Front-end Web Development: Section 9   Pages

Notes - Class 3

CSS Position property

Governs how the position of an element is calculated on the page.

static: An element's natural placement in the flow of the document. Every element is static positioned by default. The element resides in the normal page flow. There is rarely a need to set this value unless the element has been set with another matching selector to a different value and you need to set it back.

relative: A position specified relative to the element's usual place in the flow of the document. Element's original position (as if it were static) remains in the flow of the document. But now left/right/top/bottom/z-index do work. The positional properties "nudge" the element from the original position in the specified direction.

absolute: A position specified from closest parent element with either position: absolute or position: relative. If not parent element has those, use the top left of the body element. Element is removed from the flow of the document (other elements will behave as if it's not even there). All the other positional properties work on it. Essentially you are able to declare the exact position where you want the element to appear.

fixed: A position specified from the browser window, regardless of whether the user has scrolled down the page or not. Element is removed from the flow of the document like absolute positioned elements. In fact they behave almost the same, only fixed positioned elements are always relative to the document (window), not any particular parent, and are unaffected by scrolling. It's usually used to provide a persistant visual element, like navigation bar that is always visible.

Absolute or Fixed elements no longer take up space in the flow of the document. Other elements will be moved to fill the space where they would have been.

position: static; /* this is a default */
position: relative;
position: absolute;
position: fixed;

with the last 3 above you can use following properties:
top
bottom
left
right
(z-index)
these properties will adjust position of an element by a specified measure in relation to element's original position within it's parent element.
These properties, in px or %, will specify the distance from the relevant edge for your element, if you're using position relative, absolute or fixed. (left/right/top/bottom/z-index have no effect on a static positioned element)

example of use:

```
    h1 {
        position: relative;
        top: 20px; /* this will move element relative to its original position by 20 px from the top, you can also use % here, in which case it will take percentage
of the total
        width of the parent element */
    }
```

- position: absolute;   /*this stops element having effect on elements around it and positions it absolutely as specified by top, left, right or bottom measurements in relation to the parent element. This is great for making elements overlap, in which case you will find the following property to specify which element should appear on top or otherwise and in which order:

- z-index: 1; /* default value for z-index is 0, so the higher the number the higher its stacking order */

A List Apart has a great article called CSS Positioning 101 which covers the above.

****

Tools mentioned on the day:

http://wave.webaim.org/ - checks your Accessibility score

http://validator.w3.org/ - checks your HTML

http://jigsaw.w3.org/css-validator/ - checks your CSS

http://chengyinliu.com/whatfont.html - Find out which web font

http://heydonworks.com/revenge_css_bookmarklet/ - on-page highlighter for bad HTML

http://www.cornify.com/ - Unicorn and rainbow happiness

****

Please use previous week's notes for reviewing Floats and Clearing

****

We used known elements and properties to build out navigation and learnt about sprites and how they are used. So these are the main takeaways from last class:
Navigation
Sprites
Practice!

Navigation

here's an example of navigation bar:

```
<nav>
     <ul>
        <li>
            <a href="home.html">Home</a>
        </li>
        <li>
            <a href="about.html">About</a>
        </li>
        <li>
            <a href="contact.html">Contact</a>
        </li>
    </ul>
<nav>
```

we use <ul> unordered list to organise our navigation links, and we describe that this is our navigation by using <nav> element.

NB when it comes to navigation it's important that not just the text area that is a clickable link, its a good practice to ensure the area around is also clickable, this is achievable by adding adjusting padding on the <a> element to a desired "size", or if you must specify a width on it

"Nav simple" exercise (code along activity)

Notes:

- to make <li> elements of our navigation appear on one line we float them left

- browser default for links is blue, when overriding this colour its best to override using <a> selector

- because <a> elements by default are inline inline they will not take specified width, so when applying width to <a> element (to ensure the area around text link is also clickable) you need to apply "display: block;" to them and then you can select a width.

```
    nav a {
        display: block;
        width: 230px; /* by specifying width we specify the size of the area which is clickable*/
        padding: 5px; /* adding padding also increases clickable area */
        background: red;
        color: white;
        text-decoration: none;
    }
```

- in this exercise our nav links are the same width and the spaces between them are also the same

:active pseudo class:

```
    nav a:active {
        background: pink; /* the background of the link will become pink on click */
    }
```

side note: aparently a big proportion of male population does not differentiate between red and green or between red and blue, you might want to consider this when selecting colours for your site

Purple Nav exercise (independent activity)

- we don't have specify the width of nav elements in this exercise, we can just apply the same padding to all of them instead, so as a result their width is the width of the text link plus the padding around it, adding the padding does the same job as setting the width to ensure that the area around the text link is also clickable (the latter as was done in the previous exercise)

side note:
- a good practice is to make sure that all your background images are loaded by the browser prior to the user needing/activating them.

