the JSP page.
11.	session	Specifies whether or not the JSP page participates in HTTP sessions

2. The include Directive

- The include directive is used to include external files during the translation phase. The content of external files can be merged in the current JSP by the container. The include directive can be declared anywhere in the JSP page.

☞ Syntax

```
<%@ include file = "relative url" %>
```

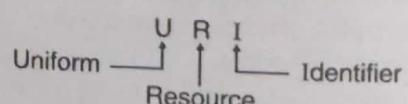
- In the include directive we have to specify the relative URL of the external file which we want to include in our JSP. If the filename is given without any specific path, then the JSP compiler search for the file in the same directory in which the current JSP is stored.

3. The taglib Directive

- This directive allows the JSP page to include and use custom user defined tag library. This tag library is a bunch of user-defined JSP tags which implement custom behavior.
- The taglib directive indicates that the JSP page is going to use some custom tags.

☞ Syntax

```
<%@ taglib uri = "uri" prefix = "prefixOfTag" %>
```



The URI (Uniform Resource Identifier) attribute identifies the location which the container understands and the prefix attribute informs prefix string which is used to define the custom tags.

3.7.7 JSP Programs

Program 3.7.1

Write a simple JSP program.

Soln. :

Simple JSP program

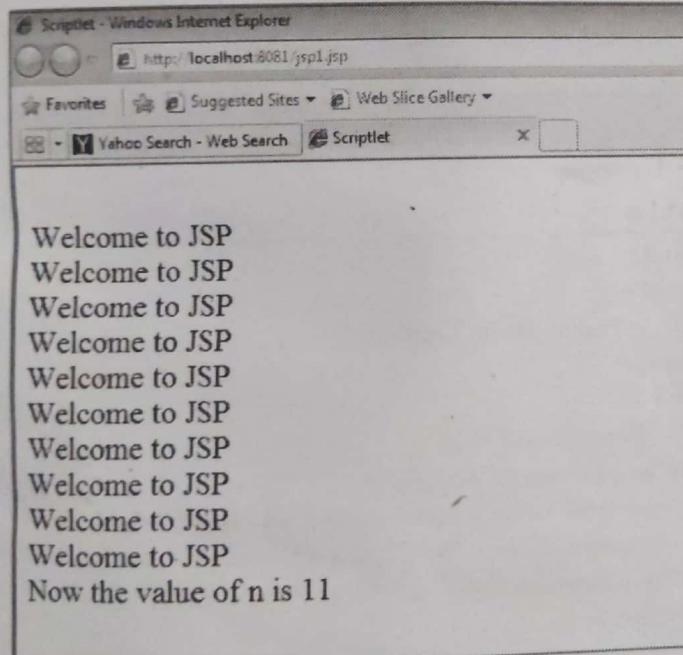
```
<html>
<head>
<title>
Scriptlet
```

```

</title>
<%!
private int n = 1;

%>
</head>
<body>
<font size=5>
<%
while(n<=10)
{
    out.println("<br>Welcome to JSP");
    n++;
}
%>
<br> Now the value of n is <%=n%>
</font>
</body>
</html>

```

Output**Program 3.7.2**

Create HTML page which should accept name from user. Send it to JSP. The JSP should display welcome message with the given name.

Soln. :

HTML

```

<html><head>
<title>Servlet</title>
</head>

<body>
<font color="blue" size=5>

```

```

<form method =
"post" ACTION="http://localhost:8081/jsp2.jsp">
Enter your name <input type="text"
name="n"><br><br>
<input type="submit" value="Submit">
</form>
</font>
</body>
</html>

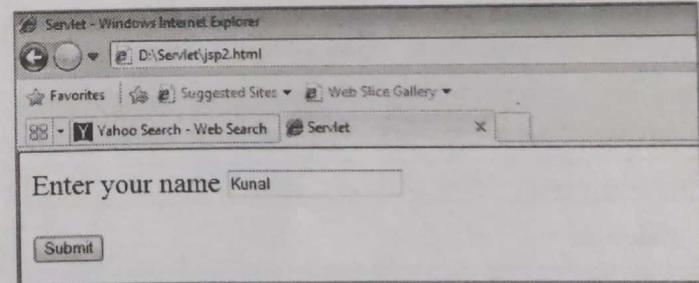
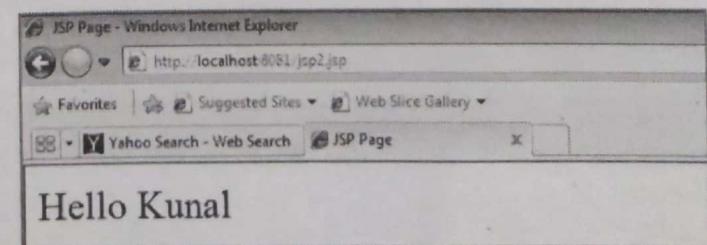
```

