

--	--	--	--	--	--	--	--	--	--

Internal Assessment Test 1 Answer Key– NOV. 2019

Sub:	Web Technologies				Sub Code:	18MCA13	Branch:	MCA
Date:	18/11/2019	Duration:	90 min's	Max Marks:	50	Sem	III	

**O1) Explain request phase and response phase of HTTP [10]**

The general form of an HTTP request is as follows:

1. HTTP method Domain part of the URL HTTP version
2. Header fields
3. Blank line
4. Message body

The following is an example of the first line of an HTTP request:

GET /storefront.html HTTP/1.1

Table 1.1 HTTP request methods

Method	Description
GET	Returns the contents of the specified document
HEAD	Returns the header information for the specified document
POST	Executes the specified document, using the enclosed data
PUT	Replaces the specified document with the enclosed data
DELETE	Deletes the specified document

The format of a header field is the field name followed by a colon and the value of the field. There are four categories of header fields:

1. General: For general information, such as the date
2. Request: Included in request headers
3. Response: For response headers
4. Entity: Used in both request and response headers

Accept: text/plain  
Accept: text/html  
Can be written as  
Accept: text/\*

A wildcard character, the asterisk (\*), can be used to specify that part of a MIME type can be anything.

The Host: host name request field gives the name of the host. The Host field is required for HTTP1.1.

The If-Modified-Since: date request field specifies that the requested file should be sent only if it has been modified since the given date. If the request has a body, the length of that body must be given with a Content-length field. The header of a request must be followed by a blank line, which is used to separate the header from the body of the request.

The general form of an HTTP response is as follows:

1. Status line
2. Response header fields
3. Blank line
4. Response body

The status line includes the HTTP version used, a three-digit status code for the response, and a short textual explanation of the status code. For example, most responses begin with the following:

HTTP/1.1 200 OK

The status codes begin with 1, 2, 3, 4, or 5. The general meanings of the five categories specified by these first digits are shown in Table 1.2.

Table 1.2 First digits of HTTP status codes

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

One of the more common status codes is one user never want to see: 404 Not Found, which means the requested file could not be found.

**Q2a) Explain Internet Protocol with appropriate example? [5]**

Internet Protocol Addresses

The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32-bit number.

- IP addresses usually are written (and thought of) as four 8-bit numbers, separated by periods.
- The four parts are separately used by Internet-routing computers to decide where a message must go next to get to its destination.
- Although people nearly always type domain names into their browsers, the IP works just as well.
- For example, the IP for United Airlines (www.ual.com) is 209.87.113.93. So, if a browser is pointed at http://209.87.113.93, it will be connected to the United Airlines Web site.

**Q2b) Explain DNS with neat diagram**

The IP addresses are numbers. Hence, it would be difficult for the users to remember IP address. To solve this problem, text based names were introduced. These are technically known as **domain name system (DNS)**.

These names begin with the names of the host machine, followed by progressively larger enclosing collection of machines, called **domains**. There may be two, three or more domain names. DNS is of the form **hostname.domainName.domainName**.

The steps for conversion from DNS to IP:

- The DNS has to be converted to IP address before destination is reached.
- This conversion is needed because computer understands only numbers.
- The conversion is done with the help of *name server*.
- As soon as domain name is provided, it will be sent across the internet to contact name servers.
- This name server is responsible for converting domain name to IP
- If one of the *name servers* is not able to convert DNS to IP, it contacts other name server.
- This process continues until IP address is generated.
- Once the IP address is generated, the host can be accessed.
- The hostname and all domain names form what is known as FULLY QUALIFIED DOMAIN NAME.

