4 - CSS

Our theme is almost complete, but it looks ugly - style sheets to the rescue!

Confluence and Theme Builder combined will by default make most content look quite nice. However, there are several things we wanted to further customise:

- the overall deign of the web page
- extra styling for tabs
- styling of widgets (in portals, eg. right sidebar panels)
- customisation of text styles such as headings and links
- customisation of the {code} macro

The entire style sheet was added to the "Custom CSS" setting on the CSS Tab. We decided to allow some minor discrepancies between browsers so we didn't use any Internet Explorer specific CSS or hacks.

Page design

Our page design requires the use of several graphics - you can see the actual graphics used by our theme on the Theme Images page. The URL's used for background images shown in the CSS below are just the links to these images taken from the attachments screen of the Theme Images page (by right-clicking and choosing "Copy Shortcut" or "Copy link location" depending on which browser you're using).

To wrap our pages in these graphics requires an understanding of the HTML output by Theme Builder - luckily we've got a page that describes it quite well: HTML Structure - you might want to open this in a new window/tab so you can refer to it for the remainder of this tutorial.

Outer Borders

The outer page is wrapped in graphical borders to give a drop-shadow effect and the subtle gradient effect behind the header panel.

First we set the width and two margins of the "atb-shell" element so there's some space around our page. We also define the left border image and the overall background colour for the transparent panels that we defined in 1 - Layout and Panel Content:

```
.atb-shell {
 margin-bottom: 10px;
 margin-top: 15px;
 width: 97%;
 background: #eee url(/download/attachments/20808042/left_border.gif) repeat-y scroll top left;
}
```

Because the atb-shell element is the outermost element we're going to customise in our style sheet, all the other elements discussed later in this tutorial appear "above" it as far as the browser is concerned, allowing us to display the rounded corner images over the left border image.

NB: It's not the order in which the various blocks of CSS are output that cause this effect - it's the structure of the HTML.

Now we add the right border:

```
.atb-p1 {
 background: transparent url(/download/attachments/20808042/right_border.gif) repeat-y scroll top right;
}
```

And then the subtle gradient background that appears behind our header panel:

```
.atb-upper {
 background: transparent url(/download/attachments/20808042/top-bg.gif) repeat-x scroll top left;
}
```

We can't add that background to the .atb-header element because it would appear in front of the rounded corners at the top of the page.

Speaking of which:

```
.atb-u1 {
 background: transparent url(/download/attachments/20808042/top_left_border.gif) no-repeat scroll top left;
}
.atb-u2 {
 background: transparent url(/download/attachments/20808042/top_right_border.gif) no-repeat scroll top right;
}
```

Next, we add the bottom border:

```
.atb-p2 {
 background: transparent url(/download/attachments/20808042/bottom_border.gif) repeat-x scroll bottom left;
}
```

And then the rounded corners at the bottom:

```
.atb-p3 {
 background: transparent url(/download/attachments/20808042/bottom_left_border.gif) no-repeat scroll bottom
left;
}
.atb-footer {
 background: transparent url(/download/attachments/20808042/bottom_right_border.gif) no-repeat scroll bottom
right;
 padding-bottom: 5px;
}
```

We've added some padding to the bottom of the footer panel to make sure the text shown in the panel doesn't overlap the bottom border.

We also need to add some padding (7px on the right, 5px on the left) to the lower portion of the page which contains the sidebars and everything between them for the same reason:

```
.atb-lower {
 padding: 0 7px 0 5px;
}
```

We then add some padding to our menu and navigation panels (5px on the left and right), again so it's content doesn't overlap the border graphics:

```
.atb-menu, .atb-navigation {
 padding: 0 5px;
}
```

Because we're using the default menu design, we also need to change the menu background and border colour:

```
.atb-navigation div.dynarch-horiz-menu {
 background-color: #eee;
 border-style: none;
 border-top: 1px solid #cacaca;
}
```

You may have noticed that our header graphics actually include a border line, but we've set our header panel height (back in 1 - Layout and Panel Content) so it doesn't get shown. Instead the border set via CSS on the top of the menu *bar* is shown. We've taken this seemingly strange approach so that when the menu *panel* is shown (ie. our optional site message banner) it has both a border at the top (from the theme graphics) and a border at the bottom (from the menu bar).

Inner border

All of the outer borders are now complete and we could have probably stopped at this point, however we wanted to have a border around the tabs and page content.

