



# CSS - tabbed navigation

» this tutorial will take you through the steps of setting up a tabbed menu for site navigation - inspired by Douglas Bowman, Founder and principal of Stopdesign and an expert in the use of CSS for "simple, clean, forward-thinking design" (quoted from 'a list apart', [www.alistapart.com](http://www.alistapart.com)).

He discusses this technique in an article and named it "the sliding doors of CSS" - due to the use of 2 images for each tab, allowing for flexible editing of button titles.

The use of tabbed navigation is very popular, especially in larger scale sites and allows for an user-friendly menu system that will remain flexible enough to add and delete menu items without major time-consuming adjustments to the overall page. One very important advantage is an easily recognisable breadcrumb by highlighting the current tab selected - allowing the site visitor to recognise at one glance which page is currently open.

As I myself am not a fan of tabs, especially the very popular rounded rectangular appearance - my graphics will look slightly different from the norm - hopefully encouraging you (given your brief allows such style) to take a more creative approach yourself ;)

This tutorial does not rely on any resource files - which you can nonetheless download if you prefer.

## page in browser window

(see website to preview sample file)



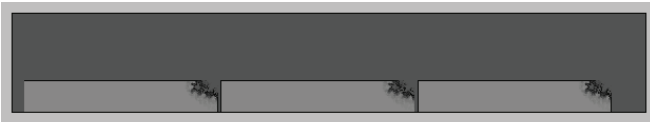
# graphics set up

## tab graphic

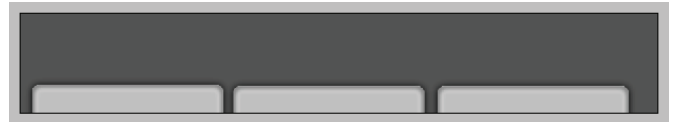
To set up the desired tab effect - we will use 1 graphic, cut into 2 different parts, one for each horizontal side of the tab. This set up will create the illusion of 1 solid graphic, ie our tab, by overlapping the 2 graphics and controlling their stacking order. Its major advantage is the flexibility in the length of each button's title - through the overlap, we can extend each tab to whichever length the text requires, without having to create several graphics which in turn would cause each update to become more and more time consuming.

Just for those of you who might need a rounded corner tab - I have included the graphics for this as well as the visuals used in this example.

sample tab - mock up



rounded corner tab - mock up



full tab graphic



full tab graphic



- create your chosen tab in Photoshop, following the settings below:

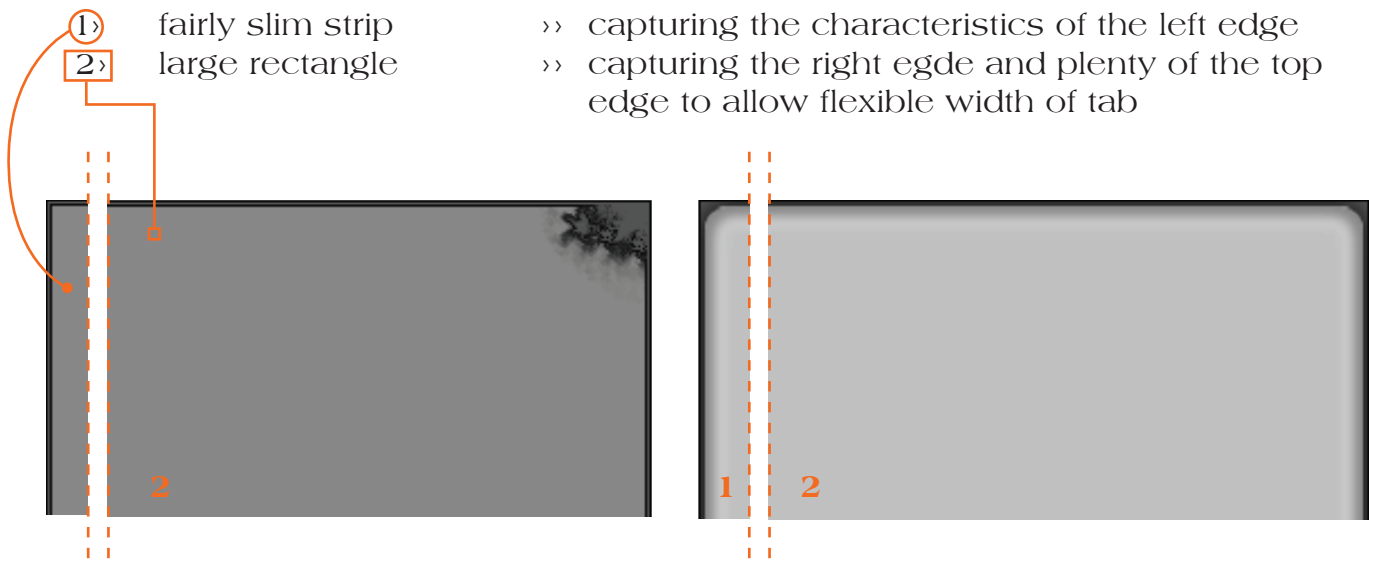
dimensions	300px x 150px
dpi	150px
colour	RGB

The dimensions are intentionally quite large to allow flexibility later on with using larger or bold fonts. You can choose to have a transparent background and compensate for this later on - in this example however, the background colour is set to the intended background colour of the overall site (#666).

