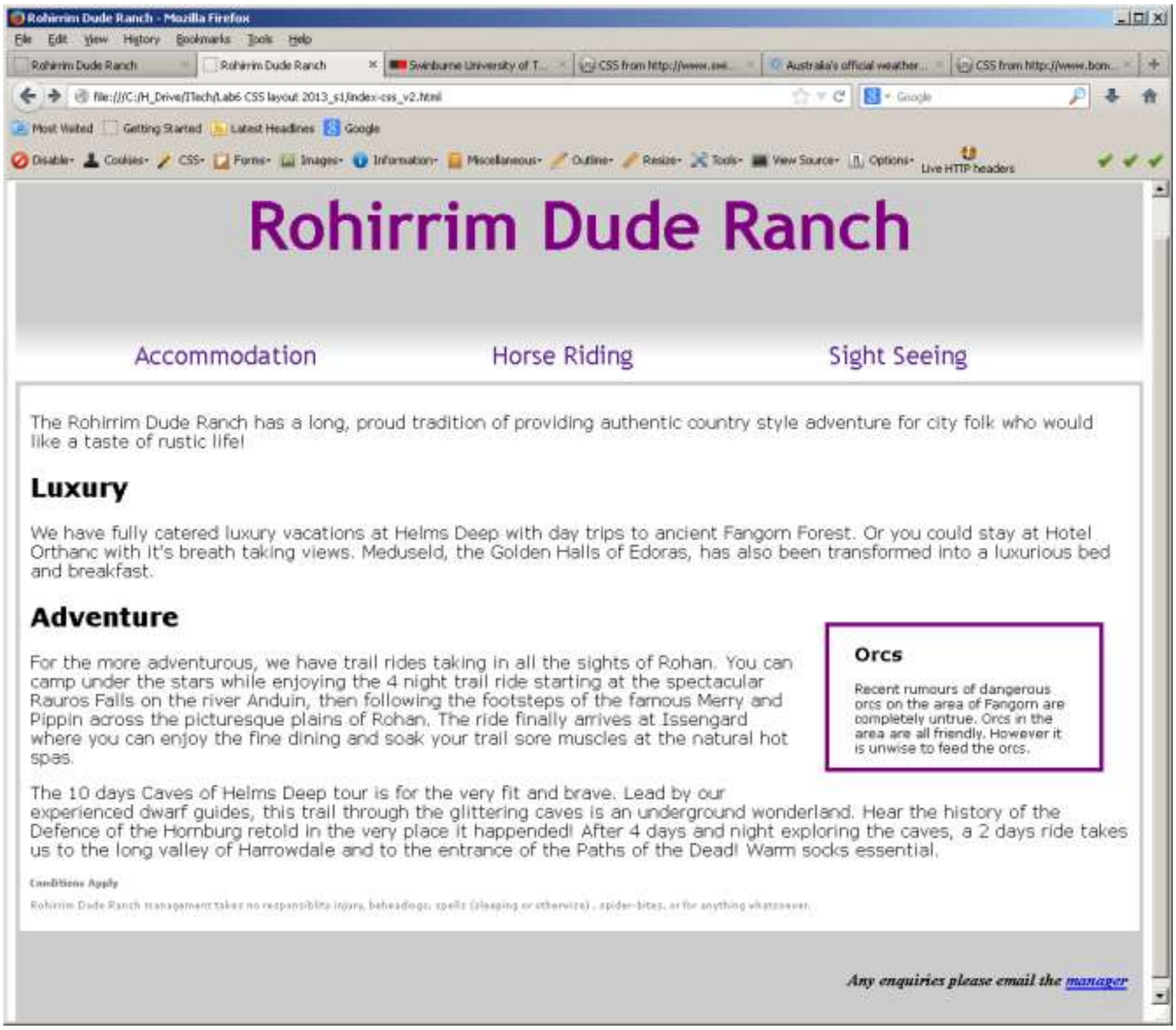
Lab 5 – CSS Page Layout Techniques

Aims:

- To learn some techniques for formatting page layout in CSS.

Task 1: CSS Layout

The purpose of this task is to give you more experience with CSS based page layout, and a chance to explore some more advanced aspects of CSS. In particular you will take a pre-existing HTML page and format it as below using the CSS box model to achieve a fluid layout.


Rohirrim Dude Ranch - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/H_Drive/ITech/Lab6 CSS layout 2013_s1/index-css_v2.html

Rohirrim Dude Ranch

Accommodation Horse Riding Sight Seeing

The Rohirrim Dude Ranch has a long, proud tradition of providing authentic country style adventure for city folk who would like a taste of rustic life!

Luxury

We have fully catered luxury vacations at Helms Deep with day trips to ancient Fangorn Forest. Or you could stay at Hotel Orthanc with it's breath taking views. Meduseld, the Golden Halls of Edoras, has also been transformed into a luxurious bed and breakfast.

