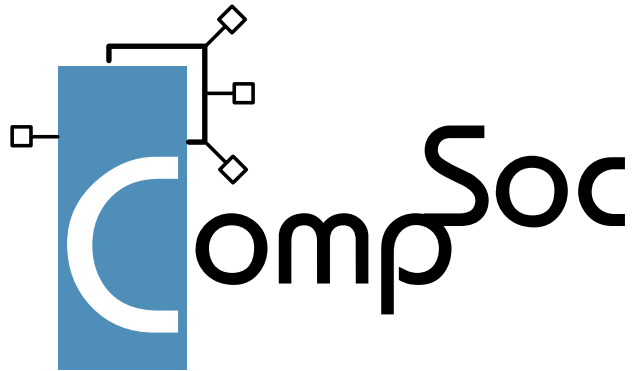


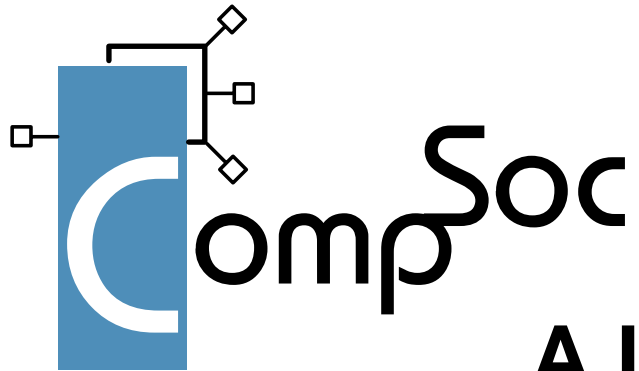
Cascade Style Sheets (CSS) Workshop

Advanced Web Design Methods and Creating Site Styles



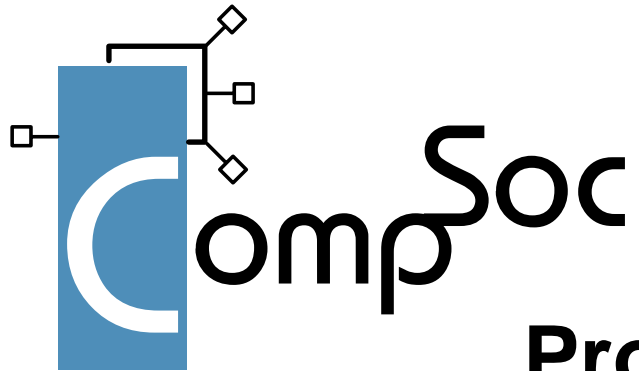
Contents

- A Little Background
- Problems that occur
- How does (CSS) fit in?
- What exactly is CSS?
- Who does it come from?
- CSS Attribute Structure
- How do we use CSS in HTML?
- The Span & Div tags
- Style Definitions
- CSS selectors
- Grouping & Inheritance
- Classes
- Applying a Class to an Element
- Cascading Concept
- Linking to an external Style Sheet
- 12 Quick CSS Effects



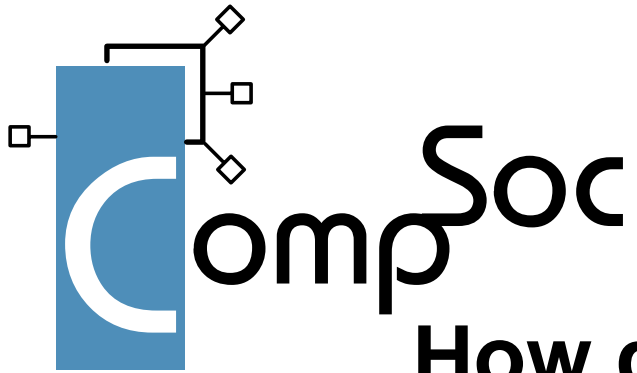
A Little Background

- Web Designers: perfectionists
- Rarely possible to get website working on all computers
- ``
Same result for everyone that looks at the page?
Mozilla & Opera? vs. IE & NN?
- Poor of sight, blind, colour blind, deaf, screen size, screen resolution, modem speed, line speed, processor speed, amount of RAM
- Arrival of digital TV, Game Consoles, PDAs, WAP



Problems that occur

- You've spent hours:
 - Positioning words precisely, placing images,
 - Balancing out tables
- To find:
 - Things wrapping around, columns pushed around, bits of text disappear in NN etc.



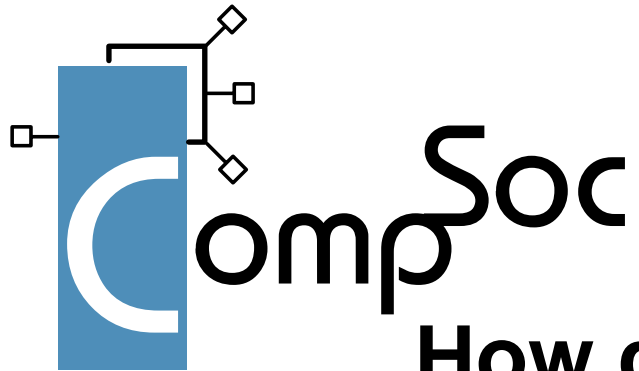
How does (CSS) fit in? (1)

- Offer more control over how elements are rendered by writing something like:

``

`This font is size 12 pixels`

- Not perfect (Netscape renders a fraction smaller) but best you'll get.

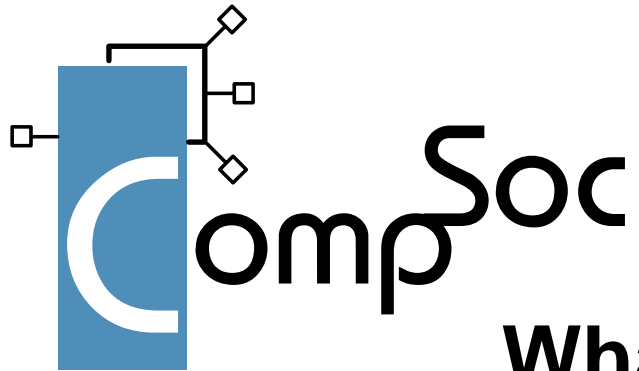


How does (CSS) fit in? (2)

- If you don't want browser to underline links and colour them red when rolled over, try this:

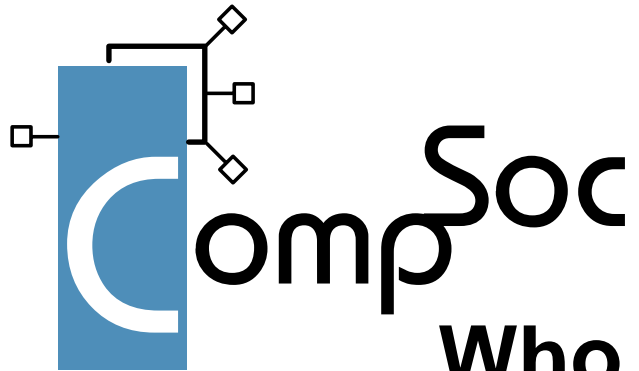
```
<a href="info.htm" style="color:black; text-decoration:none;">Info</A>
```

- First you need understand just what CSS actually **IS**, and how to use it.



