What are we having for lunch?

No, not that kind of menu! Menus here are about finding our way around our new multi-page site, our way of **navigating** the site. Just like the menu at a restaurant giving you choices of what to eat, so the menu on a Web site gives the reader choices of which pages to go to.

More than one way to cook an egg …

There are a few ways to tackle menus – you can use a simple HTML menu bar, running across the page (like the menu bars you see just below the headers on Resources4Learning.org), with some spaces and separator characters. Let's take a look at that method first:

Funky little characters

The first thing we need to learn here is how to generate some of those characters that we don't find on the keyboard – and it's useful to tell HTML when we want it to put more than one space between characters. Let's take a look at how we can get HTML to **identify** the special characters we want to use; one way is that some of the special characters have what we call **entity codes**:

`&nbsp;`	(non-breaking space)
`&loz;`	(lozenge shaped divider)

The menu bar

I'm going to start a new example Web site (in a new folder, because, like you, I'm trying to get tidier about the places I put my work on my computer's hard drive). I'm also going to create a fairly simple css for my site, so all the pages have some uniform qualities, without having to write tons of code into the header of each page.

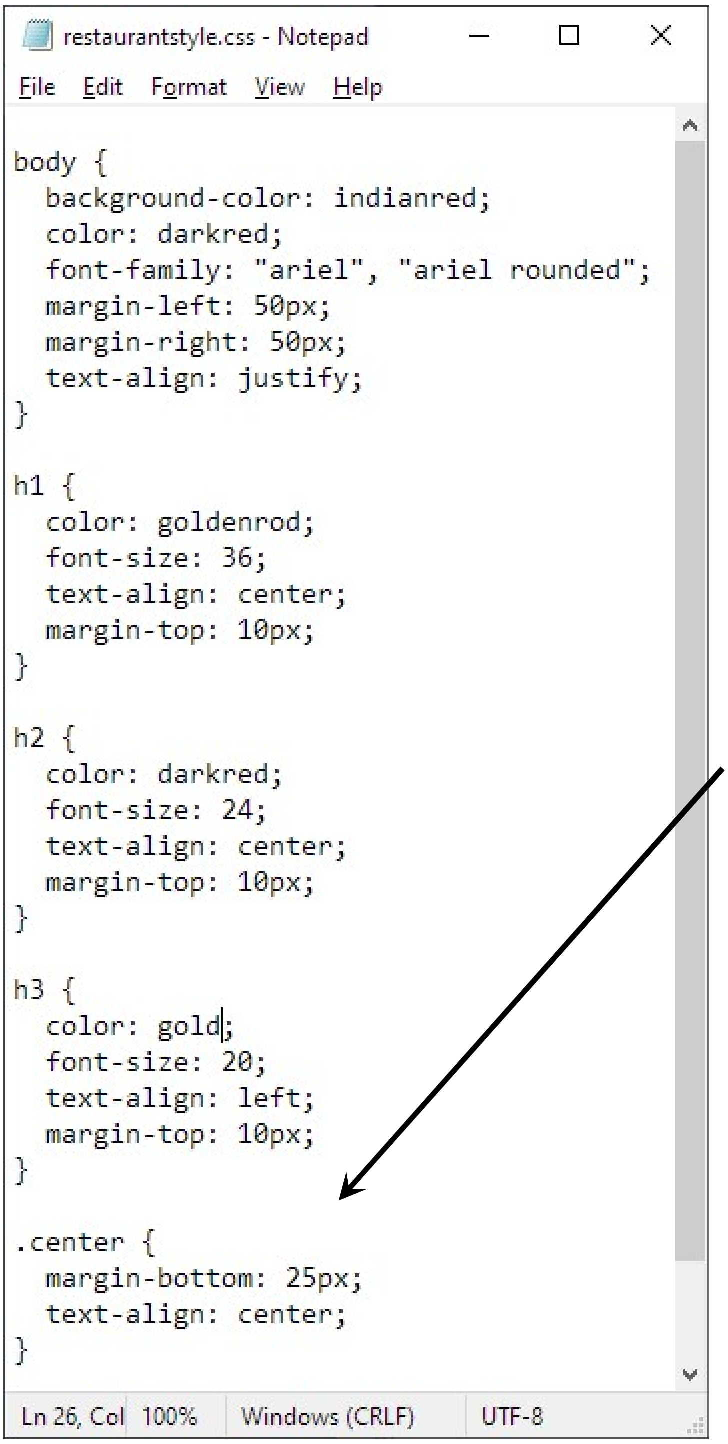
This is going to be a site for a restaurant, with a home page, and separate pages for food and drinks.

