CCA-104:Web Technologies

Assianment

Part 1

Q1:Write html and css for following using div

Header		
Navigation	Content	Sidebar
Footer		

Ans. layout.htm

```
        <html>
          <head>
                      <style>
                              #header{
                                      background-color: lightblue;
                                      width:100%;
                                      height:50px;
                                      text-align: center;
                              }
                              #sidebar-left{
                                      float:left;
                                      width:15%;
                                      background-color: red;
                              }
                              #main{
                                      float:left;
                                      width:70%;
                                      background-color: lightgray;
```

```
            }
            #sidebar-right{
                float:left;
                width:15%;
                background-color: red;
            }
            #footer{
                clear:both;
                height: 50px;
                width: 100%;
                text-align: center;
                background-color: lightblue;
            }
            #sidebar-left, #main, #sidebar-right{
                min-height: 600px
            }
        </style>
    </head>
    <body>
        <div id="header">Header</div>
        <div id="sidebar-left">Left</div>
        <div id="main">Main</div>
        <div id="sidebar-right">Right</div>
        <div id="footer">Footer</div>
    </body>
</html>
```

Q2:Write html for the following using table

<table>
<tr><th>H1</th><th>H2</th><th>H3</th></tr>
<tr><td rowspan="2">C1</td><td colspan="2">C2</td></tr>
<tr><td>C3</td><td>C4</td></tr>
</table>

Ans. HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading.

Example

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

Try it Yourself »

Note: Browsers automatically add some white space (a margin) before and after a heading.

Headings Are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

`<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.

Note: Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

Bigger Headings

Each HTML heading has a default size. However, you can specify the size for any heading with the `style` attribute, using the CSS `font-size` property:

Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

This tag is also essential for displaying information in rows and columns on your page. It is meant primarily for displaying formatted data. It can also be used to create the layout of a web page, but that use is being phased out in favor of using Cascading Style Sheets (CSS) for formatting.

The downside to using tables is that they are somewhat complicated to use and understand. The use of an HTML editor is extremely helpful when using tables. Nonetheless, understanding how tables are structured will help you fix any problems.

Deeply nested tables (tables within tables within tables, etc) have been known to cause problems with some browsers and should be avoided if possible.

All of the tags below require an end tag unless otherwise specified.

`<table>`

This tag is creates a table on the web page. But without any additional tags to determing the table rows `<tr>` and cells `<td>`, nothing will appear on your page. A table is constructed using these tags in the same way you'd build a brick wall. Each row begins with a layer of cement `<tr>` and then each brick `<td>` is placed within it.

- **Attributes for the `<table>` tag:**
    - **align** - This can force the table to align to either the left, right, or center of the browser's window.
    - **background** - This attribute points to the location of an image file that will be displayed in the background of the table. If the image is smaller in dimentions than the table's borders, the image will repeat or "tile".
    - **bgcolor** - This establishes the color that is shown in the background of the table.
    - **border** - This determins the thickness of the table's outside borders. The border will appear in a 3D style

    with the table being "raised" off of the page. The default is 0.

    - **cellpadding** - This is the empty space between the border of a table cell ad the content of the cell. Use this to create space between text and the table's borders.
    - **cellspacing** - This alters the thickness of the interior borders of the table.
    - **width** - This changes the width of the table to a value you specify. You can use either a fixed number of pixels or a percentage value.

- Example: `<table align="center" background="bgrnd.gif" border="3" cellpadding="2" cellspacing="3" width="90%">` This will create a table that is centered on the page, has a background taken from a gif image called "bgrnd.gif", a border of 3 pixels, a padding within each cell of 2 pixels, a spacing between cells of 3 pixels and a width of 90% of the browser's window.

`<tr>`

This tag creates a "row" within the table. You will place cell tags `<td>` within each of the `<tr>` tags. You will have a `<tr>` tag for each row within your table and within each `<tr>` tag, you will have a set of `<td>` tags to determine the columns.

- **Attributes for the `<tr>` tag:**
    - **align** - This establishes how the content of each row's cells are aligned. The options are left, right and center. The default is "center."
    - **bgcolor** - This is exactly as the "bgcolor" attribute for the `<table>` tag above except it only applies to the specific row this attribute is placed.
    - **valign** - This determines how the content of cells within this row are aligned vertically. The options are

top, middle, bottom and baseline. The default is "middle." The option baseline lines up text within the row's cells.