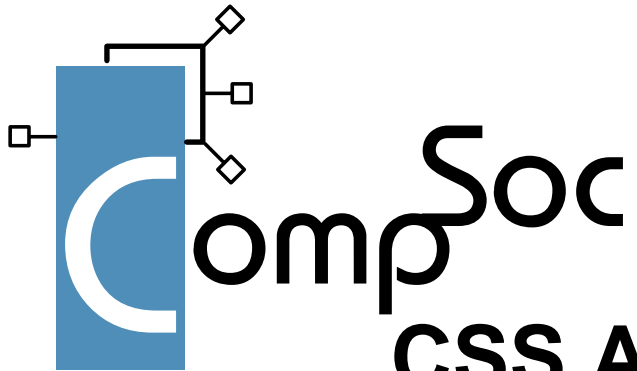
What exactly is CSS?

- CSS is a language, or a micro-language that's integrated into HTML, principally from the v4 browsers (which means IE4 and NS4).
 - It uses its own syntax
 - It doesn't interfere with old browsers
 - It offers a stronger, more predictable way to control appearances (font sizes to margins, backgrounds, forms, layers etc.)



Who does it come from?

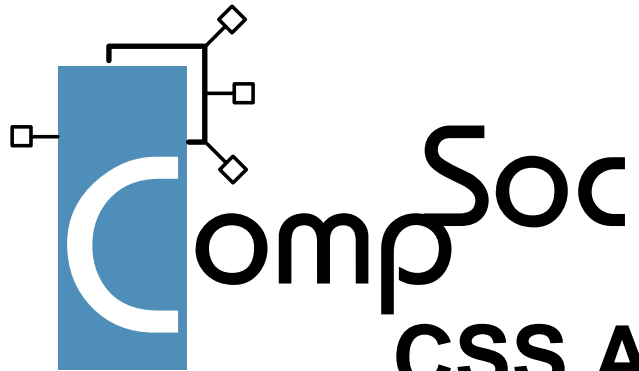
- There is only one CSS, as set down by the World Wide Web Consortium (W3C) (pay them a visit at www.w3c.org)
- These CSS standards are implemented by those who create the browsers (e.g. Mozilla projects, Microsoft, Netscape).



CSS Attribute Structure (1)

- In short, it pairs *attributes* (e.g. size, color) with *values* (e.g. 4px, #003366). An attribute picks out a property, the value says what it should be. Take this example:

```
<p style="color: #00FF00;  
margin-left: 10px">...</p>
```

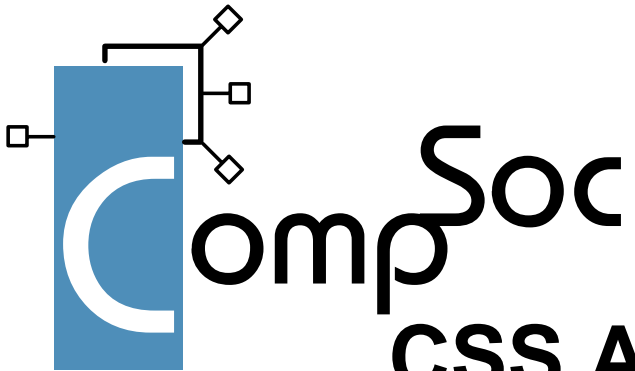


CSS Attribute Structure (2)

- Here's another example:

```
<P style="width: 50px;  
border-style: solid; border-width: 1px;  
border-color: black; padding: 5px"> Here's  
some words</P>
```

- This is fantastic! Why? Using basic HTML?



CSS Attribute Structure (3)

- The basic syntax for CSS is:

attribute: value;

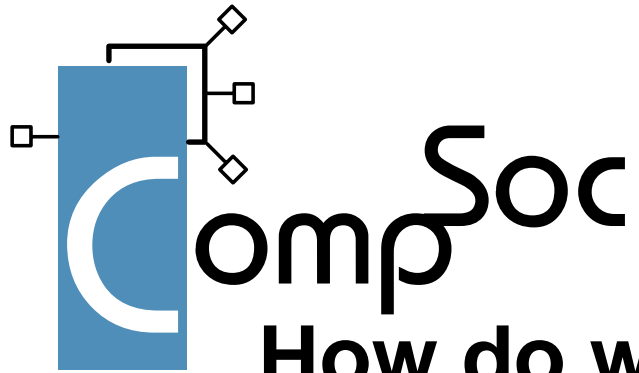
such as:

font-weight: bold

color: #330033

- With a semi-colon between pairs:

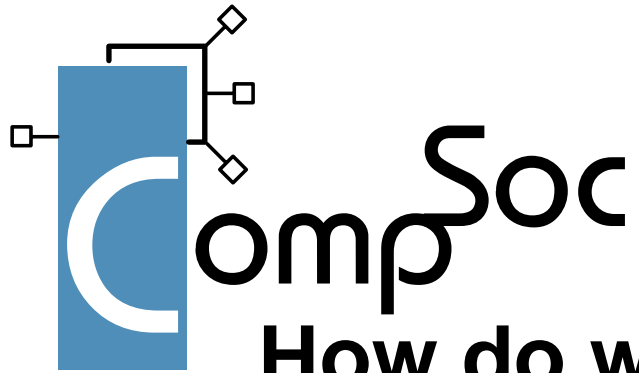
font-weight: bold; color: #330033



How do we use CSS in HTML? (1)

- You can apply a CSS style directly within a tag in your main body HTML.
- Just use **style="..."** within the tag (just as you would use **color="..."** or **href="..."**) and write your CSS between the quotes:

```
<tagname style=" ... ">  
  Values</tagname>
```



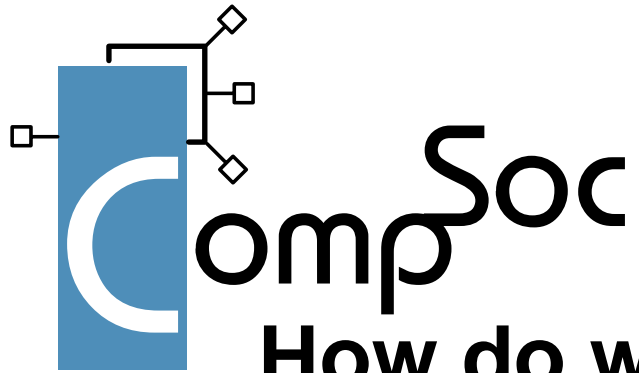
How do we use CSS in HTML? (2)

- An example of the style attribute in work with the H1 tag would be:

```
<H1 style=" ... ">
```

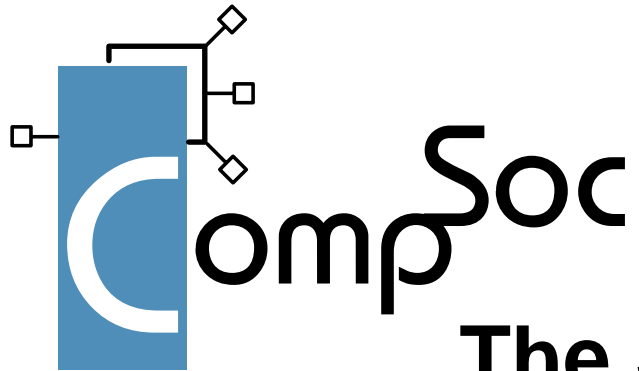
```
Your text here</H1>
```

- This method of applying a style to HTML is called inline styles, because the style attribute is in the tag it wants to apply itself to.