JSP

```

<%@page contentType="text/html"%>
<html>
<head><title>JSP Page</title></head>
<body>
<font color="blue" size=6>
<%
String nm = request.getParameter("n");
out.println("Hello " + nm);
%>
</font>
</body>
</html>

```

Output**HTML****JSP****3.7.8 Actions and Templates**

Q. Enlist and explain different types of actions in JSP.

(4 Marks)



Types of Actions

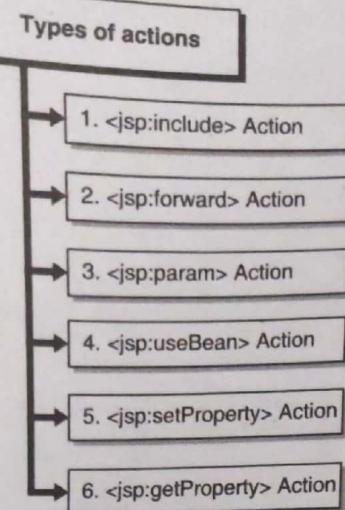


Fig. C3.10 : Types of Actions

- ☞ **Use :** JSP actions are used to handle the behavior of the Servlet engine. These actions can perform various tasks like insertion of file dynamically, forward the control to next page, reuse JavaBeans components or creation of HTML content for the Java plugin.

Syntax of actions

```
<jsp:action_name attribute = "value" />
```

→ 1. <jsp:include> Action

This is same as of include page directive i.e. it is used to insert a JSP file in another file.

<jsp:include> vs include directive

In <jsp:include> the external file is inserted at the time of request processing while in include directive it is inserted at translation phase.

☞ Syntax of <jsp:include>

```
<jsp:include page="page URL" />
```

☞ Example

```
<html>
<head>
<title>Example of JSP include Action Tag</title>
</head>
<body>
<h3>Include JSP</h3>
<jsp:include page="test.jsp" flush="false" />
</body>
</html>
```

- **page :** Page value is test.jsp. It is the page which we want to include in the current file. Only the file name indicates that the test.jsp is in the same directory.
- **flush :** Its value is false, which indicates that the resource buffer has not been flushed out before inserting in the current page.

→ 2. <jsp:forward> Action

<jsp:forward> is used for redirecting the request and transfer the control to another file.

☞ Syntax of <jsp:forward>

```
<jsp:forward page="URL_page" />
```

☞ Example

first.jsp

```
<html>
<head>
<title> Forward Action Tag</title>
</head>
<body>
<h3>Forward</h3>
<jsp:forward page="next.jsp" />
</body>
</html>
```

→ 3. <jsp:param> Action

This action is used to pass parameters to Other JSP action tags like JSP include and JSP forward tag. The request object is used to accept those parameters.

☞ Syntax of <jsp:param>

```
<jsp: param name="param_name"
value="value_parameter" />
```

☞ Example

first.jsp

```
<html>
<head>
<title> Param Action Tag</title>
</head>
<body>
<h3>Param Action</h3>
<jsp:forward page="next.jsp">
<jsp:param name ="date" value="14-08-2017" />
<jsp:param name ="time" value="10:15AM" />
<jsp:param name ="data" value="Wave" />
</jsp:forward>
</body>
</html>
```

In this example the page first.jsp is passing three parameters; date, time & data to next.jsp and next.jsp can access these parameters as follows:

```
sDate:<%= request.getParameter("date") %>
sTime:<%= request.getParameter("time") %>
sData:<%= request.getParameter("data") %>
```

→ 4. <jsp:useBean> Action :

This action is used to use Beans in JSP page. Beans are invoked by this tag.