My page names are going to match the purpose of each page, so I'll have `home.htm`, `food.htm` and `drink.htm` as my page names. I think I'll call my style sheet `restaurantstyle.css` to be nice and clear. OK, I'll also save the whole lot into a new folder.

So, my menu bar is going to need links to all three pages (yes, I know that means it can link back to the page you're already on, but making one universal menu bar means you can copy and paste it onto each page. That's not so much "cheating" as saving yourself a bit of time typing.

The parts of my site:

Here's the cascading style sheet:



restaurantstyle.css - Notepad
File Edit Format View Help
Ln 26, Col 100% Windows (CRLF) UTF-8

```
body {
  background-color: indianred;
  color: darkred;
  font-family: "ariel", "ariel rounded";
  margin-left: 50px;
  margin-right: 50px;
  text-align: justify;
}

h1 {
  color: goldenrod;
  font-size: 36;
  text-align: center;
  margin-top: 10px;
}

h2 {
  color: darkred;
  font-size: 24;
  text-align: center;
  margin-top: 10px;
}

h3 {
  color: gold;
  font-size: 20;
  text-align: left;
  margin-top: 10px;
}

.center {
  margin-bottom: 25px;
  text-align: center;
}
```

Here is my cascading style sheet for the Ernie's Eatery restaurant:

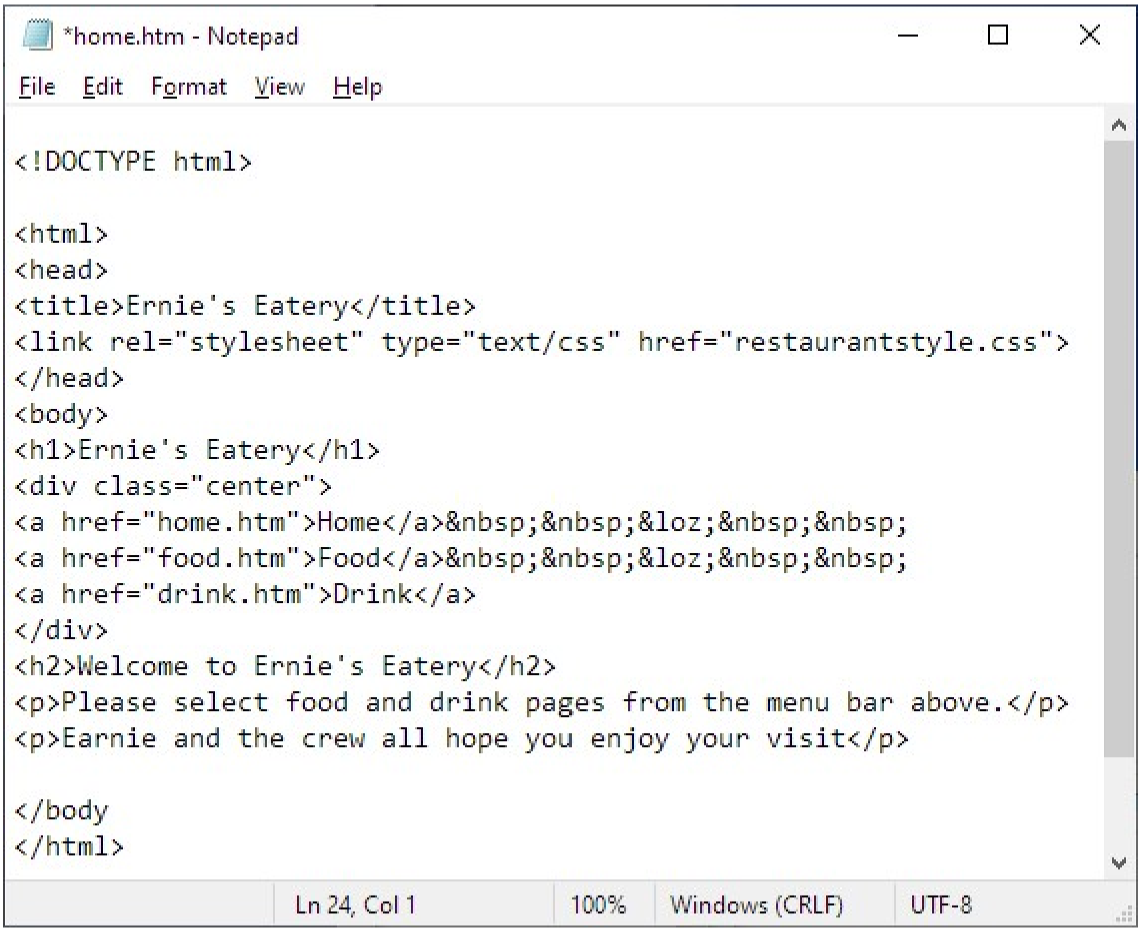
My style sheet starts off declaring some attributes I want to give the `<body>` part of every page of my site – these include the background and text colours, the text font, and some margins, just like earlier examples.

