USN | | | | | | | | | | |

Internal Assessment Test 1 – Nov 2021

Answer Scheme

Sub:	Web Technology & its Applications					Sub Code:	15CS71/17 CS71	Branch:	CSE
Date:	11-11-2021	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7-D		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain the following HTML5 tags with example. (i) headings (ii)paragraphs (iii)Inline elements (iv)image (v)time (vi)divisions	[6]	CO1	L1

(i) headings: HTML provides six levels of heading (h1 through h6), with the higher heading number indicating a heading of less importance. Headings are also used by the browser to create a document outline for the page.

Example:

```
<h1>Share Your Travels</h1>
<h2>New York - Central Park</h2>
```

(ii)paragraphs: It is the most basic unit of text in an HTML document. Notice that the <p> 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:

```
<p>Photo by Randy Connolly</p>
<p>This photo of Conservatory Pond in  <a
href="http://www.centralpark.com/">Central Park</a>
New York City was taken on October 22, 2015 with a <strong>Canon EOS
30D</strong> camera.
</p>
```

(iii)Inline elements: They do not disrupt the flow of text (i.e., cause a line break). HTML defines over 30 of these elements. Few are as follows:

Element	Description
<a>	Anchor used for hyperlinks.
<abbr>	An abbreviation
 	Line break
<cite>	Citation (i.e., a reference to another work).
<code>	Used for displaying code, such as markup or programming code.
	Emphasis
<mark>	For displaying highlighted text
<small>	For displaying the fine-print, i.e., "non-vital" text, such as copyright or legal notices.
	The inline equivalent of the element. It is generally used to mark text that will receive special formatting using CSS.
	For content that is strongly important.
<time>	For displaying time and date data

[04

iv)image: defines an image. It defines the following:

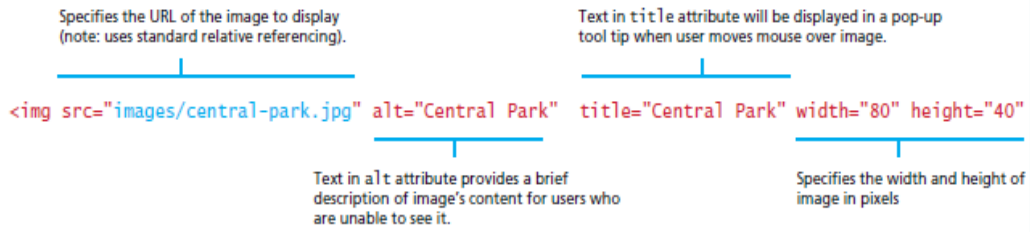


FIGURE 2.18 The <img> element

(v)time: Used to represent time.

Example:
By Susan on <time>October 1, 2015</time>

(vi) 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.

<small>Copyright © 2015 Share Your Travels</small>
</body>

(b)

Explain the following in brief with example. (i) Links to destination (ii) URL referencing [04]

CO1 L2

(i) Links to destination:
The use the anchor element to create a wide range of links. These include:

- Links to external sites (or to individual resources such as images or movies on an external site).
- Links to other pages or resources within the current site.
- Links to other places within the current page.
- Links to particular locations on another page (whether on the same site or on an external site).
- Links that are instructions to the browser to start the user's email program.
- Links that are instructions to the browser to execute a JavaScript function.
- Links that are instructions to the mobile browser to make a phone call.

FIGURE 2.16 Different link destinations

(ii) URL referencing:

Whether we are constructing links with the <a> element, referencing images with the <img> element, or including external JavaScript or CSS files, we need to be able to successfully reference files within our site. This requires learning the syntax for so-called relative referencing.

When referencing a page or resource on an external site, a full **absolute reference** is required: that is, a complete URL as described in Chapter 1 with a protocol (typically, http://), the domain name, any paths, and then finally the file name of the desired resource.

However, when referencing a resource that is on the same server as your HTML document, you can use briefer relative referencing. If the URL does not include the "**http://**" then the browser will request the current server for the file. If all the resources for the site reside within the same **directory** (also referred to as a **folder**), then you can reference those other resources simply via their file name.

