Department of Information Science and Engineering

18CS734 – Web Technology and its Applcation July / August 2022

VTU Question Paper – Scheme and Solution

Module 1

1. (a) What is HTML? Explain the structure of HTML document with an example.

Solution : HTML is a markup language

Structure of HTML Documents

A simple html program

```
<!DOCTYPE html>
<html>
<head>
<title> A simple Program
</title>
</head>
<body>
<p> A simple program to show the working of html documents</p>
</body>
</html>
```

<!DOCTYPE html> ----- Defines the type of document
<html> ------Root element – indicates that webpage is written in html
<head> ---- Contains **title** & information about this web page
<body> ------- Contents to be displayed on webpage
</html> -------- Closing of html document

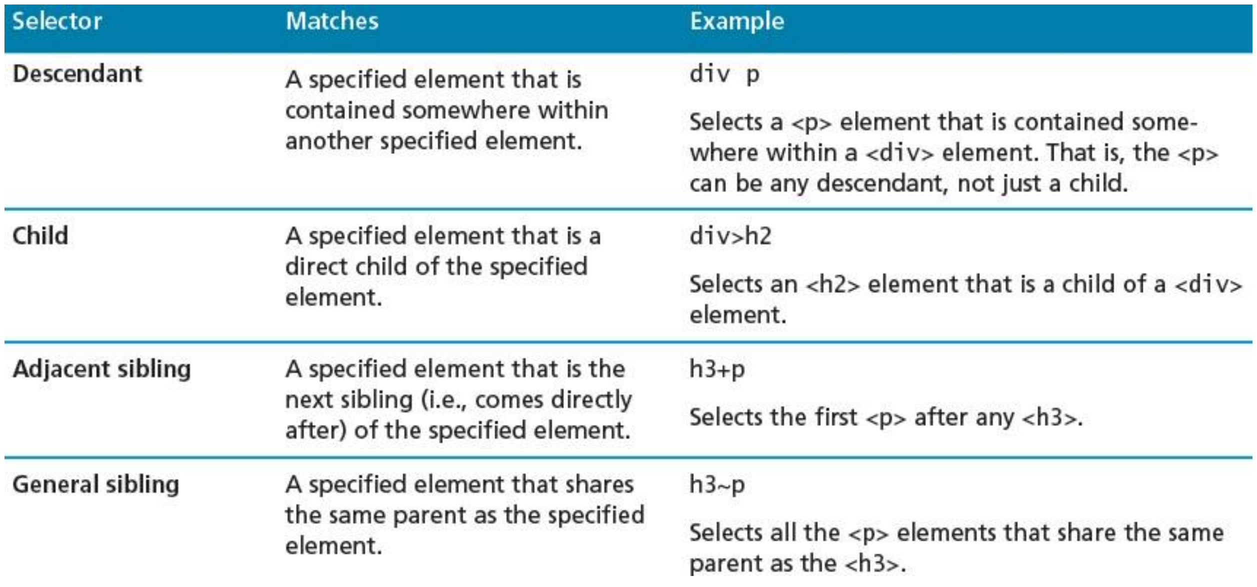
```
<!DOCTYPE html>

<!-- headings.html An example to illustrate headings-->

<head> <title> Headings </title>

</head>

<body>

        <h1> Aidan's Airplanes (h1) </h1>
        <h2> The best in used airplanes (h2) </h2>
        <h3> "We've got them by the hangarful" (h3) </h3>
        <h4> We're the guys to see for a good used airplane (h4) </h4>
        <h5> We offer great prices on great planes (h5) </h5>
        <h6> No returns, no guarantees, no refunds, all sales are final! (h6) </h6>
</body>
</html>
```

(b) What are contextual selectors? Identify and explain four different contextual selectors.

Contextual Selectors

A **contextual selector** (in CSS3 also called **combinators**) allows to select elements based on their *ancestors*, *descendants*, or *siblings*. It selects elements based on their context or relation to other elements in the document tree.

As shown in below table, **descendant selector** matches all elements that are contained within another element. The character used to indicate descendant selection is the space character.



Selector	Matches	Example
Descendant	A specified element that is contained somewhere within another specified element.	`div p` Selects a <p> element that is contained some-where within a <div> element. That is, the <p> can be any descendant, not just a child.
Child	A specified element that is a direct child of the specified element.	`div>h2` Selects an <h2> element that is a child of a <div> element.
Adjacent sibling	A specified element that is the next sibling (i.e., comes directly after) of the specified element.	h3+p Selects the first <p> after any <h3>.
General sibling	A specified element that shares the same parent as the specified element.	h3~p Selects all the <p> elements that share the same parent as the <h3>.

(c) Write the syntax of below mentioned HTML elements and briefly explain with examples.

(a) <a>   (b) <img>

Solution :

(a)<a> :

Links are an essential feature of all web pages. Links are created using the <a> element (the "a"stands for anchor).

<a> is an inline tag. A link has two main parts: the destination and the label. The label of a linkcan be text or another HTML element such as an *image*.

```
<a href = "next.html"> click here </a>

<a href = "next.html"><img src = "image1.jpg"></a>
```

- A link specifies the address of the destination. Such an address might be a file name, adirectory path and a file name, or a complete URL.
- The document whose address is specified in a link is called the target of that link.
- The value assigned to href (hypertext reference) specifies the target of the link. If the targetis in another document in the same directory, the target is just the document's file name.

<img> :

The <img> tag is the oldest method for displaying an image.

```
<img src = "park.jpg" alt = "Central Park" title = "Central-Park" width = "80" height = "40" />
```

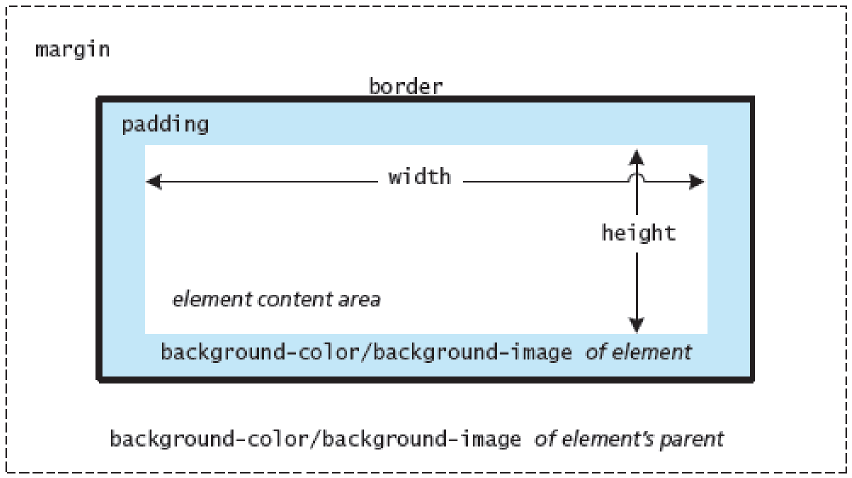
src attribute - specifies the file containing the image;
alt - specifies text to be displayed when it is not possible to display the image.
title – specifies the title to be displayed in pop-up tool tip when user moves mouse over image.
width and height - can be included to specify (in pixels) the size of the rectangle for the image
Eg:

```
<html>
<head>
<head> <title>Image display</title>
</head>
<body>
<p >Displays the image</p>
<img  src="cycle-riding.jpg" width="200" height="200">
<p >qqqqqqqq,</p>
</body>
</html>
```

2. (a) Illustrate the CSS box model. Be sure to label and briefly explain each component of the box.
Solution:

The Box Model

In CSS, all HTML elements exist within a rectangular **element box** shown in Figure.


margin
border
padding
width
height
element content area
background-color/background-image of element
background-color/background-image of element's parent

The background color or image of an element fills an element within its border.
Some of the common background properties are –

1. Property Description

background	A combined shorthand property that allows you to set multiplebackground property.
background-	Specifies whether the background image scrolls with the document (default) or remains fixed. values are: *fixed, scroll*

background-color	Sets the background color of the
background-image	Specifies the background image

1. Borders

Borders are used to visually separate elements. Borders are put around all four sides of anelement, or just one, two, or three of the sides. Various border properties are –

Property	Description
border	A shorthand property that allows to set the style, width, and color of a border in one property. The order is important and must be: border-style border-width border-color
border-style	Specifies the line type of the border. Possible values are: *solid, dotted, dashed, double, groove, ridge, inset, and outset.*
border-width	The width of the border in a unit(usually in px). A variety of keywords (thin, medium, thick etc.) are also supported.
border-color	The color of the border in a color unit.
border-radius	The radius of a rounded corner.
border-image	The URL of an image to use as a border

(b) List the HTML5 Semantic elements and explain any three with suitable elements.

