

WEB ENGINEERING

SUBJECT CODE-

CSE 421



WRITTEN BY-

ISRAT JAHAN.

PDF CREATE BY-

M.H.SUMI.

Subject Code: CSE-421**Web Engineering****CHAPTER 1 PAGE NO:07****INTRODUCTION TO WEB APPLICATION**

1. Define
 - i) Webpage (2016,2013)
 - ii) Website (2016,2013)
 - iii) Web server (2016)
 - iv) Web browser (2016)
2. Explain the basic structure of website (2017,2015 2011)
3. Which components are needed to create a fully dynamic web page?(2015)
or, write down the basic concepts of making dynamic website. (2012)
4. explain the document tree of the webpage. (2012)
5. Describe the common feature of web server. (2017,2014, 2011)
6. What are the important of !doctype and <meta> tag in webpage. (2017)
or, explain the necessity of !doctype in web page design. (2011)
or, Explain<meta>tag with example. (2011)
7. Distinguish between HTTP & HTTPS (2016,2014)
8. what is the difference among WAMP,MAMP and LAMP? (2015)
9. Name the main disadvantages of working on a remote web server. (2015)
- 10.what do you know about agile methods for developing a website? how do agile development methods overcome the problems that arise in conventional development approaches? (2015)
- 11.Describe basic concept of static and dynamic website. (2013)
- 12.Distinguish between Static website and Dynamic website. (2016,2011)
- 13.Briefly explain client – server architecture (2016, 2013)
14. Write down the problems created by Hoakers, Spyware , Virus and Pop-ups. (2016,2013)
15. write down the function of w3 system. (2013)
16. name one popular website and list out the technical features that made this site special. (2015)
17. write down the concept of web application (2014)

18. Explain the symptoms of virus attack in the computer and how to prevent
computer from virus. (2014)
19. explain how HTTP protocol works with http request and response. (2012)

CHAPTER 2 PAGE NO: 29

HTML AND INTRODUCTION TO HTML

1. What is HTML attributes ? Explain the HTML elements . (2017,2011)
2. Write a code segment that will create a password filed in an HTML form. (2017,2014)
3. Write down the difference between GET and POST method. (2017)
4. What are the advantages of HTML 5 over HTML4? Discuss any two Tag names of HTML 5 with example. (2016,2012)
5. discuss paragraph, heading and lists tag with example. 2012
6. explain the following tag with example:-
 (i) <table></table>
 (ii) <img.>.....
 (iii)<a href>..... (2012)
7. what is DOCTYPE in HTML? Why it is used? (2012)
8. what are the differences between HTML and XHTML?
9. Discuss different types of links in HTML. (2014,2012,2011)
- 10.what are the necessities of using HTML form? Explain HTML form tag with its different attributes. (2016,2014,2011)
 or, With example, describe different form input element (text field, text area, check box, radio, dropdown list, submit button, password button). (2010)
- 11.describe basic tag of HTML. (2014)
- 12.Describe tag with its common attributes (align alt, border, sre, width, use map). Give example. (2010)
- 13.How information is present using table tag and related tags and their attribute? Explain with example. (2010).
- 14.What new input types available at HTML5? Describe 4 new input types with example. (2010)

Web Engineering -3

15. Identify HTML element that you can use inside the <head></head> element and describe what they are use for. (2010)
16. How we make web page dynamic and interactive? Discuss at least 3 techniques to make web page dynamic and interactive. (2010)
17. Identify HTML element that is use to create divisions or sections within your page. (2010)
18. What are the necessities of HTML form? Explain HTML FORM tag with its different attributes. (2016)
19. Explain the way using HTML code in which data can be presented in a tabular form as given below: (2016)

CSE	
Semester	1
	2
	3

CHAPTER 3 PAGE No:63**BASIC CONCEPTS OF CSS**

1. what is CSS? Write down the CSS syntax with example. (2016,2014)
2. Why should you prefer CSS rather using only HTML in web design? (2017,2014)
3. Describe CSS inheritance with example. (2017, 2011)
4. Explain different ways of inserting a style sheet in a webpage. (2017,2012)
or, what are the ways of inserting CSS into an HTML page? Explain with example. 2015
5. or, What three ways of inserting CSS in your HTML page and their priority level? Explain with example 2010
6. compare different style sheet approaches. 2016,2015,2011
7. which CSS approaches is best and why? 2016
8. Describe different CSS selectors with example 2017
9. explain margin, border and shading with CSS code. 2012
10. explain margin, border and padding of box model with example. 2013

10. how CSS property value cascade in to one style sheets explain with example. 2013
 11. explain the contextual selection of ID and class with example. (2015)
 12. Write down the difference between CSS ID and class. 2015
 13. explain direct descendant selector with example. 2015

CHAPTER 4 PAGE NO: 81

BASIC CONCEPTS OF JAVA APPLET

1. What is Scripting Language? (2017, 2013, 2012)
2. what is JavaScript? (2015, 2010)
3. Why JavaScript is used in the web page design? (2015, 2010)
Or, What are the advantages of JavaScript? (2013)
4. Write down the differences between Programming Language and Scripting Language. (2017, 2014, 2012)
5. Write down some common mistakes in JavaScript. (2017, 2012)
6. Write a JavaScript Program to calculate the Fibonacci series (2017, 2012)
7. what is the necessity of HTML,CSS and script language used together to build a website. (2012)
8. write a JavaScript program to find the minimum number among three numbers. (2015, 2010)
9. Different between client side and server side scripting with example. (2010)
10. Different between following common attribute and describe their uses : Class, id, name.. (2010)
11. Write a JavaScript function for common input validation. . (2010)
12. what is DHTML ? (2012)
13. describe the following types of event in JavaScript with example:- (2013)
 - (i) input event
 - (ii) mouse event
 - (iii) load event
14. describe different mouse events in JavaScript. (2012)
15. describe a JavaScript function. Right line, date and time with example.
16. what is a cookie? How does one access cookies in JavaScript? (2012) (2014)

17. describe JavaScript objects briefly with examples. (2014)
18. explain the way JavaScript handles arrays with examples. (2014)
19. write a JavaScript program to find the factorial of a given number. (2014)
20. explain document . Get element By ID () function with example. (2015, 2010)
21. prove that, HTML form validation can be done by JavaScript. (2013)
22. describe various types of input control for a basic form. (2013)

CHAPTER 5 PAGE NO:108**PHP AND MY-SQL**

1. What is PHP? (2017,2014,2013)
2. How PHP code in bed with HTML? (2013)
3. What is PHP? How is it related with HTML? (2017,2011)
4. What are the basic requirements to running PHP program? (2014)
5. How array is declared in PHP? Also explain various types of array with proper example. (2017,2011)
6. What are the techniques used for form validation and verification in PHP? (2017)
7. Write short notes : (i) Session; (ii) PHP variable. (2017)
8. write down the different types of super global array with their functionality. (2013)

or, describe following global variable in PHP:-

- (i)S-GET
- (ii)S-POST
- (iii)S-REQUEST (2014)

9. explain PHP my SQL connectivity and basic connection function with example. (2013,2010)
10. explain "altering table","inserting data in table" with example. (2013)
11. what is season variable? How do you create a season in PHP? (2013)
12. what are the use of session and cookie variables in state management? (2015,2011)
13. Explain the advantages and disadvantages of MySQL. (2017,2014,2011)
14. Write down the procedure of creating database and table in MySQL. (2017, 2015,2014, 2010)

15. Discuss the steps to establish a connection from PHP to MySQL with sample code. (2017,2014,2012,2011)
Or, what is the standard PHP function for connecting to a my SQL database?
Explain the process of using my SQL with PHP using sample code. (2015)
16. what are the features that made my SQL a popular database management system for web service? (2015)
17. What is the PHP operator? Describe different type of PHP operator with example. (2010)
18. what is error handler? What is the purpose of error handler? (2012)
19. Both SELECT DISTINCT and GROUP BY causes the display to show only one output raw for each value in a column, even if multiple rows contain that value. What are the main differences between SELECTDISTINCT and GROUPBY? (2015)

CHAPTER 1

INTRODUCTION TO WEB APPLICATION

1. Define

- v) Webpage (2016,2013)
- vi) Website (2016,2013)
- vii) Web server(2016)
- viii) Web browser(2016)

Solution:

i) Webpage

A web page or webpage is a document commonly written in HTML (Hypertext Markup Language) that is accessible through the Internet or other networks using an Internet browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files. The page you are reading now is an example of a web page.

ii) Website

A site or website is a central location of various web pages that are all related and can be accessed by visiting the home page of the website using a browser.

For example, the google website address URL (Uniform Resource Locator) is <https://www.google.com>.

iii) Web server

A web server is a system that delivers content or services to end users over the internet. A web server consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication.

A web server is also known as an internet server

iv) Web browser

web browser, a browser is a software application used to locate, retrieve and display content on the World Wide Web, including webpages, images, video and other files. As a client/server model, the browser is the client run on a computer or mobile device that contacts the Web server and requests information. The web server sends the information back to the browser which displays the results on the Internet-enabled device that supports a browser.

2. Explain the basic structure of website (2017, 2015, 2011)

Solution:

A web page is structured as follows.

The Doctype

The first item to appear in the source code of a web page is the doctype declaration. This provides the web browser (or other user agent) with information about the type of markup language in which the page is written, which may or may not affect the way the browser renders the content. It may look a little scary at first glance, but the good news is that most WYSIWYG web editors will create the doctype for you automatically after you've selected from a dialog the type of document you're creating. If you aren't using a WYSIWYG web editing package, you can refer to the list of doctypes contained in this reference and copy the one you want to use. The doctype looks like this (as seen in the context of a very simple HTML 4.01 page without any content):

```
<!DOCTYPE html><html><head><title>Page  
title</title></head><body></body></html>
```

In the example above, the doctype relates to HTML 4.01 Strict. In this reference, you'll see examples of HTML 4.01 and also XHTML 1.0 and 1.1, identified as such. While many of the elements and attributes may have the same names, there are some distinct syntactic differences between the various versions of HTML and XHTML. You can find out more about this in the sections entitled HTML Versus XHTML and HTML and XHTML Syntax.

The Document Tree

A web page could be considered as a document tree that can contain any number of branches. There are rules as to what items each branch can contain (and these are detailed in each element's reference in the "Contains" and "Contained by" sections). To understand the concept of a document tree, it's useful to consider a simple web page with typical content features alongside its tree view, as shown in Figure 1. Figure 1. The document tree of a simple web page

```

<html>
  <head>
    <meta content="text/html; charset=UTF-8" http-equiv="Content-Type"/>
    <title>My lovely web page </title>
  </head>
  <body>
    <h1>This is my lovely web page!</h1>
    <p>
      It has lots of lovely content. It has some
      emphasized text <em>I will
      and look at that, a blockquote:</em>
    </p>
    <blockquote>
      <p>You feels, I will destroy you
      all!</p>
      <h2>And here's a subheading!</h2>
      <p>That about covers it, I think we
      done.
    </blockquote>
  </body>
</html>

```

This is my lovely web page

It has lots of lovely content. It has some emphasized text and look at this, a blockquote:

You feels, I will destroy you all!

And here's a subheading

That about covers it, I think

If we look at this comparison, we can see that the `html` element in fact contains two elements: `head` and `body`. `head` has two subbranches—a metaelement and a `title`. The `body` element contains a number of headings, paragraphs, and a block quote.

Note that there's some symmetry in the way the tags are opened and closed. For example, the paragraph that reads, “It has lots of lovely content ...” contains three text nodes, the second of which is wrapped in an `em` element (for emphasis). The paragraph is closed after the content has ended, and before the next element in the tree begins (in this case, it's a `blockquote`); placing the closing `</p>` after the `blockquote` would break the tree's structure.

`html`

Immediately after the doctype comes the `htmlelement`—this is the root element of the document tree and everything that follows is a descendant of that root element.

If the root element exists within the context of a document that's identified by its doctype as XHTML, then the `html` element also requires an `xmlns` (XML Namespace) attribute (this isn't needed for HTML documents):

The `html` element breaks the document into two main sections: the `head` and the `body`.

`head`

The `head` element contains metadata—information that describes the document itself, or associates it with related resources, such as scripts and style sheets.

The simple example below contains the compulsory `title` element, which represents the document's title or name—essentially, it identifies what this document is. The content inside the `title` may be used to provide a heading that appears in the browser's title bar, and when the user saves the page as a favorite. It's also a very

important piece of information in terms of providing a meaningful summary of the page for the search engines, which display the **title** content in the search results.

Here's the **title** in action:

In addition to the **title** element, the **head** may also contain:

- **base**
defines baseURLs for links or resources on the page, and target windows in which to open linked content
- **link**
refers to a source of some kind, most often to a style sheet that provides instructions about how to style the various elements on the webpage
- **meta**
provides additional information about the page; for example, which character encoding the page uses, a summary of the page's content, instructions to search engines about whether or not to index content, and soon
- **object**
represents a generic, multipurpose container for a media object
- **script**
used either to embed or refer to an external script
- **style**
provides an area for defining embedded (page-specific) CSS styles

All of these elements are optional and can appear in any order within the **head**. Note that none of the elements listed here actually appear on the rendered page, but they are used to affect the content on the page, all of which is defined inside the **body** element.

body

This is where the bulk of the page is contained. Everything that you can see in the browser window (or viewport) is contained inside this element, including paragraphs, lists, links, images, tables, and more. The **body** element has some unique attributes of its own, all of which are now deprecated, but aside from that, there's little to say about this element. How the page looks will depend entirely upon the content that you decide to fill it with; refer to the alphabetical listing of all HTML elements to ascertain what these contents might be.

3. Which components are needed to create a fully dynamic web page?(2015)

or, write down the basic concepts of making dynamic website. (2012)

Solution: The four components required to create a fully dynamic web page are

- 1) a web server
(such as Apache),
- 2) a server-side scripting language (PHP),
- 3) a database (MySQL), and
- 4) a client-side scripting language (JavaScript).