☞ Syntax of <jsp:useBean>

```
<jsp: useBean id=" bean_id"
class="package_name.class_name" />
```

☞ Example of <jsp:useBean>, <jsp:setProperty> and <jsp:getProperty>

When Bean class is instantiated as shown in above statement, we have to use jsp:setProperty and jsp:getProperty actions to use the bean's parameters.

☞ Test.jsp

```
<html>
<head>
    <title>Using Beans</title>
</head>
<body>
    <h1>Bean Test</h1>
    <jsp:useBean id="sn" class="javabn.StuBean"/>
    <jsp:setProperty name="sn" property="*"/>
    <h1>
        name:<jsp:getProperty name="sn"
property="sname"/><br>
        empno:<jsp:getProperty name="sn"
property="srollno"/><br>
    </h1>
</body>
</html>
```

☞ StudentBean.java

```
package javabn;
public class StuBean {
    public StuBean() {
    }
    private String sname;
    private int srollno;
    public void setName(String name)
    {
        this.sname = name;
    }
    public String getName()
    {
        return sname;
    }
    public void setRollno(int rollno)
    {
        this.srollno = rollno;
    }
    public int getRollno()
    {
        return srollno;
    }
}
```

→ 5. <jsp:setProperty> Action

This action tag is basically used to set the property of a Bean. We have to mention unique name for bean while using this action tag.

☞ Syntax of <jsp:setProperty>

```
<jsp: useBean id=" bean_id"
class="package_name.class_name" />
...
...
<jsp:setProperty name="bean_id"
property="property_name" />
```

OR

```
<jsp: useBean id=" bean_id"
class="package_name.class_name" />
...
...
<jsp:setProperty name=" bean_id"
property="property_name" />
</jsp:useBean>
```

It also possible to use '*' in property_name, which indicates that all the request parameters which matches to the Bean's property will be passed to the corresponding setter method.

→ 6. <jsp:getProperty> Action

This action helps to retrieve the value of Bean's property.

☞ Syntax of <jsp:getProperty>

```
<jsp: useBean id=" bean_id"
class="package_name.class_name" />
...
...
<jsp:getProperty name=" bean_id"
property="property_name" />
```

OR

```
<jsp: useBean id=" bean_id"
class="package_name.class_name" />
...
...
<jsp:getProperty name=" bean_id"
property="property_name" />
</jsp:useBean>
```

☞ Other Action Tags

There are also some other tags which are not frequently used.

7. <jsp:plugin> Action

This tag is used at the situation when there is requirement of a plugin to execute a Bean class or an Applet.

8. <jsp:body> Action

9. <jsp:element> Action

10. <jsp:text> Action

11. <jsp:attribute> Action

3.8 Implicit Objects in JSP

Q. Explain implicit objects in JSP.

(8 Marks)

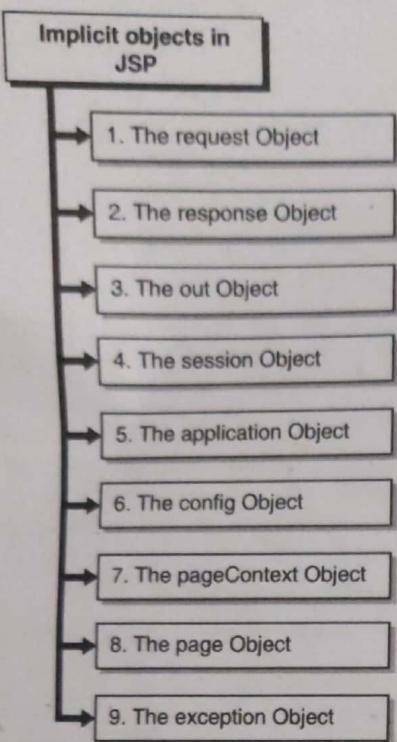


Fig. C3.11 : Implicit objects in JSP

→ 1. The request Object

- This object is an instance of a **javax.servlet.http.HttpServletRequest** object. Whenever there is request for JSP page from client, a new object is created by JSP engine to represent that request.
- The HTTP header information containing form data, cookies, HTTP methods can be accessed using request object. This object is passed to the JSP by the container as a parameter.

→ 2. The response Object

- The response object is an instance of a **javax.servlet.http.HttpServletResponse** object. When the JSP engine creates request object, at the same time it creates the response object to handle the response for client.
- The response object encapsulates the response generated by the JSP, to be sent back to client in response to request. This object is passed to the JSP by the container as a parameter.