- The new semantic elements in HTML5 are listed below -
  1. Headers and Footer
  2. Heading Groups
  3. Navigation
  4. Articles and Sections
  5. Figure and Figure Captions
  6. Aside

Header and Footer

Most website pages have a recognizable header and footer section. Typically the header contains the site logo and title (and perhaps additional subtitles or taglines), horizontal navigation links, and perhaps one or two horizontal banners. The typical footer contains less important material, such as smaller text versions of the navigation, copyright notices, information about the site’s privacy policy, and perhaps twitter feeds or links to other social sites.

Both the HTML5 <header> and <footer> element can be used not only for *page* headers and footers, but also for header and footer elements within other HTML5 containers, such as
<article> or <section>.

```
<header>
<img src="logo.gif" alt="logo" />
<h1>Fundamentals of Web Development</h1>
...
</header>
<article>
   <header>
      <h2>HTML5 Semantic Structure Elements</h2>
      <p>By <em>Randy Connolly</em></p>
      <p><time>September 30, 2015</time></p>
   </header>
   ...
</article>
```

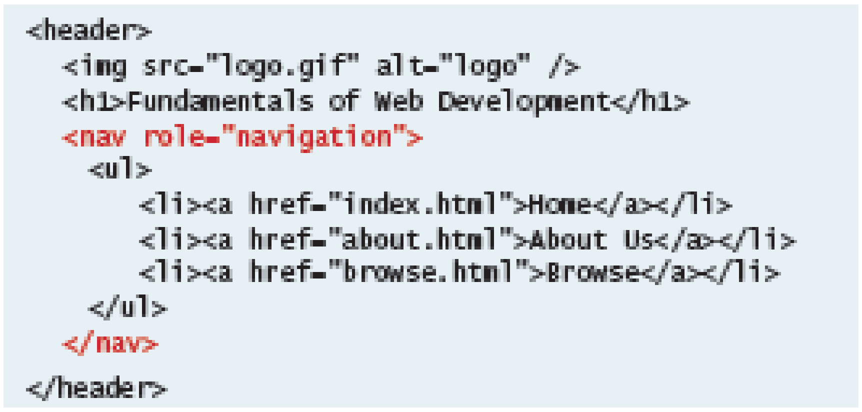
Heading Groups

A header may contain multiple headings <hgroup> element is usually used in such cases. The
<hgroup> element can be used in contexts other than a header. For instance,
one could also usean <hgroup> within an <article> or a <section> element.
The <hgroup> element can *only* contain
<h1>, <h2>, etc., elements.

```
<header>
   <hgroup>
   <h1>Chapter Two: HTML 1</h1>
   <h2>An Introduction</h2>
   </hgroup>
</header>
<article>
   <hgroup>
   <h2>HTML5 Semantic Structure Elements</h2>
   <h3>Overview</h3>
   </hgroup>
</article>
```

Navigation

The <nav> element represents a section of a page that contains links to other pages or to other parts within the same page. Like the other new HTML5 semantic elements, the browser does not apply any special presentation to the <nav> element. The <nav> element was intended to be usedfor major navigation blocks. However, like all the new HTML5 semantic elements, from the browser’s perspective, there is no definite right or wrong way to use the <nav> element. Its sole purpose is to make the document easier to understand.



```
<header>
  <img src="logo.gif" alt="logo" />
  <h1>Fundamentals of Web Development</h1>
  <nav role="navigation">
    <ul>
       <li><a href="index.html">Home</a></li>
       <li><a href="about.html">About Us</a></li>
       <li><a href="browse.html">Browse</a></li>
    </ul>
  </nav>
</header>
```

(c) Describe the embedded style sheet with example.

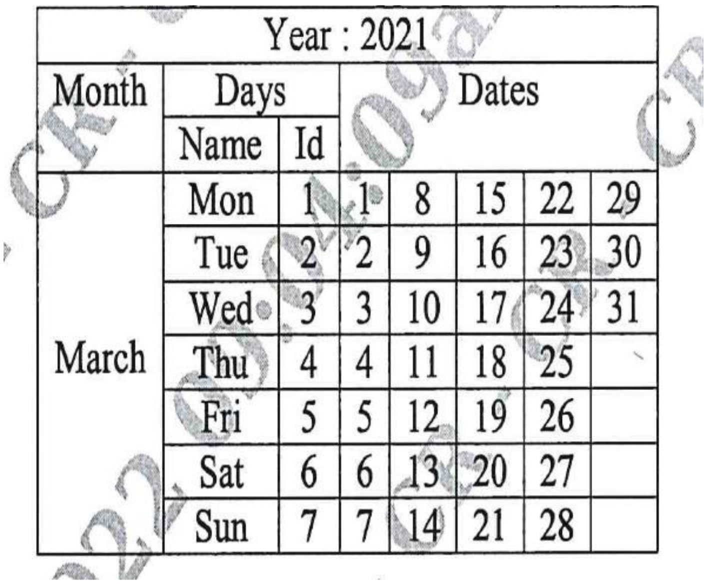
Embedded style sheets (also called **internal styles or document level styles**) are style rules placed within the <style> element (inside the <head> element of an HTML document) and apply to the whole body of the document.

The disadvantage of using embedded styles is that it is difficult to consistently style multiple documents when using embedded styles. But it is helpful when quickly testing out a style that is used in multiple places within a single HTML document. Spaces are ignored in <style> element.

```
<head>
<title>Student Data</title>
<style>
h1 { font-size: 24pt;
h2
{
font-size: 18pt;
}
</style>
</head>
<body>
<h1>Student count</h1>
<h2>CSE/ISE Department</h2>
</body></html>
```

Module 2

3. (a) Write the HTML code for table operation



Year : 2021							
Month	Days		Dates				
	Name	Id					
March	Mon	1	1	8	15	22	29
	Tue	2	2	9	16	23	30
	Wed	3	3	10	17	24	31
	Thu	4	4	11	18	25	
	Fri	5	5	12	19	26	
	Sat	6	6	13	20	27	
	Sun	7	7	14	21	28	

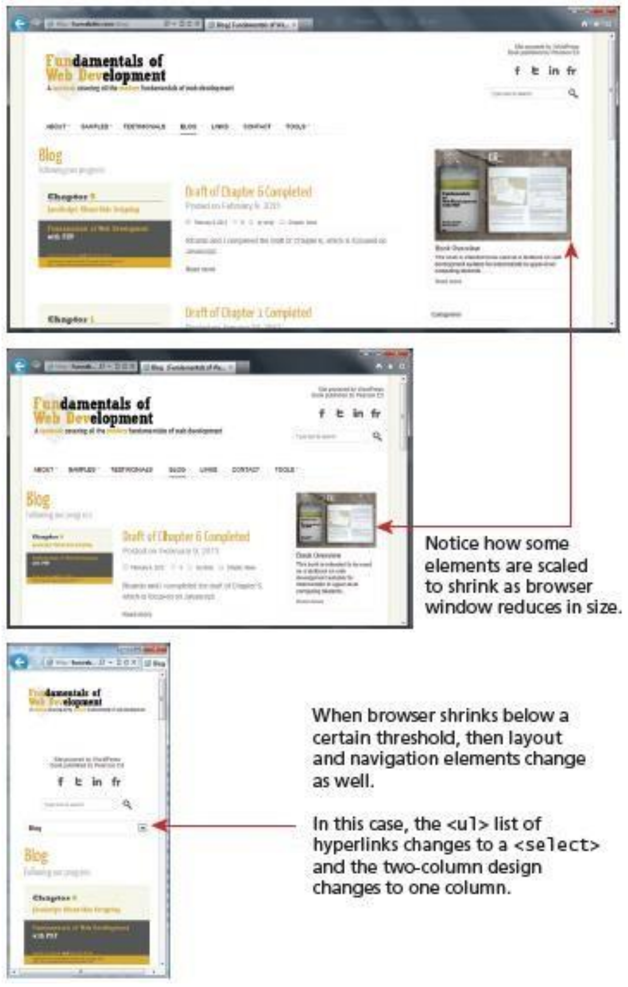
```
<html>
<body>
<table border="border">
<tr>
<th colspan="8"> Year 2021 </th>
</tr>
<tr>
<th rowspan="2">Month</th>
<th colspan="2"> Days</th>
<th rowspan="2" colspan="6"> Dates </th>
</tr>
<tr>
<th> Name</th>
<th>Id</th>
</tr>
<tr>
<th rowspan="7"> March</th>
<th> Mon </th>
<th> 1 </th>
<th> 1 </th>
```

```
<th> 8 </th>
<th> 15 </th>
<th> 22</th>
<th> 29</th>
</tr>
<tr>
<th> Tue </th>
<th> 2 </th>
<th> 2 </th>
<th> 9 </th>
<th> 16 </th>
<th> 23</th>
<th> 30</th>
</tr>
<tr>
<th> wed </th>
<th> 3 </th>
<th> 3 </th>
<th> 10 </th>
<th> 17 </th>
<th> 24</th>
<th> 31</th>
</tr>
<tr>
<th> Thur </th>
<th> 4 </th>
<th> 4 </th>
<th> 11 </th>
<th> 18 </th>
<th> 25</th>
<th> </th>
</tr>
<tr>
```

```
<th> fri </th>
<th> 5 </th>
<th> 5 </th>
<th> 12 </th>
<th> 19 </th>
<th> 26</th>
<th> </th>
</tr>
<tr>
<th> sat </th>
<th> 6 </th>
<th> 6 </th>
<th> 13 </th>
<th> 20 </th>
<th> 27</th>
<th> </th>
</tr>
<tr>
<th> sun </th>
<th> 7 </th>
<th> 7 </th>
<th> 14 </th>
<th> 21 </th>
<th> 28</th>
<th> </th>
</tr>

</body>
</html>
```

(b) What is responsive design? Explain in brief the four key components that make a responsive design work.
Solution :

In a **responsive design**, the page "responds" to changes in the browser size that go beyond the width scaling of a liquid layout.
One of the problems of a liquid layout is that images and horizontal navigation elements tend to take up a fixed size, and when the browser window shrinks to the size of a mobile browser, liquid layouts can become unusable. In a responsive layout, images will be scaled down and navigation elements will be replaced as the browser shrinks, as shown in the figure below.


