Internal Assessment Test 1 – Sept. 2019									
Web Technologies & its Applications					Sub Code:	15CS71	Branch:	CSE	
23-09-2019	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7 – A, B & C		OBE	

Answer any FIVE FULL Questions	MARKS	CO	RBT
1. Explain the following tags with examples:	[10]	CO1	L2

i)<a> ii)<img> iii) iv) v)<link>

(i) Anchor:

This tag is to create a hyperlink to a text/ image.

e.g) Go to CMRIT

(ii) image: defines an image. It defines the following:

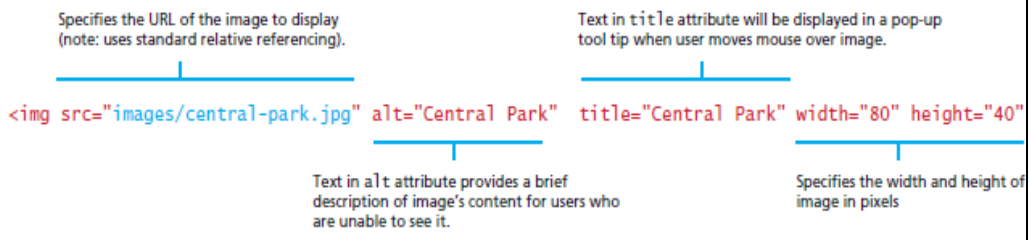


FIGURE 2.18 The <img> element

iii) paragraph: It is the most basic unit of text in an HTML document. Notice that the tag is a container and can contain HTML and other inline HTML elements. This term refers to HTML elements that do not cause a paragraph break but are part of the regular "flow" of the text.

Example:

Photo by Randy Connolly

This photo of Conservatory Pond in Central Park

New York City was taken on October 22, 2015 with a Canon EOS 30D camera.

  (iv) divisions: This element is a container element and is used to create a logical grouping of content (text and other HTML elements, including containers such as and other elements). The element has no intrinsic presentation; it is frequently used in contemporary CSS-based layouts to mark out sections. Example: By Ricardo on <time>September 15, 2015</time> Easy on the HDR buddy. By Susan on <time>October 1, 2015</time> I love Central Park. Copyright © 2015 Share Your Travels </body> **v) Link** This tag is used to link an external CSS to an HTML page. <link rel=”stylesheet” href=”a.css”>			
2. a) Write a note on XHTML and HTML5. XHTML stands for EXtensible HyperText Markup Language. It is the next step to evolution of internet. The XHTML was developed by World Wide Web Consortium (W3C). It helps web developers to make the transition from HTML to XML. Using XHTML, developers can enter the XML world with all the features of it, and they can still remain confident about the backward and future compatibility of the content. The XHTML 1.0 is the first document type in the XHTML family and it is Recommended by W3C in 26 January 2000. The XHTML 1.1 is Recommended by W3c in 31 May 2001. The XHTML5 is a standard and is used to develop an XML adaptation of the HTML5 specification. The XHTML documents contains three parts, which are discussed below: • **DOCTYPE:** It is used to declare a DTD • **head:** The head section is used to declare the title and other attributes.	[04]	CO2	L2

• **body:** The body tag contains the content of web pages. It consists many tags. Creating a XHTML web page, it is necessary to include DTD (Document Type Definition) declaration. There are three types of DTD which are discussed below: • Transitional DTD • Strict DTD • Frameset DTD **Transitional DTD:** It is supported by the older browsers which does not have inbuilt cascading style sheets supports. There are several attributes enclosing the body tag which are not allowed in strict DTD. XHTML stands for EXtensible HyperText Markup Language. It is the next step to evolution of internet. The XHTML was developed by World Wide Web Consortium (W3C). It helps web developers to make the transition from HTML to XML. Using XHTML, developers can enter the XML world with all the features of it, and they can still remain confident about the backward and future compatibility of the content. The XHTML 1.0 is the first document type in the XHTML family and it is Recommended by W3C in 26 January 2000. The XHTML 1.1 is Recommended by W3c in 31 May 2001. The XHTML5 is a standard and is used to develop an XML adaptation of the HTML5 specification. The XHTML documents contains three parts, which are discussed below: • DOCTYPE: It is used to declare a DTD • head: The head section is used to declare the title and other attributes. • body: The body tag contains the content of web pages. It consists many tags. Creating a XHTML web page, it is necessary to include DTD (Document Type Definition) declaration. There are three types of DTD which are discussed below: • Transitional DTD • Strict DTD • Frameset DTD Transitional DTD: It is supported by the older browsers which does not have inbuilt cascading style sheets supports. There are several attributes enclosing the body tag which are not allowed in strict DTD.			
2. b) With an example explain different levels of style sheets. CSS style rules can be located in three different locations. **1. Inline Styles** **Inline styles** are style rules placed within an HTML element via the style attribute,An inline style only affects the element it is defined within and overrides any other style definitions for properties used in the inline style. Using inline styles is generally discouraged since they increase bandwidth and decrease maintainability (because presentation and content are intermixed and because it can be difficult to make consistent inline style changes across multiple files.	[06]	CO3	L2