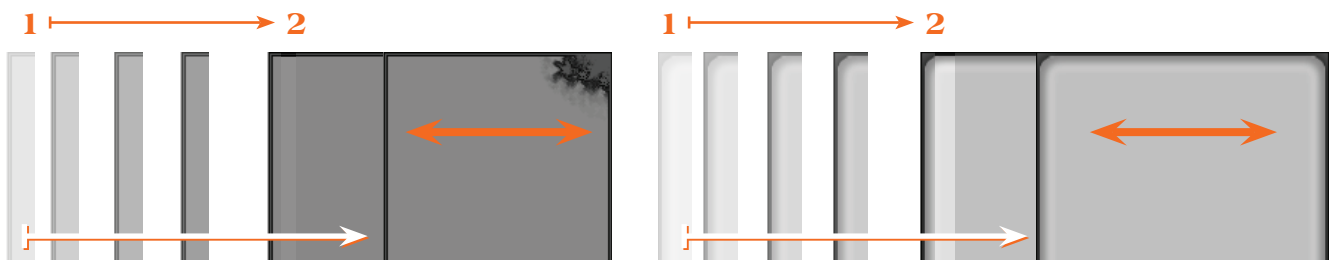
- Make sure to save a copy of this original before the next steps - this is a vital back up to make as we will now crop this image into 2 parts. If you make a mistake at this point, the graphics might not align perfectly, giving away the 2 graphics used and looking quite ugly.  
With a back up copy you can repeat this step and make sure to achieve the desired effect.

## tab graphic - split image

- leaving the image's height (150px) intact - crop the graphic into 2 parts:



By creating this set up graphic #1 will become the top most graphic, overlapping and essentially covering up graphic #2, the larger rectangle. It can therefore be positioned anywhere on the horizontal axis, creating any button from a very narrow tab to a longer tab, no wider than 300px.

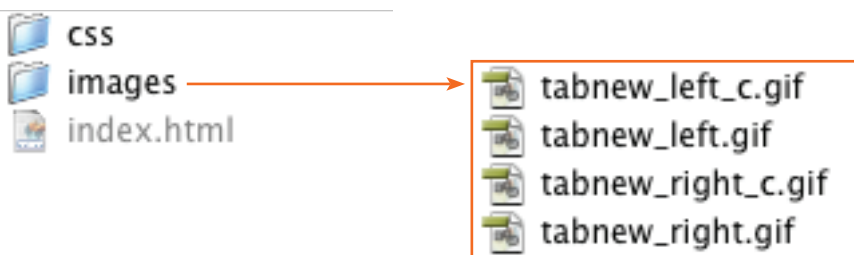


- to prepare both images for the web - use your usual method to optimise the graphics, naming both files appropriately for better workflow.

create an additional copy of both files - to become the tab appearance of any currently selected tab.

In this example - the main tab colour has been set to the overall site's colour (#666) to blend seamlessly into the current page.

- set up the root folder for your site and save the web-ready files into the relevant folder - and we're ready to set up our tabbed menu ;)





## menu set up

- create the index page in your root folder and define your document as always.

```
XHTML ><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<link rel="stylesheet" href="css/tab.css" type="text/css"
media="screen, projection" />
<meta name="description" content="tabbed navigation set up" />
<meta name="keywords" content="css, tabs, menu, breadcrumb" />
<meta name="author" content="graphicEyedea, www.graphiceyedea.co.uk."
/>
<title>the sliding doors of CSS</title>
</head>

<body>

</body>
</html>
```

By now you will be more familiar with CSS and will hopefully have no difficulties in setting up your initial style sheet rules for universal settings and any re-defined tags - I will merely focus on the tabbed menu itself (you can access the full css file on the sample online, a link is provided on each page).

- in your style sheet, add the following code to define the appearance of the listed buttons through a new ID:

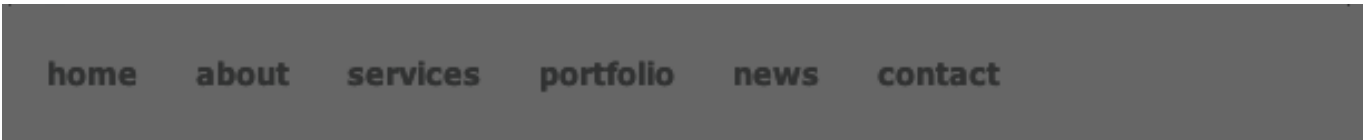
```
CSS > /* tabbed menu set up */
#nav {
    float: left;
    width: 100%;
    font-size: 0.9em;
}
#nav ul {
    margin: 0;
    padding: 10px 10px 0 10px;
    list-style: none;
}
#nav li {
    float: left;
    margin: 0;
    padding: 0;
}
```

By setting the list style to none and floating the overall navigation as well as each list item individually - each button will appear next to the previous in an horizontal line (similar to setting the display to inline).