How do we use CSS in HTML? (3)

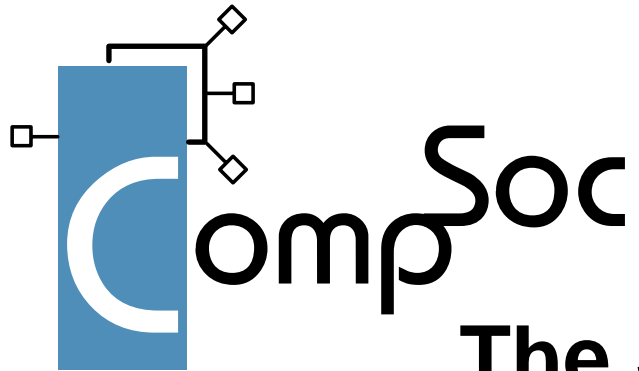
- You can use this **style="..."** syntax in any tag you want providing it makes CSS sense to apply that attribute to that tag.
- e.g. You can't apply a margin to a word in the middle of a sentence, but only to what are called '*block level elements*':
 - <P>, <H1>, <H2> etc.



The Span & Div tags (1)

- But what if you do want to apply a style to a word in the middle of a paragraph - make it blue, bold and italics, say?
- In 'traditional' HTML you would write this:

```
<FONT color="blue">  
<B><I>... </I></B>  
</FONT>
```

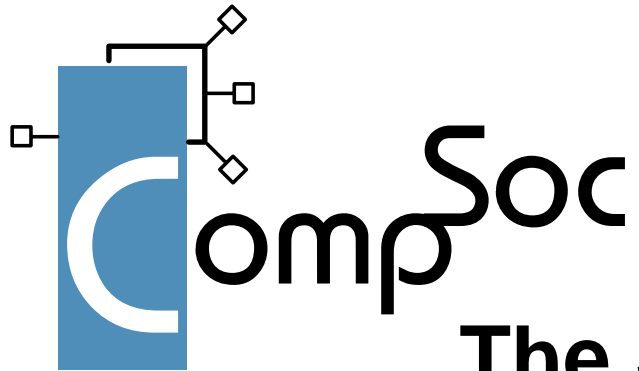


The Span & Div tags (2)

- In CSS lingo we write it as:

```
style= "color: blue; font-weight: bold;  
        font-style: italic"
```

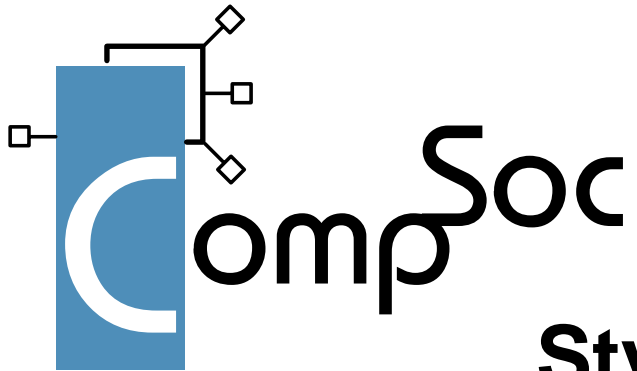
- But now you don't need any of the block tags, where do you put the style="..."?
- Could use any of these tags; but the favoured means is to use the “do nothing” **span** tag:



The Span & Div tags (3)

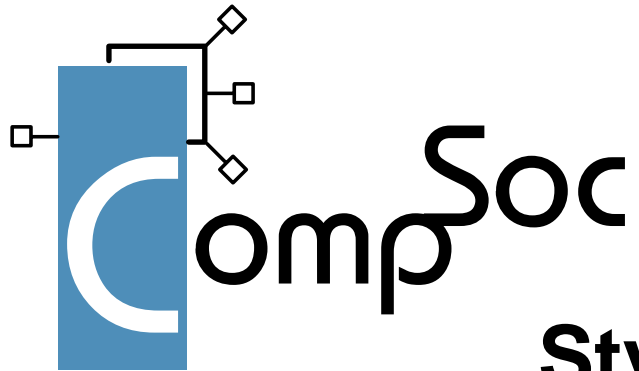
<SPAN style="color:blue;
font-weight:bold; font-style:italic">...

- has no particular meaning or attachment - it simply means: 'Do these things to whatever's within me'.
- <DIV> is similar, but is for block level elements



Style Definitions (1)

- Style Definitions are pieces of CSS code placed in the head section of your page. In them you can change a particular HTML element (e.g. `<p>`) to reflect certain styles.

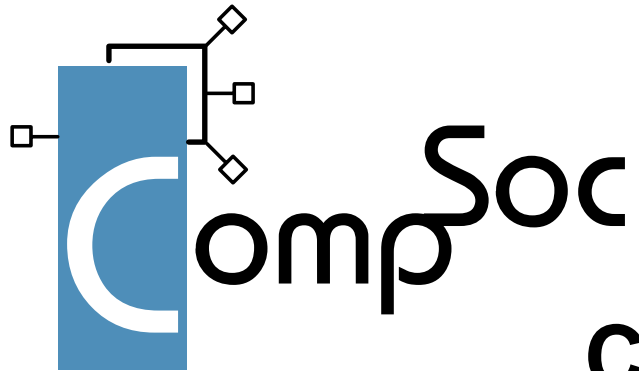


Style Definitions (2)

- A sample style definition:

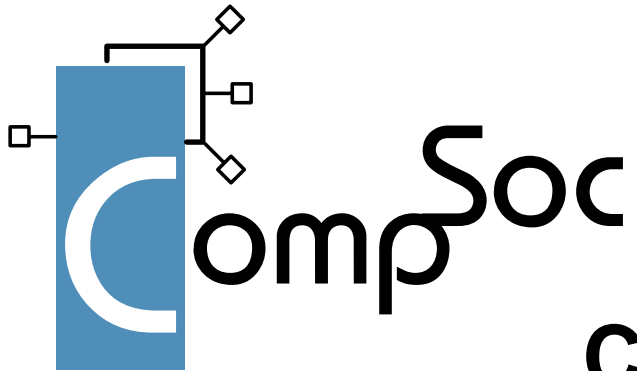
```
a {  
    text-decoration: none; color: black;  
    font-family: Verdana; font-size: 12pt;  
}
```

- Whatever styles you define for a tag in the HEAD section, will apply to all instances of that tag in your HTML body (really useful!!)