The following style creates the grey border that appears between the menu bar and the rest of the page:

```
.atb-leftslider, .atb-title {
 border-top: 1px solid #ccc;
}
```

Getting the border to appear on the left sidebar is more tricky because we need the tab images to appear on top of it. To get round this problem we made a 1 pixel background image and applied it to the sidebar:

```
.atb-leftSidebar {
 background: #eee url(/download/attachments/20808042/vert-eee-border.gif) repeat-y scroll top right;
 padding-right: 0px;
}
```

To allow our tabs to appear flush with the side of the left sidebar we also had to remove the default padding on the right hand side of it.

At this point it's also worth removing the padding from the top of the right sidebar so that the widgets in our portal are top-aligned with the main page content:

```
.atb-rightSidebar {
 padding-top: 0;
}
```

Ok, now back to our inner borders, this time the rightmost border:

```
.atb-title, .atb-content, .atb-footnote {
 border-right: 1px solid #ccc;
}
```

You might be wondering why we didn't just add that border to the left side of the right slider bar... We wanted to have a very subtle 3D effect by having an adjacent white border:

```
.atb-rightslider {
 border-left: 1px solid #fff;
}
```

This gives the very subtle appearance of the page being "inset" in to the surrounding grey borders.

Next, we need to do the bottom border:

```
.atb-leftslider, .atb-footnote {
 border-bottom: 1px solid #ccc;
}
```

Tabs

Now we've almost finished our inner border, it's time to turn our attention to the tabs.

The first thing we want to do is hide the little black arrow image that appears in the left slider bar. Although the slider will still be active (try clicking it - it'll hide the tabs!), visually we only want it to depict the colour of the selected tab so the arrow image has to be removed:

```
.atb-leftslider-img {
 visibility: hidden;
}
```

It's important to note that we've hidden it using "visibility" - this forces the browser to still render the image, ensuring that the slider bar doesn't disappear due to having no content.

We now need to make un-selected tabs look separate from the slider bar. Because the tabs appear over the inner border (which you'll remember we added using a 1 pixel background image) we need to add a real border - to the tab images:

```
.atb-leftSidebar a img {
 border: 1px solid #ccc;
 -moz-border-radius-topleft: 4px;
 -moz-border-radius-bottomleft: 5px;
 margin-bottom: 6px;
}
```

By placing the border all the way around the tab images we also complete the last part of our inner border.

For Firefox users, we also add rounded corners to the tab border to make them look a little nicer and then add a margin to the bottom to space the tabs vertically.

But what about the selected tab? It needs to join up with the left slider bar to make it look "selected". Thanks to the "stab" class we added back on page 3 - Tabs, this is fairly trivial - we simply remove the right border:

```
.atb-leftSidebar a.stab img, .atb-leftSidebar a:hover img, .atb-leftSidebar a:active img {
 border-right-style: none;
}
```

You'll notice that we also remove the right border when the mouse hovers over a tab and when it's been clicked (active).

Header panel

We next decided to improve the styling of our header panel content, starting with the links:

```
.atb-header a:link, .atb-header a:hover, .atb-header a:visited, .atb-header a:active {
 color: #526288;
 text-decoration:none;
 vertical-align:middle;
}
.atb-header a:hover {
 text-decoration:underline;
}
```

Next, we customised the space title:

```
.atb-header .space-title {
 font-size: 18px;
}
```

And then the breadcrumb trail:

```
div.breadcrumbs {
 margin-top: 4px;
}
```

Menu panel

Because our menu is using the default design, we wanted to remove the background image when the mouse hovers over items on the menu bar and instead using a dark background and white text:

```
/* clean mouse over */
.atb-navigation div.dynarch-horiz-menu table tr td.item.hover,
.atb-navigation div.dynarch-horiz-menu table tr td.item.active {
 background-color: #526288;
 background-image: none;
}
.atb-navigation div.dynarch-horiz-menu table tr td.item.hover .label,
.atb-navigation div.dynarch-horiz-menu table tr td.item.active .label {
 color: #fff;
}
```

The default menu CSS also added too much spacing either side of our icon-only links on the right side of the menu, so we added the following:

```
/* Shrunk menubar icons */
.atb-navigation div.dynarch-horiz-menu table tr td.item.shrink-width,
.atb-navigation div.dynarch-horiz -menu table tr td.hover.shrink-width,
.atb-navigation div.dynarch-horiz-menu table tr td.active.shrink-width {
 border:1px solid transparent;
 padding:2px 1px 2px 3px;
}
```