I go on to describe how I want some of the header styles to look, so I don't have to type this stuff in every page…

And then you'll see that `.center` element. Starting a declaration with a dot means I am telling the style sheet (and later the browser) how I want it to behave when I use the word `center` as a style within my HTML. I could just as easily call it `.middle` if I wanted to. You'll see how this works in a minute.

This is **declaring** (or describing a **class** that we want the Web site to use. As I said, you'll see how this works, and how we can use it, a few pages further on.

Here’s how the homepage of my site will start:



```
<!DOCTYPE html>

<html>
<head>
<title>Ernie's Eatery</title>
<link rel="stylesheet" type="text/css" href="restaurantstyle.css">
</head>
<body>
<h1>Ernie's Eatery</h1>
<div class="center">
<a href="home.htm">Home</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="food.htm">Food</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="drink.htm">Drink</a>
</div>
<h2>Welcome to Ernie's Eatery</h2>
<p>Please select food and drink pages from the menu bar above.</p>
<p>Earnie and the crew all hope you enjoy your visit</p>

</body
</html>
```

The `<head>` section should be familiar – it contains the page title and names the style sheet file we want our browser to use.

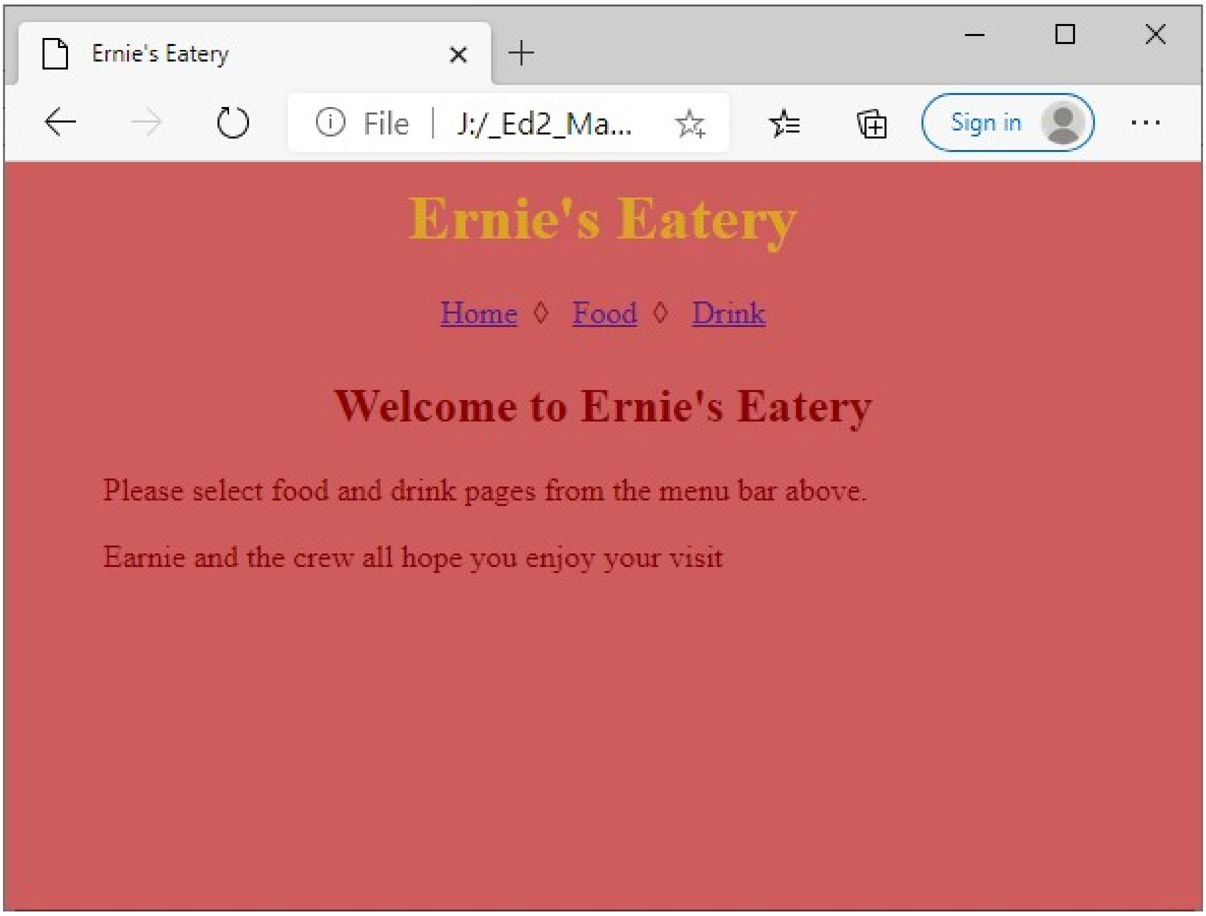
You’ll see a new pair of tags, `<div>` and `</div>` - these tell the browser where we want to put a new division in our page. Divisions work a bit like a special sort of paragraph.

Inside that `<div>` you’ll see three anchor points with links to the different pages of our site, and some `&nbsp;` and `&loz;` characters between them. This is our menu bar.

You’ll also see I have used two of the three header styles I declared in the style sheet.

Before I show you how the other two pages work, let’s just have a look at our home page, and see what that menu looks like:

Here's how that homepage looks so far:



Ernie's Eatery

Home ◊ Food ◊ Drink

Welcome to Ernie's Eatery

Please select food and drink pages from the menu bar above.