Fundamentals of Web Development
Notice how some elements are scaled to shrink as browser window reduces in size.
When browser shrinks below a certain threshold, then layout and navigation elements change as well.
In this case, the <ul> list of hyperlinks changes to a <select> and the two-column design changes to one column.

There are four key components that make responsive design work. They are:

1. Liquid layouts
2. Scaling images to the viewport size
3. Setting viewports via the <meta> tag
4. Customizing the CSS for different viewports using media queries

Responsive designs begin with a liquid layout, in which most elements have their widthsspecified as percentages. Making images scale in size is done as follows:

```
img {
max-width: 100%;
}
```

But this does not change the downloaded size of the image; it only shrinks or expands its visualdisplay to fit the size of the browser window, never expanding beyond its actual dimensions.

Setting Viewports

A key technique in creating responsive layouts is the ability of current mobile browsers to shrink or grow the web page to fit the width of the screen. The mobile browser renders the page on a canvas called the **viewport**. On iPhones, for instance, the viewport width is 980 px, and then thatviewport is scaled to fit the current width of the device. The mobile Safari browser introduced the viewport <meta> tag as a way for developers to control the size of that initial viewport.

```
<html>
<head>
<meta name="viewport" content="width=device-width" />
```

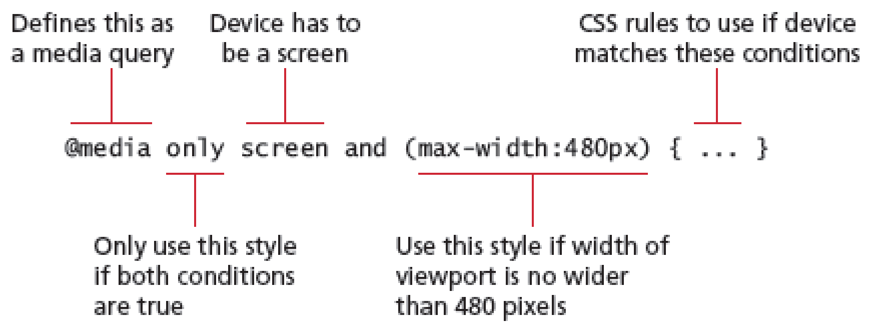
By setting the viewport as above, the page is telling the browser that no scaling is needed, and to make the viewport as many pixels wide as the device screen width. This means that if the device has a screen that is 320 px wide, the viewport width will be 320 px; if the screen is 480 px, then the viewport width will be 480 px.

Media Queries

The other key component of responsive designs is **CSS media queries**. A media query is a way to apply style rules based on the medium that is displaying the file. Use these queries to look at the capabilities of the device, and then define CSS rules to target that device.

Example of media query

@media only screen and (max-width: 480px) {......}


Defines this as a media query
Device has to be a screen
CSS rules to use if device matches these conditions
@media only screen and (max-width:480px) { ... }
Only use this style if both conditions are true
Use this style if width of viewport is no wider than 480 pixels

4. (a) Illustrate the construction of multi column layouts with example

Solution :

Constructing Multicolumn Layouts

The previous sections showed two different ways to move items out of the normal top-down flow, by using positioning (relative, absolute or fixed) and by using floats. They are the techniques that can be used to create more complex layouts. The below topics are about the creation of layout using float and positioning property of CSS.

Using Floats to Create Columns

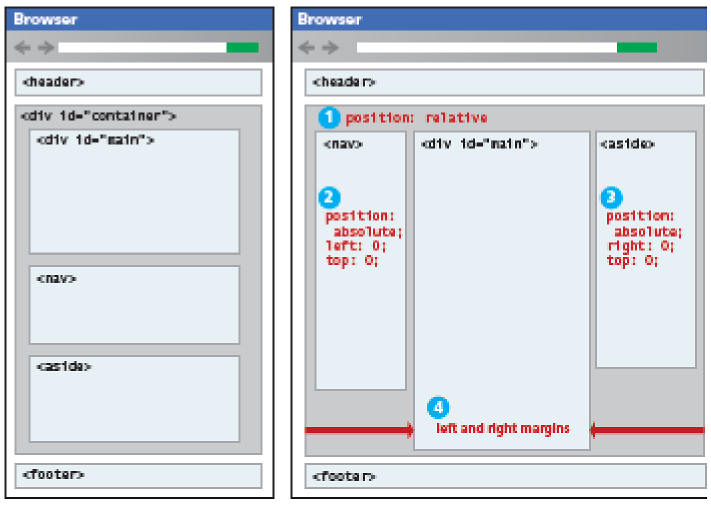
Using floats is the most common way to create columns of content. The steps for this approachare as follows –

1. float the content container that will be on the left-hand side. (the floated container needsto have a width specified).
2. The other content will flow around the floated element.

Set the left – hand side margin for the non-floated element

Using Positioning to Create Columns

Positioning can also be used to create a multicolumn layout. Typically, the approach is toabsolute position the elements.
This approach uses some type of container, in which the elements are positioned.



The following steps are followed –

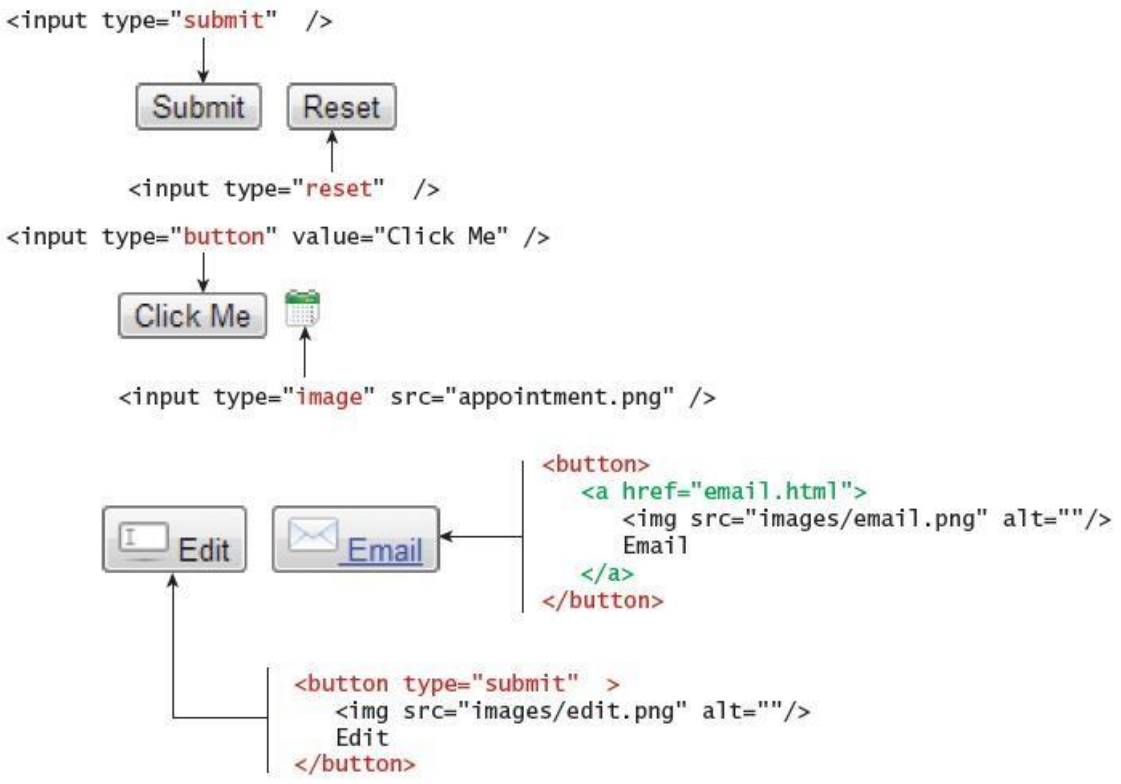
1. Position the container element (into which all other elements are positioned) with respectto the browser window.
2. Position the other elements with respect to the container element, created in step

(b) Explain the different types of buttons define in HTML

Button Controls

HTML defines several different types of buttons.

