

Making the Menu

1. What is a menu?

Menu on a web-page means a list of links which will open new content on your site. By making them as a menu you will separate them ordinary links which will lead outside from your site. Menu is often styled to look like a bunch of buttons on the left side of the main page or as it is more common today, as a drop-down menu at the top part of the page. We will look at the drop-down menus later, but today we will build a simple button-style menu.

2. Menu in the html

First write the html needed for the menu into your layout's left div (in the code after `<div id="left">`). That is a simple list:

```
<ul id="menu">
```

```
<li>Home</li>
```

```
<li>Accommodation</li>
```

```
<li>Food</li>
```

```
<li>Gallery</li>
```

```
</ul>
```

Save the page and look at it in the browser. Now you have the list, next we will add the links for each list item (``):

```
<ul>
```

```
<li><a href="home.html">Home</a></li>
```

```
<li><a href="accommodation.html">Accommodation</a></li>
```

```
<li><a href=food.html">Food</a></li>
```

```
<li><a href="gallery.html">Gallery</a></li>
```

```
</ul>
```

Be careful, don't forget to finish the links with ``! The links are targeted to pages 'accommodation.html', 'food.html' and 'gallery.html' which we don't have yet, but they will be made later. Save and look at the result in a browser. You should now see a list of underlined links.

2. Styling with CSS

Now when the menu list with the links is ready, we can concentrate on how it looks; that means of course the CSS-styles. In the html we gave for the list id 'menu' and so we can target the styling to it without affecting any other lists, if there happened to be them on the same page. `Ul id="menu"` in the html is `ul#menu`, so add the following code in the CSS-styles of your page:

```
ul#menu
```

```
{
```

```
list-style-type:none;
```

```
}
```

Save and see, how it works. List style type means the signs before the list items, other values could be for example squares or some icons, but now we need the value none.

Next we take of the underlining from the links. On a web page, underlining is a sign of a link, but because we want to make them to look like buttons that wouldn't be neat, so we take them off. See that the code is now targeted to the links inside the ul#menu, because it is the link properties we want to change this time, not the list. That is why we add the 'a' after the ul#menu: a means link (from anchor) in the html. So add to CSS:

```
ul#menu a
```

```
{
```

```
text-decoration:none;
```

```
}
```

Next we start making the list look like buttons. Add the following code for the links in ul#menu a:

```
border:5px groove #333;
```

Save and check the result in the browser. You should have now have a 5 pixel groove-styled border with color #333 (quite dark) around the links. Then add to the same place:

```
font:18px Arial;
```

```
color:#033;
```

background-color:#ddd;

padding: 3px 8px;

You should now in the menu have an 18 pixels Arial font, which is colored with #033 and on a #ddd colored background and padding (=text distance from the borders) should be three pixels on the top and bottom and eight pixels on the left and right . Check the result in the browser.

Next thing is to make the links look equally wide. We can add for them in the CSS:

width:80%;

But that won't make any effect to their width. That is because links are not so called block level elements; i.e. they do not have any space, they are in a space. We can change this by adding them

display:block;

That makes them bloc-level elements and now you can set the width for them.

The next thing is to make the links to react to the mouse arrow. That is possible, because links have properties a:hover, a:active and a:visited to define how the link will change its outlook when the mouse is over it (hover), when it is clicked (active) and after that (visited). Define the hover in the CSS:

ul#menu a:hover

{

color:#ddd;

background-color:#033;

}

Test the result with the browser. Then add some 3D by making the border and the padding to change when the link is hovered (so in ul#menu a:hover):

padding: 0px 5px;

border:8px groove #333;

Finally, add a definition for an active link:

```
ul#menu a:active
```

```
{
```

```
color:red;
```

```
}
```

If you wanted to move the whole menu's location you could give it some margins (margin-top, margin-left).

Loading new content into the page: Iframe

Now you have the menu, but if you click the links in it, you will get an error message telling that no such page exists. So your next thing is to create those pages. Open a blank page in the Notepad++ and write the basic html code:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>home</title>
```

```
</head>
```

```
<body>
```

```
<h1>Tampere, My City</h1>
```

```
</body>
```

```
</html>
```

Save that page as 'home.html' exactly in the same folder (directory) where you have your layout.html -page. Then make three other pages with names 'accommodation.html', 'food.html' and 'gallery.html'. Tip: you can clone the first page, just change the title, headline and save it as different names.

When you are ready, test, if you can open the pages from the links. If not, check your links and page names and that they really are in the same folder (directory) as your layout.html -page.

When you've got your links working, open layout.html in Notepad++ and add to html in the div middle (so after <div id="middle"> the following code:

```
<iframe name="frame" src="main.html"> </iframe>
```

That element is called iframe and it makes a frame where you can load other pages. That makes the page structure stay the same when the content is changing. Otherwise you should write the menu differently for each page or use some programming. Look in the browser, what the frame looks like. You should have a small window in the middle div and the main.html in it. Attribute src in the iframe tells which page is loaded in to it by default.

The aim is to load the linked pages into the iframe. Now if you click the links in the menu, they will not do that. To make that happen you have to add for the links in the menu a new attribute, target. So find the menu in the html and add for each link '**target="frame"**', like:

```
<li><a href="accommodation.html" target="frame">Accommodation</a></li>
```

Add that for every link! Now try clicking the menu. Your pages should now open in the frame. The links will know the iframe from the name 'frame' that we gave to it.

Finally, make the iframe window cover the whole middle div. Add definitions for the iframe in the CSS of the page (layout.html):

iframe

{

width:100%;

height:100%;

border-width:0;

}