We also felt that the separators on the menu bar were a bit "chunky" for our liking and decided to customise those as well:

```
/* Separators on the menu bar */
.atb-navigation div.dynarch-horiz-menu table tr td.separator div {
 border-left:1px solid #ACA899;
 border-right:1px solid #FFFFFF;
 border-style:none solid;
 height:100%;
 margin:2px 3px;
 opacity:0.7;
 width:0px;
}
```

To make it more obvious which items on the menu bar would take you to another page when clicked, we decided to set a the mouse cursor to "pointer" when you hover over them (hover over "Home", which will take you to the home page, and "View", which won't go anywhere, to see the effect on the cursor):

```
/* Add pointer cursor to menus which have explicit actions */
div.dynarch-horiz-menu table tr td.explicit-action {
 cursor:pointer;
}
```

Note: We've not restricted the style above to the .atb-navigation panel as we want it to take effect on all menus, regardless of where they may appear.

Footnote panel

When content appears in the footnote panel, it can sometimes look as if it's part of the main page content. To get round this issue, we added a `div` with class of "line" back in 1 - Layout and Panel Content and apply a dotted border and some padding to the top of that `div`:

```
.atb-footnote div.line {
 border-top: 1px dotted #ccc;
 padding-top: 3px;
}
```

Because we're also displaying our own "Powered By" line (a requirement of the Atlassian Confluence and Adaptavist Theme Builder EULA) we can also hide the standard "powered by" message that appears at the very bottom of the page:

```
#poweredby {
 display: none;
 visibility: hidden;
}
```

Content formatting

After completing the overall page design, we then decided to customise certain elements of the content as well...

Headings

We used custom fonts and sizes for our headings (including those within blog posts):

```
/* headings */
h1, h2, h3, h4, h5, h6, a.blogHeading {
 font-family: tahoma, arial, helvetica, sans-serif;
 font-weight: normal;
}
h1 {
 color: #0f49a2;
 font-size: 18px;
}
h2, a.blogHeading {
 color: #1d62d1;
 font-size: 16px;
}
h3 {
 color: #623b76;
 font-size: 14px;
}
```

Links

We also wanted to customise links and items in the page tree ("Navigator" widget in the right sidebar):

```
/* links */
a:link, a:visited {
 text-decoration:none;
}
.atb-page a:hover, .atb-page a:active, .atb-page a:focus,
.atb-page a.node:hover, .atb-page a.node:active, .atb-page a.node:focus,
.atb-page a.prnnode:hover, .atb-page a.prnnode:active, .atb-page a.prnnode:focus {
 background-color: #526288;
 color: #fff;
 text-decoration:underline;
}
a:hover img, a:active img, a:focus img {
 background-color: #fbfbdc;
}
/* page links, eg. wiki layouts */
.page-links {
 white-space: nowrap;
}
.page-links img {
 margin-left: 6px;
 margin-right: 2px;
}
```

Expand macro

We use the expand macro all over our site and wanted to make it far more obvious that the title can be clicked to show the content:

```
/* expand macro */
expand-title {
 color: property.style.linkcolour;
}
expand-title:hover {
 text-decoration: underline;
}
```

Code macro

We also use the code macro extensively (as you can see from all the blocks of CSS on this page!) so we wanted to make it's output look a lot nicer:

```
/* code macro */
.atb-page div.code {
 border-style: none;
 margin: 0;
}
.atb-page div.code pre {
 font-family: 'andale mono', 'lucida console', monospace;
}
.atb-content div.codeContent {
 background-color: #f3f4f3;
}
```

Portals and Widgets

We're using portals and widgets on home pages and in sidebars - all prominent places - so they also need some CSS attention:

```
/* spacing between portal columns */
.portal-column.column2 {
 padding-left: 8px;
}

/* widget */
.widget {
 background: #fff url(/download/attachments/20808042/widget-bg.gif) repeat-x scroll bottom left;
 border: 1px solid #bbb;
 margin-bottom: 6px;
 padding: 6px;
}
.atb-content .widget {
 -moz-border-radius: 4px;
 border-radius: 4px;
}
.atb-rightSidebar .widget {
 border: 1px solid #aaa;
 border-bottom-color: #fff;
 border-right-color: #fff;
 padding: 3px;
 width: 170px;
}
.atb-rightSidebar .widget.collapsed {
 background-color: #eee;
 background-image: none;
 border-style: none;
}

/* widget titles */
.widget-title {
 font-color: #526288;
 font-weight: bold;
 height: 20px;
 vertical-align: center;
}
.widget-title img {
 float: left;
}
```

And that's it. You're looking at the results!