

## IN-CLASS ACTIVITY 2:

When this activity is complete, the “Activity2.html” page will be changed by applying some advanced CSS techniques – many of them CSS3. We will proceed together through this exercise and discuss each step.

Start this activity by downloading “Activity1\_Starter.zip”, unzip the file into a new folder, open the html file “Activity2.html”, and make sure the file appears in the browser. Using the browsers developer tools, check that the file contains no errors.

Complete the following steps:

- 1) Color for List links: pseudo-classes used are :link (initial colour of the link); :visited (color after a link has been selected); and :active for when it is in the process of being clicked.

Pick three colors for each and set the pseudo-class properties for the <a> tag using the CSS3 named colours from the list at [http://www.w3schools.com/colors/colors\\_names.asp](http://www.w3schools.com/colors/colors_names.asp).

- 2) Fix the Footer: We want to fix the footer so it floats above the content by stays anchored to the bottom of the browser window. To accomplish this, we will use the position property of “fixed”. We will then anchor the footer section 0px from the bottom and 75% from the left side. We will give it a z-index a value (any value greater than 0). Finally, we only need to see the portion of the footer that has the link to w3schools so we will clip the area using the clip property and set it to the rectangle shape (the only valid shape at this time) using the following settings:

***clip: rect(top, right, bottom, left);***

where top, right, bottom, and left specify distances from the upper left corner. Play with the numbers to create an appropriate box.

- 3) a) We want to set the background property of every second <li> entry of the unordered list to the color red=240, blue= 155, and green=220. We will do this using the ***rgb(red, green, blue)*** method where the parameters are values in the range of 0 to 255. To change just every second <li> element, we will use the :nth-child(pos) pseudo-class where pos represents the row on which the property will be applied. For our purposes, 2n+1 will target the correct rows. View the result.

b) We want to introduce a gradient so that the background is not so stark. We will do this using the linear-gradient function as the value of the background property. To use the function, the values of the starting color and the ending color must be specified. “Transparent” is a legitimate color – so we will transition from the current color to the transparent color. The function’s first property indicates the angle of the gradient transition relative to the horizontal plane. 90deg will be a good value – but you can play with this.

***linear-gradient(degree, start-color, end-color)***

c) You would like the gradient to end long before the right border of the browser. To stop it earlier, you can specify a percentage value of the width where the color transition ends for the second color value. After transparent add 45%.

d) Now you notice that the fading is too pronounced before the text of the <li> tag ends, so you want the transition to start later than the very beginning – that is, you would like the new background to be solid until a certain point and then transition to transparent. A percentage value after the first color will accomplish this.

The final property should be something like:

***background: linear-gradient(90deg, rgb(240,220,155) 15%, transparent 45%);***

- 4) Larger First Letters: We can make the first letter of every paragraph three times larger by defining the pseudo-class property ::first-letter of the <p> tag. We simply change the font-size to be 3 times larger than the current font using the relative size unit of em. We can also make these letters stand out even more by floating them to the left. Doing so will cause the regular size text to come up and wrap around these larger letters. Simply set the float property to left. If the larger letter seems a bit too close to the smaller text, you can add a margin around the letter. Play with the margin property to find a comfortable setting.

- 5) Heading Properties:

Let us start by setting the font size of all the headings (H1, H2, and H3 in this case) to twice the size of the base font (ie. 2em). Also, we would like to change the margin value to 0px (no margin before or after) and make sure every word starts with a capital letter. We can do the latter by setting the text-transform property to capitalize.

Next, we want the H1 heading to be 3 times larger than the base font and we want the text to be center aligned (text-align property). Also, we want to add a background color to white – but not a solid white. The opacity needs to be set to 20%. This is accomplished using the rgba(red, green, blue, alpha) function. The alpha channel is the level of opacity stated in a range from zero to one. Zero represents totally transparent and 1 is solid. 20% would be represented as 0.2. You can play with this.

- 6) Formatting the Articles: The first section has a couple of articles and we would like them stand out and be side by side. Let's start by setting the background of this section (class = article) to red but with a 10% opacity. Also add a padding of 10px.

Next, let's format the paragraphs in this section using a contextual selector. The properties we need to set are as follows:

- i) Set the text alignment to "justify" and the font family to "New Courier", courier, or monospace.

- ii) Set the width of each paragraph to between 360 and 450px – using the max-width and min-width properties.
- iii) Set the background to the color white with 50% opacity
- iv) Set the padding of each paragraph to 40px except the right which will be 50px
- v) Set the margins to 0 on the top and left, 10 on the right and 5 on the bottom using the property ***margin: top, right, bottom, left;***
- vi) Float each paragraph to the left

There is another issue – the text from the following section is floating up around the two stylized articles. We need to stop this section from floating up by setting the clear property of the following section to the value “both”. Look in the HTML file, identify the class setting of the following section and create the property.

- 7) There are 10 other sections in the document. It would helpful if they were numbered - but I don't want to change the HTML. Let us make CSS number the sections for us. To do this:
  - i) In the body element definition, add ***counter-reset: section 0;***
  - ii) Now add the following definition:
 

```
section:not(.article) h2:before{
    counter-increment: section;
    content: counter(section) " ";
}
```
- 8) Finally, let us narrow the paragraphs of the last 10 sections to a width of 80% of the browser window. This will be accomplished using a contextual selector applying this setting to <p> elements in the <section> elements except the one marked as belonging to class="article".