Drop-down designs

Hope you enjoyed the menu bar approach to menus last time. Now you know most of how the menus on the Resources 4 Learning .org Website work. You're getting some serious Web design tools here that I use in my sites.

This session gets just a little bit more "flashy" – we're going to explore those "drop-down" menus that you see on some sites.

Take a look at the Website of one of Coventry's radio stations, Hillz FM:

https://www.hillzfm.co.uk

You'll see it's got a horizontal menu bar, just like the one we made last time. (In fact, those horizontal menu bars are so cool, even the BBC uses them!) – but now take you mouse, and **hover** (but don't click) on the menu link "support us". You should see a second style of menu, a "drop down" menu, appear.

Today's session looks at how to write those drop-down menus.

It's a good job HTML is software and not hardware – or you'd need a new, bigger toolbox every week at the moment, with the stuff we're adding!

Much more in the style sheet

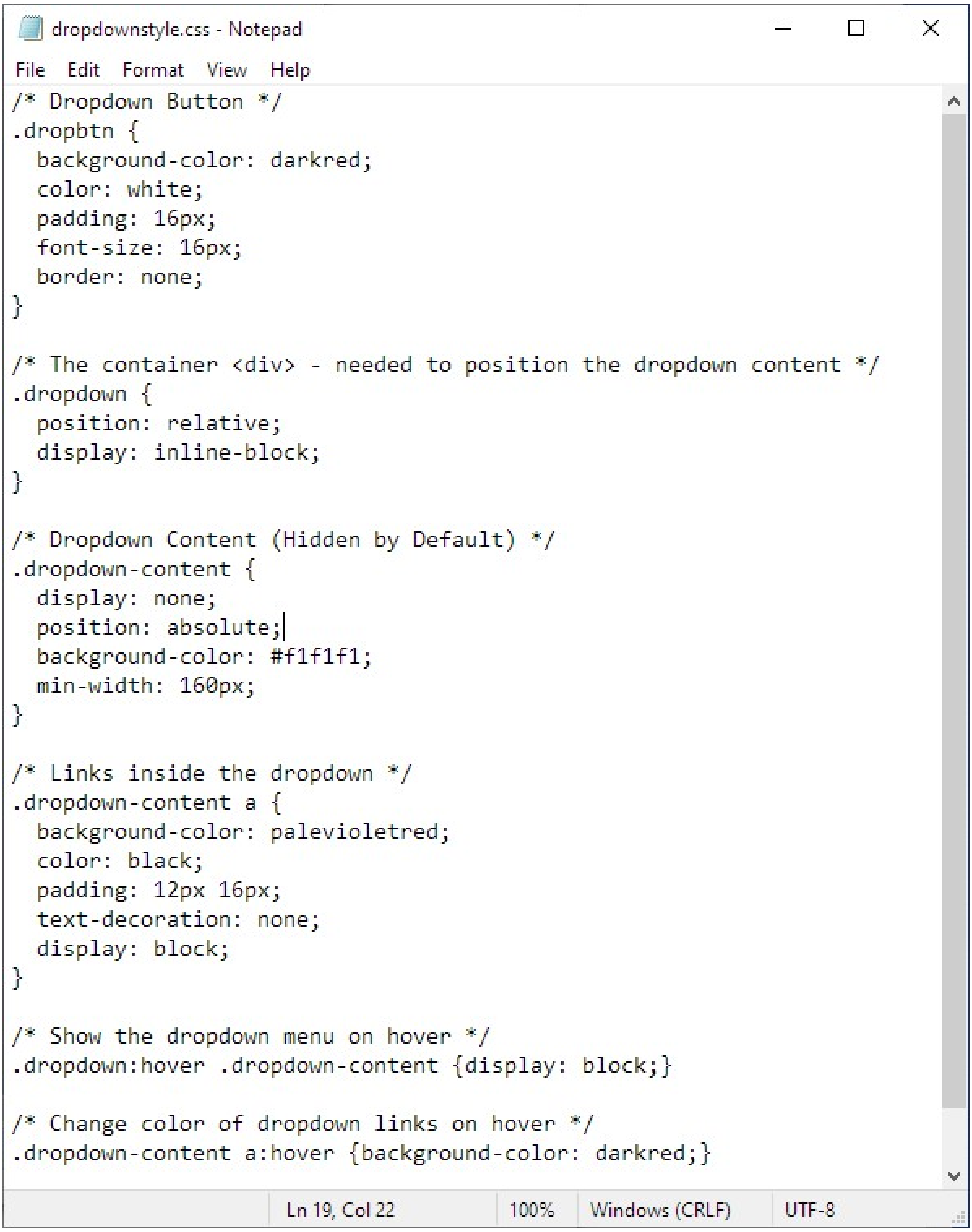
OK – deep breath – this is a bit of a toughie. We're going to look at some rather more complicated stuff in that style sheet to make this this work. I'll show you everything we need to make this thing work, and then let you add that to your existing style sheet, to use a drop-down menu (or two!) in your own site.