Internal styles example:

```
<h1>Share Your Travels</h1>
<h2>style="font-size: 24pt"Description</h2>
...
<h2>style="font-size: 24pt; font-weight: bold;">Reviews</h2>
```

2. Embedded Style Sheet

Embedded style sheets (also called **internal styles**) are style rules placed within the <style> element (inside the <head> element of an HTML document), While better than inline styles, using embedded styles is also by and large discouraged. Since each HTML document has its own <style> element, it is more difficult to consistently style multiple documents when using embedded styles.

Just as with inline styles, embedded styles can, however, be helpful when quickly testing out a style that is used in multiple places within a single HTML document.

Example:

```
<title>Share Your Travels -- New York - Central Park</title>
<style>
h1 { font-size: 24pt; }
h2 {
font-size: 18pt;
font-weight: bold;
}
</style>
</head>
<body>
<h1>Share Your Travels</h1>
<h2>New York - Central Park</h2>
...
```

3. External Style Sheet

External style sheets are style rules placed within a external text file with the **.css** extension. This is by far the most common place to locate style rules because it provides the best maintainability. When you make a change to an external style sheet, all HTML documents that reference that style sheet will automatically use the updated version. The browser is able to cache the external style sheet, which can improve the performance of the site as well.

To reference an external style sheet, you must use a <link> element (within the
<head> element),We can link to several style sheets at a time; each linked style sheet will require its own <link> element.

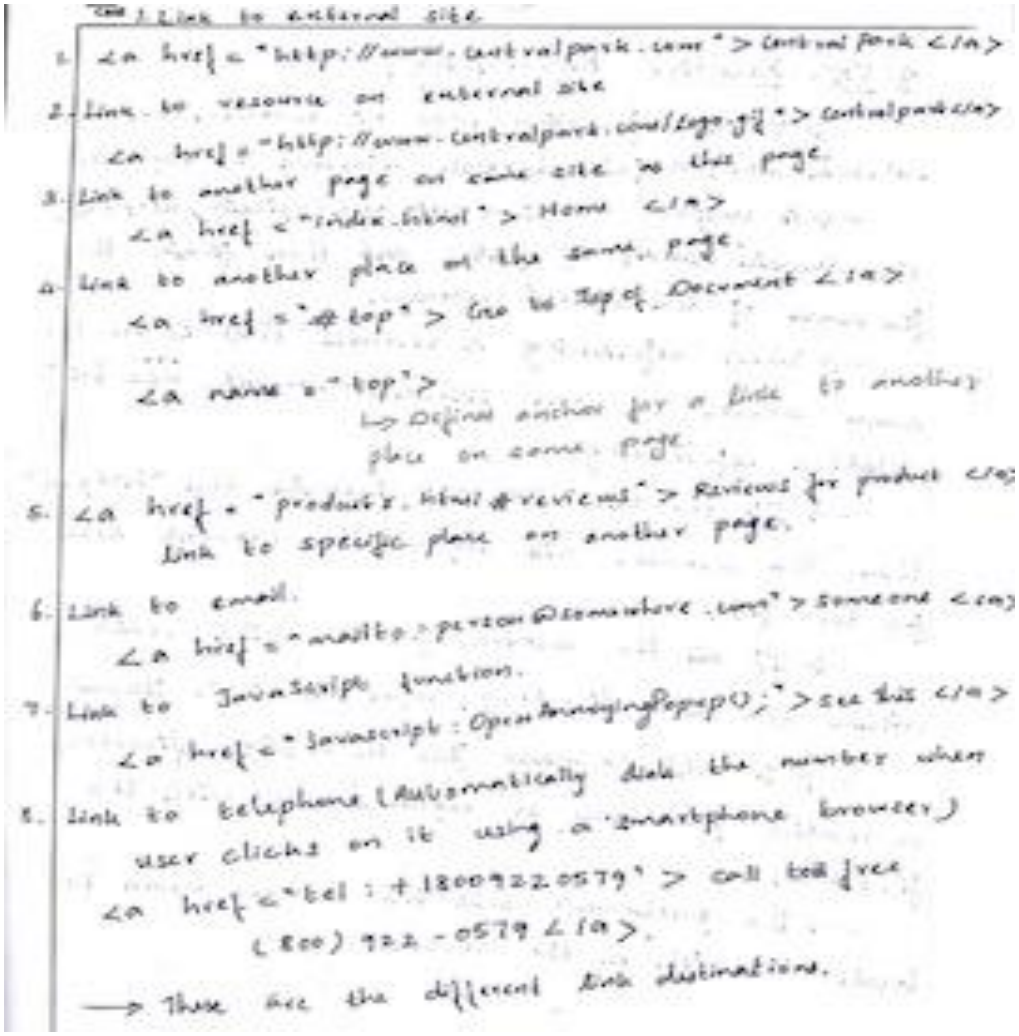
Example:

```
<head lang="en">
<meta charset="utf-8">
<title>Share Your Travels -- New York - Central Park</title>
<link rel="stylesheet" href="styles.css" />
```