Web Server

A web server is computing system designed for exchanging information from server to browser or other client making http (the basic network protocol used to distribute information on the World Wide Web) request. Web server is consist of hardware computer, an operating system and various supporting applications to process network protocal requests. Most commonly Apache is used are server application system on top of Linux or windows server operating system.

Server-side scripting is a technique used in web development which involves employing scripts on a web server which produce a response customized for each user's (client's) request to the website. The alternative is for the web server itself to deliver a static web page.

Database

Database are information house for any dynamic website, it is used to store data in different tables and extract it dynamically on demand. Database is designed to keep all relevant information and website content in different tables and with the help of database query language we can read, insert, or edit data very easily and efficiently. Most commonly MySql is used as database system as its free and widely tested and accepted. Even your favourite social network Facebook uses MySql to store and display data from all over the world.

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.

The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of **security risks** they may switch the scripting facility off.

When this is the case a message usually pops up to alert the user when script is attempting to run.

4. explain the document tree of the webpage. (2012)

Solution:

The HTML Document Tree

Each HTML document can actually be referred to as a document tree. We describe the elements in the tree like we would describe a family tree. There are ancestors, descendants, parents, children and siblings.

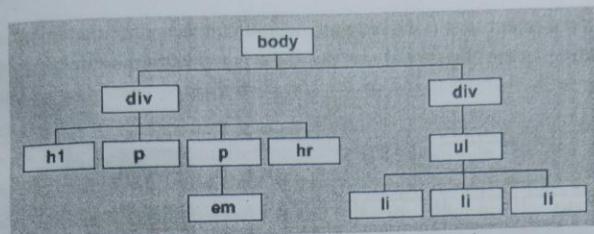
It is important to understand the document tree because CSS selectors use the document tree.

Use the sample HTML document below for these examples. The <head> section of the document is omitted for brevity.

```
<body>
  <div id="content">
    <h1>Heading here</h1>
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Lorem ipsum dolor <em>sit</em> amet.</p>
    <hr>
  </div>

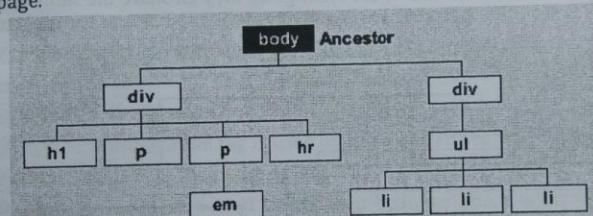
  <div id="nav">
    <ul>
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
    </ul>
  </div>
</body>
```

A diagram of the above HTML document tree would look like this.

**Ancestor**

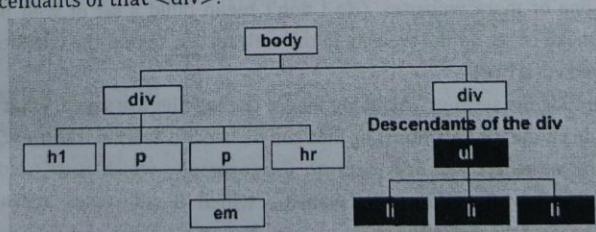
An ancestor refers to any element that is connected but further up the document tree - no matter how many levels higher.

In the diagram below, the <body> element is the ancestor of all other elements on the page.

**Descendant**

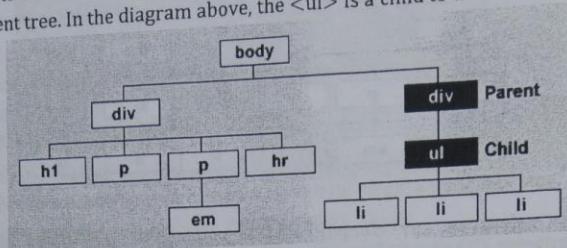
A descendant refers to any element that is connected but lower down the document tree - no matter how many levels lower.

In the diagram below, all elements that are connected below the <div> element are descendants of that <div>.

**Parent and Child**

A parent is an element that is directly above and connected to an element in the document tree. In the diagram below, the <div> is a parent to the .

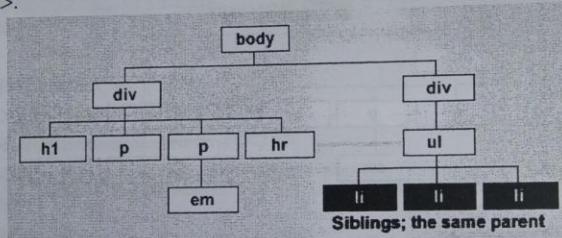
A child is an element that is directly below and connected to an element in the document tree. In the diagram above, the `` is a child to the `<div>`.



Sibling

A sibling is an element that shares the same parent with another element.

In the diagram below, the ``'s are siblings as they all share the same parent - the ``.



5. Describe the common feature of web server. (2017,2014, 2011)

Solution: web servers have features that allow you to do the following:

- Create one or more websites. (No I don't mean build a set of web pages. What I mean is, set up the website in the web server, so that the website can be viewed via HTTP)
- Configure log file settings, including where the log files are saved, what data to include on the log files etc. (Log files can be used to analyse traffic etc)
- Configure website/directory security. For example, which user accounts are/aren't allowed to view the website, which IP addresses are/aren't allowed to view the website etc.
- Create an FTP site. An FTP site allows users to transfer files to and from the site.
- Create virtual directories, and map them to physical directories

- Configure/nominate custom error pages. This allows you to build and display user friendly error messages on your website. For example, you can specify which page is displayed when a user tries to access a page that doesn't exist (i.e. a 404 error).
- Specify default documents. Default documents are those that are displayed when no file name is specified. For example, if you open `http://localhost`, which file should be displayed? This is typically `index.html` or similar but it doesn't need to be. You could nominate `index.cfm` if your website is using ColdFusion. You could also nominate a 2nd choice (in case there is no `index.cfm` file), and a 3rd choice, and so on.

6. What are the important of `!doc type` and `<meta>` tag in webpage. (2017)

or, explain the necessity of `!doc type` in web page design. (2011)

or, Explain `<meta>` tag with example. (2011)

Solution:

The `<!DOCTYPE>` Declaration

The `<!DOCTYPE>` declaration tag is used by the web browser to understand the version of the HTML used in the document. The `<!DOCTYPE>` declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

Current version of HTML is 5 and it makes use of the following declaration –

`<!DOCTYPE html>`

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing `<!DOCTYPE...>` tag along with other HTML tags.

In HTML 4.01, the `<!DOCTYPE>` declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.

HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

Metadata is data (information) about data.

The `<meta>` tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

HTML5 introduced a method to let web designers take control over the viewport (the user's visible area of a web page), through the `<meta>` tag

A meta tag can have following attributes in addition to core attributes –

Sr.No	Attribute & Description
1	Name Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
2	content Specifies the property's value.
3	scheme Specifies a scheme to interpret the property's value (as declared in the content attribute).
4	http-equiv Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

You can use `<meta>` tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```

<!DOCTYPE html>
<html>

    <head>
        <title>Meta Tags Example</title>
        <meta name = "keywords" content = "HTML, Meta Tags, Metadata" />
    </head>