Type	Description
<input type="submit">	Creates a button that submits the form data to the server
<input type="reset">	Creates a button that clears the user's already entered form data.
<input type="button">	Creates a custom button. This button may requires a script for it to actually perform any action
<input type="image">	Creates a custom submit button that uses an image for its display
<button>	Creates a custom button. The <button> element differs from <input type="button"> in that you can completely customize what appears in the button. It can be used to include both images and text.



```
<input type="submit"  />
<input type="reset"  />
<input type="button" value="Click Me" />
<input type="image" src="appointment.png" />
```

```
<button>
   <a href="email.html">
      <img src="images/email.png" alt=""/>
      Email
   </a>
</button>
```

```
<button type="submit"  >
   <img src="images/edit.png" alt=""/>
   Edit
</button>
```

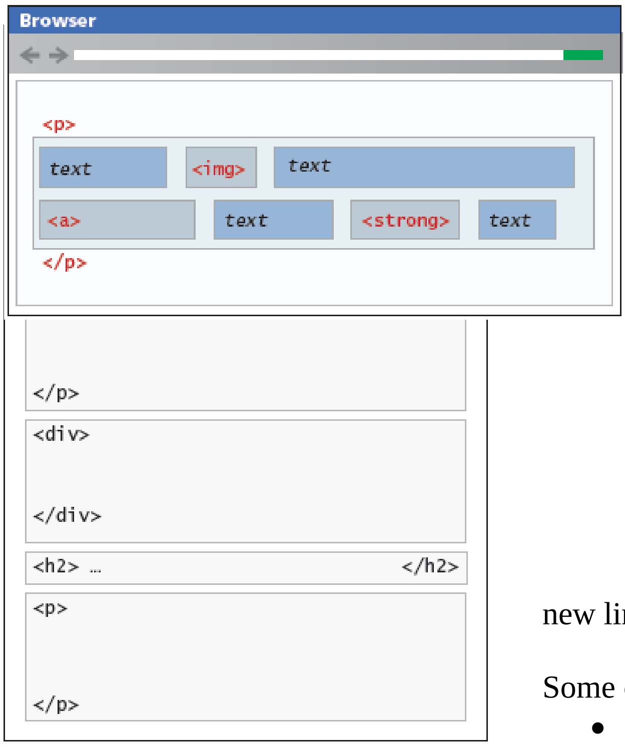
(c) How the block level and inline elements are displayed in the normal flow

Solution :

Block-level elements such as <p>, <div>, <h2>, <ul>, and <table> are elements that are contained on their own line, because block-level elements begin with a line break (new line). Two block-level elements can’t exist on the same line, without styling.

Some of the properties of block-level elements -

- Each block exists on its own line.
- It is displayed in normal flow from the browser window’s top to its bottom.
- By default each block level element fills up the entire width of its parent (browserwindow).
- CSS box model properties can be used to customize, for instance, the width of the boxand the margin space between other block level elements



Inline elements do not form their own blocks but instead are displayed within lines. Normal text in an HTML document is inline, and also elements such as <em>, <a>, <img>, and <span> are inline. Inline elements line up nextto one another horizontally from left to right on the same line, when there is no enough space left on the line, the content moves to a new line.

Some of the properties of inline elements are –

- Inline element is displayed in normal flow fromits container’s left to right.
- When a line is filled with content, the next linewill receive the remaining content, and so on.
- If the browser window resizes, then inline content will be “re-flowed” based on the new width.

5. (a) write Javascript code that uses function for the following problems:
   a. For the string input the output should be to display the position of left most vowel.
   b. For the numeric input output should be to display the reverse of a number

```
<html>
<body>
<script type="text/javascript">
function Reverse(str)
{
var num,rev=0,remainder;
num = parseInt(str);
while(num!=0)
{
remainder = num%10;
num = parseInt(num/10);
rev = rev * 10 + remainder;
}
alert("Reverse of "+str+" is "+rev);
}
function Vowel(str)
{
var str = str.toUpperCase();
for(var i = 0; i<str.length; i++)
{
var chr = str.charAt(i);
if(chr == 'A' || chr == 'E' || chr == 'I' || chr == 'O' || chr
== 'U')
break;
}
if(i<str.length )
alert("The position of the left most vowel is "+(i+1));
else
alert("No vowel found in the entered string");
}
var str = prompt("Enter the Input","");
if((isNaN(str)))
{
   Vowel(str);
}
else
{
Reverse(str);
}
</script>
</body>
</html>
```

(b)Explain the two approaches to embed php script in HTML with suitable and compare two approaches.

(c) What is DOM? Briefly explain the different types of nodes.

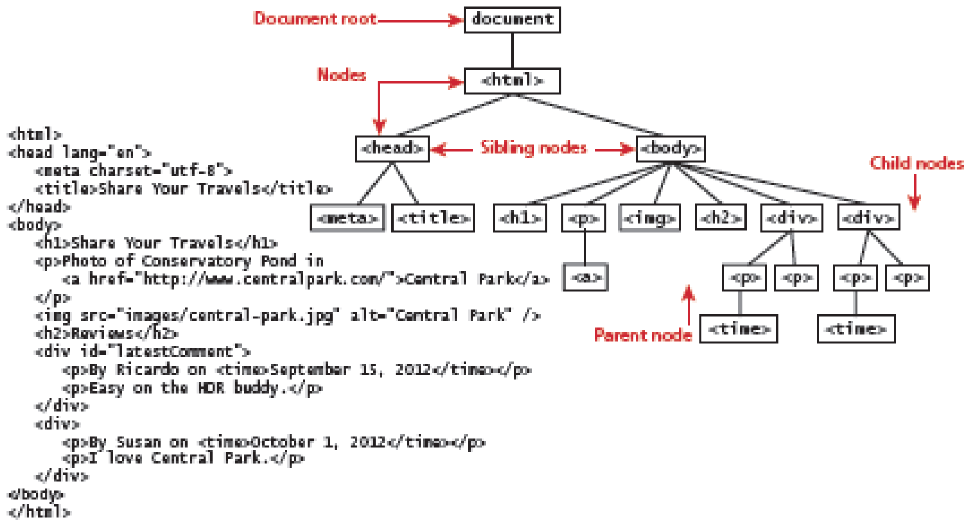
JavaScript is used to interact with the HTML document elements in which it is contained. The elements and attributes of HTML can be programmatically accessed, through an API called the **Document Object Model (DOM)**.

DOM is an API using which the javascript can dynamically access and modify html elements, its attributes and styles associated with it. Javascript can access, modify, add or delete the html elements.

Nodes

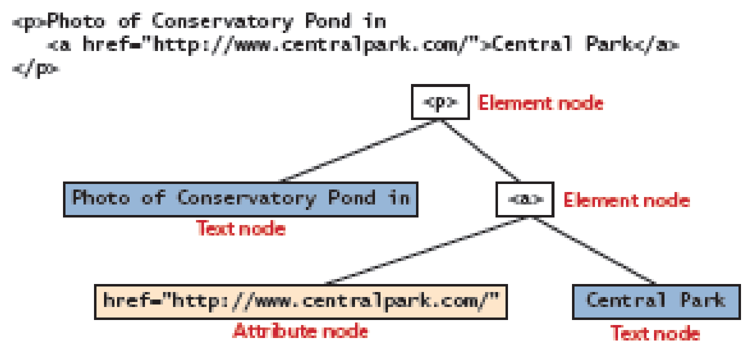
The html document with elements is considered as a tree structure called the DOM tree. Here each element of HTML document is called a **node**. Each node is an individual branch, with text nodes, and attribute nodes. Most of the tasks that we typically perform in JavaScript involve finding a node, and then accessing or modifying it via those properties and methods.

