

Web Technology Assignment Question

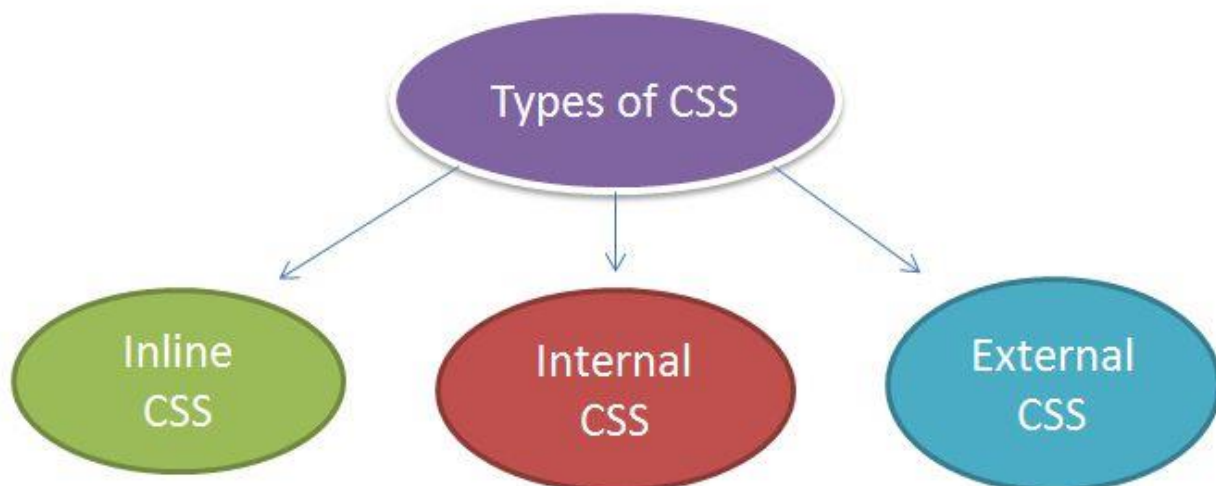
1. Define HTML tag references and its global attribute? Write 10 example of each.
2. Define different types of formatting tags?
3. Write down elements used in form with example and form properties and its types?
4. Explain three approaches of inserting css method?

Answer will be published on site www.bcanotesnepal.com

BCA

4. Explain three approaches of inserting css method?

- There are the following three types of CSS:
 - a) Inline CSS.
 - b) Internal CSS.
 - c) External CSS

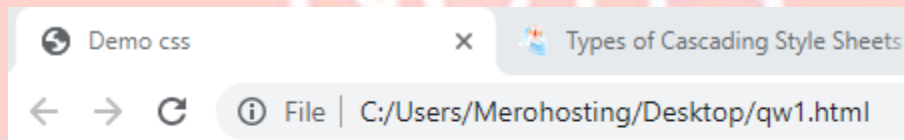


➤ Inline CSS

- For Inline CSS every style content is in HTML elements. It is used for a limited section. Whenever our requirements are very small we can use inline CSS.
- It will affect only single elements. In HTML we require that various HTML tag's views are different so then we use inline Cascading Style Sheets. There are disadvantage of inline Cascading Style Sheets. It must be specified on every HTML tag. There is a lot of time consumed by that and it is not the best practice for a good programmer and the code will be quite large and very complex.
- Inline CSS examples are given below:

```
<!DOCTYPE html>
<html xmlns="http://www.bcanotesnepal.com/xhtml">
<head>
<title>Demo css</title>
</head>
<body>
<h3 style="font-family:arial;background-color:green;">this is a test css</h3>
</body>
</html>
```

Result



this is a test css

Internal CSS

In internal CSS the style of CSS is specified in the **<head>** section. This is internal CSS, it affects all the elements in the body section. Internal CSS is used in the condition when we want a style to be used in the complete HTML body. For that we can use style in the head tag.

This style performs an action in the entire HTML body.

```
<!DOCTYPE html>
<html xmlns="http://www.bcanotesnepal.com/xhtml">
<head>
  <title>Demo css</title>
  <style>
body {font-family:'Buxton Sketch';color:red;background-color:yellow;}
p {color:blue}
</style>
</head>
<body>
  <h3 this is a test css</h3>
</body>
</html>
```

Result



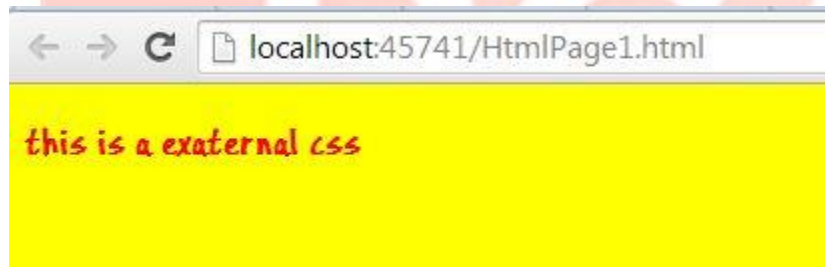
External CSS

In External CSS we create a .css file and use it in our HTML page as per our requirements. Generally external Cascading Style Sheets are used whenever we have many HTML attributes and we can use them as required; there is no need to rewrite the CSS style again and again in a complete body of HTML that inherits the property of the CSS file. There are two ways to create a CSS file. The first is to write the CSS code in Notepad and save it as a .css file, the second one is to directly add the style sheet in our Solution Explorer and direct Visual Studio to use it on our HTML page. How to create and use an External CSS.