    <body>
        <p>Hello HTML5!</p>
    </body>

</html>

```

This will produce the following result –

Hello HTML5!

7. Distinguish between HTTP & HTTPS(2016,2014)

Solution:

BASIS FOR COMPARISON	HTTP	HTTPS
DEFINITION	HTTP (Hypertext Transfer Protocol) is the base of the data communication for the web this is how the internet works when it comes to delivering the web pages.	HTTPS (Hypertext Transfer Protocol Secure) is nothing but the HTTP working in tandem with SSL (Secure Socket Layer) that is the "S" in HTTPS.
Prefix Used	Url begins with "http://"	Url begins with "https://"
Security	Unsecured.	Secured.
Operated On	Application layer	Transport layer.
Encryption	No encryption is there	Encryption is used.
Certificate	Not required.	Necessary
Port Used	Port number 80 is used for communication.	Port number 443 is used for communication.
Characteristics	It is subject to man-in-the-	It is designed to resist man-

	middle and eavesdropping attacks.	in-the-middle and eavesdropping attacks and is considered secure against such attacks.
Example	Websites like internet forums, educational sites.	Websites like Banking Websites, Payment gateway, Shopping Websites, etc.

8. what is the difference among WAMP,MAMP and LAMP? (2015)

Solution: The primary difference between WAMP, LAMP, MAMP and XAMPP is of the operating system. WAMP is used for Windows, LAMP for Linux, MAMP for Mac and XAMPP can be used for any operating system.

WAMP

- ✓ WAMP is used for Windows operating system only.
- ✓ Its full form is Windows, Apache, MySQL and PHP
- ✓ It is an open source platform.
- ✓ It uses the Apache web server.
- ✓ Relational database management system for WAMP is MySQL.
- ✓ PHP (Hypertext Preprocessor) is the object-oriented scripting language.
- ✓ WAMP is very easy to setup and configure.

LAMP

- ✓ LAMP is used for Linux operating system only.
- ✓ Its full form is Linux, Apache, MySQL and PHP
- ✓ It is also an open source platform.
- ✓ It uses the Apache web server.
- ✓ Relational database management system for LAMP is MySQL.
- ✓ PHP (Hypertext Preprocessor) is the object-oriented scripting language.

MAMP

- ✓ MAMP is used for Mac operating system only.
- ✓ Its full form is MAC, Apache, MySQL and PHP
- ✓ It is also an open source platform.
- ✓ It uses the Apache web server.
- ✓ Relational database management system for LAMP is MySQL.

- ✓ PHP (Hypertext Preprocessor) is the object-oriented scripting language.

XAMPP

- ✓ XAMPP can be used for any operating system.
- ✓ Its full form is x-os, Apache, MariaDB, PHP, and Perl.
- ✓ X-os implies that it is cross platform application and can be used for any operating system.
- ✓ It is free, open source, cross-platform web server solution pack and is developed by Apache Friends.
- ✓ XAMPP comes with extra features like supporting of perl, filezilla, mercury mail and some other scripts.
- ✓ It is simple, lightweight and easy to use for developers in creating local web server.

9. Name the main disadvantages of working on a remote web server. (2015)

Solution: Limitations of Remote Access

At the time of taking advantage of remote access to desktop don't forget that you are going to face two important drawbacks and limitations as well.

- While away from your PC, trying virtual access to its system will delay the whole course. You may feel annoyance on waiting so long for completing the whole process.
- Security is another issue should be kept in view properly. If you leave the computer without turning off the session you would have to suffer.

10. what do you know about agile methods for developing a website? how do agile development methods overcome the problems that arise in conventional development approaches?(2015)

Solution: Agile web development is actually a broad category of methodologies based on the principles outlined in the Manifesto for Agile Software Development, which was compiled by a team of professional developers in 2001. Specific methods such as scrum and XP are considered agile although they existed before the manifesto was written. Through their combined experiences of working with other developers, the authors recognized the value of adaptive planning and collaboration between self-organizing, cross-functional teams. The goal is to allow for flexibility and provide rapid and continuous improvement of

software solutions. Early delivery is also a key goal of agile development, which entails streamlining projects by eliminating time-sucking tasks.

Traditional Web Development vs Agile Web Development

Because the internet has evolved so rapidly in the past few decades, it's easy to forget that the World Wide Web isn't even 30 years old yet. During the infancy stages of web development, designers grappled with the basic task of translating the types of information you'd find in books into a website. They used paper prototypes, wire frames and flow charts to illustrate ideas. Use-case scenarios and focus groups were a primary source of feedback. Back then, CEOs would have their secretaries print out emails and then dictate responses, so there were obviously large gaps of understanding between users, developers and executives.

Developers soon discovered that creating software shouldn't be an entirely sequential process. There are always unexpected bugs as well as new technological hurdles to overcome, and anticipating the demands of users has become a science in itself. Fortunately, now that everyone is so well connected, it's easy to collect user feedback in real-time, so we have a greater understanding of how people interact with web applications.

The increase in internet users has coincided with advancements in content management systems, which has made it possible for anyone to design and edit a basic website with little training or expertise. These trends have given rise to industries that simply didn't exist a few years ago such as e-commerce. Since the process of making websites has become much more streamlined, developers have shifted focus to perfecting their methods to address the growing needs of businesses and consumers.

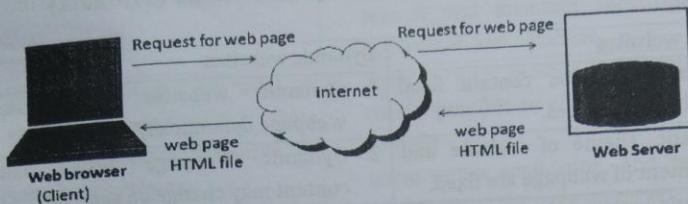
11. Describe basic concept of static and dynamic website. (2013)

Solution:

Static Websites

Static websites are also known as flat or stationary websites. They are loaded on the client's browser as exactly they are stored on the web server. Such websites contain only static information. User can only read the information but can't do any modification or interact with the information.

Static websites are created using only HTML. Static websites are only used when the information is no more required to be modified.



Dynamic Websites

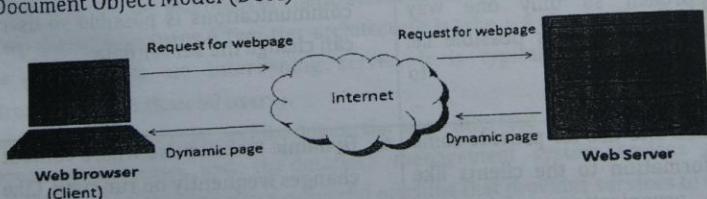
Dynamic websites show different information at different points of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using Ajax technology.

Server-side dynamic web page

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

Client-side dynamic web page

It is processed using client-side scripting such as javascript. And then passed in to Document Object Model (DOM).



12. Distinguish between Static website and Dynamic website. (2016,2011)

Solution:

Static websites	Dynamic websites
1. Static websites contain fixed number of pages.	1. Dynamic websites can create webpage dynamically.
2. Static Theme of website and content of webpage are fixed.	2. Dynamic Webpage design and content may change on run time.
3. Static websites load quickly on client browser because it has only some markup contents.	3. Dynamic sites take some time to load on client browser because it processes the request server side and create contents dynamically.
4. Static sites never use database connectivity.	4. Dynamic sites deal with database and generate the contents dynamically using database queries.
5. Static websites is highly secure than dynamic sites because it behaves as a half duplex approach so only one way communication is possible i.e. server to client.	5. Dynamic sites are less secure because it behaves as full duplex approach so both side communications is possible so user can change the server data
6. Static site use for provide some information to the clients like an organization or institute website.	6. Dynamic website use where content changes frequently on run time. Like a E-commerce site, online examination, etc.
7. Static website directly run on browser and does not require other server application language. Static website can be created from HTML and CSS.	7. Dynamic website run the application on server and the output will display on webpage. So this is require server application language like PHP , The Official Microsoft ASP.NET Site JSP etc.
8. Static sites are easy to develop and a bit experienced people	8. Dynamic websites not easy to develop because require qualify

can develop it.	developers to create it, manage it, test it and maintain security of application and database.
9. In static website if we want to change the page content then we have to upload that page on server many times.	9. Dynamic sites provide the facilities that it possible to change the page content using server application. And need not to upload the page on server.

13. Briefly explain client – server architecture (2016, 2013)

Solution:

Client/Server Architecture The client/server architecture significantly decreased network traffic by providing a query response rather than total file transfer. It allows multi-user updating through a GUI front end to a shared database. Remote Procedure Calls (RPCs) or standard query language (SQL) statements are typically used to communicate between the client and server.

The following are the examples of client/server architectures.

1) **Two tier architectures** A two-tier architecture is where a client talks directly to a server, with no intervening server. It is typically used in small environments(less than 50 users).

In two tier client/server architectures, the user interface is placed at user's desktop environment and the database management system services are usually in a server that is a more powerful machine that provides services to the many clients. Information processing is split between the user system interface environment and the database management server environment.

2) **Three tier architectures** The three tier architecture is introduced to overcome the drawbacks of the two tier architecture. In the three tier architecture, a middleware is used between the user system interface client environment and the database management server environment.

These middleware are implemented in a variety of ways such as transaction processing monitors, message servers or application servers. The middleware perform the function of queuing, application execution and database staging. In

addition the middleware adds scheduling and prioritization for work in progress.

The three tier client/server architecture is used to improve performance for large number of users and also improves flexibility when compared to the two tier approach.

The basic characteristics of client/server architectures are:

- 1) Combination of a client or front-end portion that interacts with the user, and a server or back-end portion that interacts with the shared resource. The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The server process acts as a software engine that manages shared resources such as databases, printers, modems, or high powered processors.
- 2) The front-end task and back-end task have fundamentally different requirements for computing resources such as processor speeds, memory, disk speeds and capacities, and input/output devices.
- 3) The environment is typically heterogeneous and multivendor. The hardware platform and operating system of client and server are not usually the same. Client and server processes communicate through a well-defined set of standard application program interfaces (API's) and RPC's.
- 4) An important characteristic of client-server systems is scalability. They can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multi servers.

14. Write down the problems created by Hoakers, Spyware , Virus and Pop-ups.
(2016,2013)

Solution:

Spyware is the term given to a category of software which aims to steal personal or organizational information. It is done by performing a set of operations without appropriate user permissions, sometimes even covertly. General actions a spyware performs include advertising, collection of personal information and changing user configuration settings of the computer.

15. write down the function of w3 system. (2013)

Solution: World-Wide Web (also called WWW or W3) is a hypertext-based information system. Any word in a hypertext document can be specified as a pointer to a different hypertext document where more information pertaining to that word can be found. The reader can open the second document by selecting the word (using different methods depending on the interface; in a mouse based system, a user would probably place the mouse over the word and click the mouse button); only the part of the linked document which contains relevant information will be displayed.

The second document may itself contain links to further documents. The reader need not know where the referenced documents are, because they will be obtained and presented as they are needed.

World-Wide Web uses hypertext over the Internet: the linked documents may be located at different Internet sites. WWW can handle different text formats and different methods of organizing information.

The World-Wide Web also provides access to many of the other tools described in this guide, and is becoming widely used as the major means of access to Internet resources.

Special index documents have been created in the WWW information space and these can be searched for given keyword(s). The result is a new document which contains links to documents selected from the index.

If you were reading this document on a hypertext system, instead of this all too short explanation about hypertext, you would have a selectable pointer to a complete hypertext information web with examples and more pointers to other definitions. For instance, in the first document you might read:

Web Engineering -26

The Worldwide Web (W3) is the universe of network-accessible information, an embodiment of human knowledge. It is an initiative started at "CERN", now with many participants. It has a body of software, and a set of protocols and conventions. W3 uses "hypertext" and multimedia techniques to make the web easy for anyone to roam browse, and contribute to.

16. name one popular website and list out the technical features that made this site special. (2015)

Solution:

W3Schools is a web developers site, with tutorials and references on web development languages such as HTML, CSS, JavaScript, PHP, SQL, Python, jQuery, Java, W3.CSS, and Bootstrap, covering most aspects of web programming.

The site derives its name from the World Wide Web (W3), but is not affiliated with the W3C.

W3Schools was originally created in 1998 by Refsnes Data, a Norwegian software development and consulting company.

Easy Learning

W3Schools has focus on simplicity.

W3Schools practice easy and straight-forward learning.

W3Schools uses simple code explanations with simple illustrations of how to use it.

W3Schools' tutorials start from basic level and move all the way up to professional references.

17. write down the concept of web application (2014)

Solution: Web application is the combination of server side scripts and client side scripts In server side scripting to handle the storage information but in client side scripts to presents information to user. It allows users to connect with the organization through online shopping, share information and more.

18. Explain the symptoms of virus attack in the computer and how to prevent computer from virus. (2014)

Solution: What are the signs of a computer virus

A computer virus attack can produce a variety of symptoms. Here are some of them:

- Frequent pop-up windows. Pop-ups might encourage you to visit unusual sites. Or they might prod you to download antivirus or other software programs.
- Changes to your homepage. Your usual homepage may change to another website, for instance. Plus, you may be unable to reset it.
- Mass emails being sent from your email account. A criminal may take control of your account or send emails in your name from another infected computer.
- Frequent crashes. A virus can inflict major damage on your hard drive. This may cause your device to freeze or crash. It may also prevent your device from coming back on.
- Unusually slow computer performance. A sudden change of processing speed could signal that your computer has a virus.
- Unknown programs that start up when you turn on your computer. You may become aware of the unfamiliar program when you start your computer. Or you might notice it by checking your computer's list of active applications.
- Unusual activities like password changes. This could prevent you from logging into your computer.

How to Prevent Viruses

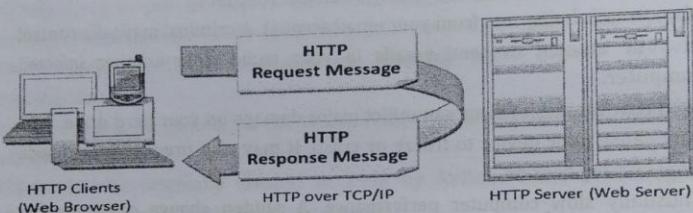
In order to prevent viruses from infecting your computer, you must follow certain best practices such as:

- Do not insert removable devices/USB devices without scanning using a virus scanner
- Abstain from downloading software from untrusted websites
- Don't open email attachments without verifying the sender id, content, and purpose
- Use a premium antivirus software rather than free software as those may not be capable of providing 100% protection.

19. explain how HTTP protocol works with http request and response. (2012)

Solution:

The operation of Hypertext Transfer Protocol (HTTP) involves the communication between a Hypertext Transfer Protocol (HTTP) client application (Usually web browser) and a Hypertext Transfer Protocol (HTTP) server application (Web servers like IIS). Hypertext Transfer Protocol (HTTP) uses Transmission Control Protocol (TCP) as the Transport Layer Protocol at Well Known port number 80. Once the TCP connection is established, the two steps in Hypertext Transfer Protocol (HTTP) communication are



- 1) **HTTP Client Request:** Hypertext Transfer Protocol (HTTP) client sends an Hypertext Transfer Protocol (HTTP) Request to the Hypertext Transfer Protocol (HTTP) Server according to the HTTP standard, specifying the information the client like to retrieve from the Hypertext Transfer Protocol (HTTP) Server.
- 2) **HTTP Server Response:** Once the Hypertext Transfer Protocol (HTTP) Request arrived at the Hypertext Transfer Protocol (HTTP) server, it will process the request and creates an Hypertext Transfer Protocol (HTTP) Response message. The Hypertext Transfer Protocol (HTTP) response message may contain the resource the Hypertext Transfer Protocol (HTTP) Client requested or information why the Hypertext Transfer Protocol (HTTP) request failed.

CHAPTER 2

HTML AND INTRODUCTION TO HTML

1. What is HTML attributes ? Explain the HTML elements . (2017,2011)

Answer:

HTML attributes

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts – a name and a value

- The name is the property you want to set. For example, the paragraph `<p>` element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.
- The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Align Attribute Example</title>
    </head>

    <body>
        <p align = "left">This is left aligned</p>
        <p align = "center">This is center aligned</p>
        <p align = "right">This is right aligned</p>
    </body>
</html>
```

HTML elements

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

Start Tag	Content	End Tag
<p>	This is paragraph content.	</p>
<h1>	This is heading content.	</h1>
<div>	This is division content.	</div>

So here <p>...</p> is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which don't need to be closed, such as <img.../>, <hr /> and
 elements. These are known as void elements. HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

2. Write a code segment that will create a password filed in an HTML form.
(2017,2014)

Answer:

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to password.

Example

Here is a basic example of a single-line password input used to take user password –

```
<!DOCTYPE html>
<html>
    <head>
        <title>Password Input Control</title>
    </head>
    <body>
        <input type="password" name="password" value="Hello World!">
    </body>
</html>
```

```

<body>
    <form>
        User ID : <input type = "text" name = "user_id" />
        <br>
        Password: <input type = "password" name = "password"
    />
    </form>
</body>

</html>

```

This will produce the following result –

User ID :	<input type="text"/>
Password:	<input type="password"/>

3. Write down the difference between GET and POST method. (2017)

Answer:

BASIS FOR COMPARISON	GET	POST
Parameters are placed inside	URI	Body
Purpose	Retrieval of documents	Updation of data
Query results	Capable of being bookmarked.	Cannot be bookmarked.
Security	Vulnerable, as present in plaintext	Safer than GET method
Form data type constraints	Only ASCII characters are permitted.	No constraints, even binary data is permitted.
Form data length	Should be kept as minimum as possible.	Could lie in any range.
Visibility	Can be seen by anyone.	Doesn't display variables in URL.
Variable size	Up to 2000 character.	Up to 8 Mb

Caching	Method data can be cached.	Does not cache the data.
---------	----------------------------	--------------------------

4. What are the advantages of HTML 5 over HTML4? Discuss any two Tag names of HTML 5 with example. (2016,2012)

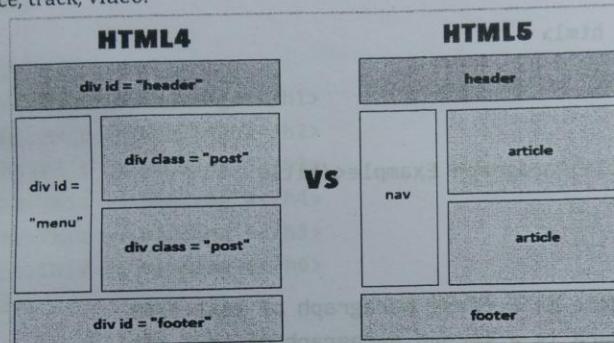
answer:

What are the main differences between HTML and HTML5

The one consistent thing about the field of information technology is that periodic updates/changes are inevitable. No language is capable of avoiding upgrades and/or new releases. HTML is no exception. HTML5 was released with the primary objective of improving the World Wide Web experience for the developers and the end users. As already mentioned, the biggest advantage that HTML5 has over its unnumbered predecessor is that it has high-level audio and video support which was not a part of the version specifications in previous HTMLs. Other differences between HTML and HTML5:

- SVG, canvas and other virtual vector graphics are supported in HTML5, whereas in HTML, using vector graphics was only possible by using it in conjunction with different technologies like Flash, VML, and Silver-light, etc.
- HTML5 uses web SQL databases, application cache for temporary storing data, meanwhile, in HTML, only browser cache could be utilized for this purpose.
- Another difference between HTML and HTML5 worth mentioning is that the former doesn't allow JavaScript to run within the web browser (it instead runs in the browser interface thread) whereas the latter provides full support for JavaScript to run in the background (This is possible courtesy to the JS web worker API of HTML5).
- HTML5 is not based on SGML, and that allows it to have improved parsing rules which provide enhanced compatibility.
- In HTML5, inline MathML and SVG can be used in text whereas this wasn't possible in HTML.
- Some of the deprecated elements that have now been dropped completely are: isindex, noframes, acronym, applet, basefont, dir, font, frame, frameset, big, center, strike, tt.

- HTML5 supports new kinds of form controls, for example: dates and times, email, number, range, tel, url, search etc.
- There are many new elements introduced in HTML. Some of the most important ones are: summary, time, aside, audio, command, data, datalist, details, embed, wbr, figcaption, figure, footer, header, article, hgroup, bdi, canvas, keygen, mark, meter, nav, output, progress, rp, rt, ruby, section, source, track, video.