The DOM tree



```
<html>
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels</title>
</head>
<body>
   <h1>Share Your Travels</h1>
   <p>Photo of Conservatory Pond in
      <a href="http://www.centralpark.com/">Central Park</a>
   </p>
   <img src="images/central-park.jpg" alt="Central Park" />
   <h2>Reviews</h2>
   <div id="latestComment">
      <p>By Ricardo on <time>September 15, 2012</time></p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <div>
      <p>By Susan on <time>October 1, 2012</time></p>
      <p>I love Central Park.</p>
   </div>
</body>
</html>
```

Example of node and its attribute and text nodes:



```
<p>Photo of Conservatory Pond in
   <a href="http://www.centralpark.com/">Central Park</a>
</p>
```

Properties of a node object

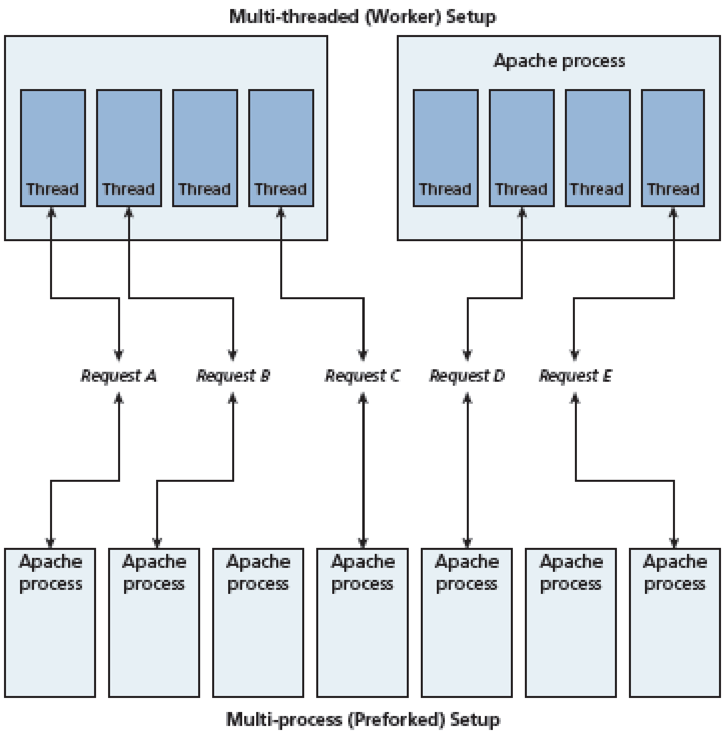
Property	Description
attributes	Collection of node attributes
childNodes	A NodeList of child nodes for this node
firstChild	First child node of this node
lastChild	Last child of this node
nextSibling	Next sibling node for this node
nodeName	Name of the node
nodeType	Type of the node
nodeValue	Value of the node
parentNode	Parent node for this node
previousSibling	Previous sibling node for this node.

6. Explain the php module in Apache and describe the differences between multi threaded and multi process setup.

Apache and PHP

As shown in the above figure, PHP is usually installed as an Apache module. The PHP module mod_php5 is sometimes referred to as the **SAPI** (Server Application Programming Interface) layer since it handles the interaction between the PHP environment and the web server environment.

Apache runs in two possible modes: **multi-process** (also called **preforked**) or **multi-threaded** (also called **worker**).



The default installation of Apache runs using the multi-process mode. That is, each request is handled by a separate process of Apache; the term **fork** refers to the operating system creating a copy of an already running process. Since forking is time intensive, Apache will prefork a set number of additional processes in advance of their being needed. A key advantage of multi- processing mode is that each process is insulated from other processes, that is, problems in one process can't affect other processes

In the multi-threaded mode, a smaller number of Apache processes are forked. Each of the processes runs multiple threads. A **thread** is like a lightweight process that is contained within an operating system process. A thread uses less memory than a process, and typically threads share memory and code; as a consequence, the multi-threaded mode typically scales better to large loads.

(b) Discuss the different ways the javascript can be included in HTML page and which is the most preferred way and why?

JavaScript can be linked to an HTML page in different ways.

1) **Inline JavaScript**

Inline JavaScript refers to the practice of including JavaScript code directly within certain HTML attributes. Use of inline javascript is general a bad practice and should be avoided.

Eg:

```
<input type="button" onclick="alert('Are you sure?');" />
```

2) **Embedded JavaScript**

Embedded JavaScript refers to the practice of placing JavaScript code within a <script> element Use of embedded JavaScript is done for quick testing and for learning scenarios, but is not accepted in real world web pages. Like with inline JavaScript, embedded scripts can be difficult to maintain.

Eg:

```
<script type="text/javascript">
        /* A JavaScript
        Comment */ alert
        ("Hello World!");
</script>
```

3) **External JavaScript**

Since writing code is a different than designing HTML and CSS, it is often advantageous to separate the two into different files. An external JavaScript file is linked to the html file as shown below.

```
<head>
        <script type="text/JavaScript" src="greeting.js">
        </script>
</head>
```

The link to the external JavaScript file is placed within the <head> element, just as was the case with links to external CSS files. It can also be placed anywhere within the <body> element. It is recommended to be placed either in the <head> element or the very bottom of the <body> element.

JavaScript external files have the extension .js. The file “greeting.js” is linked to the required html file. Here the linked is placed in the <head> tag. These external files typically contain function definitions, data definitions, and other blocks of JavaScript code. Any number of webpages can use the same external file.

(c)List the web server’s responsibilities.

- Server is responsible for answering all client requests. Even a simplest website must have a web server, to answer requests.

- Once a web server is configured and the IP address associated through a DNS server, it can then start listening for and answering HTTP requests.

- In the very simplest case the server is hosting static HTML files, and in response to a request sends the content of the file back to the requester.

- A web server has many responsibilities beyond responding to requests for HTML files. These include handling HTTP connections, responding to requests for static and dynamic resources, managing permissions and access for certain resources, encrypting and compressing data, managing multiple domains and URLs, managing database connections, cookies, and state, and uploading and managing files.

7. **(a) What are super globals? List the different super globals and briefly any two.**
Solution:

Superglobal Arrays

PHP uses special predefined associative arrays called **superglobal variables.** It allows the programmer to easily access HTTP headers, query string parameters, and other commonlyneeded information. They are called superglobal because these arrays are always accessible,from a function , class or file, without using the global keyword.

Some of the superglobal variables are –

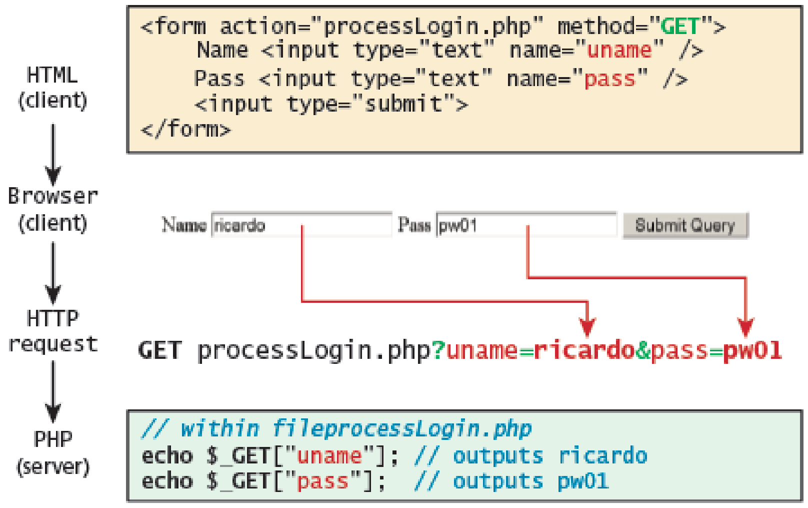
NAME	DESCRIPTION
$GLOBALS	Array for storing user data that needs superglobal scope
$_COOKIES	Array of cookie data passed to page via HTTP request
$_ENV	Array of server environment data
$_FILES	Array of file items uploaded to the server
$_GET	Array of query string data passed to the server via the URL
$_POST	Array of query string data passed to the server via the HTTP header
$_REQUEST	Array containing the contents of $_GET, $_POST, and $_COOKIES
$_SESSION	Array that contains session data

$_GET and $_POST Superglobal Arrays

The $_GET and $_POST arrays allows the programmer to access data sent by the client in a query string. They are the most important superglobal variables in PHP.

An HTML form allows a client to send data to the server. That data is formatted such that each value is associated with a name of control defined in the form. If the form was submitted usingan HTTP GET request ie. <form method=”get” action=”*server file path*” >, then the resulting URL will contain the data in the query string.
Eg-
https://name-processing.php?firstname=amith&lastname=kumar%20singh
retrieve the values sent through the URL using $_GET[“firstname”] and $_GET[“lastname”]

PHP will populate the superglobal $_GET array using the contents of this query string in the URL.

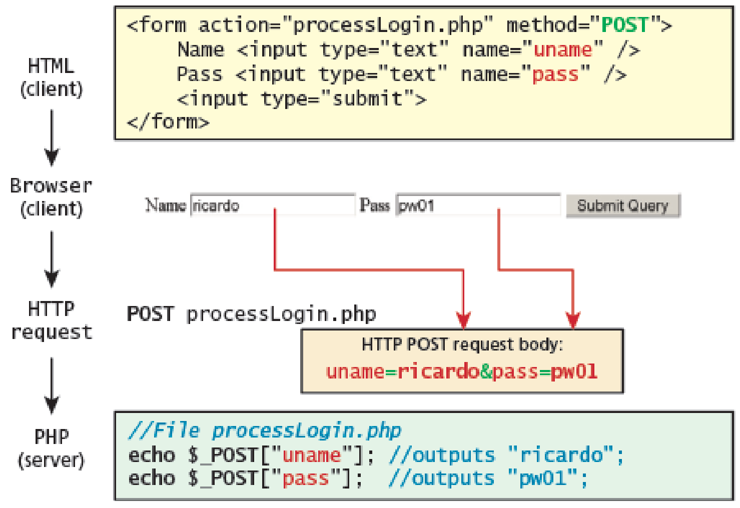


```
<form action="processLogin.php" method="GET">
    Name <input type="text" name="uname" />
    Pass <input type="text" name="pass" />
    <input type="submit">
</form>
```

```
GET processLogin.php?uname=ricardo&pass=pw01
```

```
// within fileprocessLogin.php
echo $_GET["uname"]; // outputs ricardo
echo $_GET["pass"];  // outputs pw01
```

If the form was sent using HTTP POST, then the values will be sent through HTTP POSTrequest body. The values and keys are stored in the $_POST array.



```
<form action="processLogin.php" method="POST">
    Name <input type="text" name="uname" />
    Pass <input type="text" name="pass" />
    <input type="submit">
</form>
```

```
POST processLogin.php
```

HTTP POST request body:

```
uname=ricardo&pass=pw01
```

```
//File processLogin.php
echo $_POST["uname"]; //outputs "ricardo";
echo $_POST["pass"];  //outputs "pw01";
```

Thus the values passed by the user can easily be accessed at the server side using the global arrays $_GET[] and $_POST[] for ‘get’ and ‘post’ methods respectively.