Adventure

For the more adventurous, we have trail rides taking in all the sights of Rohan. You can camp under the stars while enjoying the 4 night trail ride starting at the spectacular Rauros Falls on the river Anduin, then following the footsteps of the famous Merry and Pippin across the picturesque plains of Rohan. The ride finally arrives at Issengard where you can enjoy the fine dining and soak your trail sore muscles at the natural hot spas.

Orcs

Recent rumours of dangerous orcs on the area of Fangorn are completely untrue. Orcs in the area are all friendly. However it is unwise to feed the orcs.

The 10 days Caves of Helms Deep tour is for the very fit and brave. Lead by our experienced dwarf guides, this trail through the glittering caves is an underground wonderland. Hear the history of the Defence of the Hornburg retold in the very place it happended! After 4 days and night exploring the caves, a 2 days ride takes us to the long valley of Harrowdale and to the entrance of the Paths of the Dead! Warm socks essential.

Conditions Apply

Rohirrim Dude Ranch management takes no responsiblity injury, beheadings, spells (sleeping or otherwize) , spider-bites, or for anything whatsoever.

Any enquiries please email the manager

Note: *All the HTML markup you need is provided for this exercise. You do* ***not*** *need to edit the HTML. The only thing you need to do is write CSS!*

Step 1: Linking the External Stylesheet

Download the file "`index-css.html`" the lab folder in BlackBoard and save it to and appropriate folder on your computer. Open it. It doesn't look very nice (yet!), but we can see and select all the text, which makes the content easily "searchable" by search engines. Validate the html file before you start to ensure its ok.

Notice in the `head` section of the document, a `link` to an external stylesheet named `style.css`. Use Notepad++ or similar to create a new style sheet with that name and insert a header comment.

```
/*style.css
author: [your name]
created: [enter date]
last modified: [enter date]
description: A style sheet for index-css.html
*/
```

Step 2: Navigation

Let's fix up the navigation bar. Currently the links are contained in an unordered list. Logically this is okay since a collection of links really is just an (unordered) list of links. But we don't want the "bullet point" look so let's change it. There is already a `nav` tag around the navigation list to help identify it, so let's use that to get rid of the bullet points.

In the stylesheet create a style rule:

```
nav ul
{
      list-style:none;
}
```

This rule uses a ***contextual*** selector so that only a `ul` element which is a child of the `nav` element is affected. It sets the list style to "`none`" so the bullet points go away.

Now we want the links on a single line and to space out the menu across the page. Try adding this rule:

```
nav li
{
      float:left;
      width:30%;
      text-align: center;
      font-size: 1.5em;
      padding:5px;
}
```

Save the CSS and refresh the page in Firefox. What is the result?

Oops! They are all on the same line, but because the list items are now "floating", the rest of the document tries to fit on the same line. Let's fix this by adding a rule to the rest of the content that follows after the list:

```
article
{
      clear:both;
}
```

Better? Now let's fix up the spacing by applying some CSS formatting to the links, to the <`a`> tags themselves:

```
nav a
{
      padding:0.2em 0.6em;
      text-decoration: none;
}
```

Once again, we are using a ***contextual*** selector to affect only the `a` tags which are children of the `nav` element. Without this context, all `a` tags in the whole document would get the same style.
The "`text-decoration: none;`" property gets rid of the underline from the links. But it would still be nice to have some visible way to reinforce that fact that these are real links when the user hovers their mouse over them. To do this, we can add the following rule:

```
nav a:hover
{
      text-decoration: underline;
}
```

This rule uses the same context as the previous rule, but also uses the pseudo class (`a:hover`) to only affect anchors that are in the hover state. See how CSS lets us be highly selective in our application of rules?
We now have our underline back, but only when users hover over the anchor.

But what if the device does not have a mouse? We could add the `a:focus` pseudo class as well, so that the link is highlighted if a user 'tabs' through anchors on the web-page.

```
nav a:focus
{
      text-decoration: underline;
}
```

However, to stop bloating the CSS with too many rules, we should simplify this by *grouping* the selectors:
`nav a:hover, nav a:focus { … }`

Select the 'Accommodation' menu item and press `tab`. The focus should move to the next menu item.

Now that the links are working the way we want them to, let's improve the look of the navigation bar. With the included files is a tiny image file called "`fade.jpg`". We want to tile this image horizontally across the navigation bar. We can do this by setting it as a background image with the following rule:

```
nav
{
      background:url(fade.jpg) repeat-x top left;
      float:left;
      width:100%;
      margin:0;
      padding:5px 0;
}
```

The "`background:url(fade.jpg) repeat-x top left;`" is an example of setting multiple values with a single rule property. In this case we are setting the background image with the `url` value, and specifying that the image only repeat horizontally (along the x axis), and that it starts in the top left corner of the `div`.
Now we (should) have a pretty nice looking navigation bar.

Check that your CSS validates using the Web Developer toolbar. Fix any errors.

Step 3: Background and Font formatting

First we will set the background colour with a hexadecimal value #ccc. We will select on a HTML element called <main> that encapsulates all the content of the document. It sits directly under the <body> and you can only have one instance.