CSS selectors (1)

- A form of Style Definition used for mouse rollovers
 - Supported by later browsers
 - Can define **a:link**, **a:active**, **a:hover**,
 - Simply applies the selected style when the corresponding action is triggered (e.g. on it's own, when clicked on, on mouse over)

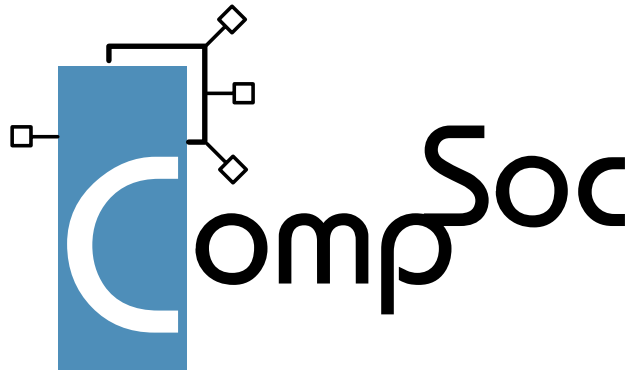


CSS selectors (2)

```
a:link { text-decoration: none;  
color: black; font-weight: bold }
```

```
a:hover {  
text-decoration: underline;  
color: darkgreen  
}
```

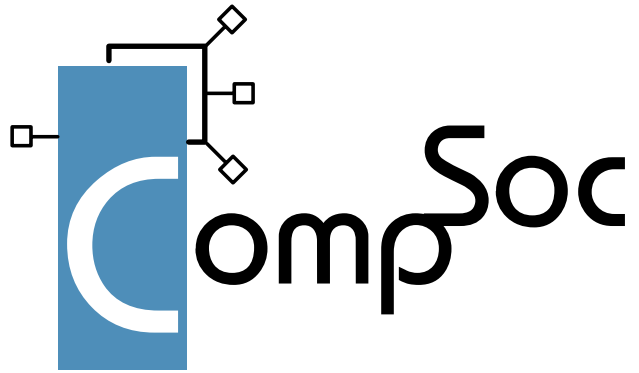
- And your plain, black, but bold link will underline and turn dark green on rollover.



Grouping

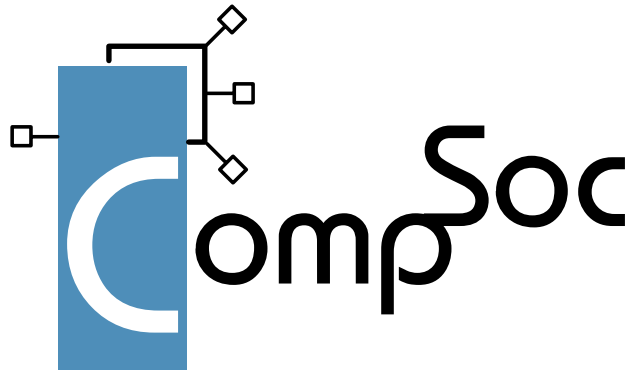
- In order to decrease repetitious statements within style sheets, grouping of selectors and declarations is allowed.

```
H1, H2, H3, H4, H5, H6 {  
    color: red;  
    font-family: sans-serif  
}
```



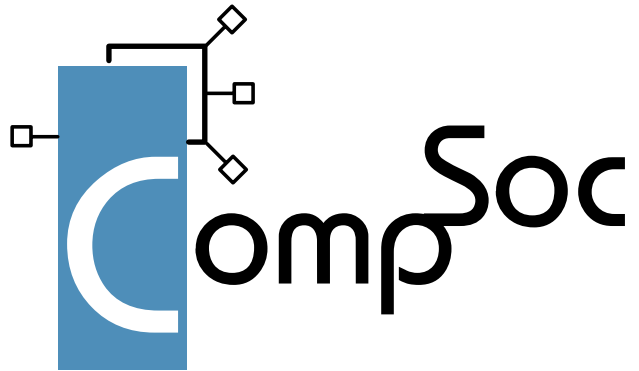
Inheritance

- Virtually all selectors which are nested within selectors will inherit the property values assigned to the outer selector unless otherwise modified.
- e.g. a colour defined for the BODY will also be applied to text in a paragraph.



Classes (1)

- There are two ways of creating styles.
 - You can redefine an existing HTML tag
e.g. <body>, <p>, <a>
 - You can create a brand new style, which you can apply to any tag you wish.
 - e.g. if I wanted a particular type of style for the sub headings in a news page, HTML does not supply me with any tag to do this.
 - So instead of redefining a <h2> tag, for example, I can create a **new style** called 'subhead'. This new type of style is called a **class**.



Classes (2)

- The definition for this subhead would be:

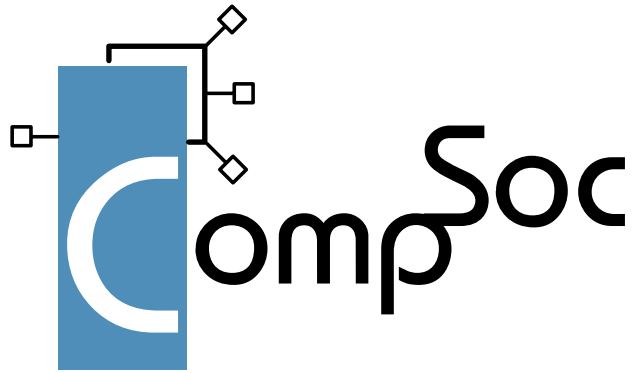
<style>

<!--

```
. subhead { font-family: Arial;  
            color: #0066CC;  
            font-size: 18pt }
```

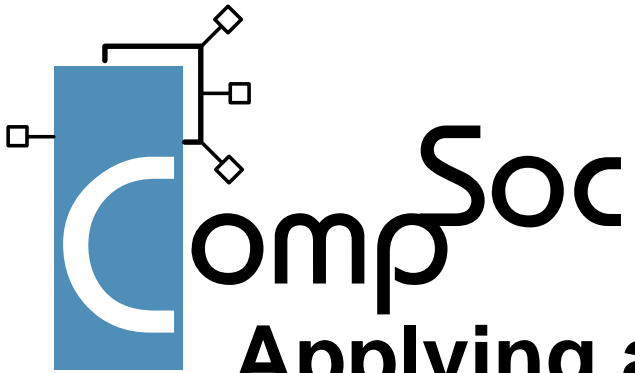
-->

</style>



Classes (3)

- Notice the '.' in front of the style name?
 - This is to indicate to the browser that this style is not redefining a HTML tag, but instead creating a class.