However, most real-world sites contain too many files to put them all within a single directory. For these situations, a relative pathname is required along with the file name. The **pathname** tells the browser where to locate the file on the server.

Pathnames on the web follow Unix conventions. Forward slashes ("/") are used to separate directory names from each other and from file names. Double-periods. ("..") are used to reference a directory "above" the current one in the directory tree.

The different types of URL referencing are:

Relative Link Type	Example
1 Same Directory To link to a file within the same folder, simply use the file name.	To link to example.html from about.html (in Figure 2.1 use: <a href="example.html">
2 Child Directory To link to a file within a subdirectory, use the name of the subdirectory and a slash before the file name.	To link to logo.gif from about.html, use: <a href="images/logo.gif">
3 Grandchild/Descendant Directory To link to a file that is multiple subdirectories *below* the current one, construct the full path by including each subdirectory name (separated by slashes) before the file name.	To link to background.gif from about.html, use: <a href="css/images/background.gif">
4 Parent/Ancestor Directory Use "../" to reference a folder above the current one. If trying to reference a file several levels above the current one, simply string together multiple "../".	To link to about.html from index.html in members, use: <a href="../about.html"> To link to about.html from bio.html, use: <a href="../../about.html">
5 Sibling Directory Use "../" to move up to the appropriate level, and then use the same technique as for child or grandchild directories.	To link to about.html from index.html in members, use: <a href="../images/about.html"> To link to background.gif from bio.html, use: <a href="../../css/images/background.gif">
6 Root Reference An alternative approach for ancestor and sibling references is to use the so-called root reference approach. In this approach, begin the reference with the root reference (the "/") and then use the same technique as for child or grandchild directories. **Note that these will only work on the server! That is, they will not work when you test it out on your local machine.**	To link to about.html from bio.html, use: <a href="/about.html"> To link to background.gif from bio.html, use: <a href="/images/background.gif">
7 Default Document Web servers allow references to directory names without file names. In such a case, the web server will serve the default document, which is usually a file called index.html (Apache) or default.html (IIS). **Again, this will only generally work on the web server.**	To link to index.html in members from about.html, use either: <a href="members"> Or <a href="/members">

2 (a) Explain the semantic markup of HTML5 and its advantages. [05] CO2 L2

Semantic markup of HTML5:

Example:

```
<!DOCTYPE html>
<html>
<head lang="en">
<meta charset="utf-8" />
<title>Share Your Travels -- New York - Central Park</title>
<link rel="stylesheet" href="css/main.css" />
<script src="js/html5shiv.js"></script>
</head>
<body>
<h1>Main heading goes here</h1>
...
</body>
</html>
```

1. DOCTYPE

DOCTYPE (short for **Document Type Definition**) element, which tells the browser (or any other client software that is reading this HTML document) what type of document it is about to process. Notice that it does not indicate what version of HTML is contained within the document: it only specifies that it contains HTML. The HTML5 doctype is quite short in comparison to one of

the standard doctype specifications for XHTML:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

2. Head and Body

HTML5 does not require the use of the <html>, <head>, and <body> elements However, in XHTML they were required, and most web authors continue to use them. The <html> element is sometimes called the **root element** as it contains all the other HTML elements in the document.

Notice that it also has a lang attributes. This optional attribute tells the browser the natural language that is being used for textual content in the HTML document, which is English in this example. This doesn't change how the document is rendered in the browser; rather, search engines and screen reader software can use this information.

HTML pages are divided into two sections: the **head** and the **body**, which correspond to the <head> and <body> elements. The head contains descriptive elements *about* the document, such as its title, any style sheets or JavaScript files it uses, and other types of meta information used by search engines and other programs. The body contains content (both HTML elements and regular text) that will be displayed by the browser.