This is as shown below:

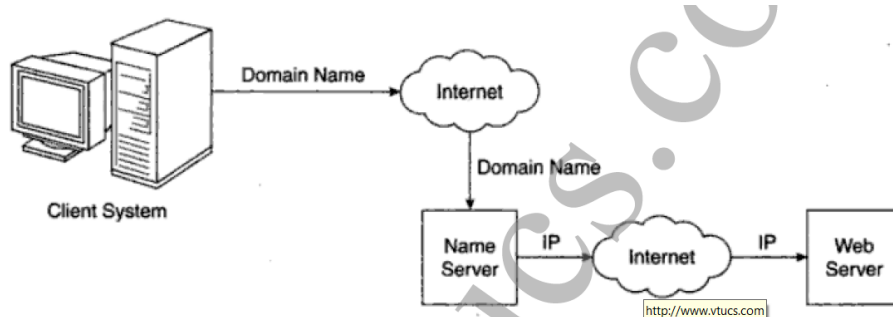


Figure 1.1 Domain name conversion

**Q3) Explain basic text markup tags with syntax.**

## 1) Paragraphs

## 2) Line Breaks

## 3) Preserving whitespace

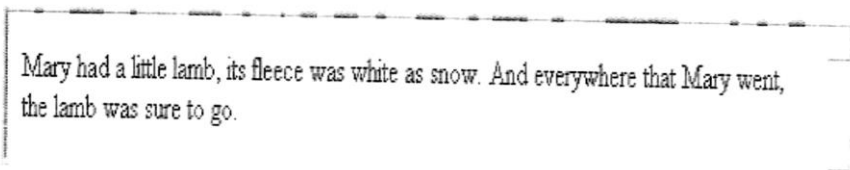
## 4) Headings

## 5) Block Quotations

### 1) Paragraphs

- XHTML does not allow text to be directly placed into the document. Text is normally organised into paragraphs.
- It begins with `<p>` and ends with `</p>`. Multiple paragraphs may appear in a single document

```
<p>
  Mary had
a
  little lamb, its fleece was white as snow. And
everywhere that
  Mary went, the lamb
was sure to go.
</p>
```



Mary had a little lamb, its fleece was white as snow. And everywhere that Mary went, the lamb was sure to go.

### 2) Line Breaks

- Sometime text requires a line break without the preceding blank line. This exactly what the break tag does.
- Break tag does not have any content therefore it is self-closing tag
- The break tag is specified as `<br />`. The slash indicates that the tag is both an opening and closing tag.

```
<p>
  Mary had a little lamb, <br />
  its fleece was white as snow.
</p>
```

### 3) Preserving Whitespace

Sometimes it is desirable to preserve the white space in text—that is, to prevent the browser from eliminating multiple spaces and ignoring embedded line breaks. This can be specified with the **<pre>** tag.

```

<p><pre>
Mary
    had a
        little
            lamb
</pre>

```

#### 4) Headings

In XHTML, there are six levels of headings, specified by the tags <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>, where <h1> specifies the highest-level heading. Headings are usually displayed in a boldface font whose default size depends on the number in the heading tag. On most browsers, <h1>, <h2>, and <h3> use font sizes that are larger than that of the default size of text, <h4> uses the default size, and <h5> and <h6> use smaller sizes. The heading tags always break the current line, so their content always appears on a new line. Browsers usually insert some vertical space before and after all headings

```

<html>
  <head><title> Headings </title></head>
  <body>
    <h1> Heading 1 </h1>
    <h2> Heading 2 </h2>
    <h3> Heading 3 </h3>
    <h4> Heading 4 </h4>
    <h5> Heading 5 </h5>
    <h6> Heading 6 </h6>
  </body>
</html>

```

#### 5) Block Quotations

The <blockquote> tag is used to make the contents look different from the surrounding text.

```
<p>Here is a quote from WWF's website:</p>
```

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

**For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.**

```
</blockquote>
```

#### About WWF

Here is a quote from WWF's website:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

#### Q4) Explain the following tags with example i)Image ii)Link iii)List

##### i)Image

- Image can be displayed on the web page using <img> tag.
- When the <img> tag is used, it should also be mentioned which image needs to be displayed. This is done using src attribute.
- Attribute means extra information given to the browser

- Whenever <img> tag is used, alt attribute is also used.
- Alt stands for alert.
- Some very old browsers would not be having the capacity to display the images.
- In this case, whatever is the message given to alt attribute, that would be displayed.
- Another use of alt is ☒ when image display option has been disabled by user. The option is normally disabled when the size of the image is huge and takes time for downloading.

```
<html>
  <head> <title>display image</title> </head>
  <body>
    
  </body>
</html>
```

## ii) Link

Hyperlinks are the mechanism which allows the navigation from one page to another.