Earnie and the crew all hope you enjoy your visit

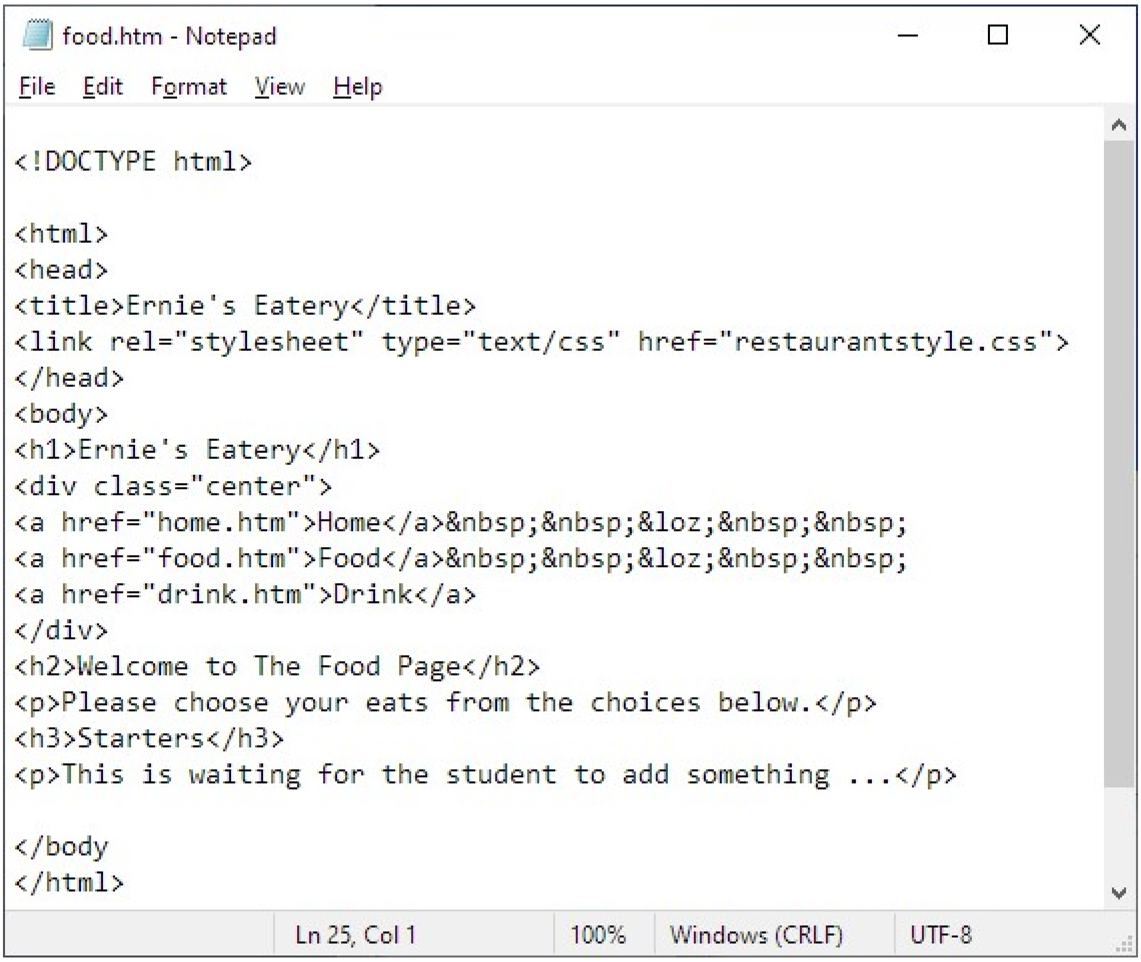
Of course, you might not like my colour scheme, or you might want to give your restaurant a different name – your call!

Now, there's a handy thing in Windows called copy and paste (if you use a Mac, I think you'll find you can do the same thing!).

When you look at the code for the `food.htm` page of our site (next page on this handout), you'll see all of the code down to the `</div>` at the end of the menu bar is the same.

So, on to our food page:

Here’s the code for our food page:



```
<!DOCTYPE html>

<html>
<head>
<title>Ernie's Eatery</title>
<link rel="stylesheet" type="text/css" href="restaurantstyle.css">
</head>
<body>
<h1>Ernie's Eatery</h1>
<div class="center">
<a href="home.htm">Home</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="food.htm">Food</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="drink.htm">Drink</a>
</div>
<h2>Welcome to The Food Page</h2>
<p>Please choose your eats from the choices below.</p>
<h3>Starters</h3>
<p>This is waiting for the student to add something ...</p>

</body
</html>
```

Like I said, the top part is the same (although you could change the `<title>` name if you want to).

