



29 January, 2011

# Create a centred horizontal navigation

Centring block level elements is easy, just define a width and set `margin:0 auto;`, but what if you don't know that fixed width? You could use `text-align:center;` but that won't work on 100%-width block-level elements either... that'll only work on text-level elements.

Defining explicit widths and heights should always be avoided wherever possible, as doing so will make the document a lot less future-proof, flexible and extensible... Suppose you have four items in your navigation menu—you can work out the width of these and use `margin:0 auto;` to centre them. Adding a fifth will increase the width, meaning you'd need to alter the CSS, too. This is far from ideal, and more so with a CMS to power the site (a client can add pages, but perhaps can't edit CSS).

However, there is a way to have a centred horizontal navigation without knowing an explicit width, and without adding CSS. It's also remarkably easy.

The markup:

```
<ul class="nav">
  <li><a href="/">Home</a></li>
  <li><a href="/about/">About</a></li>
  <li><a href="/work/">Work</a></li>
  <li><a href="/clients/">Clients</a></li>
  <li><a href="/contact/">Contact</a></li>
</ul>
```

Pretty standard, an unordered list of menu items. The CSS is where it's at. I have highlighted the bits that do the majority of the work:

```
.nav{
  border:1px solid #ccc;
  border-width:1px 0;
  list-style:none;
  margin:0;
  padding:0;
  text-align:center;
}
.nav li{
  display:inline;
}
.nav a{
  display:inline-block;
  padding:10px;
}
```

What I've done here is simply create a navigation list and given it a border top and bottom (purely to highlight its centred text). Instead of floating the *block-level* `<li>`s left I've given them **display:inline;**, that is to say they no longer occupy 100% the available width and they now stack up nicely against each other.

Next we use (the much underused) **display:inline-block;** to make sure the links themselves don't break onto new lines but still obey any padding values accordingly. Here I have given them a larger hit area by adding **padding:10px;**

You could have, if you wanted, applied inline-block to the `<li>`s. however IE6-7 will only allow **inline-block** to work on elements that are inherently inline elements. **display:inline-block;** will not work on block-level elements.

## Demo

**Here's a quick demo.** Try using Firebug or similar to add other list items on the fly, and watch as they seamlessly centre in the list. I have tested this in IE7-8 to find it works perfectly. I haven't checked IE6 but I imagine it'll be fine.

## Update

You asked and I heard; I have made a **CSS powered dropdown version of this** for you. The line **top:100%;** will make the dropdown work in IE7, but kinda ruins the experience a little in all other browsers. Whether you leave it in or not is up to you. Again, view source for the how-to...

Did you enjoy this? Hire me!

Hi there, I'm Harry. I am an **award-winning Consultant Front-end Architect, designer, developer, writer and speaker** from the UK. I **write, tweet, speak** and **share code** about authoring and scaling CSS for big websites. You can **hire me**.

Follow @csswizardry 45.9K followers

I am currently **accepting new projects** for Q2-3, 2016

Referral scheme



Avada is the best selling theme of all time on ThemeForest! Loved by 180k+ users.

ads via Carbon



Avada is the best selling theme of all time on ThemeForest! Loved by 180k+ users.

ads via Carbon



# inuitcss

ITCSS – coming soon...

css { guide:  
lines; }

## Events

Workshop

 **Topconf**, Linz (Austria), May 2016

I am available for hire to consult, advise, and develop with passionate product teams across the globe.

I specialise in large, product-based projects where performance and CSS scalability and maintainability are paramount.

I am currently considering new projects for Q2-3, 2016.