New Semantic/Structural Elements

HTML5 offers new elements for better document structure:

Tag	Description
<u><article></u>	Defines an article in a document
<u><aside></u>	Defines content aside from the page content

5. discuss paragraph, heading and lists tag with example. 2012

answer:

Paragraph Tag

The `<p>` tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening `<p>` and a closing `</p>` tag as shown below in the example –

Example

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Paragraph Example</title>
</head>

<body>
  <p>Here is a first paragraph of text.</p>
  <p>Here is a second paragraph of text.</p>
  <p>Here is a third paragraph of text.</p>
</body>
```

This will produce the following result –

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. While displaying any heading, browser adds one line before and one line after that heading.

Example

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

    <body>
        <h1>This is heading 1</h1>
        <h2>This is heading 2</h2>
        <h3>This is heading 3</h3>
        <h4>This is heading 4</h4>
        <h5>This is heading 5</h5>
        <h6>This is heading 6</h6>
    </body>
</html>
```

This will produce the following result –

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

- – An unordered list. This will list items using plain bullets.
- – An ordered list. This will use different schemes of numbers to list your items.
- <dl> – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML tag. Each item in the list is marked with a bullet.

Example

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

    <body>
        <ul>
            <li>Beetroot</li>
            <li>Ginger</li>
            <li>Potato</li>
            <li>Radish</li>
        </ul>
    </body>
</html>
```

This will produce the following result –

- Beetroot
- Ginger
- Potato
- Radish

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using `` tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with ``.

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Ordered List</title>
  </head>
  <body>
    <ol>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>
</html>
```

This will produce the following result –

1. Beetroot
2. Ginger
3. Potato
4. Radish

HTML Definition Lists

HTML and XHTML supports a list style which is called definition lists where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> – Defines the start of the list
- <dt> – A term
- <dd> – Term definition
- </dl> – Defines the end of the list

Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>HTML Definition List</title>
    </head>

    <body>
        <dl>
            <dt><b>HTML</b></dt>
            <dd>This stands for Hyper Text Markup Language</dd>
            <dt><b>HTTP</b></dt>
            <dd>This stands for Hyper Text Transfer Protocol</dd>
        </dl>
    </body>

</html>
```

This will produce the following result -

HTML
This stands for Hyper Text Markup Language
HTTP
This stands for Hyper Text Transfer Protocol

6. explain the following tag with example:- (2012)

- (i) <table></table>
- (ii) <img..>.....
- (iii)< a href>.....

Answer:

- (i) <table></table>

A simple HTML table, containing two columns and two rows:

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

The <table> tag defines an HTML table.

An HTML table consists of the <table> element and one or more <tr>, <th>, and <td> elements.

The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.

- (ii) <img..>.....

How to insert an image:

```

```

The `` tag defines an image in an HTML page. The `` tag has two required attributes: `src` and `alt`.

(iii) `<a href>.....`

A link to W3Schools.com:

```
<a href="https://www.w3schools.com">Visit W3Schools.com!</a>
```

The `<a>` tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

7. what is DOCTYPE in HTML? Why it is used? (2012)

answer:

The `<!DOCTYPE>` declaration must be the very first thing in your HTML document, before the `<html>` tag.

The `<!DOCTYPE>` declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

In HTML 4.01, the `<!DOCTYPE>` declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly.

HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>
```

```

<body>
The content of the document.....
</body>

</html>

```

8. what are the differences between HTML and XHTML?

Answer:

HTML versus XHTML comparison chart		
	HTML	XHTML
Introduction (from Wikipedia)	HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.	XHTML (Extensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.
Filename extension	.html, .htm	.xhtml, .xht, .xml, .html, .htm
Internet media type	text/html	application/xhtml+xml
Developed by	W3C & WHATWG	World Wide Web Consortium
Type of format	Document file format	Markup language
Extended from	SGML	XML, HTML
Stands for	HyperText Markup Language	Extensible HyperText Markup Language
Application	Application of Standard Generalized Markup Language (SGML).	Application of XML

Function	Web pages are written in HTML.	Extended version of HTML that is stricter and XML-based.
Nature	Flexible framework requiring lenient HTML specific parser.	Restrictive subset of XML and needs to be parsed with standard XML parsers.
Origin	Proposed by Tim Berners-Lee in 1987.	World Wide Web Consortium Recommendation in 2000.
Versions	HTML 2, HTML 3.2, HTML 4.0, HTML 5.	XHTML 1, XHTML 1.1, XHTML 2, XHTML 5.

9. Discuss different types of links in HTML. (2014,2012,2011)

Answer:

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

HTML Links - Syntax

In HTML, links are defined with the `<a>` tag:

`link text`

Example

`Visit our HTML tutorial`

Local Links

The example above used an absolute URL (a full web address).

A local link (link to the same web site) is specified with a relative URL (without `https://www....`).

Example

`HTML Images`

HTML Links - The target Attribute

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- `_blank` - Opens the linked document in a new window or tab
- `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)

- _parent - Opens the linked document in the parent frame
- _top - Opens the linked document in the full body of the window
- framename - Opens the linked document in a named frame

This example will open the linked document in a new browser window/tab:

Example

```
<a href="https://www.w3schools.com/" target="_blank">Visit  
W3Schools!</a>
```

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">  
    
</a>
```

HTML Links - Create a Bookmark

HTML bookmarks are used to allow readers to jump to specific parts of a Web page.

Bookmarks can be useful if your webpage is very long.

To make a bookmark, you must first create the bookmark, and then add a link to it.

When the link is clicked, the page will scroll to the location with the bookmark.

Example

First, create a bookmark with the id attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

```
<a href="#C4">Jump to Chapter 4</a>
```

External Paths

External pages can be referenced with a full URL or with a path relative to the current web page.

This example uses a full URL to link to a web page:

Example

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

10. what are the necessities of using HTML form? Explain HTML form tag with its different attributes. (2016,2014,2011)

or, With example, describe different form input element (text field, text area, check box, radio, dropdown list, submit button, password button). (2010)

answer:

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, dropdown menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax –

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

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form –

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms –

- Single-line text input controls – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.
- Password input controls – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input> tag.
- Multi-line text input controls – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

```
<form>
    First name: <input type = "text" name = "first_name">
/>
    <br>
    Last name: <input type = "text" name = "last_name" />
</form>

First name: 
Last name: 
```

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox..

```
<form>
    <input type = "checkbox" name = "maths" value = "on">
Maths
    <input type = "checkbox" name = "physics" value =
"on"> Physics
</form>
```

Maths Physics

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to radio.

```
<form>
    <input type = "radio" name = "subject" value =
"maths"> Maths
    <input type = "radio" name = "subject" value =
"physics"> Physics
</form>
```

Maths Physics

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

```
<select name = "dropdown">
    <option value = "Maths" selected>Maths</option>
    <option value = "Physics">Physics</option>
</select>
```

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the `<input>` element but type attribute is set to file.

```
<form>
    <input type = "file" name = "fileupload" accept =
"image/*" />
</form>
```

Choose File No file chosen

11. describe basic tag of HTML. (2014)

answer:

basic tag of HTML

HTML Documents

All HTML documents must start with a document type declaration: <!DOCTYPE html>.

The HTML document itself begins with <html> and ends with </html>.

The visible part of the HTML document is between <body> and </body>.

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important

heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Paragraphs

HTML paragraphs are defined with the <p> tag:

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

HTML Links

HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

HTML Images

HTML images are defined with the `` tag.

The source file (`src`), alternative text (`alt`), width, and height are provided as attributes:

Example

```

```

HTML Buttons

HTML buttons are defined with the `<button>` tag:

Example

```
<button>Click me</button>
```

HTML Lists

HTML lists are defined with the `` (unordered/bullet list) or the `` (ordered/numbered list) tag, followed by `` tags (list items):

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

12. Describe `` tag with its common attributes (align alt, border, sre, width, use map). Give example. (2010)

Answer:

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using `` tag. Following is the simple syntax to use this tag.

```
<img src = "Image URL" ... attributes-list/>
```

The `` tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

HTML Images Syntax

In HTML, images are defined with the `` tag.

The `` tag is empty, it contains attributes only, and does not have a closing tag.

The `src` attribute specifies the URL (web address) of the image:

```

```

The alt Attribute

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

Example

```

```

Image Size - Width and Height

You can use the `style` attribute to specify the width and height of an image.

Example

```

```

Alternatively, you can use the `width` and `height` attributes:

Example

```

```

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

Example

```

```

13. How information is present using table tag and related tags and their attribute? Explain with example. (2010).

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the `<table>` tag in which the `<tr>` tag is used to create table rows and `<td>` tag is used to create data cells. The elements under `<td>` are regular and left aligned by default
able Header, Body, and Footer

Tables can be divided into three portions – a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are –

- `<thead>` – to create a separate table header.
- `<tbody>` – to indicate the main body of the table.
- `<tfoot>` – to create a separate table footer.

A table may contain several `<tbody>` elements to indicate different pages or groups of data. But it is notable that `<thead>` and `<tfoot>` tags should appear before `<tbody>`

Example

```
<!DOCTYPE html>  
<html>
```

```
<head>
    <title>HTML Table</title>
</head>

<body>
    <table border = "1" width = "100%">
        <thead>
            <tr>
                <td colspan = "4">This is the head of the
table</td>
            </tr>
        </thead>

        <tfoot>
            <tr>
                <td colspan = "4">This is the foot of the
table</td>
            </tr>
        </tfoot>

        <tbody>
            <tr>
                <td>Cell 1</td>
                <td>Cell 2</td>
                <td>Cell 3</td>
                <td>Cell 4</td>
            </tr>
        </tbody>
    </table>
</body>

</html>
```

This will produce the following result –

This is the head of the table			
Cell 1	Cell 2	Cell 3	Cell 4
This is the foot of the table			

14. What new input types available at HTML5? Describe 4 new input types with example.(2010)

Answer:

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

New input types that are not supported by older web browsers, will behave as `<input type="text">`.

4 new input types

Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

Example

`<form>`

Select your favorite color:

```
<input type="color" name="favcolor">
</form>
```

Input Type Date

The `<input type="date">` is used for input fields that should contain a date. Depending on browser support, a date picker can show up in the input field.

Example

```
<form>
    Birthday:
    <input type="date" name="bday">
</form>
```

Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

Example

```
<form>
    E-mail:
    <input type="email" name="email">
</form>
```

Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

Example

```
<form>
    Select      a      file: <input type="file" name="myFile">
</form>
```

15. Identify HTML element that you can use inside the `<head></head>` element and describe what they are use for. (2010)

Answer:

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.

HTML metadata is data about the HTML document. Metadata is not displayed. Metadata typically define the document title, character set, styles, links, scripts, and other meta information.

The following tags describe

metadata: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.

The HTML `<title>` Element

The `<title>` element defines the title of the document, and is required in all HTML/XHTML documents.

The `<title>` element:

- defines a title in the browser tab
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine results

A simple HTML document:

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>Page Title</title>
```

```
    </head>
```

```
    <body>
```

The content of the document.....

```
    </body>
```

```
</html>
```

The HTML `<style>` Element

The `<style>` element is used to define style information for a single HTML page:

Example

```
<style>
    body {background-color: powderblue;}
    h1 {color: red;}
    p {color: blue;}
</style>
```

The HTML `<link>` Element

The `<link>` element is used to link to external style sheets:

Example

```
<link rel="stylesheet" href="mystyle.css">
```

The HTML `<meta>` Element

The `<meta>` element is used to specify which character set is used, page description, keywords, author, and other metadata.

Metadata is used by browsers (how to display content), by search engines (keywords), and other web services.

Define the character set used:

```
<meta charset="UTF-8">
```

Define a description of your web page:

```
<meta name="description" content="Free Web tutorials">
```

16. How we make web page dynamic and interactive? Discuss at least 3 techniques to make web page dynamic and interactive. (2010)

answer:

Dynamic web pages are those which display different content on the same page depending on the conditions pre defined.

For example:

News site: Keeps on changing every hour at the least.

E-commerce Site: Keeps on changing each day at minimum.

Social Site: Displays different contents to each individuals depending upon their preferences, friends circle, etc.

YouTube: Displays different video suggestions to each individuals.

Web Interactivity Today

The last decade or so has seen a focus on web standards — designing and building sites and applications for the web that conform to agreed-upon conventions and philosophies. Along with that focus has come a number of technologies for better handling client-side scripting. At the same time, better server-side tools have emerged, and the two have side of the coin have become more tightly integrated.⁵

HTML 5

HTML — Hypertext Markup Language — is the language of web sites. Originally conceived as a simply a way to prepare and present text documents for the web, it has expanded a great deal.

The latest version of HTML — HTML 5 — includes provisions for embedded media and dynamic, interactive content.

CSS

CSS — Cascading Stylesheets — is the language used for defining how HTML documents should look when displayed in the browser. A CSS file might specify the color of the page background, the font, or the size of various images.

As the language has developed, it has gotten a lot more complex and sophisticated. In addition to document styling, it provides a number of tools for handling interaction. For example, CSS can be used to animate web page elements and move them around the screen, resize them, or otherwise manipulate them.

JavaScript

JavaScript (which has no real connection to the Java language mentioned earlier) is a (mostly) client-side scripting language that allows you to write scripts (computer programs) which run inside the web browser.

Web browsers have had JavaScript capabilities for a long time. But early on, JS was just used for low-level interaction and cheap tricks. The language has evolved into a full-fledged programing language, and can now be used to build fully-featured applications.

JS applications are native to the web browser environment, and are are tightly integrated with HTML documents. These features overcome the limitations on other client-side application platforms.

Server-side scripting

A number of server-side languages have developed and evolved since the bad old days of CGI. Some of them are web-specific. Others are more generalized, but have web-facing components.

- ✓ PHP
- ✓ Python
- ✓ Ruby
- ✓ Perl
- ✓ JavaScript (via Node.js)

AJAX

AJAX isn't really a technology as much as it is an idea — a pattern for designing web applications.

AJAX stands for "Asynchronous JavaScript and XML." It works like this:

- You have a website open on your browser, with a JavaScript application running on the page
- Without reloading the page (that is, asynchronously) the JS app sends a request to the server
- The server sends back data, rather than a full document. The data was originally usually XML (hence the name AJAX), but today it is more likely to be JSON.
- The JS app parses the data and updates the currently displayed page, without the page every reloading.

