

Internal Assessment Test 1 – July. 2022						
Web Technologies				Sub Code:	20MCA23	Branch: MCA
28/7/2022	Duration:	90 min's	Max Marks:	50	Sem	II OBE

Q1) Explain request phase and response phase of HTTP.

Request Phase:

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

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 HTTP 1.1. 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 Response Phase:

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.

Q2) What is a Web Server? Explain its characteristics?

Web server operations:

- All the communications between a web client and a web server use the HTTP
- When a web server begins execution, it informs the OS under which it is running & it runs as a background process
- A web client or browser, opens a network connection to a web server, sends information requests and possibly data to the server, receives information from the server and closes the connection.
- The primary task of web server is to monitor a communication port on host machine, accept HTTP commands through that port and perform the operations specified by the commands.
- When the URL is received, it is translated into either a filename or a program name

General characteristics of web server:

- The file structure of a web server has two separate directories
- The root of one of these is called document root which stores web documents

- The root of the other directory is called the server root which stores server and its support software's
- The files stored directly in the document root are those available to clients through top level URLs
- The secondary areas from which documents can be served are called virtual document trees.
- Many servers can support more than one site on a computer, potentially reducing the cost of each site and making their maintenance more convenient. Such secondary hosts are called virtual hosts.
- Some servers can serve documents that are in the document root of other machines on the web; in this case they are called as proxy servers

Q3) Describe basic syntax and structure of xhtml5. Explain any three text markup tags

<!DOCTYPE HTML>

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

<meta charset=UTF-8" />

<title>Your Website</title>

</head>

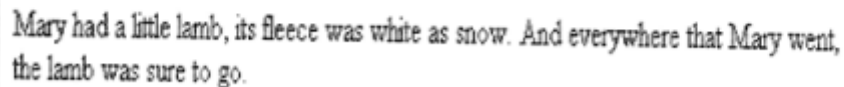
<body>

</body>

</html>

- 1) Paragraphs
 - XHTML does not allow text to be directly placed into the document. Text is normally organised into paragraphs
 - It begins with and ends with 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>
```

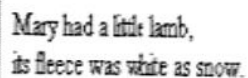


Mary had a little lamb, its fleece was white as snow. And everywhere that Mary went,
the lamb was sure to go.

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 `
` 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>
```

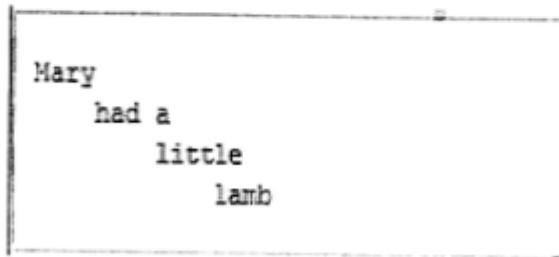


Mary had a little lamb,
its fleece was white as snow.

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>
```



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

1. Image

The Tag

The image tag, , which is an inline tag, specifies an image that is to appear in a document. In its simplest form, the image tag includes two attributes: **src**, which specifies the file containing the image; and **alt**, which specifies text to be displayed when it is not possible to display the image. If the file is in the same directory as the XHTML file of the document, the value of **src** is just the image's filename. In many cases, image files are stored in a subdirectory of the directory where the XHTML files are stored. For example, the image files might be stored in a subdirectory named **images**. If the image file's name is **stars.jpg** and it is stored in the **images** subdirectory, the value of **src** would be as follows:

```
"images/stars.jpg"
```

Two optional attributes of **img**, **width** and **height**, can be included to specify (in pixels) the size of the rectangle for the image. These can be used to scale the size of the image (that is, to make it larger or smaller). Care must be taken to ensure that the image is not distorted in the resizing. For example, if the image is square, the **width** and **height** attribute values must be equal.

The following is an example of an image element:

```
<img src = "c210.jpg" alt = "Picture of a Cessna 210" />
```

2. Link

Links are specified in an attribute of an anchor tag (<a>), which is an inline tag. The anchor tag that specifies a link is called the *source* of that link. The document whose address is specified in a link is called the *target* of that link.

As is the case with many tags, the anchor tag can include many different attributes. However, for creating links only one is required, href (an acronym for hypertext reference). The value assigned to href specifies the target of the link. If the target is in another document in the same directory, the target is just the document's filename. If the target document is in some other directory, the UNIX pathname conventions are used. So, an XHTML file named c210data.html in a subdirectory of the directory in which the source XHTML file—say, named airplanes—is specified in the href attribute as airplanes/c210data.html. This is the relative method of document addressing. Absolute file addresses could be used in which the entire pathname for the file is given. However, relative links are easier to maintain, especially if a hierarchy of XHTML files must be moved. If the document is on some other machine (not on the server providing the document that includes the link), the complete URL obviously must be used.