</head>
Explain how CSS styles interact.
The “Cascade” in CSS refers to how conflicting rules are handled. CSS uses the following cascade principles to help it deal with conflicts: inheritance, specificity, and location.

3. a) Explain different form widgets created with the <input> tag. [06] CO4 L2

Value Name	Notes
`button`	Defines a button-like input.
`checkbox`	Defines a checkbox, which the user can toggle on or off.
`file`	Defines a file upload box with a browse button.
`hidden`	Defines a field within a form that is not visible to the user.
image	Defines an image that is clicked to submit a form.
password	Displays an obfuscated password entry field.
`radio`	Defines a circular selection button in a form.

reset	Defines a button on a form that will return all fields to their default values.			
submit	Defines a button that is clicked to submit a form.			
text	Defines a text entry field in a form.			
3. b) Briefly explain the different link destinations with examples 		[04]	CO2	L2

1. Link to external site.
    Central Park
2. Link to resource on external site
    centralpark
3. Link to another page on same site as this page.
    Home
4. Link to another place on the same page.
    Go to Top of Document
   ...

   ↳ Define anchor for a link to another place on same page.
5. Reviews for product
   Link to specific place on another page.
6. Link to email.
    Someone
7. Link to JavaScript function.
    See this
8. Link to telephone (Automatically dials the number when user clicks on it using a smartphone browser)
    call toll free
   (800) 922-0579 .

→ These are the different link destinations.

4. a) Explain the benefits of CSS. Maintainability Output flexibility Improved accessibility Improved download speed	[04]	CO3	L2

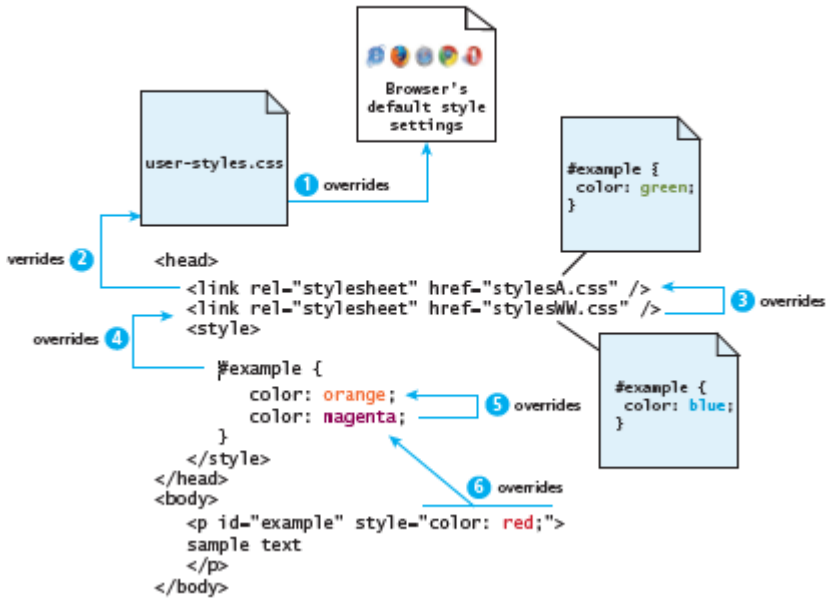
4. b) Explain how CSS styles interact.
 	