(b) Explain the support of object oriented design in PHP.

Object-Oriented Design

The object-oriented design of software offers many benefits in terms of modularity, testability, and reusability.

Objects can be reused across the program. The software is easier to maintain, as anychanges in the structure need to change only the class. OO concepts enables faster development and easier maintenance of the software.

Data Encapsulation

Object-oriented design enables the possibility of **encapsulation** (hiding), that is, restrictingaccess to an object's internal components (properties and methods). Another way of understanding encapsulation is: it is the hiding of an object's implementation details.

In properly encapsulated class, the properties and methods, are hidden using the private access specifier and these properties and methods are accessed to outside the class using the public methods. Thus we can restrict the usage of required properties and methods.

The hidden properties are accessed and modified using public methods commonly called **getters and setters** (or accessors and mutators). A getter returns a variable's value does not modify the property. It is normally called without parameters, and returns the property from within the class.Eg -

```
  public function
            getFirstNa
            me() {
            return
            $this-
            >firstName
            ;
}
```

Setter methods modify properties, and allow extra logic to be added to prevent properties frombeing set to strange values.

Eg –

```
public function setFirstName($name) {
            $this->firstName = $name;
}
```

The below example shows the modified Student class with getters and setters. Some of theproperties are private. These properties cannot be accessed or assigned from outside the class,
the getters and setters.

```
<?php
Class Student
```

```
{
        private $name;
        private $USN;
        public $address;
        public $avg;

function __construct($n,$usn,$add,$avg)
{
        $this->name = $n;
        $this->USN= $usn;
        $this->address = $add;
        $this->avg = $avg;
}

public function getname() { return
$this->name; }public function
getUSN() { return $this->USN; }

public function setname($fullname) { $this-
>name=$fullname; }public function setUSN($usn) {
$this->USN=$usn; }
}

$s1 = new Student("Sajeev", "1VA15CS013","Bangalore",66);
$s1->setname("Adithya");
$s1->avg = 77;          //setting avg directly , where as name is set using the function
$name = $s1->getname();
$usn= $s1->getUSN();
echo "name is $name <br /> USN is $usn <br />";
echo "address is $s1->address <br /> average is $s1->avg";

?>
```

Note: '$s1->address' and '$s1->avg' – can be accessed directly, without using any get function.

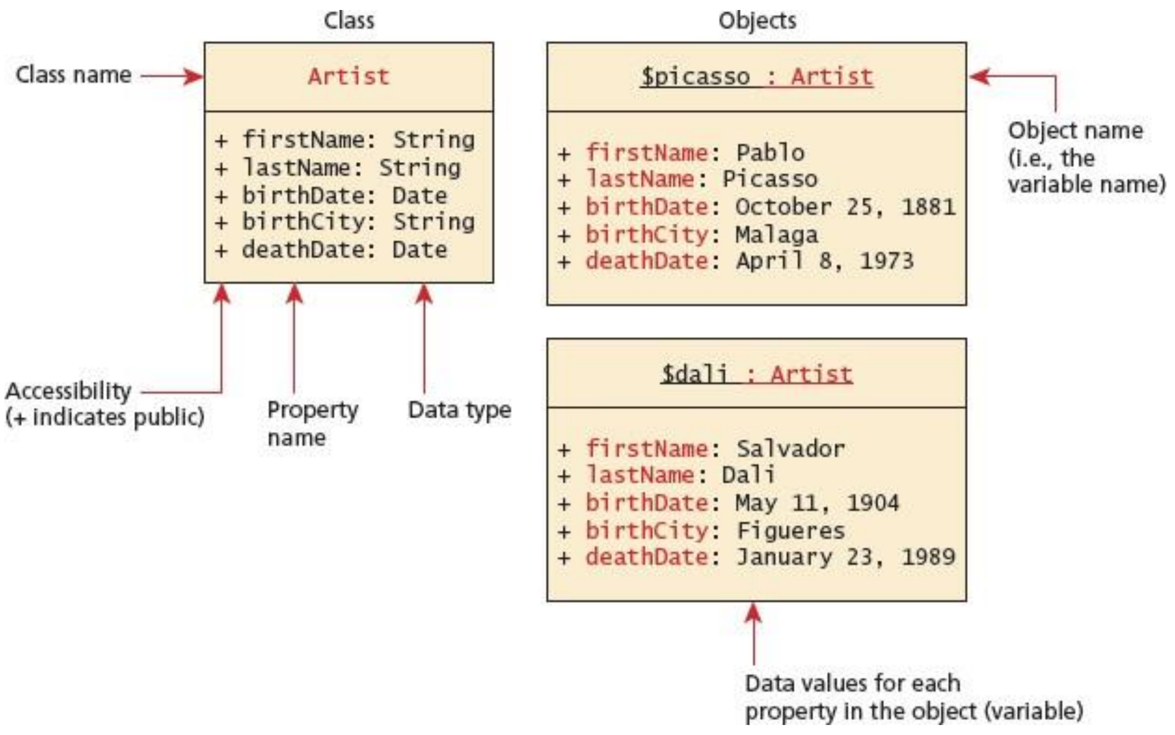
(c) Write a php code that checks for valid mime type and file extension

Limiting the Type of File Upload and extension

To check the type of file, check the file extension and the type of file using type key of $_FILESarray.

```
$validExt = array("jpg", "png");
$validMime =
array("image/jpeg","image/png");
foreach($_FILES as $fileKey =>
$fileArray )
{
        $extension = end(explode(".", $fileArray["name"]));
        if (in_array($fileArray["type"],$validMime) && in_array($extension, $validExt))
        {
      echo "all is well. Extension and mime types valid";
        }
        else

{ echo $fileKey." Has an invalid mime type or extension";
        }
        }
```

8. (a) Write a PHP program to create a class called "Artist" with suitable constructor. All its data members are accessible inside the class
Date members : First name, last name, birth city, birth date
Data functions " getters and setters
Using above class instantiate two objects and displays the artist details



(b) Explain the two techniques provided in php reading / writing files and also list comparative advantages and disadvantages

Reading/Writing Files

There are two basic techniques for read/writing files in PHP:

- **Stream access** In this technique, just a small portion of the file is read at a time. It is the most memory-efficient approach when reading very large files.
- **All-In-Memory access** In this technique, the entire file is read into memory (i.e., into a PHP variable). While not appropriate for large files, it does make processing of the file extremely easy.

Stream Access

The functions used in this technique are similar to that of C programming.

The function fopen() takes a file location or URL and access mode as parameters. The returnedvalue is a **stream resource (file handle)**.

Some of the common modes are “r” for read, “w” for write, and “c” creates a new file forwriting, “rw” for read and write.

The other functions are

– fclose(file handle)-

closing the file

fgets(file handle)- To read a single line, returns 0 if no more data fread(file handle, no.of bytes) - To read an specified amount of datafwrite(file handle, string)- To write the string to a file

fgetc(file handle) – To read a single characterfeof(file handle) – checks for EOF

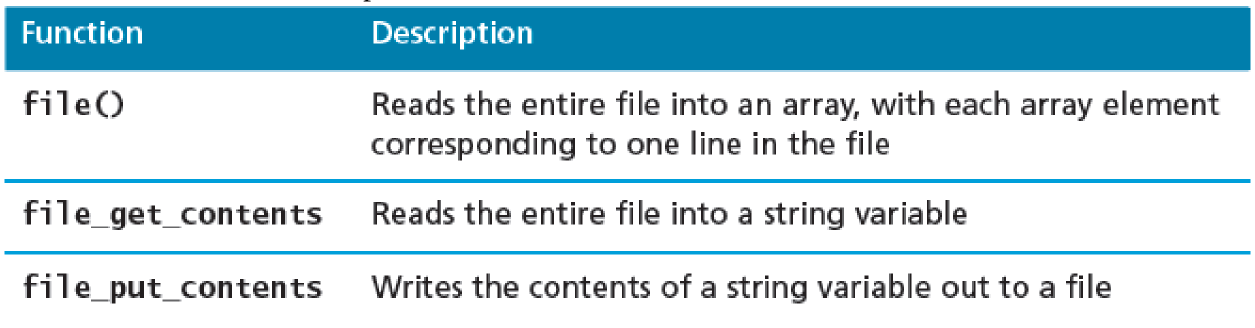
A program to read from a file and display it -

```
<?php
$f =
fopen("sample.txt
", "r");while
($line = fgets($f))
{
        echo $line . "<br>";
}
fclose($f);

      ?>
```

In-Memory File Access

While the previous approach to reading/writing files requires more care in dealing with the streams, file handles etc. The alternative simpler approach is to read/write the entire contents intothe memory with the help of variable.