```
<?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">

<!-- link.html
  An example to illustrate a link
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> A link </title>
  </head>
  <body>
    <h1> Aidan's Airplanes </h1>
    <h2> The best in used airplanes </h2>
    <h3> "We've got them by the hangarful" </h3>
    <h2> Special of the month </h2>
    <p>
      1960 Cessna 210 <br />
      <a href = "C210data.html"> Information on the Cessna 210 </a>
    </p>
  </body>
</html>
```

Links can include images in their content, in which case the browser displays the image with the link:

```
<a href = "c210data.html" >
  <img src = "small-airplane.jpg"
    alt = "An image of a small airplane" />
  Information on the Cessna 210
</a>
```

Targets within Documents

If the target of a link is not at the beginning of a document, it must be some element within the document, in which case there must be some means of specifying it. The target element can include an id attribute, which can then be used to identify it in an href attribute. Consider following example

```
<h2 id="avionics"> Avionics</h2>
```

The target is specified in the href attribute value by preceding the id value with a pound sign (#), as in the following example

```
<a href="#avionics">What about avionics? </a>
```

When target is a part or fragment of another document, the name of the part is specified at the end of URL, separated by pound sign(#)

```
<a href="AIDAN1.html#avionics">Avionics</a>
```

3. List

1) Unordered List

The tag, which is a block tag, creates an unordered list. Each item in a list is specified with an 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

```
<?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">

<!-- unordered.html
  An example to illustrate an unordered list
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Unordered list </title>
  </head>
  <body>
    <h3> Some Common Single-Engine Aircraft </h3>
    <ul>
      <li> Cessna Skyhawk </li>
      <li> Beechcraft Bonanza </li>
      <li> Piper Cherokee </li>
    </ul>
  </body>
</html>
```

2) Ordered List

Ordered lists are lists in which the order of items is important. This ordered-ness 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 . 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.

```

<?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">

<!-- ordered.html
An example to illustrate an ordered list
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Ordered list </title>
</head>
<body>
<h3> Cessna 210 Engine Starting Instructions </h3>
<ol>
<li> Set mixture to rich </li>
<li> Set propeller to high RPM </li>
<li> Set ignition switch to "BOTH" </li>
<li> Set auxiliary fuel pump switch to "LOW PRIME" </li>
<li> When fuel pressure reaches 2 to 2.5 PSI, push
starter button
</li>
</ol>
</body>
</html>

```

3) Definition List

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.

```

<?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">

<!-- definition.html
An example to illustrate definition lists
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Definition lists </title>
</head>
<body>
<h3> Single-Engine Cessna Airplanes </h3>
<dl>
<dt> 152 </dt>
<dd> Two-place trainer </dd>
<dt> 172 </dt>
<dd> Smaller four-place airplane </dd>
<dt> 182 </dt>
<dd> Larger four-place airplane </dd>
<dt> 210 </dt>
<dd> Six-place airplane - high performance </dd>
</dl>
</body>
</html>

```

Q5) Write xhtml program to describe a table with rows, columns, align, valign, colspan and rowspan attributes

```

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>MCA Department</title>
</head>
<body>
<table border="1" cellspacing="5" cellpadding="5">
<caption> Time Table</caption>
<tr>
<td></td>

```