1. Inheritance
Inheritance is the first of these cascading principles. Many (but not all) CSS properties affect not only themselves but their descendants as well. Font, color, list, and text properties are inheritable; layout, sizing, border, background,
and spacing properties are not.
 	
In the below example, without using inherit keyword, we have a single style rule that styles *all* the elements. The and <time> elements within the inherit the bold font-weight property but not the margin or border styles.
 	
Example:
 	
div {
font-weight: bold;
margin: 50px;
border: 1pt solid green;
}
p {
border: inherit;
margin: inherit;
}
 Reviews
 	
 By Ricardo on <time>September 15, 2015</time>
 Easy on the HDR buddy.
 	
 By Susan on <time>October 1, 2015</time>
 I love Central Park.
 	
However, it is possible to tell elements to inherit properties that are normally not inheritable by using the elements nested within the elements now inherit the border and margins of their parent.
 	
2. Specificity
	[06]	CO3	L2

Specificity is how the browser determines which style rule takes precedence when more than one style rule could be applied to the same element. In CSS, the more specific the selector, the more it takes precedence (i.e., overrides the previous definition).

3. Location

When inheritance and specificity cannot determine style precedence, the principle of **location** will be used. The principle of location is that when rules have the same specificity, then the latest are given more weight.

For instance, an inline style will override one defined in an external author style sheet or an embedded style sheet.



In the above example paragraph uses red color.

5. a) List the different selectors available in CSS and explain in detail. [10] CO4 L2

1 Element Selectors

Element selectors select all instances of a given HTML element You can select all elements by using the **universal element selector**, which is the * (asterisk) character. You can select a group of elements by separating the different element names with commas. This is a sensible way to reduce the size and complexity of your CSS files, by combining multiple identical rules into a single rule.

2 Class Selectors

A **class selector** allows you to simultaneously target different HTML elements regardless of their position in the document tree. If a series of HTML elements have been labeled with the same class attribute value, then you can target them for styling by using a class selector, which takes the form: period (.) followed by the class name.

3 Id Selectors

An **id selector** allows you to target a specific element by its id attribute regardless of its type or position. If an HTML element has been labeled with an id attribute, then you can target it for styling by using an id selector, which takes the form: pound/hash (#)

followed by the id name.

4 Attribute Selectors

An **attribute selector** provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute. This can be a very powerful technique, but because of uneven support by some of the browsers, not all web authors have used them. Attribute selectors can be a very helpful technique in the styling of hyperlinks and images. For instance, perhaps we want to make it more obvious to the user when a pop-up tooltip is available for a link or image. We can do this by using the following attribute selector: [title] { … } This will match any element in the document that has a title attribute.

5 Pseudo-Element and Pseudo-Class Selectors

A **pseudo-element selector** is a way to select something that does not exist explicitly as an element in the HTML document tree but which is still a recognizable selectable object. For instance, you can select the first line or first letter of any HTML element using a pseudo-element selector. A **pseudo-class selector** does apply to an

HTML element, but targets either a particular state or, in CSS3, a variety of family relationships. The most common use of this type of selectors is for targeting link states. By default, the browser displays link text blue and visited text links purple. Do be aware that this state does not occur on touch screen devices. Note the syntax of pseudo-class selectors: the colon (:) followed by the pseudo-class selector name. Do be aware that a space is *not* allowed after the colon. Believe it or not, the order of these pseudo-class elements is important. The :link and :visited pseudo-classes should appear before the others. Some developers use a mnemonic to help them remember the order. My favorite is “Lord Vader, Former Handle Anakin” for Link, Visited, Focus, Hover, Active.