The term “hyper” means beyond and “link” means connection

Whichever text helps in navigation is called hypertext

Hyperlinks can be created using <a> (anchor tag)

The attribute that should be used for <a> is href

Program: hyper.html

```
<html>
  <head>
    <title> hyperlink </title>
  </head>
  <a href = "link.html"> CLICK HERE </a>
</html>
```

## iii)List

### Unordered Lists:

The <ul> tag, which is a block tag, creates an unordered list. Each item in a list is specified with an <li> tag (li is an acronym for list item). Any tags can appear in a list item, including nested lists. When displayed, each list item is implicitly preceded by a bullet.

```
<html>
  <head> <title> Unordered      List </title> </head>
  <body>
    <h1>Engine Aircraft </h1>
    <ul>
      <li>Cessna Skyhawk</li>
      <li>Beechcraft Bonanza</li>
    </ul>
  </body>
</html>
```

### Ordered Lists:

Ordered lists are lists in which the order of items is important. This orderedness of a list is shown in the display of the list by the implicit attachment of a sequential value to the beginning of each item. The default sequential values are Arabic numerals, beginning with 1. An ordered list is created within the block tag <ol>. The items are specified and displayed just as are those in unordered lists, except that the items in an ordered list are preceded by sequential values instead of bullets.

```
<html>
  <head> <title> Unordered      List </title> </head>
  <body>
    <h1>Engine Aircraft </h1>
    <ol>
      <li>Cessna Skyhawk</li>
```

```

        <li>Beechcraft Bonanza</li>
    </ol>
</body>
</html>

```

#### Definition Lists:

As the name implies, definition lists are used to specify lists of terms and their definitions, as in glossaries. A definition list is given as the content of a <dl> tag, which is a block tag. Each term to be defined in the definition list is given as the content of a <dt> tag. The definitions themselves are specified as the content of <dd> tags. The defined terms of a definition list are usually displayed in the left margin; the definitions are usually shown indented on the line or lines following the term.

```

<html>
<head> <title> Definition List </title> </head>
<body>
    <h1> Cessna Skyhawk </h1>
    <dl>
        <dt>152 </dt>
        <dd>Two-place trainer</dd>
        <dt> 172 </dt>
        <dd>Smaller four-place airplane</dd>
        <dt> 120 </dt>
        <dd>Six-place airplane- high performance</dd>
    </dl>
</body>
</html>

```

### Q5) Explain Frameset and frame tags with examples

#### 1) Framesets:

- The number of frames and their layout in the browser window are specified with the <frameset> tag.
- A frameset element takes the place of the body element in a document. A document has either a body or a frameset but cannot have both.
- The <frameset> tag must have either a *rows* or a *cols* attribute. (or both)
- To create horizontal frames, *rows* attribute is used.
- To create vertical frames, *cols* attribute is used.
- The values for these attributes can be numbers, percentages and asterisks. Two or more values are separated by commas & given in quoted string.
  - Number- specify height of one row in pixels.
 

```
<frameset rows="200, 300, 400">
```
  - Percentages – Given as number followed immediately by percentage sign. Percentage value specify the percentage of total browser window height that a row should occupy.
 

```
<frameset rows="22%, 33%, 45%">
```
  - Asterisks – it means the remainder of window height
 

```
<frameset rows="22%, 33%, *">
```
- The *cols* attribute is very much like the *rows* attribute, except it specifies the number of columns of frames.
 

```
<frameset rows = " 23%, 33%, 33% " cols= " 25%, *" >
```

Content of frame 1	Content of frame 2
Content of frame 3	Content of frame 4
Content of frame 5	Content of frame 6

