Create a three column layout using CSS, divs and “floating”

Tasks:

- Encode page using HTML5, use semantic tags, ext CSS etc.
- Use a container div with a width, centered to create the web page layout
- Create a 3 column style layout using the “float” technique
- Add content to footer, header and the three columns
- Add padding to each of the columns
- Provide an example of floating images that wrap text
- Add a header and footer tag style
**Design Specifications:**

In this assignment we will be creating a multi-column layout using a technique known as “float” or “floating”. This assignment is very similar to our last assignment where we created a multi-column layout. Before the new CSS technique (used last week) multi-column layouts were created using “floated” divs. This assignment will explain how to work with floating and divs in order to create another type of multicolumn layout.

Each of the main content areas like the header, footer, and the three columns need to have sample content. Add sample content to webpage and there is no required theme on this assignment. You can use the sample or dummy text by using a website like [http://www.lipsum.com/](http://www.lipsum.com/) which generates dummy text.

Make sure to create an index.html and save inside a new a7 folder and use an ext CSS.
Requirements:

Step 1

Required: create a div container layout, use semantic tags and the div column layout as shown below

```html
<div class="container">
  
  <header>
  </header>

  <nav>
  </nav>

  <article>
    
    <div class="leftColumn">
      Left column content
    </div>

    <div class="centerColumn">
      center column content
    </div>

    <div class="rightColumn">
      Right column content
    </div>

  </article>

  <footer>
  </footer>

</div>
```
Step 2

Create a container class in order to create the basic layout. The left and right margins set to auto will center the layout in the page. Notice how the container is set to 1000 pixels wide. The value will determine how wide each of our three columns can be in a later step. For this assignment you are only required to create one web page and it is easiest to create an ext CSS. Optionally you can instead use an internal CSS within the head of the webpage. You must not use both techniques, use one or the other type of CSS.

```css
.container {
  width: 1000px;
  margin-right: auto;
  margin-left: auto;
}
```

Required: make sure to also add a background color or/and background image to the container style shown above. Make sure there is high contrast between the text color and the background so that the text is highly readable.

```css
background-color: #E4FFE4;
background-image: url("paper.png");
```

Required: add a border and rounded corners effect by adding these properties with similar values to your container style. Your values don’t have to match this exactly.

```css
border: 2px solid;
border-radius: 25px;
```
Step 3

Create the tag style for the new semantic tags to act like block elements in older browsers

```css
header, footer, nav, article {
    display: block;
}
```

Step 4

Create a tag style in order to style the body. Define the normal and essential properties that we are used to defining. Make sure to use different values than the example below.

```css
body {
    font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
    font-size: 20px;
    color: #00FF00;
    background-color: #E4FFE4;
}
```

Required: add a line height property and value to the body so that text has more height between lines making text much easier to read. Note there are different types of sizing in CSS and em is known as a scaling factor sort of like saying make the line height 1.6 times the default line height.

```css
line-height: 1.6em;
```

Required: add a background graphic that repeats nicely and is not too bright. Don’t use a very bright jpg image (like a photograph) and do not just use a solid color for your background.

```css
background-image: url("paper.png");
background-repeat: repeat-x;
```
Step 5
Create the .leftColumn class in order to style the div that represents the leftColumn. The style defines a width and it will float left. Remember our container is 1000 pixels wide. 1000 – 300 = 700 pixel width available for our remaining two columns.

.leftColumn {
  float: left;
  width: 300px;
}

Create the .centerColumn class in order to style the div that represents the centerColumn. The style defines a width and it will also float left. 700 – 400 = 300 pixel width available for our remaining center column.

.centerColumn {
  float: left;
  width: 400px;
}

Create the .rightColumn class in order to style the div that represents the rightColumn. The style defines a width and it will also float left. 300 – 300 = 0 pixel width available for our three-column layout, meaning we have used up the exact amount of pixel width available from the original container.