It is said to be "asynchronous" because new content can be loaded into a page at a different time than the page is loaded into the browser.

AJAX-style interaction drives many of the web applications you probably use all the time — Facebook, Twitter, GMail.

17. Identify HTML element that is use to create divisions or sections within your page. (2010)

Answer:

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The `<div>` element is a block-level element.

Example

```
<div>Hello</div>
<div>World</div>
```

The `<div>` Element

The `<div>` element is often used as a container for other HTML elements.

The `<div>` element has no required attributes, but style, class and id are common.

When used together with CSS, the `<div>` element can be used to style blocks of content:

Example

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>
```

The `` Element

The `` element is often used as a container for some text.

The `` element has no required attributes, but style, class and id are common.

When used together with CSS, the `` element can be used to style parts of the text:

Example

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

Section

The `<section>` tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document. Most browsers will display the `<section>` element with the following default values:

Example

```
section {  
    display: block;  
}
```

18. What are the necessities of HTML form? Explain HTML FORM tag with its different attributes. (2016)

Answer:

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML `<form>` tag is used to create an HTML form and it has following syntax –

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

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes –

Sr.No	Attribute & Description
1	action Backend script ready to process your passed data.
2	method Method to be used to upload data. The most frequently used are GET and POST methods.
3	target Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	enctype You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are — application/x-www-form-urlencoded — This is the standard method most forms use in simple scenarios. multipart/form-data — This is used when you want to upload binary data in the form of files like image, word file etc.

19. Explain the way using HTML code in which data can be presented in a tabular form as given below: (2016)

CSE	
Semester	
	1
	2
	3

Answer:

```

<!DOCTYPE html>
<html>

<head>
    <title>HTML Table</title>
</head>

<body>
```

```
<table border="1" width = "150px">
    <thead>
        <tr>
            <td colspan = "4">CSE</td>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td> Semester </td>
            <td> 1 </td>
        </tr>
        <tr>
            <td> </td>
            <td> 2 </td>
        </tr>
        <tr>
            <td> </td>
            <td> 3 </td>
        </tr></tr>

        </tbody>
    </table>
</body>

</html>
```

CHAPTER 3

BASIC CONCEPTS OF CSS

1. what is CSS? Write down the CSS syntax with example. (2016,2014)

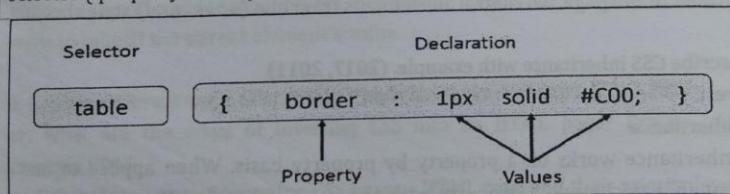
solution:

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

- **Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
- **Value** – Values are assigned to properties. For example, colorproperty can have value either red or #F1F1F1 etc.

You can put CSS Style Rule Syntax as follows –

```
selector { property: value }
```



Example – You can define a table border as follows –

```
table{ border :1px solid #C00; }
```

Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

2. Why should you prefer CSS rather using only HTML in web design? (2017,2014)

Answer:

Here are several reasons why everyone should use CSS in web design:

CSS Provides Efficiency in Design and Updates

With CSS, we are able to create rules, and apply those rules to many elements within the website. This approach offers many advantages when site-wide changes are required by a client. Since the content is completely separated from the design, we can make those changes in our Style Sheet and have it effect every applicable instance.

CSS Use Can Lead To Faster Page Downloads

Since rules are only downloaded once by the browser, then are cached and used for each page load, the use of CSS can lead to lighter page loads, and improved performance. This contributes to lighter server load and lower requirements, which overall saves money for our clients.

CSS is Easy to Work With

Because we are able to keep every visual aspect of the website completely separated from the content, using CSS when designing our websites allows us to quickly create layouts, and troubleshoot any problems. We know that regardless of the page, we control all elements from one (or several) style sheet.

3. Describe CSS inheritance with example. (2017, 2011)

Answer:

CSS Inheritance

CSS inheritance works on a property by property basis. When applied to an element in a document, a property with the value 'inherit' will use the same value as the parent element has for that property.

For example, given this style sheet:

```
.foo {  
background-color: white;  
color: black;  
}  
  
.bar {  
background-color: inherit;  
color: inherit;
```

```
font-weight: normal;  
}
```

And this HTML fragment:

```
<div class="foo">  
  <p class="bar">  
    Hello, world. This is a very short  
    paragraph!  
  </p>  
</div>
```

The background colour of the div element is white, because the background-color property is set to white. The background colour of the paragraph is also white, because the background colour property is set to inherit, and the background colour of the parent element (the div) is set to white.

The inherit value does not require that the parent element have the same property set explicitly; it works from the computed value. In the above example, the color property of the paragraph has a value of "inherit", but the computed value is "black" because it inherits.

This might seem like a lot of typing, but the default value for many properties is already inherit, and for most others (border for instance) you wouldn't usually want to inherit the parent element's value.

4. Explain different ways of inserting a style sheet in a webpage. (2017,2012 or, what are the ways of inserting CSS into an HTML page? Explain with example.2015

or, What three ways of inserting CSS in your HTML page and their priority level?

Explain with example 2010

Answer:

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the `<link>` element. The `<link>` element goes inside the `<head>` section:

Example

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a `.css` extension.

Here is how the "mystyle.css" looks:

```
body {
    background-color: lightblue;
}

h1 {
    color: navy;
    margin-left: 20px;
}
```

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:

Example

```
<head>
<style>
body {
    background-color: linen;
}
```

```
h1 {
    color: maroon;
    margin-left: 40px;
```

```

}
</style>
</head>
```

Inline Styles

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a `<h1>` element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

5. compare different style sheet approaches. 2016,2015,2011

To add CSS styles to your website, you can use three different ways to insert the CSS. You can Use an "External Stylesheet", an "Internal Stylesheet", or in "Inline Style". The benefit for using each depends on what you are doing with the Style.

The following table explains the difference between them.

Different CSS style linking	
Internal Style sheet	An internal stylesheet holds the CSS code for the webpage in the head section of the particular file. This makes it easy to apply styles like classes or id's in order to reuse the code. The downside of using an internal stylesheet is that changes to the internal stylesheet only effect the page the code is inserted into.
External Stylesheet	The External Stylesheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will affect every page in your website. This prevents you from having to make many code changes in each page. This is for "global" site changes.
Inline Styles	The Inline style is specific to the tag itself. The inline style uses the HTML "style" attribute to style a specific tag. This is not recommended, as every CSS change has to be made in every tag that has the inline style applied to it. The Inline style is good for one an individual CSS change that you do not use repeatedly through the site.

6. which CSS approaches is best and why? 2016

answer:

External Style sheet

The External Style sheet is a .css file that you link your website to. This makes it so that whatever you change in the .css sheet, will effect every page in your website. This prevents you from having to make many code changes in each page. This is for "global" site changes.

7. Describe different CSS selectors with example(2017)

Answer:

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The element Selector

The element selector selects elements based on the element name.

You can select all `<p>` elements on a page like this (in this case, all `<p>` elements will be center-aligned, with a red text color):

Example

```
p {  
    text-align: center;  
    color: red;  
}
```

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element should be unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The style rule below will be applied to the HTML element with `id="para1"`:

Example

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

The class Selector

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and center-aligned:

Example

```
.center {  
    text-align: center;  
    color: red;  
}
```

In the example below, only <p> elements with class="center" will be center-aligned:

Example

```
p.center {  
    text-align: center;  
    color: red;  
}
```

HTML elements can also refer to more than one class.

In the example below, the <p> element will be styled according to class="center" and to class="large":

Example

```
<p class="center large">This paragraph refers to two classes.</p>
```

Grouping Selectors

If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}
```

```
h2 {  
    text-align: center;  
    color: red;  
}
```

```
p {  
    text-align: center;  
    color: red;  
}
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

8. explain margin, border and shading with CSS code. 2012

answer:

The margin property defines the space around an HTML element. It is possible to use negative values to overlap content.

The values of the margin property are not inherited by the child elements. Remember that the adjacent vertical margins (top and bottom margins) will collapse into each other so that the distance between the blocks is not the sum of the margins, but only the greater of the two margins or the same size as one margin if both are equal.

We have the following properties to set an element margin.

- ✓ The margin specifies a shorthand property for setting the margin properties in one declaration.
- ✓ The margin-bottom specifies the bottom margin of an element.
- ✓ The margin-top specifies the top margin of an element.
- ✓ The margin-left specifies the left margin of an element.
- ✓ The margin-right specifies the right margin of an element.

Now, we will see how to use these properties with examples.

The Margin Property

The margin property allows you set all of the properties for the four margins in one declaration. Here is the syntax to set margin around a paragraph –

Here is an example –

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

  <body>
    <p style = "margin: 15px; border:1px solid black;">
      all four margins will be 15px
    </p>

    <p style = "margin:10px 2%; border:1px solid black;">
      top and bottom margin will be 10px, left and right margin will be 2%
      of the total width of the document.
    </p>

    <p style = "margin: 10px 2% -10px; border:1px solid black;">
      top margin will be 10px, left and right margin will be 2% of the
      total width of the document, bottom margin will be -10px
    </p>

    <p style = "margin: 10px 2% -10px auto; border:1px solid black;">
      top margin will be 10px, right margin will be 2% of the total
      width of the document, bottom margin will be -10px, left margin
      will be set by the browser
    </p>
  </body>
</html>
```

The border properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change –

- ✓ The border-color specifies the color of a border.
- ✓ The border-style specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- ✓ The border-width specifies the width of a border.

The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties –

- ✓ border-bottom-color changes the color of bottom border.
- ✓ border-top-color changes the color of top border.
- ✓ border-left-color changes the color of left border.
- ✓ border-right-color changes the color of right border.

The border-style Property

The border-style property allows you to select one of the following styles of border –

- ✓ none – No border. (Equivalent of border-width:0;)
- ✓ solid – Border is a single solid line.
- ✓ dotted – Border is a series of dots.
- ✓ dashed – Border is a series of short lines.
- ✓ double – Border is two solid lines.
- ✓ groove – Border looks as though it is carved into the page.
- ✓ ridge – Border looks the opposite of groove.
- ✓ inset – Border makes the box look like it is embedded in the page.
- ✓ outset – Border makes the box look like it is coming out of the canvas.
- ✓ hidden – Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties –

- ✓ border-bottom-style changes the style of bottom border.
- ✓ border-top-style changes the style of top border.
- ✓ border-left-style changes the style of left border.
- ✓ border-right-style changes the style of right border.

This is a border with none width.

This is a solid border.

This is a dashed border.

This is a double border.

This is a groove border.

This is a ridge border.

This is a inset border.

This is a outset border.

This is a hidden border.

This is a border with four different styles.

The border-width Property

The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to thin, medium or thick.

You can individually change the width of the bottom, top, left, and right borders of an element using the following properties –

- ✓ border-bottom-width changes the width of bottom border.
- ✓ border-top-width changes the width of top border.
- ✓ border-left-width changes the width of left border.
- ✓ border-right-width changes the width of right border.

This is a solid border whose width is 4px.

This is a solid border whose width is 4pt.

This is a solid border whose width is thin.

This is a solid border whose width is medium;

This is a solid border whose width is thick.

This is a border with four different width.

divided as follows –

- Text shadow
- Box Shadow

Text shadow

CSS3 supported to add shadow effects to text. Following is the example to add shadow effects to text –

```
<html>
<head>
<style>
h1 {
    text-shadow: 2px 2px;
}
h2 {
    text-shadow: 2px 2px red;
}
h3 {
    text-shadow: 2px 2px 5px red;
}
h4 {
    color: white;
    text-shadow: 2px 2px 4px #000000;
}
h5 {
    text-shadow: 0 0 3px #FF0000;
}
h6 {
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
p {
    color: white;
    text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
}
</style>
</head>
```

```
<body>
  <h1> EXAMPLE </h1>
  <h2> EXAMPLE </h2>
  <h3> EXAMPLE </h3>
  <h4> EXAMPLE </h4>
  <h5> EXAMPLE </h5>
  <h6> EXAMPLE </h6>
  <p> EXAMPLE </p>
</body>
</html>
```

It will produce the following result –

EXAMPLE

EXAMPLE

EXAMPLE

EXAMPLE

EXAMPLE

EXAMPLE

EXAMPLE

box shadow

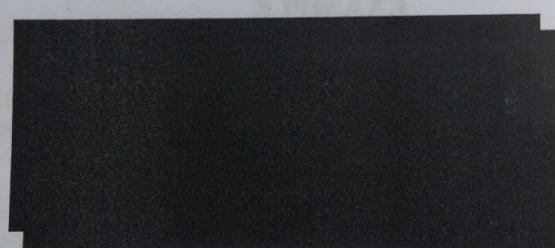
Used to add shadow effects to elements. Following is the example to add shadow effects to element.