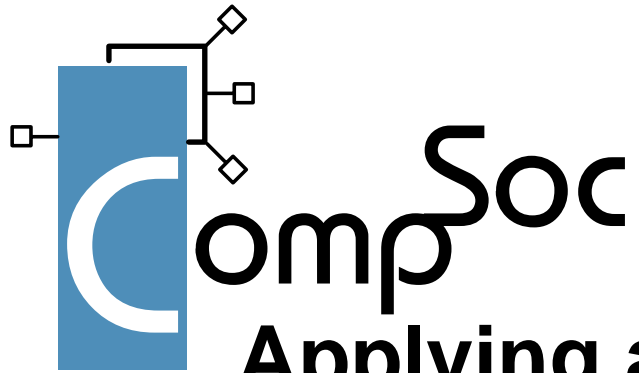
Applying a Class to an Element (1)

- Surprisingly enough, it's done through the `class=" ... "` attribute. For example:

```
.warning {  
  color: lime; background: #ff80c0 }
```

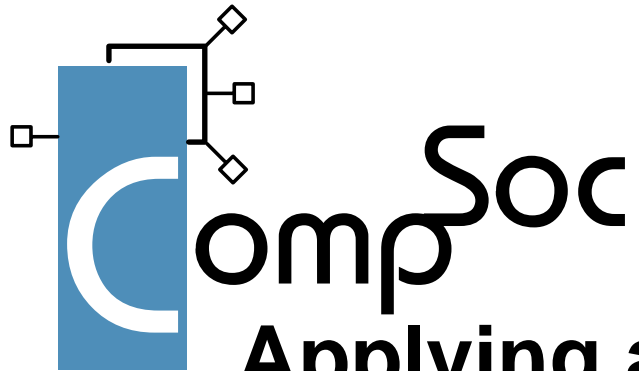
... used in HTML with the CLASS attribute ...

```
<h1 CLASS="warning">STOP!</h1>
```



Applying a Class to an Element (2)

- In this example, the warning class may be applied to any BODY element since it does not have an HTML element associated with it in the style sheet.
- Notice also that we left out the '.' when using the class attribute? (Rule: only put in when defining)

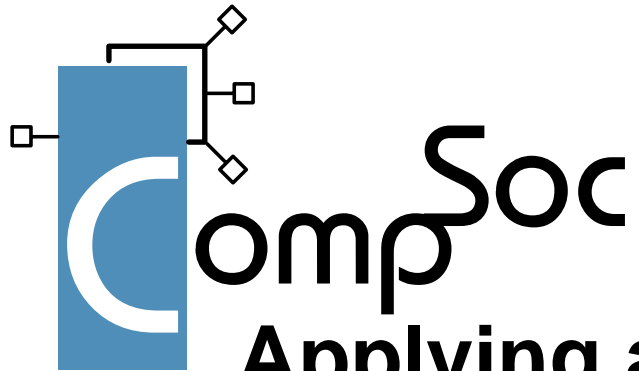


Applying a Class to an Element (3)

- We can also define classes to work only with certain tags. For example:

```
p.news {  
font-weight: bolder;  
color: red; background: white  
}
```

- In this example, the news class may only be applied to the P element.

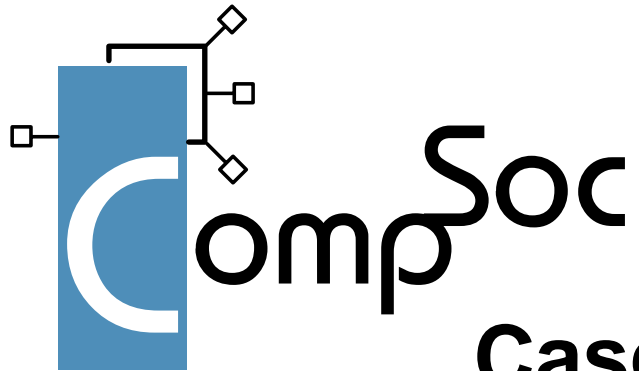


Applying a Class to an Element (4)

`<p class="news">`Due to the recent traffic crisis in Galway city, the Government have decided to abolish the road taxes, and instead introduce rent.`</p>`

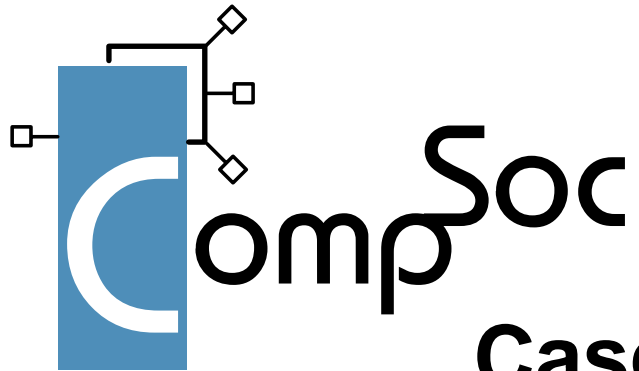
- The next example will not work:
 - the news class only works with the `<p>` tag

`<h1 class="news">`STOP!`</h1>`



Cascading Concept (1)

- As a final note, throughout all this wondrous stuff, you may have been asking: "Yes, but what the hell is '*cascading*'? "
- CSS has a tree-like structure, and one definition or class effects the things that come lower in the structure, or within its scope, unless another CSS rule comes into play, saying something different.



Cascading Concept (2)

- So, for instance, if you define:
 - The `<p>` tag as: black 12px Arial
 - A `.link` class as: blue Verdana
- Then think about this:

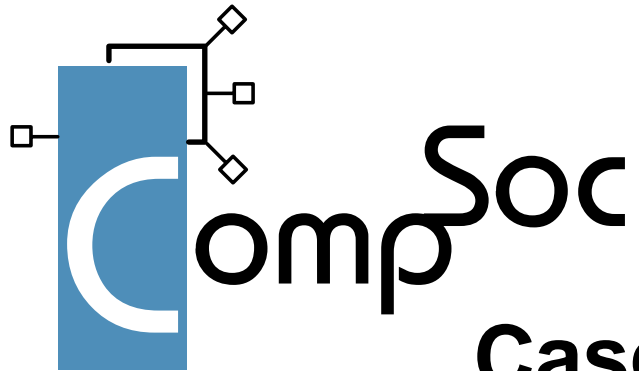
`<p>`For a more info,

`<a href="..."`

`class="link"`

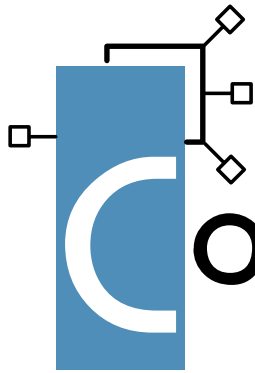
`style="color: green">click here`

`or else </p>`



Cascading Concept (3)

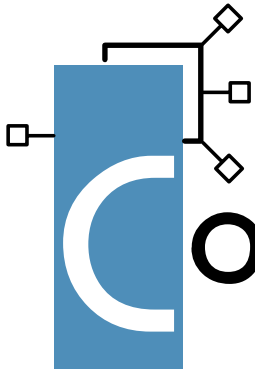
- What's going on?!
- This may seem difficult, but in practice it isn't, and makes good sense
 - Plus it gives great flexibility to the designer to manipulate elements at any stage
- So go forth and use it!



