CSS

Cascade, Fonts, Text, Colors

Cascading and Order

- ➢ UA stylesheets
- ➢ User stylesheets
- ➢ Author styles

Cascading and Order

- **UA stylesheets**

  The CSS rules that are built into the browser. The UA stylesheet has absolute priority for the appearance of elements unless its rules are directly contradicted by declarations in the user or author stylesheets below. The rules in the UA stylesheet determine how the browser presents every website by default; subtle differences in the UA stylesheets of different vendors is the reason many web developers prefer to start the CSS for their sites with reset(reset.css) or normalize declarations.

Cascading and Order

- **User stylesheets**

  The user may set their own rules for how they wish to see HTML content. This may include setting the zoom level on pages, increasing the font size, or using high-contrast colors.

Cascading and Order

- **Author styles**

  Divided into three sub-levels:

  - Inline styles for individual elements.
  - In-page styles.
  - Linked CSS rules from an external stylesheet.

Cascading and Order

- **Author styles**

  Precedence of styling

  1) Inline of element

  2) In-page css styling within the <HEAD>..</HEAD> of the document

  3) Linked external css file

Cascading and Order

➢ **The !important Keyword**

Will give the targeted element style precedence over any css styling that targets the same element.

```
.classname {
   color: red !important;
}
```

Specificity: More specific, not more important

Specificity is the means by which a browser decides which property values are the most relevant to an element and gets to be applied. Specificity is only based on the matching rules which are composed of selectors of different sorts.

Ascending order of specificity

- Universal selectors
- Type selectors
- Class selectors, Attributes selectors, Pseudo-classes
- ID selectors
- Inline style

Specificity

A selector's specificity is calculated as follows:

- Count the number of ID attributes in the selector (= a)
- Count the number of classes, attribute and pseudo-classes(:hover,:focus,..) in the selector (= b)
- Count the number of element names in the selector (= c)
- Ignore pseudo-elements.

Concatenating the three numbers a-b-c gives the specificity.

Specificity

Exercise: write a specific rule that colors td-22 in red

```
<table>
<tbody>
<tr id=”tr1”>
	<td>td-11</td>
	<td class=”tdClass”>td-12</td>
</tr>
<tr id=”tr2”>
	<td>td-21</td>
	<td class=”tdClass”>td-22</td>
</tr>
</tbody>
</table>
```

Inherit

The inherit keyword specifies that a property should inherit its value from its parent element.

Fonts

- Web fonts:
  - TTF/OTF (True type fonts/ Open type fonts)
  - EOT (Embedded open type, IE)
  - SVG (Scalable vector graphic)
  - WOFF (Web open font format)

- Adding fonts:

```
@font-face { font-family: myFirstFont;
             src: url(fileLocation.ttf)
           }
```

Ex:https://github.com/dasrick/npm-font-open-sans/blob/master/open-sans.css

Fonts

- font-family:
  - Possible values: serif, sans-serif, monospace, ‘Times New Roman’...
  - Can be stacked(font1, font2,...)
- font-size:
  - em, %: scalable
  - pt: printable good for print style sheets
  - px fixed pixels

1em

= 12pt

= 16px

= 100%

Fonts

- ➢ font-style:

  Possible values: italic, oblique, normal

- ➢ font-weight:

  Possible values: bold, bolder, lighter, normal, numeric value

- ➢ font-variant:

  Possible values: normal, small-caps

Exercise: Download and use a web font.

Text

- ➢ letter-spacing
- ➢ word-spacing
- ➢ line-height: Specifies the line height
- ➢ text-transform

  Possible Values: none, capitalize, uppercase, lowercase

- ➢ text-align

  Possible Values: left, right, center, justify

Text

- vertical-align

  Possible Values: super, sub, middle

- text-indent

- white-space

  Possible Values: normal, nowrap, pre, pre-wrap, pre-line, initial

- text-decoration:

  Possible Values: none, underline, line-through, overline

- color:#000;

Color

- Color name
- Long format hex #0000ff
- Short format hex #00f
- rgb(255, 255, 255)
- hsl(306, 50%, 51%)

colrd.com/create/palette

Color

- Hue: is a degree on the color wheel; 0 (or 360) is red, 120 is green, 240 is blue. Numbers in between reflect different shades.
- Saturation: is a percentage value; 100% is the full colour.
- Lightness: is also a percentage; 0% is dark (black), 100% is light (white), and 50% is the average.

Color

To add transparency, we use the rgba() function to define the color stops.

The last parameter in the rgba() function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

- ➢ rgba(255, 255, 255, 0.8)
- ➢ hsla(306, 50%,51%, 1) http://standardista.com/webkit/ch7/hsla.html

Opacity

- The opacity CSS property specifies the transparency of an element.

  The value applies to the element as a whole, including its contents, even though the value is not inherited by child elements.

  If you do not want apply opacity to child element - use rgba instead

0	The element is fully transparent (that is, invisible).
1	The element is fully opaque (solid).

Developer tools & code formatting

- ➢ Chrome developer tools
- ➢ Importance of styling the code so it is easy to read