CSS Intermediate

Review

- What is the DOM?
- What is separation of concerns : HTML/CSS : what advt ?
- So what is job of HTML?
- Is it a programming language?
- Ruled based ?
- Advt of moving styles into a separate file.
- Review COLOUR : RGBA..
- Units for font-size : EMs.
- Why is it cascading?
- What are stylistic hooks? e.g

A **rule based** language for style/presentation of a web page.

Multiple DOM elements

```
/* Style all paras */
p {
    color:green;
}
```

Single ID

```
#special {
    font-size : 24px;
}
```

Multiple classes

```
.offer {
    border: 1px solid black;
}
```

When to use each approach

- Use elements for general rules, higher up the CSS document.
- IDs can be used for specific structural elements, like headers/footers.
- Classes can be used on multiple different element types.
- Classes encourage **loose coupling**.

Nesting/Contextual

We can apply styling, based on relative position of an element in the DOM tree.

```
div p {
    text-decoration: underline;
}
```

Importance of space character

```
/* WITHIN : Any element with a class of offer insider a DIV *
div .offer {}

/* WITH : A div with a class of offer */
div.offer {}
```

[COURSE MANUAL p8 : selector examples]

[EXERCISE]

```
- intro/tower
- intro/cornwall
```

Block and inline elements

Block elements

- Start on a new line
- Occupy 100% width of containing element, or browser window

- Box Model properties changeable
- DIVs, Ps

```
WIDTH OF A CONTAINING ELEMENT

<div class="fred">      CSS : .fred{ width:50%; }
    <p>Hello</p>
</div>
<div class="jane">
    <p>Goodbye</p>
</div>
```

Relevance of this to **responsive design**, and the **mobile first** concept > Luke Wroblewski

Inline elements

- Same line
- Box model properties not changeable, unless display:block
- Width based on content or image
- SPANs, INPUTs

Switching the DISPLAY type

Margins top/bottom do not show on **INLINE** elements. They do show if switch to **INLINE-BLOCK**. Change to **BLOCK**, and it adds new line. Change to **NONE**, and they dissapear. Hiding block-level elements, will hide their containing inline elements.

[EXERCISE]

```
- block_inline
```

Box Model

The **content** of every DOM element is surrounded by a **padding**, **border** and **margin**.

What is the total width of an element ? Width of element either set explicitly, or

implied from width of containing element.

```
width element + padding + border + margin
```

View the **Chrome developer** tools.

Box-sizing

This default width behaviour can be overriden in IE8+.

```
box-sizing: border-box;
```

This command changes the box model, to shrink content, as padding and border increase, to keep the overall element width the same.

T-R-B-L

Margins and padding can be assigned individually, for each of the four sides.

Borders

Borders need width, style and colour.

```
/* solid, dashed, double dotted */
border: 4px solid green;
```

[EXERCISE]

```
- boxmodel/cards
- boxmodel/box_sizing
```

Inheritance

Specificity

- IDs more specific than
- CLASSES more specific than
- Contextual selectors

- Element selectors

Some stylesheet properties inherit, others do not. Text properties (font size and colour) are passed down. Box model properties like border and margin do not.

!important allows you to override the default inheritance rules.

```
p {color: blue !important;}
```

The **more specific** a selector, the more weight it is given in the cascade.

Text-related CSS commands are inherited by descendant elements

```
letter-spacing: 0.1em;
font-size: 2em;
text-align: right;
color: white;
```

Layout-related commands aren't inherited.

```
border: 1px solid black;
padding: 1em;
background-image: url( ../media/linen.png );
width:500px;
```

The background-image will show through on inherited elements. The width of child elements expands to 100% the width of their parents, by default.

[EXERCISE]

```
- Inheritance
- View the **Chrome developer** tools.
```

Planning

- Define **general rules** at the top : typography, colour.
- Logical **naming**
- Dont name things based on colour or layout position

- **Comment**
- CSS in same **order** as HTML : e.g. header/footer
- Dont use inline styles in HTML
- Be concise
- Take advantage of the cascade
- Judgement call about IDs and CLASSES.
- Avoid complex selectors : tight coupling

Text

Text can use **relative** units like % and EM.

Text can use **absolute** units like px.

```
/*
We use a percentage on the top element.
This defaults to 16px on most browsers.
*/

body {
    font-size: 100%;
}
```

A font-size of 62.5% would set this to 10 pixels.

NOTE : EM sizes are relative to their parent element.

[EXERCISE]

```
- fonts/poem
```

Links

a:link is for not visited, **a:visited** is for visited, and just "a" applies to both.

Rules need to be applied in the **LVHA** order, to work.

```
a:link { }
```

```
Unvisited link.

a:visited { }
Visited links.

a:hover { }
The user mouses over a link.

a:focus { }
The user clicks on a link.

a:active { }
The user has clicked a link.
```

Backgrounds

To style an icon

```
<a class="icon" href="http://www.adobe.com">Adobe</a>
```

Add CSS background image and padding

```
.icon {
    background: url( ../media/icon.png) no-repeat 0 0;
    color: black;
    display: block;
    font-size: 4em;
    margin: 0.5em;
    padding-left: 1.25em;
    text-decoration: none;
}
```

[EXERCISE]

```
- background/fish
- icon/icon
```

Creating boxes with curves

This used to require applying four classes around a paragraph, with four images attached.

Border-radius solves this with 1 line of CSS.

On IE8, this reverts to square corners. CSSPie is a polyfill.

(Used in boxmodel/cards example)

Layout

```
* Relative absolute
* Navigation top
* Z Index
* Nested
```

By using absolute positioning for navigation at the top of the screen, the content can come first in the code, thereby offering usability (the content displays first), accessibility (there's no need for skip to content links) and SEO (search engines can more easily access page content) benefits.

[EXERCISE]

```
- layout/relativeAbsolute.
- layout/navigationTop
- layout/nested
- layout/zIndex
```