.rightColumn {
  float: left;
  width: 300px;
}

Important note: if we add any border, padding or margins to any of the three columns they would no longer fit within the 1000 pixel width of our container. We will explain how to add some padding in a later step.
**Step 6**

Add sample content to each of our three columns. There is no required theme on this assignment. You can use the sample text by using a website like [http://www.lipsum.com/](http://www.lipsum.com/) which generates dummy text or use your own sample text or come up with a new theme.

Required: Add an image to each of the three main columns.

Preview page in browser, you should have three columns each displayed next to each other. Unlike the last assignment where we had one column that wrapped and displayed content into three columns this layout has three unique div’s that have their own content with the option of having their own unique properties and styling.

**Step 7**

One of the pitfalls of floating is that any additional content that we don’t want to potentially be floated, in this case the footer, may unintentionally appear and display around the floated area. Anything after a float that you don’t want floated should have a class which contains a property clear:both. In our example, this says not to display the footer until after anything is finished floating. It keeps the footer underneath the floated div’s. If we were not using the full 1000 pixel width of the container the footer would try and display next to the right column which is being floated. This style keeps the footer properly underneath the three floated columns or divs.

```css
footer {
    clear: both;
}
```
Step 8

Required: add a padding property to each of the three column styles. The following property and values will add 10 pixels to all of the sides of a column style. Padding adds spacing inside of the column or div. Margin would add spacing outside of the div. Adding a border would put a border around the div. Margin is the space outside of the border and padding is the space inside of the border.

```
.leftColumn {
  float: left;
  width: 300px;
  padding: 10px;
}
.rightColumn {
  float: left;
  width: 300px;
  padding: 10px;
}
.centerColumn {
  float: left;
  width: 400px;
  padding: 10px;
}
```

Preview page in browser, you should now only have two columns displayed next to each other. The third div should now be displayed below the other two div’s. The reason it is no longer displaying properly is because the container width is no longer wide enough to accommodate the new width (we just added padding) of all three columns. The problem is adding padding increases the total width of a column. Important note: if we add any border, padding or margins to any of the three columns they would no longer fit within the 1000 pixel width of our container. Container = 1000 pixels wide. Left column 300 pixels wide + 20 pixels (padding on the left and right side of the left column) = 320 pixels actual width. Right column 300 pixels wide + 20 = 320 pixels wide. Center column 400 pixels wide + 20 = 420 pixels wide. Total width of all three columns 320 + 320 + 420 = 1060 pixels wide.
Modify the container style to accommodate the new width.

```css
.container {
    width: 1060px;
    margin-right: auto;
    margin-left: auto;
}
```

Preview page in browser, you should now again have the three columns properly displayed next to each other. If you added additional margins or borders to the columns you would have to adjust the container to accommodate the new total width.

**Step 9**

Create a float image class in order to float and add box styling to each of the three columns. Once floated, any content after the image wraps around the floated image. Float value can be set to left or right.

```css
.floatLeftImageBorder {
    float: left;
    border: 3px solid #999;
}
```

Required: add margins (in the style shown above) around the floated image in order to put spacing around image and border in order to make text more readable. Remember you are required to add images in each of the three columns and apply floating examples to each of these images. See example on the next page.
Step 10

Create a footer tag style

`footer {
}
`

Required: Add at least two properties with values to the above style. Note: content in the footer is not as important as the main content inside the article tag so you should try and make the text smaller and less visible like using a dark bg with semi dark text may be?

Create a header tag style

`header{
}
`

Required: Add at least two properties with values to the above style. Note: content in the header is considered more important than the main content inside the article tag so you should try and make the text bigger and more visible. Try adding a bg color to the header may be bt make sure text is very readable.
Step 11

Make sure you have added sample content to webpage. For example, add sample content inside each of the main content areas: header, footer and the three columns.

Make sure you have saved page as index.html and save inside an a7 folder. Reminder, it is best to name files and folders using all lowercase and with no spacing in the naming.

Step 12

Upload assignment to your website, add link on your course management page and then email Instructor your assignment’s URL.

Extra credit:

(not required)

- Add a theme and create multiple pages. (but content must be dif than past assignments)
- Add a CSS style nav bar
- Add a web font to your body style, visit [http://www.google.com/fonts#](http://www.google.com/fonts#)