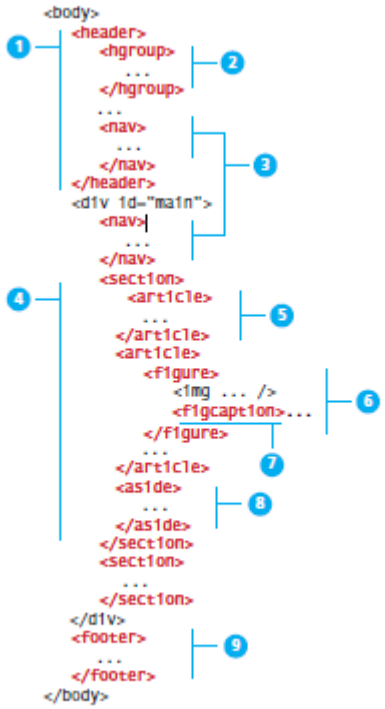
6 Contextual Selectors

A **contextual selector** (in CSS3 also called **combinators**) allows you to select elements based on their *ancestors*, *descendants*, or *siblings*. That is, it selects elements based on their context or their relation to other elements in the document tree. While some of these contextual selectors are used relatively infrequently, almost all

web authors find themselves using descendant selectors.A **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 somewhere within a 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 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>.			
6 (a)	Explain HTML5 semantic structure elements with example. **Semantic Structure of HTML5:**  Sample layout using new HTML5 semantic structure elements **1. 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. **2. Heading Groups** The <hgroup> element can be used in such a circumstance to group them together within one container.The <hgroup> element can be used	[10]	CO3	L2

in contexts other than a header.
For instance,one could also use an <hgroup> within an <article> or a <section> element as well. The <hgroup> element can *only* contain <h1>, <h2>, etc., elements.
Following illustrates the usage of <hgroup> element.

Example:
```
<header>
<hgroup>
<h1>Chapter Two: HTML 1</h1>
<h2>An Introduction</h2>
</hgroup>
</header>
```

3. 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.
Following illustrates a typical example usage of the <nav> element:

Example:
```
<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>
```

4. Articles and Sections
<section> is a much broader element, while the <article> element is to be used for blocks of content that could potentially be read or consumed independently of the other content on the page.
If some block of content could theoretically exist on another website (as if it were syndicated) and still make sense in that new context, then wrap that content within an <article> element. If a block of content has some type of heading associated with it, then consider wrapping it within a <section> element.

5. Figure and Figure Captions
Diagrams or photographs that are separate from the text (but related to it), which are described by a caption, and which are given the generic name of *Figure.*
<figure> element was used to indicate important information whose location on the page is somewhat unimportant.

	Example: This photo was taken on October 22, 2011 with a Canon EOS 30D camera. <figure> <figcaption>Conservatory Pond in Central Park</figcaption> </figure> It was a wonderfully beautiful autumn Sunday, with strong sunlight and expressive clouds. I was very fortunate that my one day in New York was blessed with such weather! **6 Aside** Used for marking up content that is separate from the main content on the page. <aside> element "represents a section of a page that consists of content that is tangentially related to the content around the aside element" (from WHATWG specification). The <aside> element could thus be used for sidebars, pull quotes, groups of Advertising images, or any other grouping of non-essential elements.			
7 (a)	Explain BOX model with the help of an example. **Background** In contemporary web design, it has become extremely common to use CSS to display purely presentational images (such as background gradients and patterns, decorative images, etc.) rather than using the <img> element. Table 3.7 lists the most common background properties. **Property Description** background A combined shorthand property that allows you to set multiple background values in one property. While you can omit properties with the shorthand, do remember that any omitted properties will be set to their default value. background-attachment Specifies whether the background image scrolls with the document (default) or remains fixed. Possible values are: fixed, scroll. background-color Sets the background color of the element. You can use any of the techniques shown in Table 3.2 for specifying the color. background-image Specifies the background image (which is generally a jpeg, gif, or png file) for the element. Note that the URL is relative to the CSS file and not the HTML. CSS3 introduced the ability to specify multiple background images.	[05]	CO3	L2

background-position Specifies where on the element the background image will be placed. Some possible values include: bottom, center, left, and right. You can also supply a pixel or percentage numeric position value as well. When supplying a numeric value, you must supply a horizontal/vertical pair; this value indicates its distance from the top left corner of the element,

background-repeat Determines whether the background image will be repeated. This is a common technique for creating a tiled background Possible values are: repeat, repeat-x, repeat-y, and no-repeat.

background-size New to CSS3, this property lets you modify the size of the background image.