Don't be surprised if this is one of the sheets you need to read through more than once before everything clicks.

Before we start, let me also explain we can put **comment** or **remark** lines into style sheets. These are lines of code which the computer and Web browser ignore, but which help us to remember things, or to explain things to other humans reading our code.

Any line that begins `/*` and ends `*/` in a style sheet is not code – the sheet will work perfectly without those lines, but, with those lines in, a human can remember what that bit does or, in this case, I can use it to explain things to you.

That style sheet:



```
/* Dropdown Button */
.dropbtn {
  background-color: darkred;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
}

/* Links inside the dropdown */
.dropdown-content a {
  background-color: palevioletred;
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {display: block;}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: darkred;}
```

Ouch! OK – Let me take it apart, a chunk at a time, and explain what's going on – it's not as bad as it first looks.

Pulling the style sheet apart

The dropdown button element

```
/* Dropdown Button */
.dropbtn {
  background-color: darkred;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
}
```

Here I've defined an element named `dropbtn` to be our dropdown button. Most of the attributes should be familiar to you. The `padding` attribute tells the browser how many pixels of space to leave around the text inside the button – in other words, how much border between the text and the edge of the button.

The container element

```
/* The container <div> - needed to position the dropdown
content */
.dropdown {
  position: relative;
  display: inline-block;
}
```

This is just telling the browser to put the dropdown button inline with anything else we might want to put in the same line of the page – for example, to make sure in goes next to any other menu buttons we might want to create (yes, you can get really complex with a bar of dropdown menus, on a very big site).

The dropdown-content element

```
/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
}
```

This is a kind of container, which we will put the contents of the dropdown menu into. Normally it is hidden, until the reader hovers the mouse over the menu button.

The links inside the dropdown

```
/* Links inside the dropdown */
.dropdown-content a {
  background-color: palevioletred;
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}
```

This defines the links or anchor points `<a href=`… within our dropdown menu. So, `dropdown-content a` is referring to the `a` in our anchor point references to the links in our menu.

The `padding` attribute again defines the padding vvertically and horizontally around the text in each menu option.

… and the last two items …

```
/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {display: block;}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: darkred;}
```

The first of these is telling the browser to display the menu when the mouse pointer is hovered over the menu button.