→ 3. The out Object

The out implicit object is an instance of a **javax.servlet.jsp.JspWriter** object. It represents output stream and is used to send content in response.

→ 4. The session Object :

The session object is an instance of **javax.servlet.http.HttpSession**. The session object

track the client session between different client requests. It is valid only for HTTP requests. Sessions are created automatically hence this variable exists even if there was no incoming session reference.

→ 5. The application Object

- It is an instance of a **javax.servlet.ServletContext** object. It represents the context within which the JSP is executing.
- In the lifecycle of JSP, this object represents the JSP page all the time. When the JSP page is initialized, this object is created and destroyed when JSP page is removed by the **jspDestroy()** method.

→ 6. The config Object

The config object is an instance of **javax.servlet.ServletConfig**. It represents the Servlet configuration. This object has page scope.

→ 7. The pageContext Object

- The pageContext object is an instance of a **javax.servlet.jsp.PageContext** object. It has page scope. It encapsulates the page context for the specific JSP page.
- This object is used to access information about the page while avoiding most of the implementation details.
- For all the requests, this object holds the references to the request and response objects. The **out**, **application**, **session** and **config** objects are derived by retrieving attributes of this object.

→ 8. The page Object

This object is considered as real reference to the instance of the JSP page. It is an object which represents the complete JSP page.

→ 9. The exception Object

- This object is an instance of class **java.lang.Throwable**. It refers to the runtime exception which is resulted in the error page being invoked. This object has page scope.
- The exception object is a container which holds the exception which is thrown by the preceding page. It generates suitable response to the error situation.

Syllabus Topic : JDBC Connectivity with JSP

3.9 JDBC Connectivity with JSP

→ (May 2015)

Program 3.9.1 SPPU - May 2015, 8 Marks

Explain how can you access a database from a JSP page?
OR

Accept employee number from user on html page. Send it to JSP. The JSP should search the record from database and print it.

Soln. :

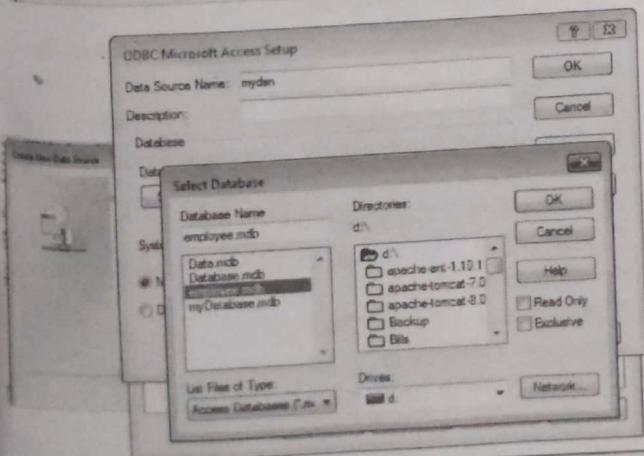
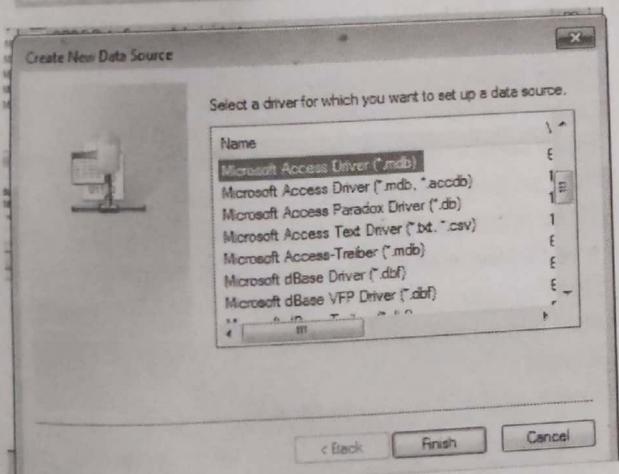
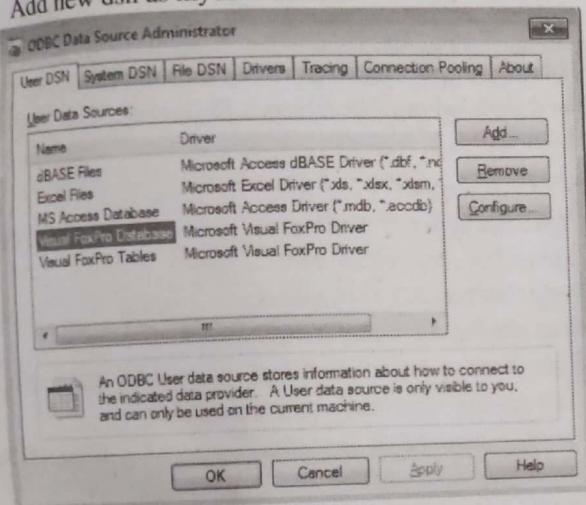
Here we consider the database created in MS access as employee.mdb having table emp.