The first of these is the <meta> element declares that the character encoding for the document is UTF-8. Character encoding refers to which character set standard is being used to encode the characters in the document. As you may know, every character in a standard text document is represented by standardized bit pattern. The original ASCII standard of the 1950s defined English (or more properly Latin) upper and lowercase letters as well as a variety of common punctuation symbols using 8 bits for each character.

UTF-8 is a more complete variable-width encoding system that can encode all 110,000 characters in the Unicode character set (which in itself supports over 100 different language scripts).

It also specifies an external CSS style sheet file that is used with this document. Virtually all commercial web pages created in the last decade make use of style sheets to define the visual look of the HTML elements in the document.

Styles can also be defined within an HTML document for consistency's sake, most sites place most or all of their style definitions within one or more external style sheet files.

Notice that in this example, the file being referenced (**main.css**) resides within a subfolder called **css**. This is by no means a requirement. It is common practice, however, for web authors to place additional external CSS, JavaScript, and image files into their own subfolders.

Finally, It also references an external JavaScript file. Most modern commercial sites use at least some JavaScript. Like with style definitions, JavaScript code can be written directly within the HTML or contained within an external file.

Advantages:

HTML markup has a variety of important advantages:

■ **Maintainability**. Semantic markup is easier to update and change than web pages that contain a great deal of presentation markup.

■ **Faster**. Semantic web pages are typically quicker to author and faster to download.

■ **Accessibility**. Not all web users are able to view the content on web pages. Users with sight disabilities experience the web using voice reading software. Visiting a web page using voice reading software can be a very frustrating experience if the site does not use semantic markup. As well, many governments insist that sites for organizations that receive federal government funding must adhere to certain accessibility guidelines. For instance, the United States government has its own Section 508 Accessibility Guidelines.

- **Search engine optimization**. For many site owners, the most important users of a website are the various search engine crawlers. These crawlers are automated programs that cross the web scanning sites for their content, which is then used for users' search queries. Semantic markup provides better instructions for these crawlers: it tells them what things are important content on the site.

(b) Explain the benefits of CSS. [05] CO2 L2

The benefits of CSS include:

■ **Improved control over formatting.** The degree of formatting control in CSS is significantly better than that provided in HTML. CSS gives web authors fine-grained control over the appearance of their web content.

■ **Improved site maintainability**. Websites become significantly more maintainable because all formatting can be centralized into one CSS file, or a small handful of them. This allows you to make site-wide visual modifications by changing a single file.

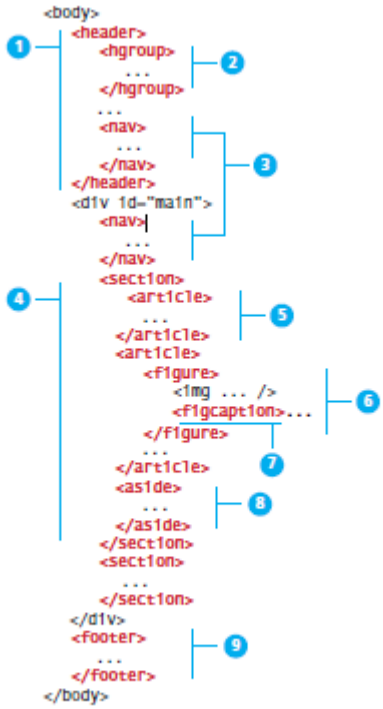
■ **Improved accessibility**. CSS-driven sites are more accessible. By keeping presentation out of the HTML, screen readers and other accessibility tools work better, thereby providing a significantly enriched experience for those reliant on accessibility tools.

■ **Improved page download speed**. A site built using a centralized set of CSS files for all presentation will also be quicker to download because each individual HTML file will contain less style information and markup, and thus be smaller.

■ **Improved output flexibility**. CSS can be used to adopt a page for different output media. This approach to CSS page design is often referred to as responsive design. Figure 3.1 illustrates a site that responds to different browser and window sizes.

3 (a) Explain HTML5 semantic structure elements with example. [10] CO3 L2

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 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:

```
<p>This photo was taken on October 22, 2011 with a Canon EOS 30D camera.</p>
<figure>
<img src="images/central-park.jpg" alt="Central Park" /><br/>
<figcaption>Conservatory Pond in Central Park</figcaption>
</figure>
<p>
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!
</p>
```

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.