Because background images are cropped off if they go beyond the border of elements for which they are used as a background, you can easily set the background image earlier in your css with the position for it to be such that its not visible to user, in this case browser will already have it loaded loaded it, so that when requested by the user there's no delay at all; all you will need is to change position to the appropriate coordinates to show desirable portion of image.

so in our 2nd exercise instead of having this:

```
nav ul li a {
    display: block;
    padding: 30px 18px;
    text-decoration: none;
    color: white;
}

nav ul li a:hover {
    background: url(../images/bg_nav_rollover.png) repeat-x;
}
```

we could have this:

```
nav ul li a {
    display: block;
    padding: 30px 18px;
    text-decoration: none;
    color: white;
    background: url(../images/bg_nav_rollover.png) repeat-x; /* this allows the browser to load the image prior to user activating the link */
    background-position: 100% 100%;
}

nav ul li a:hover {
    background-position: 0 0;
}
```

Some reading on background-position property
https://developer.mozilla.org/en-US/docs/Web/CSS/background-position

Sprite social exercise

Notes:

- spritecow.com - great for working out the position of your sprite

- sprites generally need to be .png type as they need to have transparency preserved

Blue nav exercise - homework

using class "selected to apply desired styling to the nav link of the page you are on

and don't forget about:

Image replacement technique

This technique is intended to keep the page accessible to users of screen readers, text-only web browsers, or other browsers where support for images or style sheets is either disabled or nonexistent, while allowing the image to differ between styles.

```
text-indent: -100%;
overflow: hidden;
white-space: nowrap;
```

This means that all of you remember that you worked on few exercises. We used known elements and properties to build out navigation and learnt about sprites and how they are used. We also used image replacement technique. So these are the main takeaways form Wednesday class: • Navigation • Sprites • Image replacement technique • Practice!
Navigation

here's an example of navigation bar:

```
<nav>     <ul>        <li>            <a href="home.html">Home</a>         </li>         <li>            <a href="about.html">About</a>         </li>
   <li>            <a href="contact.html">Contact</a>         </li>     </ul><nav>
```

we use <ul> unordered list to organise our navigation links, and we describe that this is our navigation by using <nav> element.
NB when it comes to navigation it's important that not just the text area that is a clickable link, its a good practice to ensure the area around is also clickable

"Nav simple" exercise (code along activity)

Notes:
- to make <li> elements of our navigation appear on one line we float them left
- browser default for links is blue, when overriding this colour its best to override using <a> selector
- because <a> elements by default are inline inline they will not take specified width, so when applying width to <a> element (to ensure the area around text link is also clickable) you need to apply "display: block;" to them and then you can select a width.
nav a { display: block; width: 230px; /* by specifying width we specify the size of the area which is clickable*/ padding: 5px; /* adding padding also increases clickable area */ background: red; color: white; text-decoration: none; }
- in this exercise our nav links are the same width and the spaces between them are also the same
:active pseudo class: nav a:active { background: pink; /* the background of the link will become pink on click */ }
side note: a big proportion of male population does not differentiate between red and green or between red and blue, you might want to consider this when selecting colours for your site

Purple Nav exercise (independent activity)
- we don't specify the width of nav elements in this exercise, we just apply the same padding to all of them instead, so as a result their width is the width of the text link plus the padding around it, adding the padding does the same job as setting the width to ensure that the area around the text link is also clickable (the latter as was done in the previous exercise)
- try making sure that all your background images are loaded by the browser prior to the user needing/activating them. Because background images are cropped off if they go beyond the border of elements for which they are used as a background, you can easily set the background image earlier in your css with the position for it to be such that its not visible to user, in this case browser will already have it loaded loaded it, so that when requested by the user there's no delay at all; all you will need is to change position to the appropriate coordinates to show desirable portion of image.
so in our 2nd exercise instead of having this:
nav ul li a { display: block; padding: 30px 18px; text-decoration: none; color: white; }
nav ul li a:hover { background: url(../images/bg_nav_rollover.png) repeat-x; }
we could have this:
nav ul li a { display: block; padding: 30px 18px; text-decoration: none; color: white; background: url(../images/bg_nav_rollover.png) repeat-x; /* this allows the browser to load the image prior to user activating the link */ background-position: 100% 100%; }
nav ul li a:hover { background-position: 0 0; }
Some reading on background-position propertyhttps://developer.mozilla.org/en-US/docs/Web/CSS/background-position

Sprite social exercise (code along activity)
Notes:
- spritecow.com - great for working out the position of your sprite
- sprites generally need to be .png as they need to have transparency preserved

Image replacement technique
This technique is intended to keep the page accessible to users of screen readers, text-only web browsers, or other browsers where support for images or style sheets is either disabled or nonexistent, while allowing the image to differ between styles. text-indent: -100%; overflow: hidden; white-space: nowrap;

- Simon mentioned that .png's end up being smaller in size if they are arranged such that they are wider rather than taller