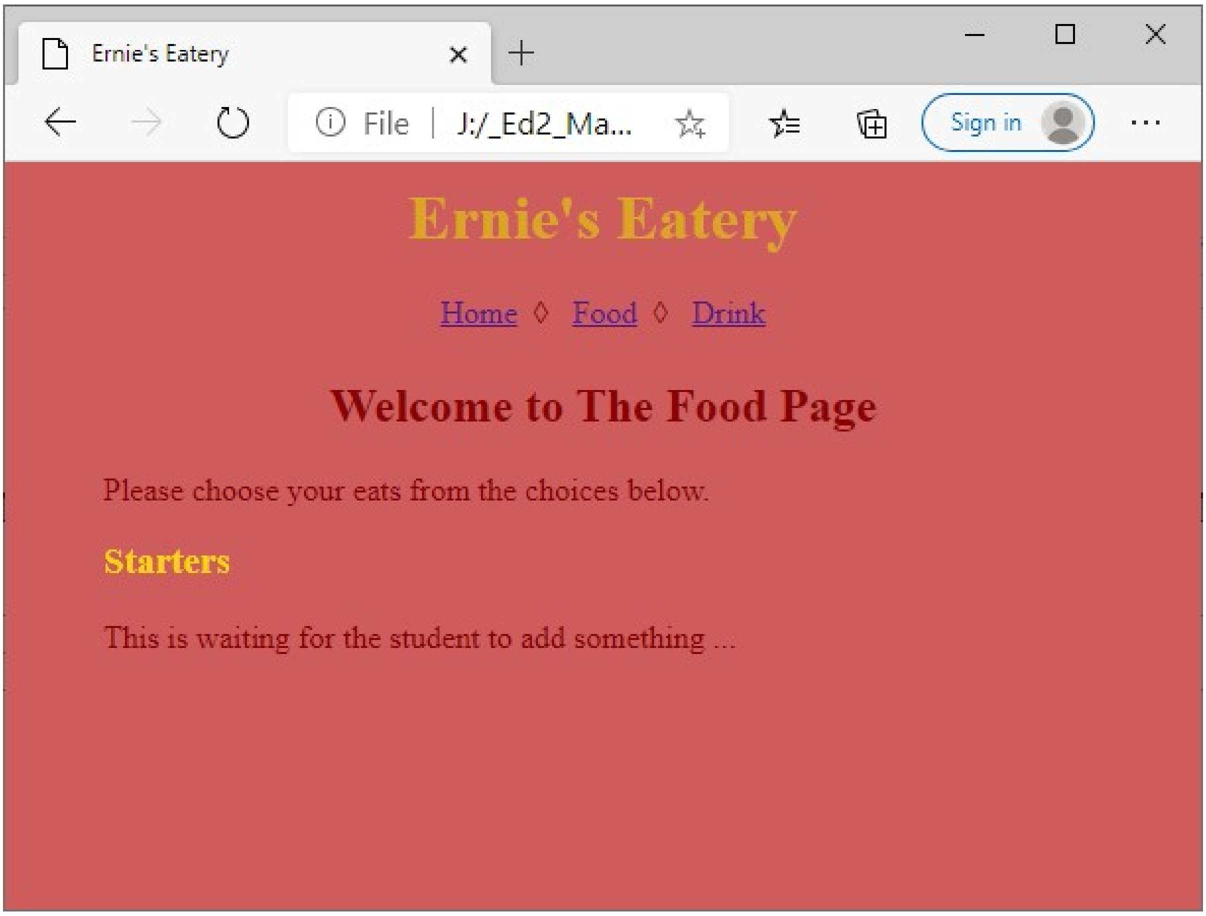
You’ll see the `<div>` for the menu is the same.

By the way, take another look at that `<div>` statement. The `class=”center”` tells it to use the `center` class that we declared at the bottom of the style sheet.

[I said you’d see how it’s used …]

You’ll also I’ve now used the `<h3>` style we declared in our style sheet.

… and here's how our Food page looks so far:



Ernie's Eatery

Home ◊ Food ◊ Drink

Welcome to The Food Page

Please choose your eats from the choices below.

Starters

This is waiting for the student to add something ...