The alternative functions to process the file are –



Function	Description
`file()`	Reads the entire file into an array, with each array element corresponding to one line in the file
`file_get_contents`	Reads the entire file into a string variable
`file_put_contents`	Writes the contents of a string variable out to a file

The file_get_contents() and file_put_contents() functions allow you to read or write an entire filein one function call. To read an entire file into a variable you can simply

use:
$fileAsString = file_get_contents(FILENAME);

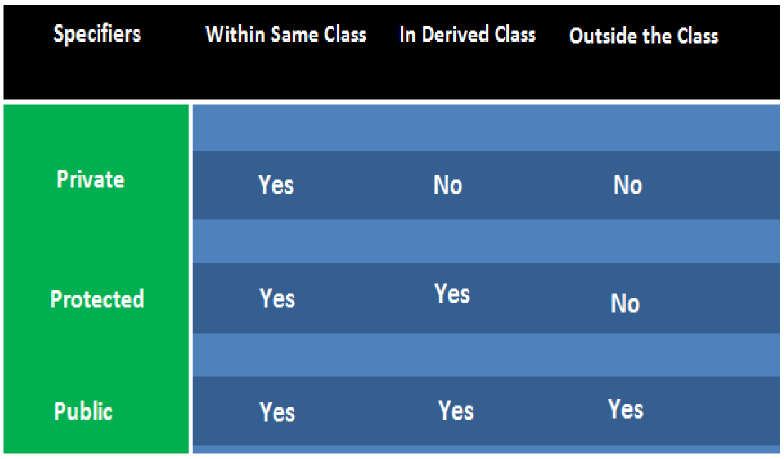
To write the contents of a string $writeme
to a file, use
file_put_contents(FILENAME, $writeme);

(c) What is a visibility of class member? Briefly explain the different levels of visibility.

Visibility

The **visibility** of a property or method determines the accessibility of a **class member** (i.e., aproperty or method) and can be set to public, private, or protected.
The public keyword means that the property or method is accessible to any code anywhere that has a reference to the object.
The private keyword sets a method or variable to only be accessible from within the class. This means that we cannot access or modify the property from outside of the class, even if it is referenced with an object.
The protected keyword sets a method or variable to be accessible from within the class and from its derived classes.

In UML, the "+" symbol is used to denote public properties and methods, the "–" symbol for private members, and the "#" symbol for protected members.



Specifiers	Within Same Class	In Derived Class	Outside the Class
Private	Yes	No	No
Protected	Yes	Yes	No
Public	Yes	Yes	Yes

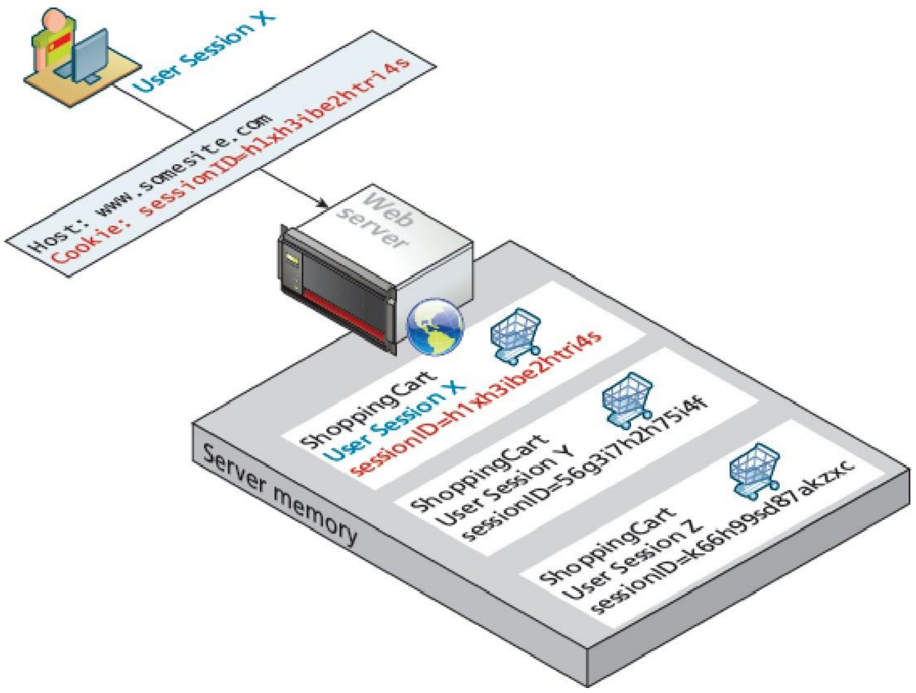
9. (a) What is session state? How does session state works with suitable example.

Session state is a server-based state mechanism that allows web applications to store and retrieveobjects of any type for each unique user session. That is, each browser session has its own session state stored as a serialized file on the server, which is deserialized and loaded into memory as needed for each request.

As server storage is a finite resource, objects loaded into memory are released when the request completes, making room for other requests and their session objects. This means there can be more active sessions on disk than in memory at any one time. Session state is ideal for storing more complex objects or data structures that are associated with a user session.

How Does Session State Work?

The session state works within the same HTTP context as any web request. Sessions in PHP are identified with a unique session ID. In PHP, this is a unique 32-byte string that is by default transmitted back and forth between the user and the server via a session cookie.



After generating or obtaining of a session ID for a new session, PHP assigns an initially empty dictionary-style collection that can be used to hold any state values for this session. When therequest processing is finished, the session state is saved to some type of state storage mechanism,called a session state provider. Finally, when a new request is received for an already existingsession, the session's dictionary collection is filled with the previously saved session data fromthe session state provider.

(b) Demonstrate the manipulation of attributes, properties and styles of the element using jQuery with suitable examples.

jQuery Attributes

Any set of elements from a web page can be selected. In order to fully manipulate the elements,one must understand an element’s *attributes* and *properties*.

HTML Attributes

The core set of attributes related to DOM elements are the ones specified in the HTML tags. In jQuery we can both set and get an attribute value by using the attr() method on any elementfrom a selector. This function takes a parameter to specify which attribute, and the optional

second parameter is the value to set it to. If no second parameter is passed, then the return valueof the call is the current value of the attribute. Some example usages are:

// var link is assigned the href attribute of the first <a> tag

var link = $("a").attr("href");

// change all links in the page to http://funwebdev.com

$("a").attr("href","http://funwebdev.com");

// change the class for all images on the page to fancy

$("img").attr("class","fancy");

HTML Properties

Many HTML tags include properties as well as attributes, the most common being the *checked* property of a radio button or checkbox. In early versions of jQuery, HTML properties could be set using the attr() method. However, since properties are not technically attributes, this resulted in odd behavior. The prop() method is now the preferred way to retrieve and set the value of a property although, attr() may return some (less useful) values.
To illustrate this subtle difference, consider a DOM element defined by
<input class ="meh" type="checkbox" checked="checked">
The value of the attr() and prop() functions on that element differ as shown below.

```
var theBox = $(".meh");
theBox.prop("checked") //
evaluates to TRUE
theBox.attr("checked") //
evaluates to "checked"
```

Changing CSS

Changing a CSS style is syntactically very similar to changing attributes. jQuery provides the extremely intuitive css() methods. There are two versions of this method (with two different method signatures), one to get the value and another to set it. The first version takes a single parameter containing the CSS attribute whose value you want and returns the current value.

```
$color = $("#colourBox").css("background-color"); // get the color
```

To modify a CSS attribute you use the second version of css(), which takes two parameters: thefirst being the CSS attribute, and the second the value.

```
// set color to red
$("#colourBox").css("background-color", "#FF0000");
```

If you want to use classes instead of overriding particular CSS attributes individually, have a look at the additional shortcut methods described in the jQuery documentation.

Shortcut Methods

jQuery allows the programmer to rely on foundational HTML attributes and properties exclusively as described above. However, as with selectors, there are additional functions that provide easier access to common operations such as changing an object's class or the text within an HTML tag.
The html() method is used to get the HTML contents of an element (the part between the <> and
</> tags associated with the innerHTML property in JavaScript).

(c) What is JASON? Explain with the code example, how to convert string to Jason and vice versa.