- Example: `<tr align="right" bgcolor="yellow" valign="top">` This will create a row within the table that will have all content aligned to the right. The row's cells will all have a background of yellow and the content will all appear at the top of the row's border.

`<td>`

This is the tag for the table's data cells and where you will actually place the content you want within your table. These tags must be placed within a `<tr>` pair of tags. The `<td>` tags determine how many columns the table has. Also, unless altered by an attribute, all `<tr>` tags within a table must contain the same number of `<td>` tags. Thus if you wanted to create a table with three rows and four columns, the table must have three sets of `<tr>` tags that each contain four `<td>` tags. Any attributes created for a `<td>` tag will override any indentical attributes created by the enclosing `<tr>` or `<table>` tags.

- **Attributes for the `<td>` tag:**
  - **align** - This aligns the content of the `<td>` tag. The options are center, left, right and justify. The default is "left." Justify will make text so that it is aligned to both the right and left edges of the cell.
  - **background** - As with the background attribute for the `<table>` tag above except is only applies to the specific table cell.
  - **bgcolor** - As with the `<table>` and `<tr>` tags above except it only applies to the specific table cell.
  - **height** - This determines the height of the cell with either a specific number in pixels or by a percentage of the height of the table.

- **nowrap** - This forces the table cell to be shown as wide as possible so as to display text on as few lines as possible. There are no perametors for nowrap.
- **colspan** - This is where it gets tricky. What this attribute tells the browser is to assume that this cell is spanning more than one cell location within the table. Or, in other words, colspan is used to create a multi-column cell. The default colspan is, obviously, 1. The tricky part is to make sure that the colspan numbers are identicall for each row of the table. If you have a four cells in one row of a table, then you can have two cells in another row as long as the colspan numbers for that row of cells equal four. Thus that row of cells could have one cell with a colspan of 3 and another with the default of 1. The first cell would cover the first three columns of the table and the last cell would match the last, or fourth column. See additional information below.
- **rowspan** - This is exactly as "colspan" above, except it applies to a row of the table instead of the column. It is the number of rows through wich the current table cell extends downward through the table. This number should not be greater than the amount of remaining `<tr>` tags in the table. In other words, there should be "room" for the cell to extend downwards. Also, in all subsequent rows after a rowspan attribute has been used, you need one less `<td>` tag. See addintianal information below.
- **width** - This determines the width of the specific table cell with either a number in pixels or a percentage of the width of the table.

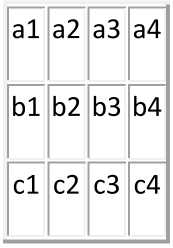
`<th>`

This is a table header cell tag. It is identical to the `<td>` tag in all respects except that any text placed within a `<th>` tag is automatically bolded and centered in most browsers. You should usually use this in the first row to create headers for your columns within a table.

Examples of colspan and rowspan

Here is a table with three rows and four columns. Each row within the table will look identical to the first row (except for the content of the cells).

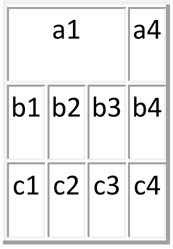
Example of first row: `<tr><td>a1</td><td>a2</td><td>a3</td><td>a4</td></tr>`



a1	a2	a3	a4
b1	b2	b3	b4
c1	c2	c3	c4

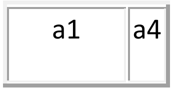
Here's the same table with the "a1" cell given a colspan of 3. Notice that the cell covers what was once cells "a2" and "a3" and there are now a total of only two cells in the first row.

Example of first row: `<tr><td colspan="3">a1</td><td>a4</td></tr>`

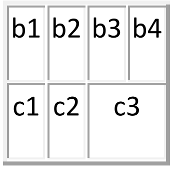


a1			a4
b1	b2	b3	b4
c1	c2	c3	c4

Here's the previous table with "c3" given a colspan of 2. Notice that the colspan spans to the right and not to the left of "a1" and "c3"



a1			a4



b1	b2	b3	b4
c1	c2	c3	

Here's the original table with "a2" given a rowspan of 2. Notice that the second or "b" row looks identical to the first row, but the code is very different.