## 2) Frames

- The content of frame is specified with <frame> tag, which can appear only in the content of a frameset element.
- The frames in the frameset appear by rows.  

```
<frame src="apples.html">
```
- The content of a frame is specified as the value of the src attribute in the <frame> tag.
- If <frame> tag has no src attribute browser displays empty frame.
- If the content of frame does not fit in given frameset scroll bars are implicitly included.
- If you want frame to have scroll bar, regardless of the size of its content, the <frame> attribute scrolling can be set to yes

**Q6) Write a XHTML program to create following table[10]**

Time Table					
	Mon	Tue	Wed	Thu	Fri
Hours	Science	Maths	Science	Maths	Arts
	Social	History	English	Social	Sports
	Lunch				
	Science	Maths	Science	Maths	Project
	Social	History	English	Social	

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Time table</title>
  </head>
  <body>
    <table border="1" cellpadding="0" cellspacing="0">
      <tr align="center">
        <th colspan="6">Time Table </th>
      </tr>
      <tr>
        <th rowspan="6">Hours</th>
        <th>Mon</th>
        <th>Tue</th>
```

```

        <th>Wed</th>
        <th>Thu</th>
        <th>Fri</th>
    </tr>
    <tr>
        <td>Science</td>
        <td>Maths</td>
        <td>Science</td>
        <td>Maths</td>
        <td>Arts</td>
    </tr>
    <tr>
        <td>Social</td>
        <td>History</td>
        <td>English</td>
        <td>Social</td>
        <td>Sports</td>
    </tr>
    <tr align="center">
        <td colspan="5">Lunch</td>
    </tr>
    <tr>
        <td>Science</td>
        <td>Maths</td>
        <td>Science</td>
        <td>Maths</td>
        <td rowspan="2">Project</td>
    </tr>
    <tr>
        <td>Social</td>
        <td>History</td>
        <td>English</td>
        <td>Social</td>
    </tr>
</table>
</body>
</html>

```

**Q7) Explain HTML 5 Media Elements (<audio> & <video>) and form element attributes (search, range, number, email) with appropriate example[10]**

### audio

```

<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>

```

The <audio> tag defines sound, such as music or other audio streams.

Currently, there are 3 supported file formats for the <audio> element: MP3, WAV, and OGG

### Video

```

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">

```



```
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

The <video> tag specifies video, such as a movie clip or other video streams.

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg

- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec

Ogg = Ogg files with Theora video codec and Vorbis audio codec

## **Q8) Explain HTML 5 semantic elements (<nav>, <section>, <article>, <aside>, <header>, <footer> ) with example.[10]**

### **1) Nav**

The <nav> tag is a new element in HTML5. It is used to define a block of navigation links, either within the current document or to other documents. Examples of navigation blocks are menus, tables of contents, and indexes.

One HTML document may contain several <nav> tags, for example, one for site navigation and one for intra-page navigation.

Note that not all links in the HTML document can be placed inside the <nav> element; it can only include major navigation blocks. For example, the <nav> tag is not placed in the <footer> tag for defining links in the footer of the website.

Example:

```
<nav>
  <a href="/learn-html.html">HTML</a> | <a href="/learn-css.html">CSS</a> | <a href="/learn-
  javascript.html">JavaScript</a> | <a href="/learn-php.html">PHP</a> |
</nav>
```

### **2) Section**

The HTML <section> tag specifies a section in a document.

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Section Tag</title>
  </head>

  <body>
    <section>
      <h1>Java</h1>
      <h3>Inheritance</h3>
      <p>Inheritance defines the relationship between superclass and subclass.</p>
    </section>
  </body>

</html>
```

### 3) Article

The <article> tag specifies independent, self-contained content.

An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

Potential sources for the <article> element:

Forum post  
Blog post  
News story  
Comment

```
<article>
  <h1>Google Chrome</h1>
  <p>Google Chrome is a free, open-source web browser developed by Google, released in 2008.</p>
</article>
```

### 4) Aside

The <aside> tag defines some content aside from the content it is placed in.

The aside content should be related to the surrounding content.