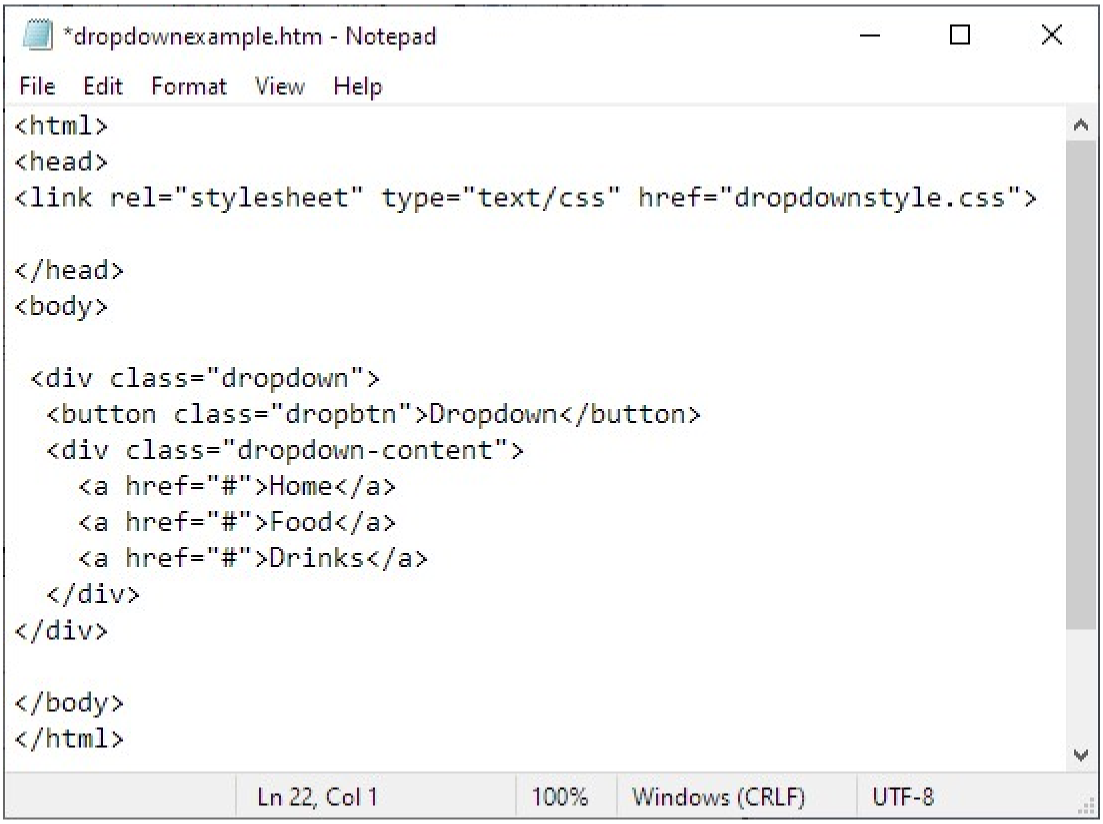
The second tells the browser to change the colour of a menu otion when that is hovered over.

Putting it all together

OK – that's quite a lot to take in – it'll become clearer once we see the HTML (actually quite a small and tidy piece of code) and see how the pieces slot-in together.

It may help to re-read this section after you've seen how the HTML works. It may also be helpful to take a short break before re-reading.

Here's the HTML code it needs:



```
<html>
<head>
<link rel="stylesheet" type="text/css" href="dropdownstyle.css">

</head>
<body>

 <div class="dropdown">
  <button class="dropbtn">Dropdown</button>
  <div class="dropdown-content">
    <a href="#">Home</a>
    <a href="#">Food</a>
    <a href="#">Drinks</a>
  </div>
</div>

</body>
</html>
```

One of the first things you'll see is that a `<div>` can be placed inside another `<div>`. Putting tags inside one another is a technique we call **nesting**. When you program in other languages, you'll find other examples of nesting.