Live Demo

```
<html>
  <head>
    <style>
      div {
```

```
width: 300px;  
height: 100px;  
padding: 15px;  
background-color: red;  
box-shadow: 10px 10px;  
}  
</style>  
</head>  
  
<body>  
<div>This is a div element with a box-shadow</div>  
</body>  
</html>
```

It will produce the following result –



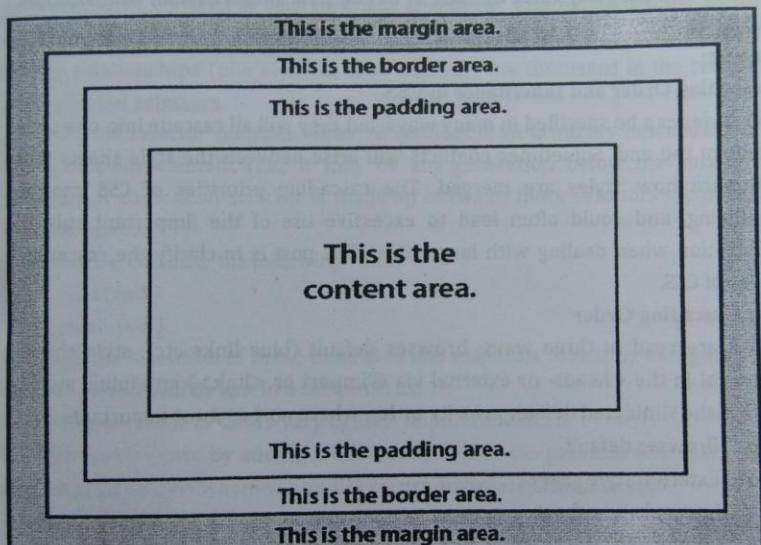
9. explain margin, border and padding of box model with example. 2013

answer:

The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- Content - The content of the box, where text and images appear
- Padding - Clears an area around the content. The padding is transparent
- Border - A border that goes around the padding and content
- Margin - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example

```
div {  
    width: 300px;  
    border: 15px solid green;  
    padding: 50px;  
    margin: 20px;  
}
```

10. how CSS property value cascade in to one style sheets explain with example.

2013

answer:

Cascading Order and Inheritance in CSS

CSS styles can be specified in many ways, but they will all cascade into one style sheet in the end. Sometimes conflicts will arise between the style sheets that influence how styles are merged. The cascading priorities of CSS can be confusing, and could often lead to excessive use of the !important rule in frustration when dealing with large sites. This post is to clarify the cascading order of CSS.

The Cascading Order

Styles are read in three ways: browser default (blue links etc), style sheets (internal in the <head> or external via @import or <link>) and inline styles. Here is the simulated default priority order, where no.4 as most important:

1. Browser default
2. External style sheet
3. Internal style sheet
4. Inline style

If two rules have the same weight, the latter wins. Imported style sheets and internal style sheets actually carry the same weight, but since imported styles are considered to be before any rules in the style sheet itself, the latter will win (the internal). This is not the case if you f.ex have two <style> tags inside your <head>, where second one uses the @import rule to import an external style sheet. That could be considered invalid according to the specs: In CSS1, all '@import' statements must occur at the start of a style sheet, before any declarations.

Also worth noting in this list, is that any inline style will carry the same weight as #id definitions inside the style sheet. But again, since the inline style attribute comes latter, it will win the battle.

10. explain the contextual selection of ID and class with example. (2015)

ANSWER:

At times, authors may want selectors to match elements that appear in a certain context, such as "only those EM elements that are contained by an H1 element". In these cases, contextual selectors add specificity. Context is defined as an ancestor/descendent relationship between elements in the document tree. Sibling relationships (one element after another) are discussed in the section on sequential selectors.

A contextual selector matches when an element is an arbitrary descendent of some ancestor element (i.e., it may be any generation below the ancestor element). A contextual selector is made up of two or more selectors separated by white space.

For example, consider the following rules:

```
H1 { color: red }  
EM { color: red }
```

Although the intention of these rules is to add emphasis to text by changing its color, the effect will be lost in a case such as:

```
<H1>This headline is <EM>very</EM> important</H1>
```

We address this case by adding a contextual rule to the previous two that sets the text color to blue whenever an EM occurs anywhere within an H1:

```
H1 { color: red }  
EM { color: red }  
H1 EM { color: blue }
```

The third rule will also match the following fragment:

```
<H1>This  
    <SPAN class="myclass">headline is <EM>very</EM>  
    important</SPAN></H1>
```

A contextual selector may also contain attribute selectors.

For example, the following matches any element with an "href" attribute inside a P with class "myclass" inside any DIV. Note that the space after "myclass" is

Web Engineering -80

essential: without it the selector would match a P with both a class and an "href":

DIV P.myclass [href]
Contextual selectors may be grouped according to the rules for grouping listed above.

11. Write down the difference between CSS ID and class.2015

Answer:

i d I N C S S

V E R S U S

c l a s s I N C S S

i d

A selector in CSS that styles the element with a specified id

Syntax is #id{ css declarations ; }

Used to apply styling to one specific element

class

A selector in CSS that styles the selected elements with a specified class

Syntax is .class { css declarations; }

Used to apply styling to multiple elements

Visit www.PEDIAA.com

12. explain direct descendant selector with example.2015

answer:

Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

Example

```
div p {  
    background-color: yellow;  
}
```

CHAPTER 4

BASIC CONCEPTS OF JAVA APPLET

1. What is Scripting Language? (2017, 2013, 2012)

Answer:

A script or scripting language is a computer language with a series of commands within a file that is capable of being executed without being compiled. Good examples of server-side scripting languages include Perl, PHP, and Python. The best example of a client side scripting language is JavaScript. A full list of scripting languages and other programming languages can be found through our programming language definition.

2. what is JavaScript? .(2015, 2010)

answer:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

3. Why JavaScript is used in the web page design? .(2015, 2010)

Or, What are the advantages of JavaScript?(2013)

Answer:

Advantages of JavaScript

The merits of using JavaScript are –

- **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

4. Write down the differences between Programming Language and Scripting Language. (2017,2014, 2012)

Answer:

Comparison between Programming Language and Scripting Language:

	Programming Language	Scripting Language
Definition	A programming language is an artificial language designed to communicate instructions to a machine, particularly a computer.	A scripting language, script language or extension language is a programming language that allows control of one or more applications.
Type	Compiler-based language	Interpreter based language
Usage	Developing something from scratch	Used to combine existing components
Interpretation	Programmed languages are compiled into a more compact form that does not need to be interpreted by	Scripted languages are interpreted within another program (like JavaScript is put within HTML and then

	another application in the same way. The compiled result is stand-alone.	interpreted by the browser).
Running	Run independent of an exterior (or parent) program	Run inside another program
Design	Designed to get full usage of a language	Designed to make coding fast and simple
Conversion	Converts the whole program into machine language in one shot	Converts high level instructions into machine language
Creation	Creates a .exe file	Does not create a .exe file
Compilation	Needs to compile the program	There is no need to compile the program
Coding	Programming is making a full code of program	Scripts are just a piece of code
Temperament	Harder to code. Needs numerous lines of code for each function	Easier to code. Needs only a few short lines of code for each function
Complexity	Are complex	Are easy to use and easy to write
Development Time	Takes longer to develop as more code needs to be written.	Takes less time to code as it needs less coding.
Support	<ul style="list-style-type: none"> • Explicit support of Data Types • Rich support for User Interface Design • Rich support for Graphic Design 	<ul style="list-style-type: none"> • Implicit support of Data Types • Limited support for User Interface Design • Limited or no support for Graphic Design
Hosting	Does not require a host. Is self – executable	Requires a host
Cost	Increased maintenance cost	Reduced maintenance cost

Examples	C, C++, C#, Java, VC++, VB, Basic, COBOL, Pascal, etc.	JavaScript, VB Script, Shell, Perl, Python, Ruby, Rexx, PHP, GameMonkey, Lua, etc.
----------	--	--

5. Write down some common mistakes in JavaScript. (2017, 2012)

Answer:

Limitations of JavaScript

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

6. Write a JavaScript Program to calculate the Fibonacci series(2017, 2012)

Answer:

```
<html>
<head><title>Fibonacci Series</title></head>
<body>
    <script type="text/javascript">
        <!--
            var var1 = 0;
            var var2 = 1;
            var var3;
            var num = prompt("Enter the limit to generate fibonacci
no",0);
        -->
    </script>
</body>
</html>
```

```
document.write(var1+"<br />");  
document.write(var2+"<br />");  
  
for(var i=3; i <= num;i++)  
{  
    var3 = var1 + var2;  
    var1 = var2;  
    var2 = var3;  
  
    document.write(var3+"<br />");  
}  
// -->  
</script>  
</body>  
</html>
```

7. what is the necessity of HTML,CSS and script language used together to build a website. (2012)

answer:

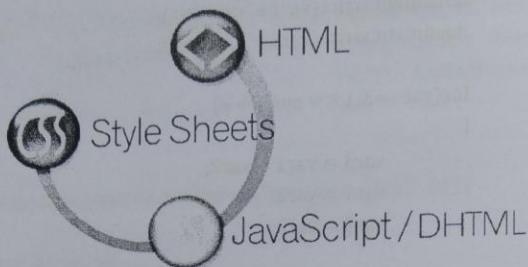
The three main technologies used to create web pages (HTML, CSS and JavaScript) each do different special jobs.

- ✓ HTML should be used only for structuring content.
- ✓ Cascading Style Sheets should be used for applying all visual styles.
- ✓ JavaScript should be used for (almost) all interactive functionality, and should always be referenced in separate files, never written into HTML.

How HTML, CSS and JS work together

The web page you see in your browser may be a combination of structure, style and interactivity.

These jobs are undertaken by 3 different technologies, HTML, Javascript, and CSS which your browser knows how to interpret.

**HTML**

marks the content up into different structural types, like paragraphs, blocks, lists, images, tables, forms, comments etc.

CSS

tells the browser how each type of element should be displayed, which may vary for different media (like screen, print or handheld device)

JavaScript

tells the browser how to change the web page in response to events that happen (like clicking on something, or changing the value in a form input)

8. write a JavaScript program to find the minimum number among three numbers. .(2015, 2010)

answer:

```
<html>
<head>
<title>
Javascript Program to find smallest from 3 numbers
<script language="javascript">

function smaller()
{
var num1=parseInt(document.f1.t1.value);
var num2=parseInt(document.f1.t2.value);
var num3=parseInt(document.f1.t3.value);
if(num1<num2 && num1<num3)
{
```

```
        alert(num1 + "  is smallest number");
    }
    else if(num2<num1 && num2<num3)

    {
        alert(num2 + "  is smallest number");
    }
    else
    {
        alert(num3 + "  is smallest number");
    }
}
</script>
</head>
<body>
<h1 align="center"><font color="red">Javascript for find
smallest</font></h1>
<form name=f1>
num1: <input type="textbox" name=t1><br>
num2: <input type="textbox" name=t2><br>
num3: <input type="textbox" name=t3><br><br><br>
<input type="button" value="smallest"
onclick="smaller()">
</form>
</body>
</html>
```

9. Different between client side and server side scripting with example. (2010)

Answer:

Differences between client-side scripting and server-side scripting

- A browser runs the script for client-side scripting that is already present in the user's computer. A web server runs the script for server-side scripting that creates the page which needs to be sent to the browser.
- Client-side scripting happens when the browser possesses all the codes and the page is later changed according to the user's input. Server-side scripting happens when a user's browser initiates a server request. Dynamic pages are then created based on several conditions.
- A browser can perform the client-side scripting after receiving the page sent by the server. A server can carry out a server-side script, but cannot perform the client side scripting.
- Server-side scripting helps in connecting to the databases that are already present in the web server. Client-side scripting does not connect to the databases that are on the web server.
- The processing of the client-side scripting occurs on the end-users' computers. The language for scripting has to be enabled on the client computer.
- Server-side scripting has access to all the files present in the web server. Client-side scripting has no such access.
- The scripting process for the server side is done on remote computer and hence the response is comparatively slower than the client sided one. In case of server-side scripting, the response is quicker as the scripts are executed on a local computer.
- Languages used in server scripting are **Ruby**, **PHP**, **ASP**, etc. Languages commonly used for client-side scripting are **Javascript**, **VB Script**, etc.
- Client-side scripting is excellent for any case which requires user interaction. Server-side scripting is excellent for any area that requires loading of dynamic data.

10. Different between following common attribute and describe their uses : Class, id, name.. (2010)

answer:

Class Selectors

In CSS we have this concept of selectors, in order for us to be able to style our html elements correctly, we have to ensure we use the right selector in the right place. Say, for example, we wanted to style all `<div>` elements that feature the class equal to blue with a blue background. We'd have to create a CSS rule that applies to the `.blue` selector.

```
.blue {  
    background-color: blue;  
}
```

With this `.blue` rule above, only the following html element would be styled with a blue background.

```
<div class="blue"></div>
```

ID Selectors

If we want to style a html component that features an id attribute that equals blue then we have to use the `# id` selector which will style the single element tagged with this id.

```
#blue {  
    background-color: blue;  
}
```

The above rule will only style the element with the attribute `id="blue"` like our `<div>` tag does below.

```
<div id="blue"></div>
```

11. Write a JavaScript function for common input validation.. (2010)

Answer:

Form validation normally used to occur at the server, after the client had entered all the necessary data and then pressed the Submit button. If the data entered by a client was incorrect or was simply missing, the server would have to send all the data back to the client and request that the form be resubmitted with correct information. This was really a lengthy process which used to put a lot of burden on the server.

an example to understand the process of validation. Here is a simple form in html format.

```
<html>
    <head>
        <title>Form Validation</title>
        <script type = "text/javascript">
            <!--
                // Form validation code will come here.
            //-->
        </script>
    </head>

    <body>
        <form action = "/cgi-bin/test.cgi" name = "myForm"
onsubmit = "return(validate());">
            <table cellspacing = "2" cellpadding = "2" border =
"1">

                <tr>
                    <td align = "right">Name</td>
                    <td><input type = "text" name = "Name" /></td>
                </tr>

                <tr>
                    <td align = "right">EMail</td>
```

```
<td><input type = "text" name = "EMail" /></td>
</tr>

<tr>
    <td align = "right">Zip Code</td>
    <td><input type = "text" name = "Zip" /></td>
</tr>

<tr>
    <td align = "right">Country</td>
    <td>
        <select name = "Country">
            <option value = "-1" selected>[choose
yours]</option>
            <option value = "1">USA</option>
            <option value = "2">UK</option>
            <option value = "3">INDIA</option>
        </select>
    </td>
</tr>

<tr>
    <td align = "right"></td>
    <td><input type = "submit" value = "Submit"
/></td>
</tr>

</table>
</form>
</body>
</html>
```

Output**Basic Form Validation**

First let us see how to do a basic form validation. In the above form, we are calling validate() to validate data when onsubmit event is occurring. The following code shows the implementation of this validate() function.

```
<script type = "text/javascript">
<!--
// Form validation code will come here.

function validate() {

    if( document.myForm.Name.value == "" ) {
        alert( "Please provide your name!" );
        document.myForm.Name.focus() ;
        return false;
    }
    if( document.myForm.EMail.value == "" ) {
        alert( "Please provide your Email!" );
        document.myForm.EMail.focus() ;
        return false;
    }
    if( document.myForm.Zip.value == "" || isNaN(
document.myForm.Zip.value ) ||
        document.myForm.Zip.value.length != 5 ) {

        alert( "Please provide a zip in the format #####." );
        document.myForm.Zip.focus() ;
        return false;
    }
    if( document.myForm.Country.value == "-1" ) {
        alert( "Please provide your country!" );
        return false;
    }
    return( true );
}
```

```
    }
  //-->
</script>
```

12. what is DHTML ? (2012)

answer:

DHTML stands for Dynamic HTML, it is totally different from HTML. The browsers which support the dynamic HTML are some of the versions of Netscape Navigator and Internet Explorer of version higher than 4.0. The DHTML is based on the properties of the HTML, javascript, CSS, and DOM (Document Object Model which is used to access individual elements of a document) which helps in making dynamic content. It is the combination of HTML, CSS, JS, and DOM. The DHTML make use of Dynamic object model to make changes in settings and also in properties and methods. It also makes uses of Scripting and it is also part of earlier computing trends.