1. Right-click on your application name in Visual Studio 2012.
2. Add a style sheet.
3. Write your CSS code and save it.
4. Open your HTML page and drag down the style sheet.
5. Save and RUN.

```
<!DOCTYPE tml>
<html xmlns="http://www.bcanotesnepal.com/xhtml">
<head>
  <title> Demo css </tyle>
  <link href="mystyle.css" rel="stylesheet"/>
</head>
</body>
<h3> this is a external css </h3>
</body>
</html>
```

Result



2. Define different types of formatting tags?

- There are different types of HTML tags are used to format the appearance of the text on your web page

1. Create Headings - The <h1> to <h6> Elements:

Any document starts with a heading. You can use different sizes for your headings. There are 6 levels of headings available, from h1 for the largest and most important heading, down to h6 for the smallest heading. Here is an example of the code for all the headline sizes:

```
<h1>Level 1 Headline</h1>
<h2>Level 2 Headline</h2>
<h3>Level 3 Headline</h3>
<h4>Level 4 Headline</h4>
<h5>Level 5 Headline</h5>
<h6>Level 6 Headline</h6>
```

2. Create Paragraph - The <p> Element:

The <p> element offers a way to structure your text into paragraphs. Each paragraph of text should go in between an opening <p> and closing </p> tag as shown below in the example:

```
<p>Here is a paragraph of text.</p>  
<p>Here is a second paragraph of text.</p>  
<p>Here is a third paragraph of text.</p>
```

3. Create Line Breaks - The
 Element:

Whenever you use the
 element, anything following it starts on the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

Note: The
 element has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break.

Example of basic HTML Program:

```
<html>  
<body>  
  Hello<br />  
  This is an example of Line break.<br />  
  Thank you<br />  
</body>  
</html>
```

This will produce the following result:

```
Hello  
This is an example of a line break.  
Thank you
```

4. Bold - :

The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.

5. Italic - <i> </i>:

Also working the same way as a word processor, italics displays the text at a slight angle.

6. Underline - `<u>` `</u>`:

Again, the same as underline in a word processor. Note that html links are already underlined and don't need the extra tag.

7. Strike-out - `<strike>` `</strike>`:

Puts a line right through the centre of the text, crossing it out. Often used to show that text is old and no longer relevant.

8. Preformatted Text - `<pre>` `</pre>`:

Any text between the pre tags, including spaces, carriage returns and punctuation, will appear in the browser as it would in a text editor

9. Source Code - `<code>` `</code>`:

Similar to tt the text is displayed in a fixed-width font, and is commonly used to show source code. I have used it on this site, along with stylesheets, to show all tags.

10. Typewriter Text - `<tt>` `</tt>`:

The text appears to have been typed by a typewriter, in a fixed-width font. For example: This text is written using the `<tt>``</tt>` tags.

11. Block Quote - `<blockquote>` `</blockquote>`:

Defines a long quotation, and the quote is displayed with an extra wide margin on the left hand side of the block quote.

12. Small - `<small>` `</small>`:

Instead of having to set a font size, you can use the small tag to render text slightly smaller than the text around it. Useful for displaying the 'fine-print'.

13. Centre - `<center>` `</center>`:

A useful tag, as it says, it makes everything in between the tags centered (in the middle of the page).

14. Emphasis - `` ``:

Used to emphasize text, which usually appears in italics, but can vary according to your browser.

15. Strong Emphasis - `` ``:

Used to emphasize text more, which usually appears in bold, but can vary according to your browser.

3. Write down elements used in form with example and form properties and its types?

➤ HTML form elements are used as follows:

The `<input>` Element

The most important form element is the `<input>` element.

The `<input>` element can be displayed in several ways, depending on the `type` attribute. Eg.

```
<input name="firstname" type="text">
```

The `<select>` Element

The `<select>` element defines a **drop-down list**:

Example

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The `<textarea>` Element

The `<textarea>` element defines a multi-line input field (**a text area**):

Example

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The `<button>` Element

The `<button>` element defines a clickable **button**:

Example

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

HTML Form Elements

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

Types of Http Methods

- a. Get
- b. Post
- c. Put
- d. Head
- e. Delete
- f. Patch

- g. Options
 - a. Get Method

The GET Method

GET is used to request data from a specified resource.

GET is one of the most common HTTP methods.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

```
/test/demo_form.php?name1=value1&name2=value2
```

Some other notes on GET requests:

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

b. Post Method

The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1  
Host: bcanotesnepal.com  
name1=value1&name2=value2
```

POST is one of the most common HTTP methods.

Some other notes on POST requests:

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length