eno	ename	sal
101	Savita	9000
102	Sharvari	8000
103	Mayuri	8500
104	Sagar	9700
105	Vinay	8500

- Create dsn (data source name for it)
- Path is as follows :
 - Control Panel → All Control Panel Items
 - Administrative Tools → Data sources(ODBC) → user dsn

Add new dsn as mydsn and link it to the mdb file.



HTML

```
<html>
<head>
<title>JSP</title></head>

<body>
<font color="blue" size=6>
<form method="get"
ACTION="http://localhost:8081/jsp3.jsp">
```

Enter Employee number

```
<input type="text" name="empno">
<br><br>
<input type="submit" value="Submit">
</form></font></body></html>
```

JSP

```
<%@page contentType="text/html"
import="javax.servlet.* , java.sql.* , java.io.* "%>
```

```
<html>
<head><title>JSP Page</title></head>
```

```
<body>
```

```
<%

```

```
Connection dbCon;
```

```
int eno;
```

```
try
```

```
{
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

Registration
of drivers

```
String dbURL = "jdbc:odbc:mydsn";
```

mydsn is Data Source
Name connected
to database

```
dbCon = DriverManager.getConnection(dbURL);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
System.out.println("Database connection failed");
```

Connection object
is initialized

```
}
```

Exception handling in
case of connection failure

```
System.out.println(e.toString());
```

```
return;
```

```
}
```

```
response.setContentType("text/html");
```



```

eno = Integer.parseInt(request.getParameter("empno"));

try {
    Statement s = dbCon.createStatement();

    ResultSet rs = s.executeQuery("select * from emp where
        eno=" + eno);

    out.println("<font size=6>");
    while (rs.next())
    {
        out.println(rs.getString(1) + " - " + rs.getString(2) + " - "
        + rs.getString(3));
    }
}

