# Web Development 1 – Lab Session 8b

**CSS for multi-column page layouts**

Today you will be using CSS styling to create multi-column layouts useful for nav-bars, headers and footers, and columns in the main section of web pages.

**Exercise 1: Create 2 sections, each containing a heading and a paragraph**

Create a new HTML page containing the following in the ‘body’

**<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* section - block \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->**

**<section>**

**<h2>BLOCK 1</h2>**

**<p>**

**content content**

**</p>**

**</section>**

**<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* section - block \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->**

**<section>**

**<h2>BLOCK 2</h2>**

**<p>**

**content content content**

**</p>**

**</section>**

Add some style, i.e. background /border so we can see the extend of each block:

**section {**

**background-color: lightgray;**

**border: 0.1rem red solid;**

**padding: 1rem;**

**margin: 1rem;**

**}**

View the page in Google Chrome:



**Exercise 2: Style a flex ‘container’ parent ‘div’ around the two sections**

Add ‘div’ tags around your two sections, giving this parent ‘flex container’ div the class “column\_container”:

**<div class="column\_container">**

**<section>**

**etc.**

**...**

**</div>**

Add a CSS rule to style this div as a flex container:

**.column\_container {**

**display: flex;**

**}**

You should now see how the default layout for flex ‘items’ – the child blocks inside a flex ‘container’ – is to lay them out left-to-right in the same horizontal section (row):



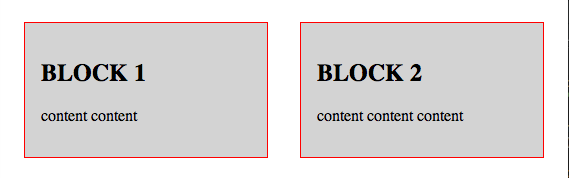
**Exercise 3: Style the section flex ‘items’ to stretch equally using up any spare width**

Add a CSS rule to style each section to stretch equally:

**.column\_container section {**

**flex: 1;**

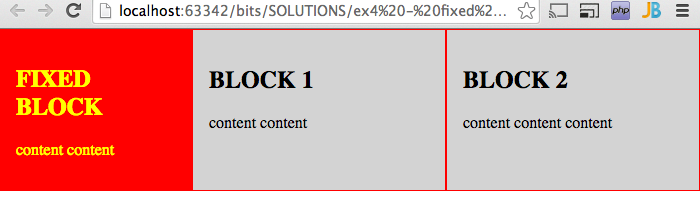
**}**



**Exercise 4: Create a fixed width section**

Copy and adapt the previous exercise:

* Remove body spacing
* Add an ‘aside’ into your DIV
* Style the aside to be a fixed width of 10rem, with a red background and 1rem padding



Hints:

**<div class="column\_container">**

**<!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* section - block \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->**

**<aside>**

**<h2>FIXED BLOCK</h2>**

**<p>**

**content content**

**</p>**

**</aside>**

**...**

**.column\_container aside {**

**background-color: red;**

**color: yellow;**

**width: 10rem;**

**padding: 1rem;**

**Exercise 5: Create a left / right aligned footer using 2 flex items**

NOTE – start a new HTML file and new CSS file for this exercise, named exercise5.html and exercise5.css.

Write HTML for a footer, containing 2 sections. The first (left) section should be a flex item with default left-aligned content. The section (right) section should be a flex item, with added CSS stye rule that right-aligns text content.

Hints:

**<footer class=”column\_container”>**

**<section>**

**Fantastic Page Layouts Limited, Dublin, Ireland.**

**</section>**

**<section class="right">**

**Copyright &copy; 2009-2013**

**.right {**

**text-align: right;**

* Add other style rules for footer dark background, light text colour, padding etc.

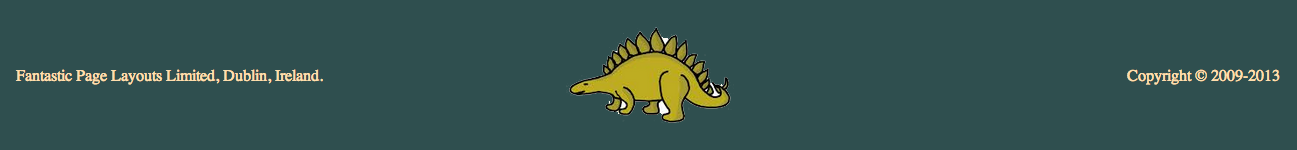
Your page footer should look as follows:

Macintosh HD:Users:matt:Copy:2014_TEACHING:comp_1_webDev1:labs:lab08_page_layouts:lab08_page_layouts:images:ex4.png

**Exercise 8: Vertically center contents of 3 flex items in a footer**

(build on your answer ot previous exercise …\_)

Write code for a footer that has content of different heights, which we wish to style with vertical alignment. Add 2 third (middle) section to your footer from Exercise 4 …



Hints:

**footer {**

**...**

**align-items: center;**