13. describe the following types of event in JavaScript with example:- (2013)

- (i) input event
- (ii) mouse event
- (iii) load event

Answer:

(i) input event

onblur - When a user leaves an input field

onchange - When a user changes the content of an input field

onchange - When a user selects a dropdown value

onfocus - When an input field gets focus

onselect - When input text is selected

onsubmit - When a user clicks the submit button

onreset - When a user clicks the reset button

onkeydown - When a user is pressing/holding down a key

onkeypress - When a user is pressing/holding down a key

onkeyup - When the user releases a key

onkeyup - When the user releases a key

onkeydown vs onkeyup - Both

(ii) mouse event

onmouseover/onmouseout - When the mouse passes over an element
onmousedown/onmouseup - When pressing/releasing a mouse button
onmousedown - When mouse is clicked: Alert which element
onmousedown - When mouse is clicked: Alert which button
onmousemove/onmouseout - When moving the mouse pointer over/out of an image
onmouseover/onmouseout - When moving the mouse over/out of an image
onmouseover an image map

(iii) load event

onload - When the page has been loaded

onload - When an image has been loaded

onerror - When an error occurs when loading an image

onunload - When the browser closes the document

onresize - When the browser window is resized

14. describe different mouse events in JavaScript. (2012)

answer:

mouse event

onmouseover/onmouseout - When the mouse passes over an element
onmousedown/onmouseup - When pressing/releasing a mouse button
onmousedown - When mouse is clicked: Alert which element
onmousedown - When mouse is clicked: Alert which button
onmousemove/onmouseout - When moving the mouse pointer over/out of an image
onmouseover/onmouseout - When moving the mouse over/out of an image
onmouseover an image map

15. describe a JavaScript function. Right line, date and time with example. (2012)

answer:

The Date object is a datatype built into the JavaScript language. Date objects are created with the new Date() as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so JavaScript can represent date and time till the year 275755.

Syntax

You can use any of the following syntaxes to create a Date object using Date() constructor.

```
new Date( )
new Date(milliseconds)
new Date(datestring)
new Date(year,month,date[,hour,minute,second,millisecond ])
```

Note – Parameters in the brackets are always optional.

Here is a description of the parameters –

- **No Argument** – With no arguments, the Date() constructor creates a Date object set to the current date and time.
- **milliseconds** – When one numeric argument is passed, it is taken as the internal numeric representation of the date in milliseconds, as returned by the getTime() method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/70.
- **datestring** – When one string argument is passed, it is a string representation of a date, in the format accepted by the Date.parse() method.
- **7 arguments** – To use the last form of the constructor shown above. Here is a description of each argument –
 - **year** – Integer value representing the year. For compatibility (in order to avoid the Y2K problem), you should always specify the year in full; use 1998, rather than 98.

- o **month** – Integer value representing the month, beginning with 0 for January to 11 for December.
- o **date** – Integer value representing the day of the month.
- o **hour** – Integer value representing the hour of the day (24-hour scale).
- o **minute** – Integer value representing the minute segment of a time reading.
- o **second** – Integer value representing the second segment of a time reading.
- o **millisecond** – Integer value representing the millisecond segment of a time reading.

16. what is a cookie? How does one access cookies in JavaScript? (2014)

answer: Cookies are data, stored in small text files, on your computer.

When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user.

Cookies were invented to solve the problem "how to remember information about the user":

- ✓ When a user visits a web page, his/her name can be stored in a cookie.
- ✓ Next time the user visits the page, the cookie "remembers" his/her name.

Cookies are saved in name-value pairs like:

username = John Doe

A Function to Get a Cookie

Then, we create a function that returns the value of a specified cookie:

Example

```
function getCookie(cname) {  
    var name = cname + "=";  
    var decodedCookie = decodeURIComponent(document.cookie);  
    var ca = decodedCookie.split(';');  
    for(var i = 0; i < ca.length; i++) {  
        var c = ca[i];  
        while (c.charAt(0) == ' ') {  
            c = c.substring(1);  
        }  
        if (c.indexOf(name) == 0) {  
            return c.substring(name.length, c.length);  
        }  
    }  
}
```

```
    return "";
}
```

Function explained:

Take the cookie name as parameter (cname).

Create a variable (name) with the text to search for (cname + "=").

Decode the cookie string, to handle cookies with special characters, e.g. '\$'.

Split document.cookie on semicolons into an array called ca (ca = decodedCookie.split(';')).

Loop through the ca array (*i* = 0; *i* < ca.length; *i*++), and read out each value c = ca[i]]).

If the cookie is found (c.indexOf(name) == 0), return the value of the cookie (c.substring(name.length, c.length)).

If the cookie is not found, return "".

A Function to Check a Cookie

Last, we create the function that checks if a cookie is set.

If the cookie is set it will display a greeting.

If the cookie is not set, it will display a prompt box, asking for the name of the user, and stores the username cookie for 365 days, by calling the setCookie function:

Example

```
function checkCookie()
{
    var username           =           getCookie("username");
    if (username           != "")           {
        alert("Welcome again " + username);
    } else {
        username = prompt("Please enter your name:", "");
        if (username != "" && username != null) {
            setCookie("username",           username, 365);
        }
    }
}
```

17. describe JavaScript objects briefly with examples. (2014)

answer:

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers –

- **Encapsulation** – the capability to store related information, whether data or methods, together in an object.
- **Aggregation** – the capability to store one object inside another object.
- **Inheritance** – the capability of a class to rely upon another class (or number of classes) for some of its properties and methods.
- **Polymorphism** – the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object, otherwise the attribute is considered a property.

Object Properties

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used throughout the page.

The syntax for adding a property to an object is –

```
objectName.objectProperty = propertyValue;
```

For example – The following code gets the document title using the "title" property of the **document** object.

```
var str = document.title;
```

Object Methods

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the **this** keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

For example – Following is a simple example to show how to use the `write()` method of document object to write any content on the document.

```
document.write("This is test");
```

18. explain the way JavaScript handles arrays with examples. (2014)

answer:

An array is a special variable, which can hold more than one value at a time. If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
var array_name = [item1, item2, ...];
```

Example

```
var cars = ["Saab", "Volvo", "BMW"];
```

Access the Elements of an Array

You access an array element by referring to the **index number**.

This statement accesses the value of the first element in cars:

```
var name = cars[0];
```

Changing an Array Element

This statement changes the value of the first element in cars:

```
cars[0] = "Opel";
```

19. write a JavaScript program to find the factorial of a given number. (2014)

answer:

```
<!doctype html>
<html>
<head>
<script>
function show()
{
    var i, no, fact;
    fact=1;
    no=Number(document.getElementById("num").value);
    for(i=1; i<=no; i++)
    {
        fact= fact*i;
    }
    document.getElementById("answer").value= fact;
}
</script>
</head>
<body>
Enter Num: <input id="num">
<button onclick="show()">Factorial</button>
<input id="answer">
</body>
</html>
```

20. explain document . Get element By ID () function with example . .(2015, 2010)

answer:

Document

A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content. The way a document content is accessed and modified is called the Document Object Model, or DOM. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

- Window object – Top of the hierarchy. It is the outmost element of the object hierarchy.
- Document object – Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
- Form object – Everything enclosed in the <form>...</form> tags sets the form object.
- Form control elements – The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

getElementById()

The getElementById() method returns the element that has the ID attribute with the specified value.

This method is one of the most common methods in the HTML DOM, and is used almost every time you want to manipulate, or get info from, an element on your document.

Returns null if no elements with the specified ID exists.

An ID should be unique within a page. However, if more than one element with the specified ID exists, the getElementById() method returns the first element in the source code.

Syntax

```
document.getElementById(elementID)
```

Parameter Values

Parameter	Type	Description
elementID	String	Required. The ID attribute's value of the element you want to get

21. prove that, HTML form validation can be done by JavaScript. (2013)

answer:

JavaScript Form Validation

HTML form validation can be done by JavaScript.

If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

JavaScript Example

```
function validateForm()
{
    var x      = document.forms["myForm"]["fname"].value;
    if (x      == "") {
        alert("Name must be filled out");
        return false;
    }
}
```

The function can be called when the form is submitted:

HTML Form Example

```
<form name="myForm" action="/action_page.php" onsubmit="return
validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
```

22. describe various types of input control for a basic form. (2013)

answer:

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form –

- ✓ Text Input Controls
- ✓ Checkboxes Controls
- ✓ Radio Box Controls
- ✓ Select Box Controls
- ✓ File Select boxes
- ✓ Hidden Controls
- ✓ Clickable Buttons
- ✓ Submit and Reset Button

Text Input Controls

There are three types of text input used on forms –

- **Single-line text input controls** – This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.
- **Password input controls** – This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag.
- **Multi-line text input controls** – This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML `<input>` tag.

Example

Here is a basic example of a single-line text input used to take first name and last name –

```
<!DOCTYPE html>
<html>

    <head>
```

```
<title>Text Input Control</title>
</head>

<body>
<form >
    First name: <input type = "text" name = "first_name"
/>
    <br>
    Last name: <input type = "text" name = "last_name" />
</form>
</body>

</html>
```

This will produce the following result –

First name:
Last name:

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML `<input>` tag but type attribute is set to password.

Example

Here is a basic example of a single-line password input used to take user password –

```
<!DOCTYPE html>
<html>

    <head>
        <title>Password Input Control</title>
    </head>

    <body>
        <form >
```

```
User ID : <input type = "text" name = "user_id" />
<br>
Password: <input type = "password" name = "password"
/>
</form>
</body>

</html>
```

This will produce the following result –

```
User ID: _____
Password: _____
```

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML `<textarea>` tag.

Example

Here is a basic example of a multi-line text input used to take item description –

```
<!DOCTYPE html>
<html>

    <head>
        <title>Multiple-Line Input Control</title>
    </head>

    <body>
        <form>
            Description : <br />
            <textarea rows = "5" cols = "50" name = "description">
                Enter description here...
            </textarea>
        </form>
    </body>
```

```
</html>
```

This will produce the following result –

Description:

```
Enter description here...
```

CHAPTER 5
PHP AND MY-SQL

1. What is PHP? (2017,2014,2013)

solution: PHP is an acronym for "PHP: Hypertext Preprocessor" PHP is a widely-used, open source scripting language are executed on the server. PHP is free to download and use

2. How PHP code in bed with HTML? (2013)

solution: "Show PHP" tool makes it easy to learn PHP, it shows both the PHP source code and the HTML output of the code.

Example

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My           first           PHP           script!";
?>

</body>
</html>
```

3. What is PHP? How is it related with HTML? (2017,2011)

solution: short for PHP: Hypertext Preprocessor, an open source, server-side, HTML embedded scripting language used to create dynamic Web pages.

In an HTML document, PHP script (similar syntax to that of Perl or C) is enclosed within special PHP tags. Because PHP is embedded within tags, the author can jump between HTML and PHP (similar to ASP and Cold Fusion) instead of having to rely on heavy amounts of code to output HTML. And, because PHP is executed on the server, the client cannot view the PHP code.

PHP can perform any task that any CGI program can do, but its strength lies in its compatibility with many types of databases. Also, PHP can talk across networks using IMAP, SNMP, NNTP, POP3, or HTTP.

4. What are the basic requirements to running PHP program? 2014

solution:

To start using PHP, you can:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL

Use a Web Host With PHP Support

If your server has activated support for PHP you do not need to do anything. Just create some .php files, place them in your web directory, and the server will automatically parse them for you. You do not need to compile anything or install any extra tools. Because PHP is free, most web hosts offer PHP support.

Set Up PHP on Your Own PC

However, if your server does not support PHP, you must:

- install a web server
- install PHP
- install a database, such as MySQL

5. How array is declared in PHP? Also explain various types of array with proper example. (2017,2011)

solution: An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables it's easy to define an array of 100 length.

In PHP, the array() function is used to create an array:

array();

In PHP, there are three types of arrays:

- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays

Indexed arrays

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

There are two ways to create indexed arrays:

The index can be assigned automatically (index always starts at 0), like this:

```
$cars = array("Volvo", "BMW", "Toyota");
```

or the index can be assigned manually:

```
$cars[0] = "Volvo";
```

```
$cars[1] = "BMW";
```

```
$cars[2] = "Toyota";
```

The following example creates an indexed array named \$cars, assigns three elements to it, and then prints a text containing the array values:

Example

```
<?php
```

```
$cars = array("Volvo", "BMW", "Toyota");
```

```
echo "I like ". $cars[0] . ", ". $cars[1] . " and ". $cars[2] . " .";
```

```
?>
```

Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

Associative arrays are arrays that use named keys that you assign to them.