CompSoc

Linking to an external Style Sheet (1)

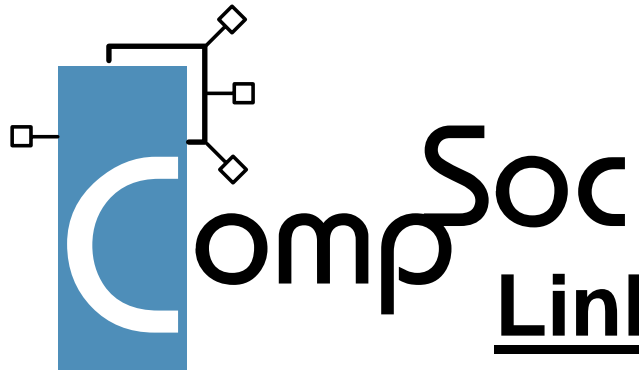
- This method of defining styles in the `<head>` section of the page is a lot more efficient than defining each item individually, as well as decreasing file size. If we have a large website however, we don't want to have to define these styles on every page.



CompSoc

Linking to an external Style Sheet (2)

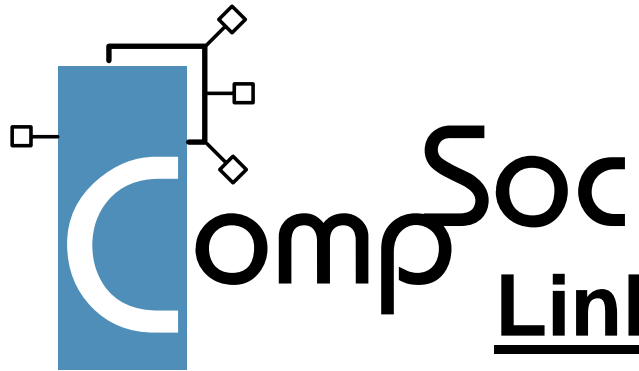
- CSS lets us create a file, where we can create our styles, and link this style file or **style sheet** to every page in our website, thus eliminating the need to type out the styles for every page.



Linking to an external Style Sheet (3)

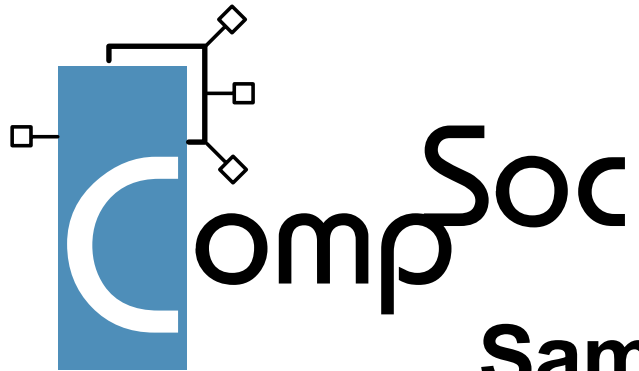
- To link a file to a style sheet, place the following code in the head section of our page:

```
<LINK  
  REL="stylesheet"  
  TYPE="text/css"  
  HREF="style.css">
```



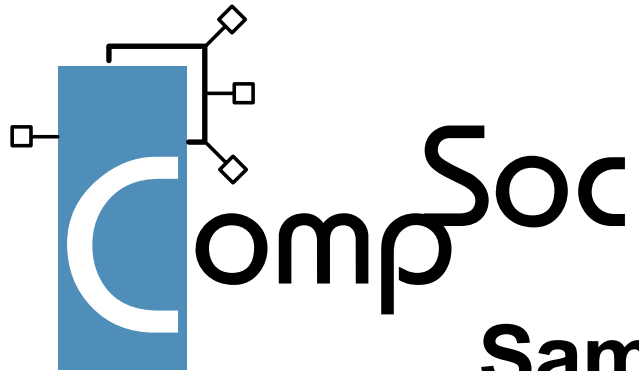
Linking to an external Style Sheet (4)

- This creates a link between our webpage and the style sheet, and all styles in the style sheet are adopted for the current webpage.
- It is important to know that external Style Sheets can only contain CSS specific mark-up. In otherwords, it CANNOT contain any HTML, only what you would normally place in the head section when defining styles.



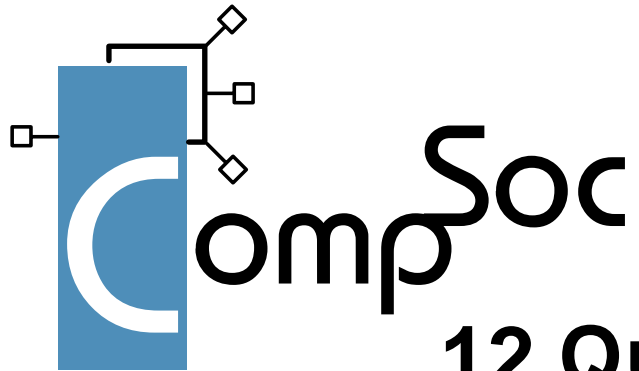
Sample Style Sheet (1)

```
/* This is a comment */  
a {    font-family: Verdana; font-size: 10pt;  
       color: #0000FF; text-decoration: none    }  
a:visited {    color: #0000FF;  
              text-decoration: none    }  
a:hover {    color: #0000FF;  
            text-decoration: underline    }  
a:active {    color: #FF8000    }
```



Sample Style Sheet (2)

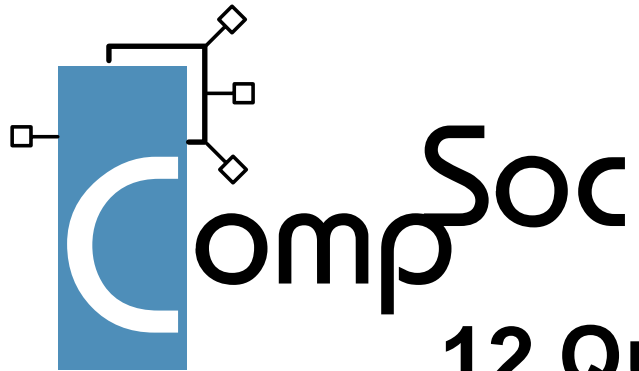
- `body { font-family: Verdana; font-size: 10pt; font-style: normal; font-weight: normal; color: #000000; background-color: #FFFFFF }`
- `td { font-family: Verdana; font-size: 10pt; font-style: normal; font-weight: normal; color: #000000; background-color: #FFFFFF }`
- `h1 { font-family: Verdana, "Times New Roman", Times, serif; font-size: 9px; color: #666666 }`