2 Borders

Borders provide a way to visually separate elements. You can put borders around all four sides of an element, or just one, two, or three of the sides. Border widths are perhaps the one exception to the general advice against using
the pixel measure. Using em units or percentages for border widths can result in unpredictable widths as the different browsers use different algorithms (some round up, some round down) as the zoom level increases.

3 Margins and Padding

Margins and padding are essential properties for adding white space to a web page, which can help differentiate one element from another

7 (b) Write HTML code for the following table: [05] CO3 L3

Time Day		9.00 am to 1.15 pm	2.00 am to 5.00 pm
Mon to Fri	Sub	Theory class	ML/WTS Lab
	F1	ABC/EFG/XYZ	AD block, 1st * floor
Sat	Sub	Extra Curricular Activity	
	F1		

```
<!DOCTYPE html>
<html>
 <head>
  <title>table</title>
   </head>
      <body>
        <table border="2">
```

	<tr> <th colspan="2">Time Day</th> <th>9.00 am to 1.15 pm</th> <th>2.00 am to 5.00 pm</th> </tr> <tr> <th rowspan="2">Mon to Fri</th> <td>sub</td> <td>Theory class</td> <td>ML/WTS Lab</td> </tr> <tr> <td>F1</td> <td>ABC/EFG/XYZ</td> <td>AD block, 1stFloor</td> </tr> <tr> <th rowspan="2">Sat</th> <td>sub</td> <td rowspan="2" colspan="2">Extra Curricular Activity</td> </tr> <tr> <td>F1</td> </tr> </table> </body> </html>			
8 (a)	Explain table and form accessibility.	[05]	CO4	L2

→ The most important guidelines in that docum

* provide text alternatives for any nontext c so that it can be changed into other forms pe need, such as large print, braille, speech, symbols simpler language.

* Create content that can be presented in different ways without losing information or struc

* Make all functionality available from a keyboa

* provide ways to help users navigate, find con and determine where they are.

*Accessible Tables :-

→ Users who rely on visual readers can fi pages with many tables especially difficult to us

→ To improve the situation is to only use ta for tabular data, not for layout.

→ Using the following accessibility features tables in HTML can improve the experience for those users.

(1) Describe the table's content using the <
element. This provides the user with the abil
discover what the table is about before having
listen to the content of each and every cell i
the table.

→ For long description for the table, put
table within a <figure> element and use th
<figcaption> element to provide the description

(2) Connect the cells with a textual descripti
the header.