```

<td>8-9</td>
<td>9-10</td>
<td>10-10:30</td>
<td>10.30-11.30</td>
<td>11.30-12:30</td>
<td>12:30-1:30</td>
<td>1:30-2:30</td>
<td>2:30-3:30</td>
</tr>
<tr valign="bottom">
  <td>Day 1</td>
  <td>web</td>
  <td>unix</td>
  <td rowspan="3">Break</td>
  <td>dms</td>
  <td>co</td>
  <td rowspan="3">Break</td>
  <td>ds</td>
  <td>unix</td>
</tr>
<tr valign="top">
  <td>Day 2</td>
  <td>web</td>
  <td>unix</td>
  <td>dms</td>
  <td>co</td>
  <td>ds</td>
  <td>unix</td>
</tr>
<tr align="right">
  <td>Day 3</td>
  <td>web</td>
  <td>unix</td>
  <td>dms</td>
  <td>co</td>
  <td colspan="2">Lab</td>
</tr>
</table>
</body>
</html>

```

Time Table

	8-9	9-10	10-10:30	10.30-11.30	11.30-12:30	12:30-1:30	1:30-2:30	2:30-3:30
Day 1	web	unix	Break	dms	co	Break	ds	unix
Day 2	web	unix		dms	co		ds	unix
Day 3	web	unix		dms	co		Lab	

Q6) Create frames as shown in the figure using frameset tag



```

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>MCA Department</title>
  </head>

  <frameset cols="10%,*">
    <frame src="1.html"/>

    <frameset rows="20%,20%,*">
      <frame src="2.html"/>
      <frame src="3.html"/>
      <frameset cols="50%,*">
        <frame src="4.html"/>
        <frame src="5.html"/>
      </frameset>
    </frameset>

  </frameset>

</frameset>

</html>

```

Q7) Explain the tags nav, section, article, aside, footer

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) 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

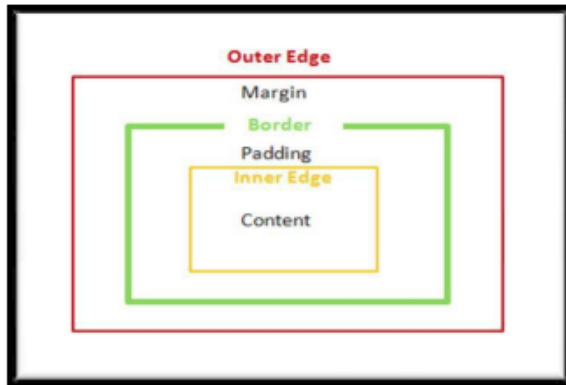
back to top links

related documents

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

Q8) Explain box model in CSS.

- On a given web page or a document, all the elements can have borders.
- The borders have various styles, color and width.
- The amount of space between the content of the element and its border is known as *padding*.
- The space between border and adjacent element is known as *margin*.



Borders:

- 1) **Border-style property** controls whether the elements content has a border, as well as the style of the border
It can be dotted, dashed, double
The styles of one of the four sides of an element can be set with
 - Border-top-style
 - Border-bottom-style
 - Border-left-style
 - Border-right-style
- 2) **Border-width** is used to specify the thickness of a border
It can be thin, medium, thick or any length value
The width of each of the four borders of an element specified with:
 - Border-top-width
 - Border-bottom-width
 - Border-left-width
 - Border-right-width
- 3) **Border-color** control color of a border
The width of each of the four borders of an element specified with:
 - Border-top-color
 - Border-bottom-color
 - Border-left-color
 - Border-right-color

```

<?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">

<!-- borders.html
      An example of a simple table with various borders
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Table borders </title>
    <style type = "text/css">
      table {border-top-width: medium;
              border-bottom-width: thick;
              border-top-color: red;
              border-bottom-color: blue;
              border-top-style: dotted;
              border-bottom-style: dashed;
            }
      p {border-style: dashed; border-width: thin;
         border-color: green
      }
    </style>
  </head>
  <body>
    <table border = "5">
      <caption> Fruit Juice Drinks </caption>
      <tr>
        <th> </th>
        <th> Apple </th>
        <th> Orange </th>
        <th> Screwdriver </th>
      </tr>
      <tr>
        <th> Breakfast </th>
        <td> 0 </td>
        <td> 1 </td>
        <td> 0 </td>
      </tr>

      <tr>
        <th> Lunch </th>
        <td> 1 </td>
        <td> 0 </td>
        <td> 0 </td>
      </tr>
      <tr>
        <th> Dinner </th>
        <td> 0 </td>
        <td> 0 </td>
        <td> 1 </td>
      </tr>
    </table>
    <p>
      Now is the time for all good Web programmers to
      learn to use style sheets.
    </p>
  </body>
</html>

```

Margins and Padding:

The margin properties are named margin, which applies to all four sides of an element: margin-left, margin-right, margin-top, and margin-bottom.

The padding properties are named padding, which applies to all four sides: padding-left, padding-right, padding-top, and padding-bottom.