4 (a)	Explain Location of styles with examples.	[05]	CO3	L2

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.

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>
```

(b) Explain how CSS styles interact. [05] CO3 L2

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.

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 <p> 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;
}
<h3>Reviews</h3>
<div>
<p>By Ricardo on <time>September 15, 2015</time></p>
<p>Easy on the HDR buddy.</p>
</div>
<hr/>
<div>
<p>By Susan on <time>October 1, 2015</time></p>
<p>I love Central Park.</p>
</div>
<hr/>
```

However, it is possible to tell elements to inherit properties that are normally not inheritable by using the <p> elements nested within the elements

now inherit the border and margins of their parent.

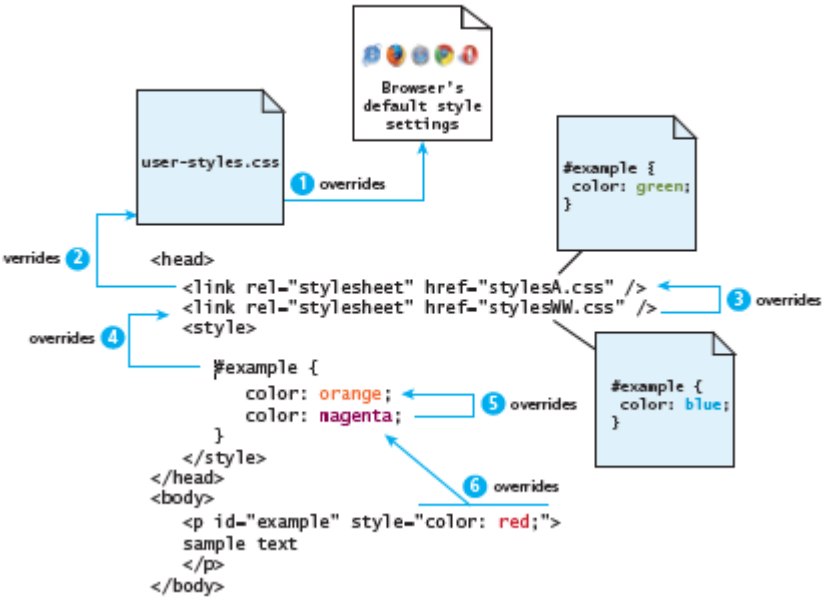
2. Specificity

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) Briefly explain Selectors in CSS with examples. [10] CO3 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 BOX model with the help of an example. [05] CO4 L2

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.

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.

(b) Briefly explain CSS properties and its values. [05] CO4 L2

Properties

Each individual CSS declaration must contain a property. These property names are predefined by the CSS standard. The CSS2.1 recommendation defines over a hundred different property names, so some type of reference guide, whether in a book, online, or within your web development software, can be helpful.5 This chapter and
the next one on CSS (Chapter 5) will only be able to cover most of the common CSS properties. Table 3.1 lists many of the most commonly used CSS properties.

Property Type Property

Fonts font font-family font-size font-style font-weight @font-face
Text letter-spacing line-height text-align text-decoration text-indent
Color and background background background-color background-image background-position background-repeat color
Borders border border-color border-width border-style border-top border-top-color border-top-width etc.

Property Type Property

Spacing padding padding-bottom, padding-left, padding-right, padding-top margin margin-bottom, margin-left, margin-right, margin-top
Sizing height max-height max-width min-height min-width width **Layout** bottom, left, right, top clear display float overflow position visibility z-index
Lists list-style list-style-image list-style-type

Values

Each CSS declaration also contains a value for a property. The unit of any given value is dependent upon the property. Some property values are from a predefined list of keywords. Others are values such as length measurements, percentages, numbers without units, color values, and URLs. Colors would seem at first glance to be the most clear of these units. CSS supports a variety of different ways of describing color;