There are two ways to create an associative array:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

or:

```
$age['Peter'] = "35";
```

```
$age['Ben'] = "37";
```

```
$age['Joe'] = "43";
```

The named keys can then be used in a script:

Example

```
<?php
```

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

```
echo "Peter is " . $age['Peter'] . " years old.";  
?>
```

Multidimensional Arrays

A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

**6. What are the techniques used for form validation and verification in PHP?
(2017)**

solution:

Validation means check the input submitted by the user. There are two types of validation are available in PHP. They are as follows –

- Client-Side Validation – Validation is performed on the client machine web browsers.
- Server Side Validation – After submitted by data, The data has sent to a server and perform validation checks in server machine.

The HTML form we will be working at in these chapters, contains various input fields: required and optional text fields, radio buttons, and a submit button:

PHP Form Validation Example

* required field

Name:

E-mail:

Website:

Comment:

Gender: Female Male Other *

Your Input:

The validation rules for the form above are as follows:

Field	Validation Rules
Name	Required. + Must only contain letters and whitespace
E-mail	Required. + Must contain a valid email address (with @ and .)
Website	Optional. If present, it must contain a valid URL
Comment	Optional. Multi-line input field (textarea)
Gender	Required. Must select one

First we will look at the plain HTML code for the form:

Text Fields

The name, email, and website fields are text input elements, and the comment field is a textarea. The HTML code looks like this:

Name: <input type="text" name="name">

E-mail: <input type="text" name="email">

Website: <input type="text" name="website">

```
Comment:      <textarea          name="comment"          rows="5"
cols="40"></textarea>
```

Radio Buttons

The gender fields are radio buttons and the HTML code looks like this:

Gender:

```
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
```

The Form Element

The HTML code of the form looks like this:

```
<form         method="post"         action="php
echo
htmlspecialchars($_SERVER["PHP_SELF"]);?&gt;"&gt;</pre
```

When the form is submitted, the form data is sent with method="post".

7. Write short notes : (i) Session; (ii) PHP variable. (2017)

solution:

(i) Session

What is a PHP Session

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So, Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: `$_SESSION`.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

Get PHP Session Variable Values

Next, we create another page called "demo_session2.php". From this page, we will access the session information we set on the first page ("demo_session1.php").

Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (`session_start()`).

Also notice that all session variable values are stored in the global `$_SESSION` variable:

Example

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php  
// Echo session variables that were set on previous page  
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";  
echo "Favorite animal is " . $_SESSION["favanimal"] . ".  
?>  
  
</body>  
</html>
```

Modify a PHP Session Variable

To change a session variable, just overwrite it:

Example

```
<?php  
session_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// to change a session variable, just overwrite it  
$_SESSION["favcolor"] = "yellow";  
print_r($_SESSION);  
?>
```

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

Destroy a PHP Session

To remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()`:

Example

```
<?php  
session_start();  
?>  
<!DOCTYPE html>
```

```
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

(ii) PHP variable

Variables are "containers" for storing information.

Creating (Declaring) PHP Variables

In PHP, a variable starts with the \$ sign, followed by the name of the variable:

Example

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for PHP variables:

- A variable starts with the \$ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (\$age and \$AGE are two different variables)

Output Variables

The PHP echo statement is often used to output data to the screen.

The following example will show how to output text and a variable:

Example

```
<?php  
$txt = "W3Schools.com";  
echo "I love $txt!";  
?>
```

PHP Variables Scope

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

- local
- global
- static

Global and Local Scope

A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function:

Example

```
<?php  
$x = 5; // global scope
```

```
function myTest() {  
    // using x inside this function will generate an error  
    echo "<p>Variable x inside function is: $x</p>";  
}  
myTest();  
  
echo "<p>Variable x outside function is: $x</p>";
```

8. write down the different types of super global array with their functionality.
 (2013)

9. describe following global variable in PHP:-

- (i)S-GET
- (ii)S-POST
- (iii)S-REQUEST 2014

answer:

PHP Global Variables - Superglobals

Several predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

```
$GLOBALS  
$_SERVER  
$_REQUEST  
$_POST  
$_GET  
$_FILES  
$_ENV  
$_COOKIE  
$_SESSION
```

PHP \$GLOBALS

\$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

PHP stores all global variables in an array called \$GLOBALS[index]. The index holds the name of the variable.

The example below shows how to use the super global variable \$GLOBALS:

Example

```
<?php  
$x = 75;  
$y = 25;
```

```
function addition() {  
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];  
}  
addition();  
echo $z;  
?>
```

PHP \$_SERVER

`$_SERVER` is a PHP super global variable which holds information about headers, paths, and script locations.

The example below shows how to use some of the elements in `$_SERVER`:

Example

```
<?php  
echo $_SERVER['PHP_SELF'];  
echo "<br>";  
echo $_SERVER['SERVER_NAME'];  
echo "<br>";  
echo $_SERVER['HTTP_HOST'];  
echo "<br>";  
echo $_SERVER['HTTP_REFERER'];  
echo "<br>";  
echo $_SERVER['HTTP_USER_AGENT'];  
echo "<br>";  
echo $_SERVER['SCRIPT_NAME'];  
?>
```

PHP \$_REQUEST

`PHP $_REQUEST` is used to collect data after submitting an HTML form.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the `<form>` tag. In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable `$_REQUEST` to collect the value of the input field:

Example

```
<html>
<body>

<form method="post" action=<?php echo $_SERVER['PHP_SELF'];?>>
Name: <input type="text" name="fname">
<input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_REQUEST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

PHP \$_POST

PHP \$_POST is widely used to collect form data after submitting an HTML form with method="post". \$_POST is also widely used to pass variables.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable \$_POST to collect the value of the input field:

Example

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if(empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

PHP \$_GET

PHP \$_GET can also be used to collect form data after submitting an HTML form with method="get".

\$_GET can also collect data sent in the URL.

Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>
<body>
<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>
</body>
</html>
```

10. explain PHP my SQL connectivity and basic connection function with example.(2013,2010)

answer:

In this, and in the following chapters we demonstrate three ways of working with PHP and MySQL:

- MySQLi (object-oriented)
- MySQLi (procedural)
- PDO

MySQLi Installation

For Linux and Windows: The MySQLi extension is automatically installed in most cases, when php5 mysql package is installed.

Syntax

```
bool mysql_close ( resource $link_identifier );
```

If a resource is not specified, then the last opened database is closed. This function returns true if it closes the connection successfully otherwise it returns false.

Example

Try the following example to connect to a MySQL server –

Example (MySQLi Object-Oriented)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>

Example (PDO)
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB",
    $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
}
catch(PDOException $e)
{
    echo "Connection failed: " . $e->getMessage();
}
?>
```

11. Explain "altering table", "inserting data in table" with example. (2013)

answer:

SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

ALTER TABLE - ADD Column

To add a column in a table, use the following syntax:

ALTER TABLE table_name

ADD column_name datatype;

The following SQL adds an "Email" column to the "Customers" table:

Example

ALTER TABLE Customers

ADD Email varchar(255);

The SQL INSERT INTO Statement

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways.

The first way specifies both the column names and the values to be inserted:

INSERT INTO table_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

INSERT INTO table_name

VALUES (value1, value2, value3, ...);

INSERT INTO Example

The following SQL statement inserts a new record in the "Customers" table:

Example

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

12. what is session variable? How do you create a session in PHP? (2013)

answer:

Get PHP Session Variable Values

Next, we create another page called "demo_session2.php". From this page, we will access the session information we set on the first page ("demo_session1.php").

Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (`session_start()`).

Also notice that all session variable values are stored in the global `$_SESSION` variable:

Example

```
<?php  
session_start();  
?  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Echo session variables that were set on previous page  
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";  
echo "Favorite animal is " . $_SESSION["favanimal"] . ".  
?  
  
</body>  
</html>
```

create a session in PHP

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So, Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: `$_SESSION`.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example

```
<?php  
// Start the session  
session_start();  
?  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Set session variables  
$_SESSION["favcolor"] = "green";  
$_SESSION["favanimal"] = "cat";  
echo "Session variables are set.";  
?  
  
</body>  
</html>
```

13. what are the use of session and cookie variables in statemanagement?
(2015,2011)

answer:

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Create Cookies With PHP

A cookie is created with the `setcookie()` function.

Syntax

`setcookie(name, value, expire, path, domain, secure, httponly);`

Only the name parameter is required. All other parameters are optional.

PHP Session

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: `$_SESSION`.

14. Explain the advantages and disadvantages of MySQL. (2017,2014,2011)

Answer:

MySQL:

Disadvantage:

- Error are misleading
- Group by function does not work as the way they are intended to
- Correlated Subquery does not work as intended

Advantage:

- LIMIT clause is nice
- Engines implemented in MySQL can just work amazing. If properly used

15. Write down the procedure of creating database and table in MySQL. (2017, 2015,2014, 2010)

Answer:

Create a MySQL Database

The CREATE DATABASE statement is used to create a database in MySQL.

The following examples create a database named "myDB":

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
// Create database  
$sql = "CREATE DATABASE myDB";  
if ($conn->query($sql) === TRUE) {  
    echo "Database created successfully";
```

```

} else {
    echo "Error creating database: " . $conn->error;
}
$conn->close();
?>

```

Create a MySQL Table

The CREATE TABLE statement is used to create a table in MySQL.

We will create a table named "MyGuests", with five columns: "id", "firstname", "lastname", "email" and "reg_date":

```

CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP
)
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (

```

```

        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP
    )";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>

```

16. Discuss the steps to establish a connection from PHP to MySQL with sample code. (2017,2014,2012,2011)

Or, what is the standard PHP function for connecting to a my SQL database? Explain the process of using my SQL with PHP using sample code. (2015)

Open a Connection to MySQL

Before we can access data in the MySQL database, we need to be able to connect to the server:

Example (MySQLi Object-Oriented)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {

```

```
die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

17. what are the features that made my SQL a popular database management system for web service?2015

answer:

- ✓ MySQL is a database system used on the web
- ✓ MySQL is a database system that runs on a server
- ✓ MySQL is ideal for both small and large applications
- ✓ MySQL is very fast, reliable, and easy to use
- ✓ MySQL uses standard SQL
- ✓ MySQL compiles on a number of platforms
- ✓ MySQL is free to download and use
- ✓ MySQL is developed, distributed, and supported by Oracle Corporation
- ✓ MySQL is named after co-founder Monty Widenius's daughter: My

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

18. What is the PHP operator? Describe different type of PHP operator with example. (2010)

Answer:

PHP Operators

Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

- ✓ Arithmetic operators
- ✓ Assignment operators
- ✓ Comparison operators
- ✓ Increment/Decrement operators
- ✓ Logical operators
- ✓ String operators
- ✓ Array operators

Arithmetic Operators

There are following arithmetic operators supported by PHP language –

Assume variable A holds 10 and variable B holds 20 then –

Show Examples

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide numerator by de-numerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

Comparison Operators

There are following comparison operators supported by PHP language

Assume variable A holds 10 and variable B holds 20 then –

Show Examples

Operator	Description	Example
<code>==</code>	Checks if the value of two operands are equal or not, if yes then condition becomes true.	$(A == B)$ is not true.
<code>!=</code>	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	$(A != B)$ is true.
<code>></code>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	$(A > B)$ is not true.
<code><</code>	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	$(A < B)$ is true.
<code>>=</code>	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	$(A >= B)$ is not true.
<code><=</code>	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	$(A <= B)$ is true.

Logical Operators

There are following logical operators supported by PHP language

Assume variable A holds 10 and variable B holds 20 then –

Show Examples

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero then condition becomes true.	(A && B) is true.
	Called Logical OR Operator. If any of the two operands are non zero then condition becomes true.	(A B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false.

Assignment Operators

There are following assignment operators supported by PHP language –

Show Examples

Assignment Operator	Meaning
=	Assigns values from right side to left side operand
+=	Adds right side value to left side value and assign result to left side operand
-=	Subtracts right side value from left side value and assign result to left side operand
*=	Multiplication of right side value with left side value and assign result to left side operand
/=	Division of left side value with right side value and assign result to left side operand
%=	Modulus of left side value with right side value and assign result to left side operand
**=	Exponent of left side value with right side value and assign result to left side operand

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	$C = A + B$ will assign value of $A + B$ into C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	$C += A$ is equivalent to $C = C + A$
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	$C -= A$ is equivalent to $C = C - A$
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	$C *= A$ is equivalent to $C = C * A$
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	$C /= A$ is equivalent to $C = C / A$
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	$C \%= A$ is equivalent to $C = C \% A$

Conditional Operator

There is one more operator called conditional operator. This first evaluates an expression for a true or false value and then execute one of the two given statements depending upon the result of the evaluation. The conditional operator has this syntax –

Show Examples

Operator	Description	Example
? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y

19. what is error handler? What is the purpose of error handler? (2012)

answer:

Error handling is the process of catching errors raised by your program and then taking appropriate action. If you would handle errors properly then it may lead to many unforeseen consequences.

Its very simple in PHP to handle an errors.

This function must be able to handle a minimum of two parameters (error level and error message) but can accept up to five parameters (optionally: file, line-number, and the error context) –

Syntax

```
error_function(error_level,error_message,
error_file,error_line,error_context);
```

Sr.No	Parameter & Description
1	error_level Required - Specifies the error report level for the user-defined error. Must be a value number.
2	error_message Required - Specifies the error message for the user-defined error
3	error_file Optional - Specifies the file name in which the error occurred
4	error_line Optional - Specifies the line number in which the error occurred

5

error_context

Optional - Specifies an array containing every variable and their values in use when the error occurred

20. Both SELECT DISTINCT and GROUP BY causes the display to show only one output raw for each value in a column, even if multiple rows contain that value. What are the main differences between SELECT DISTINCT and GROUP BY? (2015)

Answer:

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

GROUP BY Syntax

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```