catch (SQLException e)
{
}

out.close();
%>
</body>
</html>

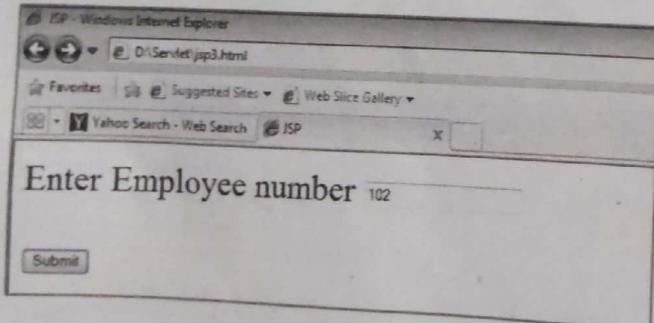
```

Annotations:

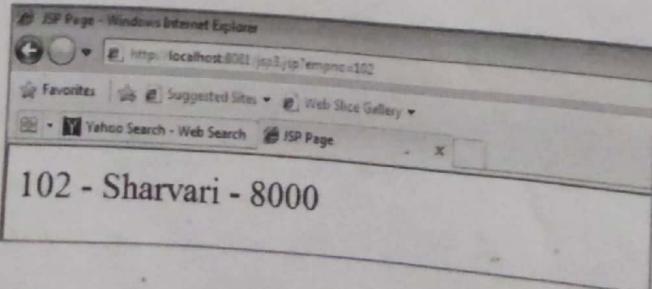
- eno is name of text field of HTML page
- Statement object is used to execute SQL queries
- ResultSet object stores result of SQL query
- getString(1) – Value of first column

Output

HTML



JSP



3.10 Exam Pack (University and Review Questions)

- Syllabus Topic : Introduction to Server Side Technology and TOMCAT**
- Q. Write short note on Server Side Technology.
(Refer section 3.1.1) (4 Marks)
- Q. Compare Client and Server Side Technologies.
(Refer section 3.1.1) (4 Marks)
- Q. Write purposes and uses of Server Side Technology.
(Refer section 3.1.2) (4 Marks)
- Q. What is Tomcat? (Refer section 3.1.3) (2 Marks)
- Q. Explain Components of Tomcat.
(Refer section 3.1.3) (4 Marks)
- Syllabus Topic : Servlet - Introduction to Servlet, Need and Advantages, Servlet Lifecycle, Creating and Testing of Sample Servlet, Session Management**
- Q. What is a Servlet?
(Refer section 3.2) (2 Marks) (Dec. 2014)
- Q. Explain the need of Servlet.
(Refer section 3.2.1) (4 Marks)
- Q. What are the advantages of Servlet?
(Refer section 3.2.2) (4 Marks) (Dec. 2014)
- Q. What are the disadvantages of servlets?
(Refer section 3.2.3) (4 Marks) (Dec. 2014)
- Q. How do servlets work?
(Refer section 3.3) (4 Marks) (Dec. 2014)
- Q. Explain the architecture of Servlet.
(Refer section 3.3) (4 Marks)
- Q. Explain the lifecycle of Servlet.
(Refer section 3.4) (6 Marks) (May 2016, Dec. 2016)
- Q. Write an example (program) of Servlet.
(Refer program 3.5.1) (4 Marks) (May 2016, Dec. 2016)
- Q. How can we read client data using Servlet?
(Refer program 3.5.2) (8 Marks) (May 2017)
- Q. What is the difference between doGet() and doPost() in Servlet?
(Refer section 3.5.3(A)) (8 Marks)
- Q. What is session? (Refer section 3.6.1) (2 Marks) (May 2016, May 2017)
- Q. List the session tracking techniques.
(Refer section 3.6.2) (2 Marks) (Dec. 2014)
- Q. Explain Hidden Form Fields. List their advantages and disadvantages.
(Refer section 3.6.2) (4 Marks)
- Q. Explain "URL Rewriting". List its advantages and disadvantages.
(Refer section 3.6.2) (4 Marks)
- Q. What are commonly used methods of HttpSession interface? (Refer section 3.6.2) (4 Marks)



- Q. How cookies are used to track a session?
(Refer section 3.6.2) (4 Marks) **(Dec. 2014)**
- Q. Write advantages and disadvantages of Cookies.
(Refer section 3.6.2) (2 Marks)
- Syllabus Topic : JSP – Introduction to JSP, Advantages of JSP over Servlet**
- Q. What is JSP?
(Refer section 3.7.1) (2 Marks) **(Dec. 2015)**
- Q. Write advantages JSP over Servlet.
(Refer section 3.7.2) (4 Marks)
- Q. Enlist advantages of JSP over CGI?
(Refer section 3.7.3) (8 Marks) **(Dec. 2015)**
- Q. Write advantages of JSP over different technologies.
(Refer section 3.7.4) (4 Marks)
- Q. What do you mean by JSP processing? How JSP pages are handled? Explain with architecture.
(Refer section 3.7.5) (8 Marks) **(Dec. 2015)**

- ☞ Syllabus Topic : Elements of JSP Page: Directive, Comments, Scripting Elements, Actions and Templates**
- Q. Explain different types of scripting elements in JSP.
(Refer section 3.7.6(A)) (4 Marks)
- Q. Explain various JSP directives.
(Refer section 3.7.6(B)) (4 Marks) **(Dec. 2015)**
- Q. Explain different attributes of page directive in JSP.
(Refer section 3.7.6(B)) (4 Marks)
- Q. Write a program to illustrate Client-Server application in JSP. *(Refer Program 3.7.2) (4 Marks)*
- Q. Enlist and explain different types of actions in JSP.
(Refer section 3.7.8) (4 Marks)
- Q. Explain implicit objects in JSP.
(Refer section 3.8) (8 Marks)
- ☞ Syllabus Topic : JDBC Connectivity with JSP**
- Q. Explain how can you access a database from a JSP page? *(Refer program 3.9.1) (8 Marks)* **(May 2015)**

□□□