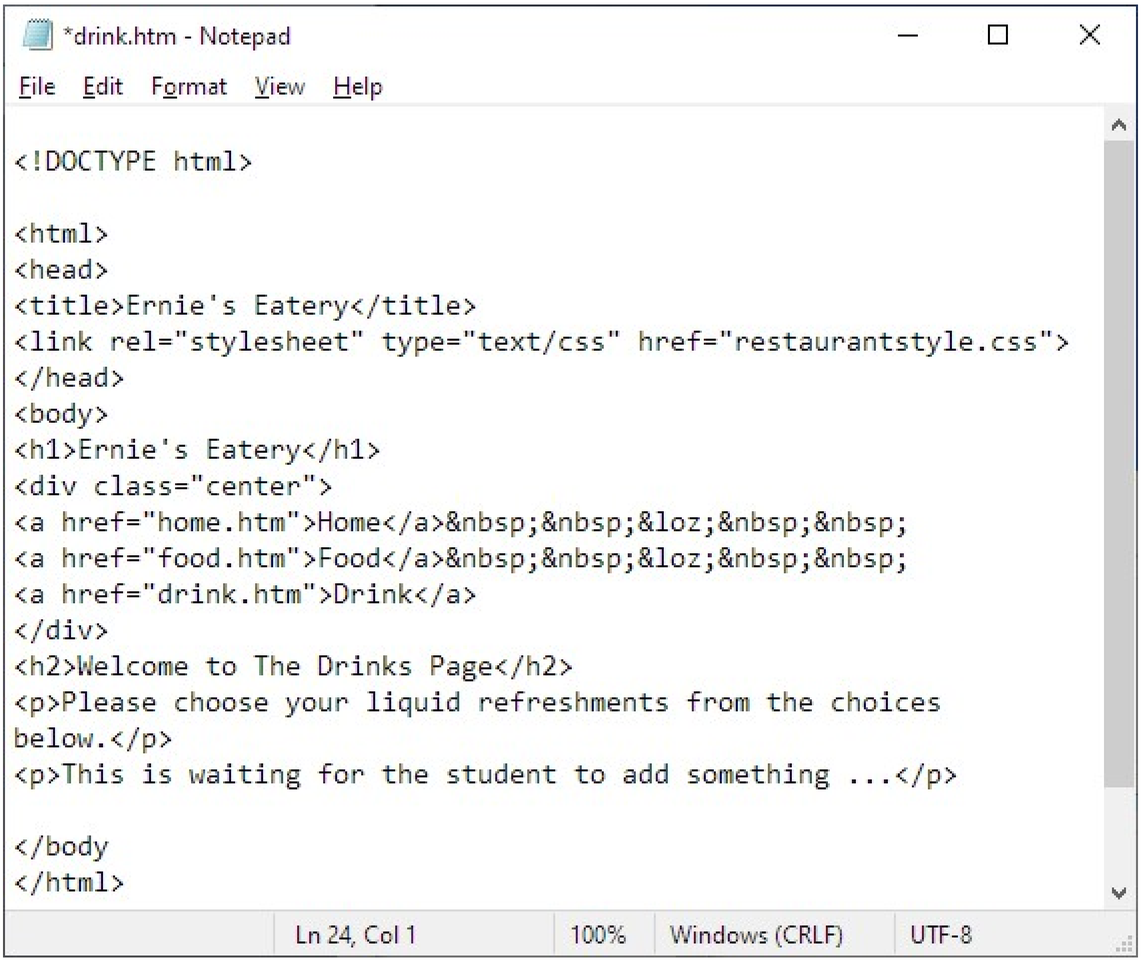
See how all three of the header styles, from `<h1>` through `<h2>` down to `<h3>` all behave as we defined them in the style sheet.

The top two are centre-aligned,; while `<h3>` is aligned to the left.

Our top header, title and menu bar are all the same as before, which keeps our page nicely uniform.

You've probably got the idea of how the code for the Drink page might look – but I'll put it onto the next page here, along with a picture, just so you can see the whole thing.