```
<p>My family and I visited The Epcot center this summer.</p>
```

```
<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
```

### 5) Header

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

one or more heading elements (<h1> - <h6>)

logo or icon

authorship information

You can have several <header> elements in one document.

Note: A <header> tag cannot be placed within a <footer>, <address> or another <header> element.

```
<article>
  <header>
    <h1>Most important heading here</h1>
    <h3>Less important heading here</h3>
    <p>Some additional information here</p>
  </header>
  <p>Lorem Ipsum dolor set amet....</p>
</article>
```

### 6) Footer

The <footer> tag defines a footer for a document or section.

A <footer> element should contain information about its containing element.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

### **Q9) Explain the various levels of stylesheet with example for each[10]**

1)Inline Style Specification: appears as values of the style attribute of a tag, the general form is as follows:

Style = "Property1 : Value1; Property2 : Value2; Property3 : Value3; ..... Property\_n:Value\_n;"  
It is recommended that last property/value pair be followed by a semicolon.

Eg: <h1 style ="font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red;">Web Technology</h1>

2)Document Style Specification: appears as the content of a style element within the header of a document, general form of the content of a style element is as follows:

<style type = "text/css"> Rule list </style>

Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property–value pairs. The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

Selector { Property1 : Value1; Property2 : Value2; Property3 : Value3; ..... Property\_n:Value\_n; }

Eg: <style type = "text/css"> h1 { font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red; } </style>

3) External Style Sheet: have a form similar to that of document style sheets. The external file consists of a list of style rules.

Eg <head> <link rel="stylesheet" type="text/css" href="cssfile.css"> </head>

Cssfile.css P{ Background-color:blue; Color:red; }

### **Q10) Explain the various selector forms with examples[10]**

1) Simple Selector Forms:

In case of simple selector, a tag is used. If the properties of the tag are changed, then it reflects at all the places when used in the program. The selector can be any tag. If the new properties for a tag are not mentioned within the rule list, then the browser uses default behaviour of a tag. Eg: h1 { font-size : 24pt; } h2, h3{ font-size : 20pt; } body b em { font-size : 14pt; } Only applies to the content of 'em' elements that are descendent of bold element in the body of the document. This is a contextual selector

2) Class Selectors:

Class selectors are used to allow different occurrences of the same tag to use different style specifications. Eg <head> <style type = "text/css"> p.one { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; }

```
p.two{ font-family: 'Monotype Corsiva'; font-size: 50pt; color: green; } </style> </head> <body> <p class = "one">Web Technology</p> <p class = "two">Web Technology</p> </body>
```

3) Generic Selectors:

Sometimes it is convenient to have a class of Style specification that applies to the content of more than one kind of tag. This is done by using a generic class, which is defined without a tag name in its name. In place of the tag name, you use the name of the generic class, which must begin with a period. Eg <head> <style type = "text/css"> .sale{ font-family: 'Monotype Corsiva'; color: green; } </style> </head> <body> <p class = "sale">Weekend Sale</p> <h1 class = "sale">Weekend Sale</h1> <h6 class = "sale"> Weekend Sale</h6> </body>

4) id Selectors:

An id selector allows the application of a style to one specific element. Eg: <head> <style type = "text/css"> #one { font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; } #two { font-family: 'Monotype Corsiva'; font-size: 50pt; color: green; } </style> </head> <body> <p id = "one">Web Technology</p> <p id = "two">Web Technology</p> </body>

5) Universal Selectors:

The universal selector, denoted by an asterisk (\*), applies its style to all elements in a document. <head> <style type = "text/css"> \*{ font-family: 'Lucida Handwriting'; font-size: 25pt; color: Red; } </style> </head> <body> <p>Web Technology</p> <p>Web Technology</p> </body>

6) Pseudo Classes:

Pseudo class selectors are used if the properties are to be changed dynamically. For example: when mouse movement happens, in other words, hover happens or focus happens. <head> <style type = "text/css"> input:focus { font-family: 'lucida calligraphy'; color: purple; font-size:100; } input: hover { font-family: 'lucida handwriting';color: violet; font-size:40; } </style> </head> <body> <form action = " "> <p><label> NAME: <input type = "text" /></label></p> </form> </body>