Note: In some browsers '`main`' might not render as a block level element, `display:block;` fixes that.

```
main{ background-color: #ccc; display:block; }
```

We will use one `font-family` and `color` for all our headings using a ***grouping*** selector. Note the use of alternate and default fonts, and the need to use "quotation marks" if there is whitespace in the font name:

```
h1, h2, h3, nav li
{
      font-family : "trebuchet ms" , Arial, sans-serif;
      color: purple;
}
```

Validate and refresh the page

Make the <h1> stand out more with an ***element specific selector*** that increases the font size and centres it.

```
h1 {
      font-size: 4em;
      text-align: center;
}
```

To style the text in the <article>, add some extra property values, and set the border, padding and background:

```
article
{
      clear:both;
      font-family: Verdana, Arial, sans-serif;
      border: 4px solid #ccc;
      padding: 10px;
      background-color: white;
}
```

Note that the font of the `article` does not override the fonts of the headings. Why not? Because <h1>, <h2> etc. are children of `article` but the rules with more specialised selectors have precedence.

Validate and refresh the page

Now set the font for the “conditions” (the management don’t want to make these too prominent! ☺).
Here we will use a ***common class selector*** i.e. the class attribute of the <h3> and <p> elements in the conditions section:

```
.fineprint
{
      color: gray;        /*note: CSS3 validates “grey” and “gray”; CSS2 only “gray”*/
      font-size: xx-small;
}
```

We will right align the text in the footer and give it a bit of padding space around it.
We will use the short form `font` property to format the font:

```
footer
{
      text-align : right;
      font: italic bold 1em Verdana, Arial, sans-serif;
      padding:1em;
}
```

Validate and refresh the page

Step 4: Layout

In this step you will create a formatted `<aside>` using the CSS `float` property and use `min-width` to preserve the integrity of the layout when the page is resized.

Look at the layout of the page at the top of this Task. An aside box typically floats, and the body text wraps around it. Let’s format it, by adding the following rule:

```
aside {     float: right;
            font-size: 75%;
            width: 20%;
            border: 4px solid purple;
            padding: 0 2em 0 2em;
            margin: 2em;
      }
```

Note: We have padded the border so there is no padding on the Top or Bottom and 2em on the Right and Left (mnemonic = TRouBLe. Top Right Bottom Left)

Finally, we want to maintain the integrity of the layout when the page is resized. What happens to the menu is narrowed? Notice the menu becomes unusable. To prevent the overlapping we will add some `min-width` properties.

```
main, nav {
      min-width: 40em;
      }
```

Different layout rules would of course need to be created for different devices like smart phones.

Validate and refresh the page to retest the resizing.

Step 5: Comments and Tidy up

Rearrange the CSS rules so they are logically grouped.

- Place the most general selectors at the top of the stylesheet, and cluster selectors referring to particular elements together. It can help readability if the order of the selectors follows the structure of the document, for example:

```
Generic element selectors (h1, p, a, etc.)
#danger /* id style */
.last  /* class style */
etc…
```

- If a selector appears more than once combine all the `property:value` pairs into a single rule.
- Alternatively it may be clearer to have a more general structure that separates functional grouping such as *typography, layout*, etc, and then order the selectors within those functional groups.

Where you use numeric values for colours, indicate in comments a name for the colour.
This could be inline or a header comment such as:

```
/*
swatch colors
-------------
1d74be - mid blue
e1e1e1 - light grey
a3a4a4 - mid grey
8a8a8a - grey
======== */
```

Include line comments for any rules whose intent might be unclear. In particular, it may be helpful to explain the use of properties such as float or min-width / max-width:

```
e.g   min-width: 40em; /*minimum content size to ensure menu is readable */
```

Validate and refresh the page to make sure you have not made any errors cutting and pasting.

Step 6: Deploy

Upload your files to Mercury and validate your CSS rules with the W3C online CSS validator http://jigsaw.w3.org/css-validator/ or 'Validate CSS' using the Web Developer toolbar

Your completed Web page and CSS should be checked by your tutor.

Other Links/Resources

- HTML Validator http://validator.w3.org
- Max Design CSS resources http://css.maxdesign.com.au
- W3Schools CSS reference http://www.w3schools.com/css/css_reference.asp
- HTMLRef (includes CSS reference) http://www.htmlref.com
- http://www.onextrapixel.com/2012/06/05/15-best-css-practices-to-make-your-life-easier/

This exercise is similar to an example on the WebCredible web site.
http://www.webcredible.co.uk/user-friendly-resources/css/css-navigation-menu.shtml

Validating as CSS3:
By default, the "Web Developer Extension" Toolbar validates as CSS2.1 with 'Validate CSS' and as CSS3 with 'Validate Local CSS'.
You can set the "Web Developer" Add-on in Firefox, (or the similar add-on in 'Google Chrome'), to validate against CSS3, by setting the 'Option' to 'Validate CSS' to 'CSS3' with the URL
http://jigsaw.w3.org/css-validator/validator?profile=css3&warning=0&uri=