12 Quick CSS Effects (1)

- Control your text size
 - Ever get really miffed that setting `` for your text never properly controls the result? And moreover, the sizes 1, 2, 3, 4 and so on don't leave much room for precision. The CSS `font-size` property offers far more control. Try this:

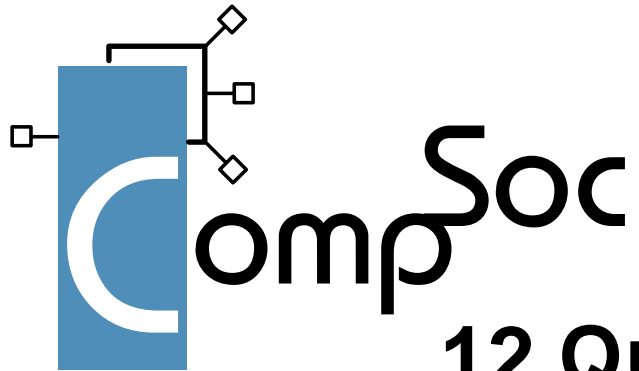
```
<p style="font-size: 12px">  
  Put your text in here</p>
```



12 Quick CSS Effects (2)

- Format all your text in one fell swoop
 - Why mess up your HTML applying font tags to every line of body text? This style rule should do for just about every line of body text you use:

```
p { font: normal 11px  
      Verdana, Arial, Helvetica, sans-serif }  
td { font: normal 11px  
      Verdana, Arial, Helvetica, sans-serif }
```



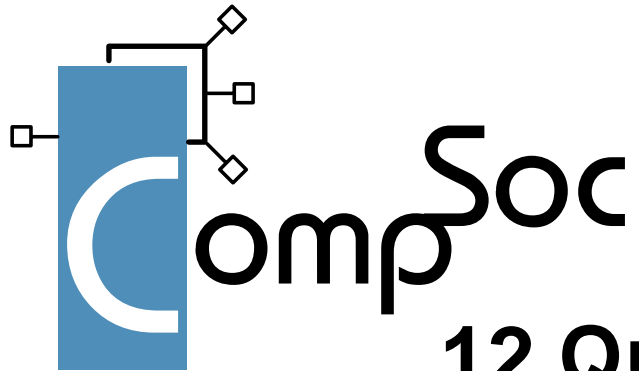
12 Quick CSS Effects (3)

- Centralise your heading styles
 - Now you can set up sub-styles for headings, subheadings and so on in just one place - so you can reformat the whole lot by changing just one line. Add this to your style sheet:

```
.subhead { font-size: 14px; font-weight: bold; }
```

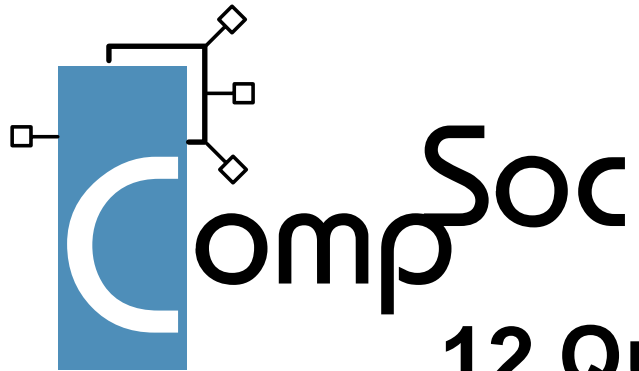
- Then for the subhead, write:

```
<p class="subhead">Subhead in here</p>
```



12 Quick CSS Effects (4)

- Get clever with links
 - The default behaviour for links - underlined and turning red on rollover in IE isn't to every designer's taste.
- ```
a { color: #003366; text-decoration: none }
a:hover { color: white;
background-color: #003366 }
```
- Now there's no underline, but the links have a blue-green background colour when you roll over.

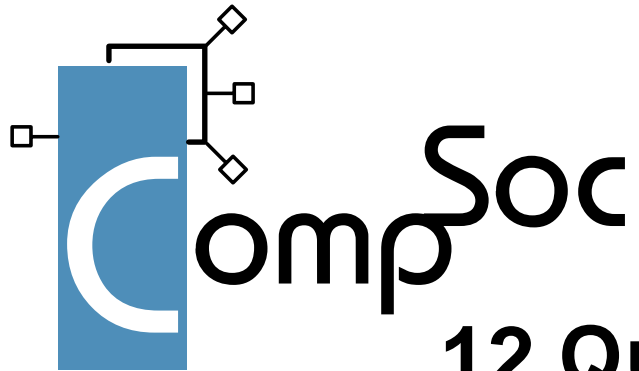


## 12 Quick CSS Effects (5)

- Create text margins
  - Margins are a nightmare to create in old HTML, fiddling about with tables and never quite knowing how the widths are going to turn out. Forget it: turn to CSS instead:

```
<p style="margin-left: 10px">...</p>
```

- Simple, eh?

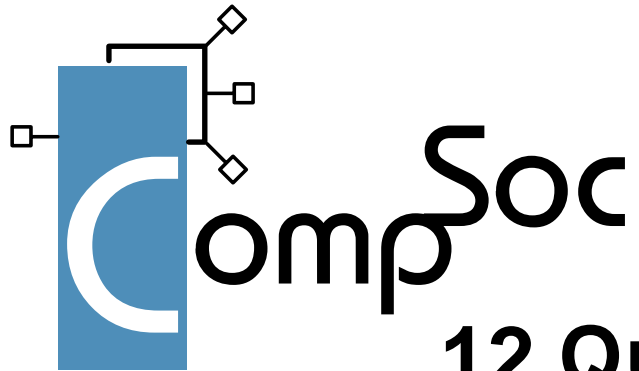


## 12 Quick CSS Effects (6)

- Funky IE cursor effects
  - In Internet Explorer, you can change the cursor that appears when you roll over a link - or indeed, images, text and so on that don't have links. How? Try this style rule:

```
a { cursor: crosshair }
```

- Alternatives to crosshair include hand, text, help, wait, and various resize options: n-resize (for a North angle), ne-resize (for Northeast), and so on.

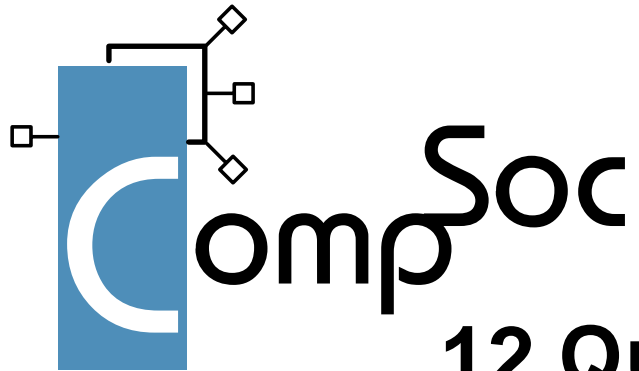


## 12 Quick CSS Effects (7)

- Juicy quote

- Ever wanted to have tempting quotes appearing in large italics through the main text of your articles (or callouts as they call them in the magazine business)? No probs. Just put this before a `<P>` tag:

```
<DIV style="
width: 130px;
float: right;
color: maroon;
font-size: 18px;
font-style: italic;
font-weight: bold">
"Here's your juicy
quote" </div>
```



## 12 Quick CSS Effects (8)

- Designer forms
  - Web forms look pretty dull in the default scheme of things, but you can easily style them up a bit with CSS. It only works in IE and Netscape after V6, but it's well worth it.
  - The effect appears as a black form input field and the text comes out white.