```
<?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">

<!-- marpads.html
  An example to illustrate margins and padding
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Margins and Padding </title>
  <style type = "text/css">
    p.one   {margin: 0.2in;
             padding: 0.2in;
             background-color: #C0C0C0;
             border-style: solid;
            }
    p.two   {margin: 0.1in;
             padding: 0.3in;
             background-color: #C0C0C0;
             border-style: solid;
            }
    p.three {margin: 0.3in;
             padding: 0.1in;
             background-color: #C0C0C0;
             border-style: solid;
            }
```

```

    p.four {margin:0.4in;
           background-color: #C0C0C0;}
    p.five {padding: 0.4in;
           background-color: #C0C0C0;
           }
</style>
</head>
<body>
  <p>
    Here is the first line.
  </p>
  <p class = "one">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.2in,
    padding = 0.2in]
  </p>
  <p class = "two">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.1in,
    padding = 0.3in]
  </p>
  <p class = "three">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.3in,
    padding = 0.1in]
  </p>
  <p class = "four">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [margin = 0.4in,
    no padding, no border]
  </p>
  <p class = "five">
    Now is the time for all good Web programmers to
    learn to use style sheets. <br /> [padding = 0.4in,
    no margin, no border]
  </p>
  <p>
    Here is the last line.
  </p>
</body>
</html>

```

Q9) Write a program to illustrate all the CSS color, alignment of text and background image properties

Q10) Discuss all the CSS font properties..

1) Font Families:

The font-family property is used to specify a list of font names. The browser uses the first font in the list that it supports.

For example, the property:

font-family: Arial, Helvetica, Futura

tells the browser to use Arial if it supports that font.

If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can.

If the browser does not support any of the specified fonts, it will use an alternative of its choosing. If a font name has more than one word, the whole name should be delimited by single quotes, as in the following example:

font-family: 'Times New Roman'

Generic Name	Examples
serif	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Caflisch Script, Zapf-Chancery
fantasy	Critter, Cottonwood
monospace	Courier, Prestige

2) Font Sizes:

The font-size property does what its name implies.

For example, the following property specification sets the font size for text to 10 points: **font-size: 10pt**

Many relative font-size values are defined, including xx-small, x-small, small, medium, large, x-large, and xx-large. In addition, smaller or larger can be specified. Furthermore, the value can be a percentage relative to the current font size.

3) Font Variants:

The default value of the font-variant property is normal, which specifies the usual character font. This property can be set to small-caps to specify small capital characters. These characters are all uppercase, but the letters that are normally uppercase are somewhat larger than those that are normally lowercase.

4) Font Styles:

The font-style property is most commonly used to specify italic, as in **font-style: italic**

5) Font Weights:

The font-weight property is used to specify the degree of boldness, as in

font-weight: bold

Besides bold, the values normal, bolder, and lighter can be specified. Specific numbers also can be given in multiples of 100 from 100 to 900, where 400 is the same as normal and 700 is the same as bold.

6) Font Shorthands:

If more than one font property must be specified, the values can be stated in a list as the value of the font property. The order in which the property values are given in a font value list is important. The order must be as follows: The font names must be last, the font size must be second to last, and the font style, font variant, and font

weight, when they are included, can be in any order but must precede the font size and font names.

font: bold 14pt 'Times New Roman'

```
<?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">

<!-- fonts.html
     An example to illustrate font properties
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Font properties </title>
    <style type = "text/css">
      p.major {font-size: 14pt;
               font-style: italic;
               font-family: 'Times New Roman';
              }
      p.minor {font: 10pt bold 'Courier New';}
      h2 {font-family: 'Times New Roman';
          font-size: 24pt; font-weight: bold}
      h3 {font-family: 'Courier New'; font-size: 18pt}
    </style>
  </head>
  <body>
    <p class = "major">
      If a job is worth doing, it's worth doing right.
    </p>
    <p class = "minor">
      Two wrongs don't make a right, but they certainly
      can get you in a lot of trouble.
    </p>
    <h2> Chapter 1 Introduction </h2>
    <h3> 1.1 The Basics of Computer Networks </h3>
  </body>
</html>
```

7) Text Decoration:

The text-decoration property is used to specify some special features of text. The available values are line-through, overline, underline, and none, which is the default.

```
<style type = "text/css">
  p.delete {text-decoration: line-through}
  p.cap {text-decoration: overline}
  p.attention {text-decoration: underline}
</style>
```