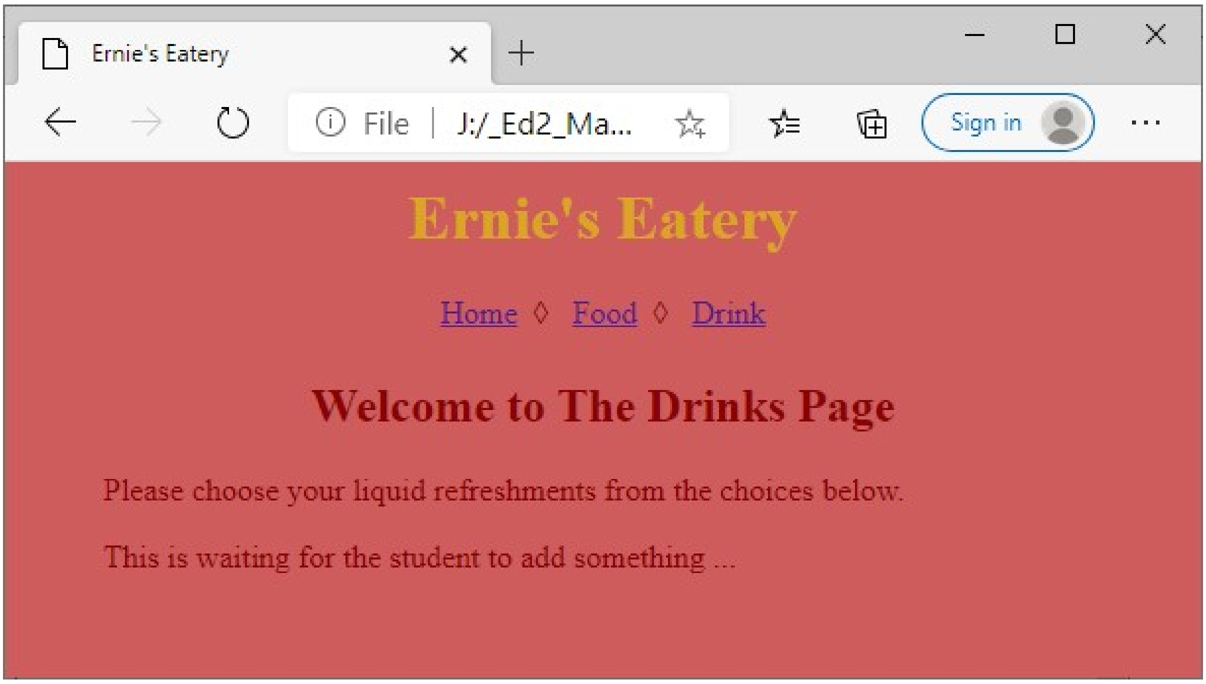
Here's that Drink page:



```
<!DOCTYPE html>

<html>
<head>
<title>Ernie's Eatery</title>
<link rel="stylesheet" type="text/css" href="restaurantstyle.css">
</head>
<body>
<h1>Ernie's Eatery</h1>
<div class="center">
<a href="home.htm">Home</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="food.htm">Food</a>&nbsp;&nbsp;&loz;&nbsp;&nbsp;
<a href="drink.htm">Drink</a>
</div>
<h2>Welcome to The Drinks Page</h2>
<p>Please choose your liquid refreshments from the choices
below.</p>
<p>This is waiting for the student to add something ...</p>

</body
</html>
```



Ernie's Eatery

Welcome to The Drinks Page

Please choose your liquid refreshments from the choices below.

This is waiting for the student to add something ...

Over to you

Have a go at adding some new pages to a site you're already working on, or start a new site with more than one page, and give it a menu bar.

You'll find it suddenly adds a lot to the things you can do with your sites, and makes them a lot more interesting for the reader. People visiting Web sites generally like them to be interesting!

.

Next session

In the next session, we're going to look at something just a little bit funky: drop-down menus. Yet another really cool, "Pro" feature to find a space for in your HTML toolbox.

Have a play with menu bar for now though – there are more new things you've covered today than you realise – and they're all worth exploring, practising and playing with so you remember them and how to use them.

Stay safe, happy coding – and see you next time.