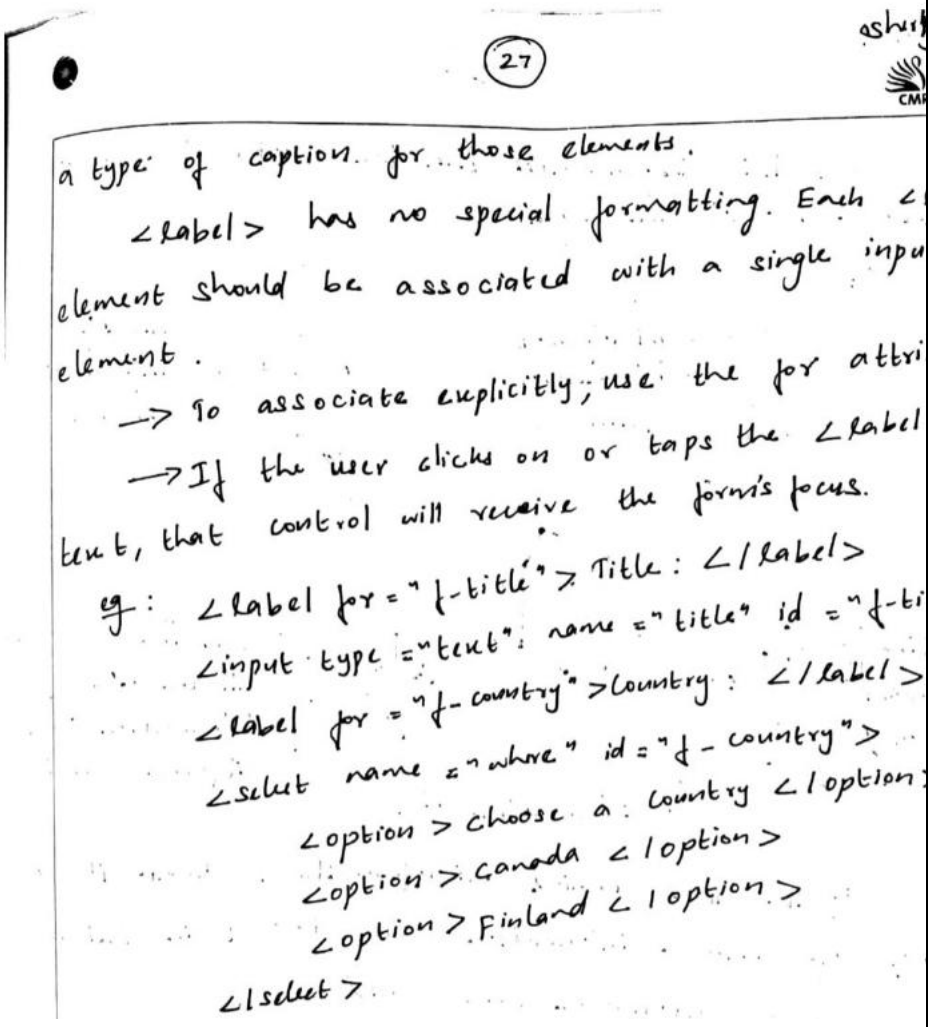
→ users rely on visual reader, this is not
easy task.

→ To listin to reader software, the conten
a table sounds like "row 3, cell 4 : 45.56; rou
cell 5 : Canada".

*Accessible forms :-

→ Browser provide some unique formatti
the <fieldset> and <legend> elements.

→ their main purpose is to logically group
form input elements together with the <legend> |



27

a type of caption for those elements.

<label> has no special formatting. Each <

element should be associated with a single inpu

element.

→ To associate explicitly, use the for attri

→ If the user clicks on or taps the <label

text, that control will receive the form's focus.

eg: <label for="f-title">Title:</label>

<input type="text" name="title" id="f-ti

<label for="f-country">Country:</label>

<select name="where" id="f-country">

<option>Choose a country</option>

<option>Canada</option>

<option>Finland</option>

</select>

8 (b)	Write short notes on micro formats with example. HTML and provide an API to be used by search engines, aggregators, and other tools. These minimal patterns of HTML are used for marking entities that range from fundamental to domain-specific information, such as people, organizations, events, and locations. Micro formats are supported by all major search engines. The data is carried in the class property that can be added to any HTML element. In addition to being machine-readable, their format is designed to be easily read by humans. An author of a webpage can add micro formats to their HTML. For example if they wanted to identify themselves they could use an h-card such as: `<a class="h-card" href="http://example.com">Joe Bloggs</a>`	[05]	CO4	L2

When a parser encounters this data, it will know that this page contains a "card" which describes a person or organization named Joe Blogger with a URL of http://example.com/. The parser makes this data available via APIs that can be used for different applications.

As in this example, some markup patterns require only a single microformat root class name, which parsers use to find a few generic properties such as name, url, photo.

All microformats class names use prefixes. Prefixes are syntax independent from vocabularies, which are developed separately.

"h-*" for root class names, e.g. "h-card", "h-entry", "h-feed", and many more. These top-level root classes usually indicate a type and corresponding expected vocabulary of properties. For example:

h-card describes a person or organization

h-entry describes episodic or date stamped online content like a blog post

h-feed describes a stream or feed of posts

example:

```
<p class="h-card">
  <img class="u-photo"
src="http://example.org/photo.png" alt="" />
  <a class="p-name u-url"
href="http://example.org">Joe Bloggs</a>
  <a class="u-email"
href="mailto:joebloggs@example.com">joebloggs@example.
com</a>,
  <span class="p-street-address">17
Austerstræti</span>
  <span class="p-locality">Reykjavík</span>
  <span class="p-country-name">Iceland</span>
</p>
```