JSON is a data serialization format, like XML. That is, it is used to represent object data in a text format so that it can be transmitted from one computer to another. Many REST web services encode their returned data in the JSON data format instead of XML. While **JSON** stands for **JavaScript Object Notation**, its use is not limited to JavaScript. It was originally designed to provide a lightweight serialization format to represent objects in JavaScript. While it doesn't have the validation and readability of XML, it has the advantage of generally requiring significantly fewer bytes to represent data than XML.

1. Using JSON in PHP

PHP comes with a JSON extension . Converting a JSON string into a PHP object is quitestraightforward:

```
<?php
// convert JSON string into PHP object
$text = '{"artist": {"name":"Manet","nationality":"France"}}';
$anObject =
json_decode($text);
echo $anObject->artist-
>nationality;
 // convert JSON string into PHP associative array

$anArray =
json_decode($text, true);
echo
$anArray['artist']['nation
ality'];
?>
```

The json_decode() function can return either a PHP object or an associative array. Since JSON data is often coming from an external source, we should check for parse errors before using it, which can be done via the json_last_error() function:

```
<?php
 // convert JSON string into PHP object

$text = '{"artist": {"name":"Manet","nationality":"France"}}';
$anObject = json_decode($text);
 // check for parse errors

if (json_last_error() ==
JSON_ERROR_NONE) {echo
$anObject->artist->nationality;
}
?>
```

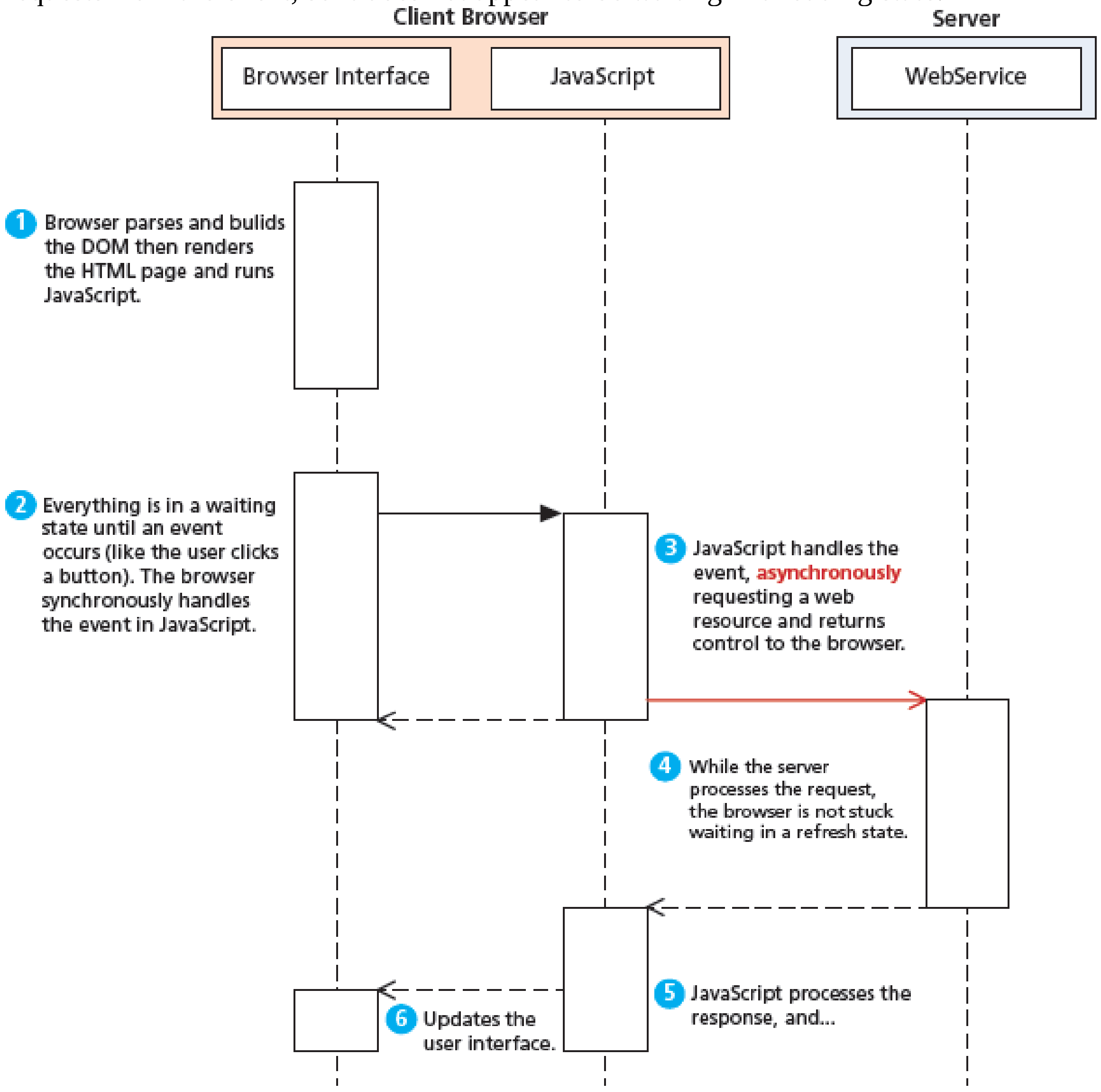
To convert a PHP object into a JSON string, use the json_encode() function.

```
 // convert PHP object into a JSON string

$text = json_encode($anObject);
```

10. (a) What is AJAX? Using an XML diagram, explain how the synchronous requeste is handled.

Asynchronous JavaScript with XML (AJAX) is a term used to describe a paradigm that allows a web browser to send messages back to the server without interrupting the flow of what's being shown in the browser. This makes use of a browser's multi-threaded design and lets one thread handle the browser and interactions while other threads wait for responses to asynchronous requests.

The below figure annotates a UML sequence diagram where the white activity bars illustrate where computation is taking place. Between the request being sent and the response being received, the system can continue to process other requests from the client, so it does not appear to be waiting in a loading state.



Client Browser
Server
Browser Interface
JavaScript
WebService
1 Browser parses and bulids the DOM then renders the HTML page and runs JavaScript.
2 Everything is in a waiting state until an event occurs (like the user clicks a button). The browser synchronously handles the event in JavaScript.
3 JavaScript handles the event, asynchronously requesting a web resource and returns control to the browser.
4 While the server processes the request, the browser is not stuck waiting in a refresh state.
5 JavaScript processes the response, and...
6 Updates the user interface.

Responses to asynchronous requests are caught in JavaScript as events. The events can subsequently trigger changes in the user interface or make additional requests. This differs from the typical synchronous requests we have seen thus far, which

require the entire web page to refresh in response to a request.
Another way to contrast AJAX and synchronous JavaScript is to consider a web page that displays the current server time. If implemented synchronously, the entire page has to be refreshed from the server just to update the displayed time. During that refresh, the browser enters a waiting state, so the user experience is interrupted.

(b) Explain the loading and processing of an xml document in javascript with suitable example.

1. XML Processing in JavaScript

All modern browsers have a built-in XML parser and their JavaScript implementations support an **in-memory** XML DOM API, which loads the entire document into memory where it is transformed into a hierarchical tree data structure.

The DOM functions such as getElementById(), getElementsByTagName(), and createElement() are used to access and manipulate the data.
For instance, the below code shows the loading of an XML document into an XML DOM object,and it displays the id attributes of the <painting>elements as well as the content of each painting's <title> element

```
<script>
 // load the external XML file
 xmlhttp.open("GET","art.x
 ml",false);xmlhttp.send();
 xmlDoc=xmlhttp.responseX
 ML;


 // now extract a node list of all <painting> elements

paintings = xmlDoc.getElementsByTagName("painting");

if (paintings) {
        // loop through each painting element

        for (var i = 0; i < paintings.length; i++)
        {
                // display its id attribute

                alert("id="+paintings[i].getAttribute("id"));
                // find its <title> element

                title =
                paintings[i].getElementsByTagName(
                "title");if (title) {
                    // display the text content of the <title> element
                    alert("title="+title[0].textContent);
                    }
          }
}
</script>
```

JavaScript supports a variety of node traversal functions as well as properties for accessinginformation within an XML node.

alert() function – displays the string in a
alert dialog box.GetAttribute(id) - returns
the value of the attribute.

getElementsByTagName(tag) – returns an array of tags with the specified title.textContent() – returns the html content in the specified tag

(c) Using functions, emulate a class with data members and member functions in javascript.

Emulate Classes through Functions

Although a formal *class* mechanism is not available to us in JavaScript,
it is possibleto get close by using functions to encapsulate variables and
methods together, as shown below.

```
function
Die(col) {
this.color=
col;
this.faces=[
1,2,3,4,5,6]
;
}
```

The ‘this’ keyword inside of a function refers to the instance, so that every reference
to internal properties or methods manages its own variables, as is the case with PHP.
One can create an instance of the object as follows, very similar to PHP.

```
var oneDie = new Die("0000FF");
```