7 (a) Write a HTML5 program for the following table. [05] CO4 L3

19th Century Fresh Paintings

Artist	Title	Year
Jacques-Louis David	The Death of Marat	1793
	The Intervention of the Sabine Women	1799
	Napoleon Crossing the Alps	1800

[05] CO1 L2

```
<!DOCTYPE html>
<html>
<head>
    <title>Table</title>
</head>
<body>
    <table border="2">
        <caption>19th Century Fresh Paintings</caption>
```

```
            <tr>
                <th>Artist</th>
                <th>Title</th>
                <th>Year</th>
            </tr>
            <tr>
                <td rowspan="3">Jacques-Louis David</td>
                <td>The Death of Marat</td>
                <td>1793</td>
            </tr>
            <tr>
                <td>The Intervention of the Sabine Women</td>
                <td>1799</td>
            </tr>
            <tr>
                <td>Napoleon Crossing the Alps</td>
                <td>1800</td>
            </tr>
        </table>
</body>
</html>
```

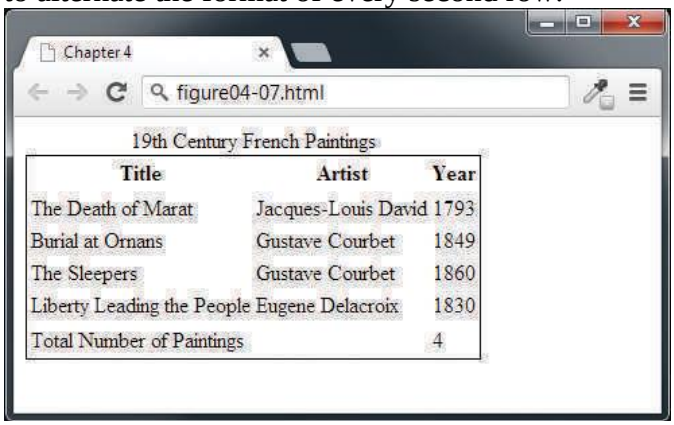
(b) Explain the following i)Table Borders ii) Boxes and zebras.

Table Borders

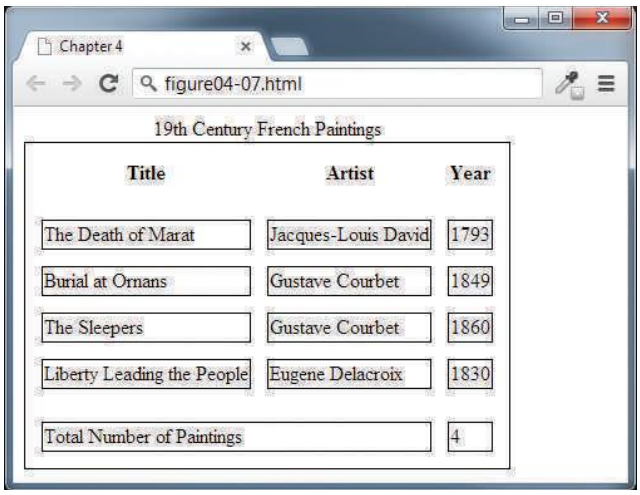
Interestingly, borders cannot be assigned to the <tr>, <thead>, <tfoot>, and <tbody> elements. Notice as well the border-collapse property. This property selects the table's border model. In this approach, each cell has its own unique borders. You can adjust the space between these adjacent borders via the border-spacing property, as shown in the final screen capture in

Boxes and Zebras

While there is almost no end to the different ways one can style a table, there are a number of pretty common approaches. We will look at two of them here. The first of these is a box format, in which we simply apply background colors and borders in various ways. We can then add special styling to the :hover pseudo-class of the <tr> element, to highlight a row when the mouse cursor hovers over a cell. That figure also illustrates how the pseudo-element nth-child can be used to alternate the format of every second row.




```
table {
border: solid 1pt black;
}
```




```
table {
border: solid 1pt black;
border-spacing: 10pt;
}
td {
border: solid 1pt black;
}
```