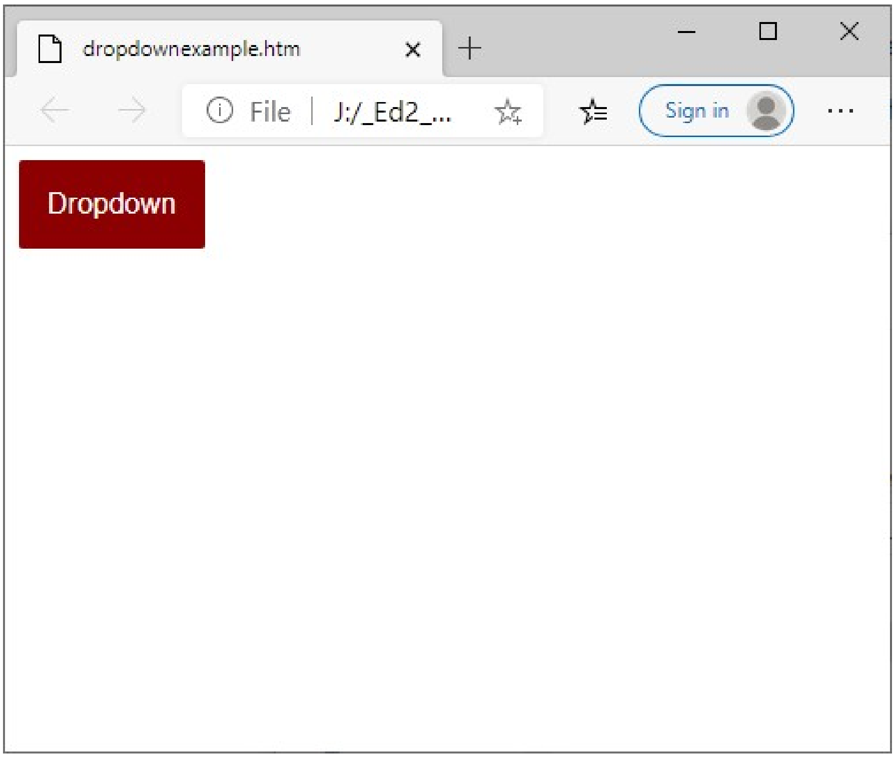
You'll see, too, that the **indent** (or left-hand margin) of the code gets slightly deeper with each **level of nesting**. Writing the code in that style makes it a lot easier for humans to understand. In HTML it actually makes no difference to the way the computer sees it. (But wait until you start working with Python!)

When you use this menu design in your HTML code, you'll have to put the relevant page file names or URLs in the `<a href="#">` parts, in place of the `#` character.

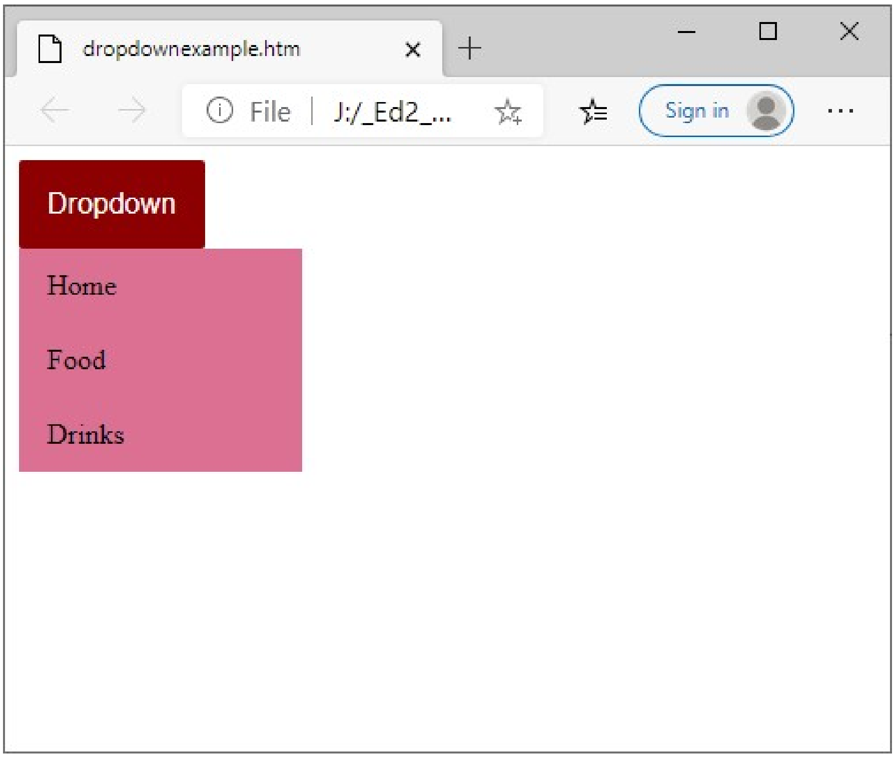
The "outer" division uses our container element to hold the whole menu:
`<div class="dropdown">`