- add your menu items as list to the XHTML page:

```
XHTML > <div id="nav">
  <ul>
    <li><a href="#">home</a></li>
    <li><a href="#">services</a></li>
    <li><a href="#">portfolio</a></li>
    <li><a href="#">news</a></li>
    <li><a href="#">contact</a></li>
  </ul>
</div>
```

your menu should now display something like this:



- to add the graphics for the tab - return to your style sheet and make the following additions (highlighted in bold orange);

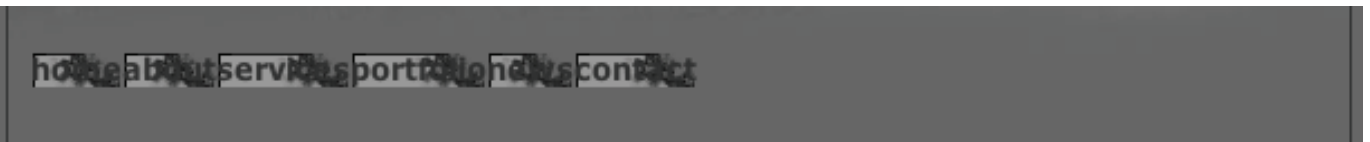
```
CSS > #nav li {
  background: url(../images/tabnew_right.gif) no-repeat right top;
  float: left;
  margin: 0;
  padding: 0;
}
#nav a {
  float: left;
  display: block;
  background: url(../images/tabnew_left.gif) no-repeat left top;
}
```

↓  
image at the back - aligned into the top **right** corner

↑  
image at the front - aligned into the top **left** corner

Notice that the order of the images is quite deliberate - giving the list item the larger graphic (#2), adding the second graphic (#1) to the link itself, ie it will therefore appear on top, covering the larger scale graphic up.

Not quite finished yet - as the tab size at the moment crowds both the text and the graphic together into a confusing lump - as shown here:



- with all elements in place - we will need to adjust padding inside the HREF:

```
CSS > #nav a {
  float: left;
  display: block;
  background: url(../images/tabnew_left.gif) no-repeat left top;
  padding: 12px 20px 2px 6px;
}
```

depending on your own graphic - you will have to adjust these settings to fit your own design.



## breadcrumb set up

In order to clearly indicate the selected tab - we will define an extra ID for the specific list item. This will be exactly the same as any of the others - only changing the tab image displayed.

- in your style sheet - add the following code:

```
CSS > /* current tab selected */
#nav #current {
    background: url(../images/tabnew_right_c.gif) no-repeat right top;
}

#nav #current a {
    background: url(../images/tabnew_left_c.gif) no-repeat left top;
    color: #fff;
    padding: 12px 20px 2px 6px;
}
```

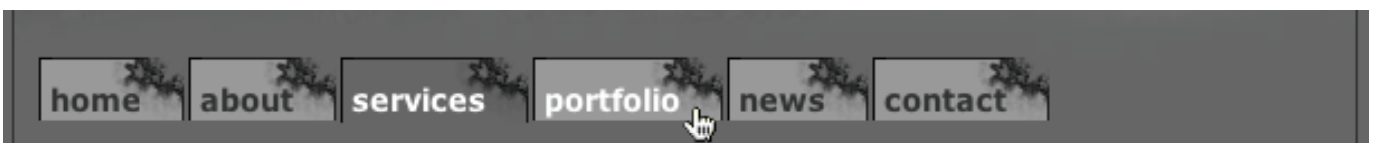
- next, edit your page to display the relevant tab as shown here

```
XHTML > <li id="current"><a href="index.html" target="_self" onfocus = "this.
blur()" id="current">home </a></li>
```

your menu should now look something like this:



on any following pages - this detail will need to be set manually and added to the relevant button title to remain consistent throughout the site.





## final touch - the bottom line

To complete the look - we will now add a 1 pixel border to the bottom of the tabs - allowing the current tab to appear to be at the front of all others.

Trying to set this up with different coloured borders at the bottom of either suitable item would end in frustration as the alignment would be almost impossible, an offset of about 1 pixel unavoidable.

We will therefore set up a small graphic, containing a 1 pixel border at the very bottom - which will then be used as background graphic - the tabs being shifted by the same amount to create the illusion.

- in your style sheet - add the following code to the main #nav ID:

```
CSS > #nav {  
    background: url(../images/bg.gif) repeat-x left bottom;  
    padding: 0;  
    float: left;  
    width: 100%;  
    font-size: 0.9em;  
}
```

- to bring the currently selected tab to the front, ie to overlap this new background image - edit the padding to allow the 1 pixel offset

take away 1 pixel from your previously set padding at the bottom:

```
CSS > #nav a {  
    float: left;  
    display: block;  
    background: url(../images/tabnew_left.gif) no-repeat left top;  
    padding: 12px 20px 1px 6px;  
    border: none;  
}
```

CSS > leaving 1 extra pixel for the currently selected tab:

```
#nav #current a {  
    background: url(../images/tabnew_left_c.gif) no-repeat left top;  
    color: #fff;  
    padding: 12px 20px 2px 6px;  
}
```

.... and you should be done ;)

*have fun ;)*

home

about

services

portfolio

news

contact