**<FORM>**

```
<INPUT type="text"
name="textfield" style="
```

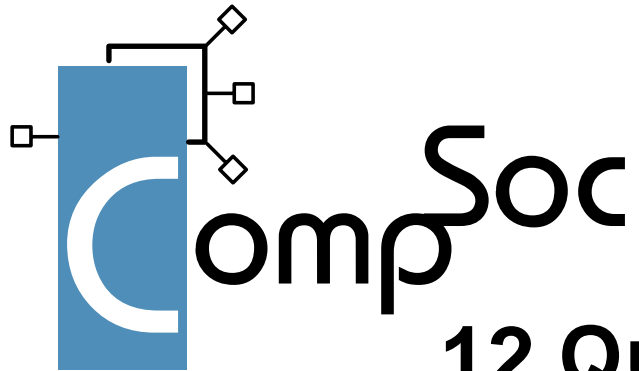
```
color: white;
```

```
background-color: black;
```

```
font: 11px Verdana,Helvetica"
```

```
value="Enter your keywords"
size="30">
```

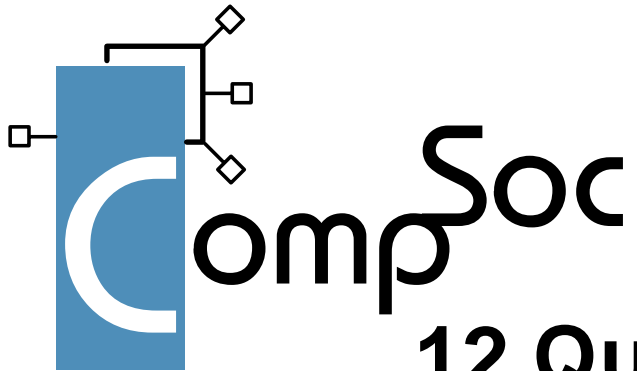
**</FORM>**



## 12 Quick CSS Effects (9)

- Form sizes
  - Another form-related stroke of CSS genius: if you test your work on all the browsers and platforms, you'll find it's nigh on impossible to get the widths of inputs, dropdowns and so on the same for all viewers - which is a pain if you're after well-spaced design. You can take 95 per cent control of this using the width CSS property, which works on forms for IE, and using the traditional size="" for NS:

```
<INPUT type="text"
 style="width: 200px" size="30">
```

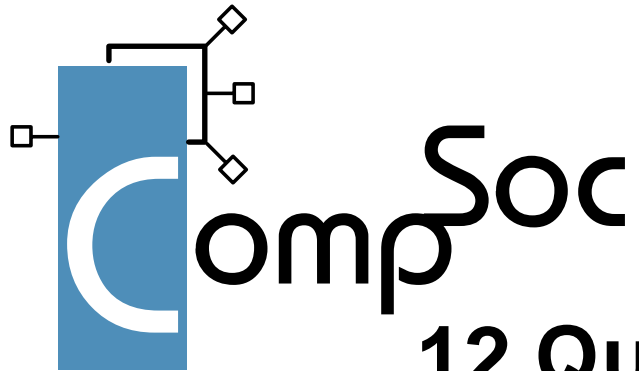


## 12 Quick CSS Effects (10)

- Enter keywords...
  - Not really CSS this, but while we're on forms, here's a useful JavaScript trick that people always ask about: how to include words in a form field, such as 'Enter keywords', which clear when they click in the space. Simple:

```
<INPUT type="text" name="textfield"
value="Enter your keywords"
onfocus="this.value=" size="30">
```

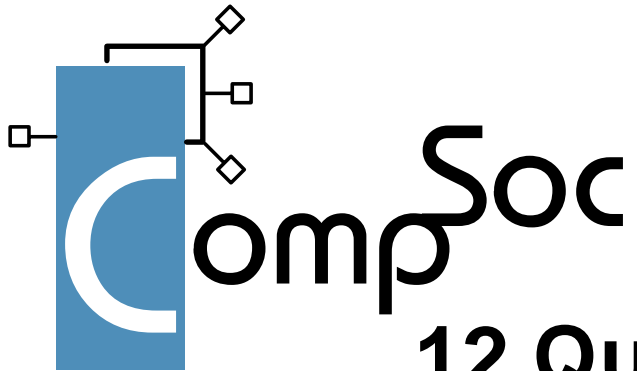
- (After this.value=, that's two single quotes followed by a double quote)



## 12 Quick CSS Effects (11)

- Cool scrollbars
  - Did you know that in IE you can change the colours of the scollbars at the right and bottom of your page? Pretty cool.
  - Twiddle the colour # references to suit yourself.

```
body {
 scrollbar-face-color: #2A314C;
 scrollbar-shadow-color: #2A314C;
 scrollbar-highlight-color: #2A314C;
 scrollbar-3dlight-color: #9AB6C4;
 scrollbar-darkshadow-color: #20253A;
 scrollbar-track-color: #20253A;
 scrollbar-arrow-color: #CCCCCC
}
```



## 12 Quick CSS Effects (12)

- Background control
  - People often ask about how the pros create the effect where you have a background graphic which doesn't tile and repeat; or which doesn't move when you scroll the page. The answer, of course, is using CSS, and here it is:

```
body {
```

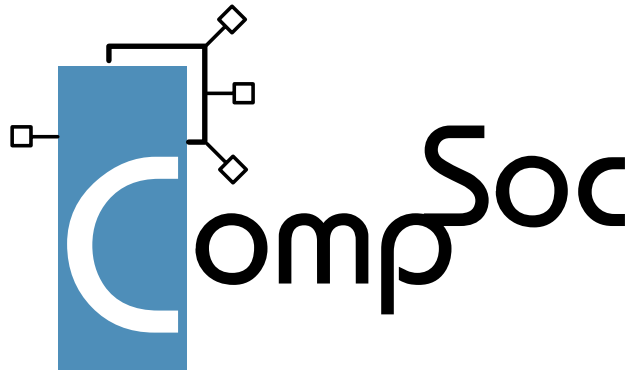
```
background-image: url(background.gif);
```

```
background-repeat: no-repeat;
```

```
background-attachment: fixed
```

```
}
```

- Replace background.gif with the URL of the image.
- background-repeat can be no-repeat, repeat, repeat-x or repeat-y. Attachment can be fixed or scroll.



**That's it folks!**

- That's more or less it as regarding CSS!
- [www.w3schools.com](http://www.w3schools.com)
- Any Questions—after workshop!