The division inside that holds the links to the places we want our menu to link to:
`<div class="dropdown-content">`

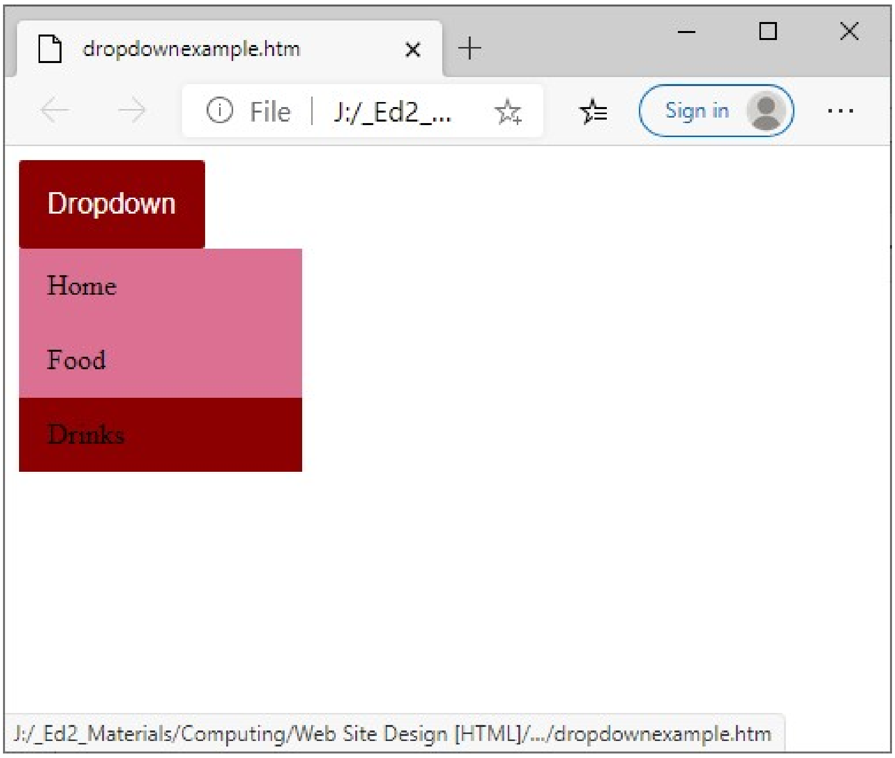
Here's what our drop-down menu looks like …



.. and here it is when we hover the mouse over it:



… and here when we hover over one of the menu options


dropdownexample.htm
File | J:/_Ed2_...
Sign in
Dropdown
Home
Food
Drinks
J:/_Ed2_Materials/Computing/Web Site Design [HTML]/.../dropdownexample.htm

Cool? Well, we quite like it.

Over to you

See if you can get something similar into a Web site of yours. Remember you'll need something like the stuff in this style sheet, **as well as** the stuff telling the browser how you want your `<body>` `<h1>`, `<h2>` and so 0n to look.

Credit

The model used in this lesson is based on an example from the w3school – you'll see I've added a few colour bits, and ripped-out some of the slightly more fancy parts, to make it a bit easier to digest first time around.

https://www.w3schools.com/howto/howto_css_dropdown.asp

Tip:

If you're ever stuck for how to do something, that site's a very good starting place to try to find the answer.