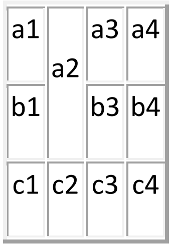
Example of first row:

```
<tr><td>a1</td><td rowspan="2">a2</td><td>a3</td><td>a4</td></tr>
```

Example of second row:

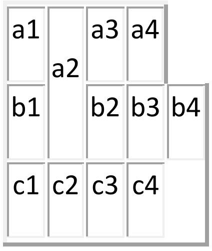
```
<tr><td>b1</td><td>b3</td><td>b4</td></tr>
```

Notice that the second `<td>` (the "b2" cell) is missing in code for the second row. It must be removed because otherwise there would be too many cells in the second row.



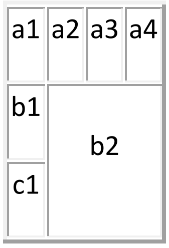
a1	a2	a3	a4
b1		b3	b4
c1	c2	c3	c4

Here is the previous table with the second `<td>` **not** removed in the second row (and thus creating an error in the code for the table). Notice how the row is offset and the "b4" cell is pushed out into a non-existant column.



a1	a2	a3	a4	
b1		b2	b3	b4
c1	c2	c3	c4	

You can also combine colspan and rowspan. This table's "b2" cell has a colspan of 3 and a rowspan of 2.



a1	a2	a3	a4
b1	b2		
c1			

PART-2

Q3:Write a program to display count, from 5to15 using PHP loop as given below.

Ans. Description:

Write a Program to display count, from 5 to 15 using PHP loop as given below.

Rules & Hint

- You can use “for” or “while” loop
- You can use variable to initialize count
- You can use html tag for line break

View Solution/Program

```php
<?php

$count = 5;

while($count <= 15)

{

  echo $count;

  echo "<br>" ;



  $count++;

}

?>
```

PHP

Copy

Tutorials Class - Output Window

Q4:Write a program to display in javascript for unit Conversion from kilometre (km)to centimetre (cm) use of message box is necessary.

Ans. Program to convert centimeter into meter and kilometer

Last Updated: 07-08-2018

Given length in centimeter, the task is to convert it into meter and kilometer.

Examples :

Input: Length in centimeter = 1000
Output:
Length in meter = 10 m
Length in kilometer = 0.01 km

Input: Length in centimeter =6540
Output:
Length in meter = 65.4 m
Length in kilometer = 0.0654 km

Formula to be used:

```
1m = 100cm

                  1km = 100000cm
```

- C++
- Java
- Python3
- C#

- PHP

filter_none

edit

play_arrow

brightness_4

```
// C++ program to convert centimeter

// into meter and kilometer

#include <bits/stdc++.h>

using namespace std;

int main()

{

    float cm, meter, kilometer;

    cm = 1000;


    // Converting centimeter into meter

    // and kilometer

    meter = cm / 100.0;

    kilometer = cm / 100000.0;


    cout << "Length in meter = " << meter << "m"

         << "\n";


    cout << "Length in Kilometer = " << kilometer << "km"

         << "\n";


    return 0;

}
```

Output:

```
Length in meter = 10 m

Length in kilometer = 0.01 km
```

Recommended Posts:

- Program to convert Centimeter to Feet and Inches
- Program to Convert BCD number into Decimal number
- Convert given time into words
- Convert the undirected graph into directed graph such that there is no path of length greater than 1
- Check whether it is possible to convert A into B
- Minimum number of operations to convert array A to array B by adding an integer into a subarray
- Minimum splits required to convert a number into prime segments
- Min steps to convert N-digit prime number into another by replacing a digit in each step
- Minimum flips required to convert given string into concatenation of equal substrings of length K
- Convert 1 into X in min steps by multiplying with 2 or 3 or by adding 1
- Convert given Decimal number into an irreducible Fraction
- Convert vowels into upper case character in a given string
- Minimum number of operations to convert a given sequence into a Geometric Progression | Set 2
- Program for converting hours into minutes and seconds
- Program to Convert Km/hr to miles/hr and vice versa
- Program to convert given number of days in terms of Years, Weeks and Days
- Program to convert speed in km/hr to m/sec and vice versa
- Program to convert KiloBytes to Bytes and Bits
- Program to convert a given number to words
- Program to convert a given number to words | Set 2

---

mansibansal2112