SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10026
Computing Technology Inquiry Project

Lecture 5

CSS 2 – CSS Properties

Topics

- CSS properties define which aspect of the *selected* HTML element will be changed or styled
  - □ Size
  - □ Color
  - □ Typography
  - □ List
  - □ Display and Alignment

CSS: Property Groups

- Animation
- **Background**
- **Border and outline**
- **Box**
- **Color**
- Content Paged Media
- Dimension
- Flexible Box
- **Font**
- Generated content
- Grid
- Hyperlink
- Linebox
- **List**
- **Margin**
- Marquee
- Multi-column
- **Padding**
- **Paged Media**
- **Positioning**
- Print
- Ruby
- Speech
- Table
- Text
- 2D/3D Transform
- Transition
- User-interface

Size measurement

- ***Relative*** is used for styling screen webpages
  `e.g. h1 { letter-spacing: .5em;}`

Unit	Abbr	Description	Example
EM	em	Height of the current font's default size	p {padding: 2em;}
Percentage	%	Works like em, where 100% is the default font size	p {line-height: 100%;}
Ex	ex	Height of letter ***x*** in the current font	p {margin: 25ex;}
Pixel	px	Pixel size of screen	p {font-size: 12px;}

- Avoid units *absolute* or *print* measurements:
  `cm, in, mm, pt`

Topics

- CSS properties define which aspect of the *selected* HTML element will be changed or styled
  - □ Size
  - □ Color
  - □ Typography
  - □ List
  - □ Display and Alignment

Color

- We can specify **`color:`** in the following four basic ways:

Format	Description and Examples
`name`	Colour **names**. There are 16 basic colours (from the Windows VGA palette) Many others are now accepted by popular browsers, but best to use ‘hex’ colors.

	`h1 {color: red}`
	`p {color: green}`

`#rrggbb` *or* **`#rgb`**	Red, green and blue values in **hexadecimal** format Written in “hex’ format in 6 or concise 3 character versions. Colour values between 00 and FF (or 0 and F)

	`hr {color: #FF0000} /* red */`
	`td {color: #00F} /* blue concise format - saves bandwidth */`

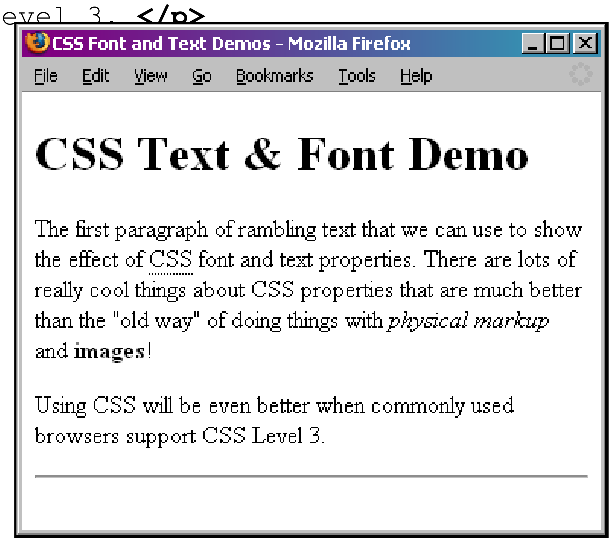
`rgb(r,g,b)`	rgb (red,green,blue) values in **decimal** with the **`rgb()`** command. Units between 0 and 255
	`.info {color: rgb(255,0,255); } /* purple info class */`
`rgb(r%,g%,b%)`	rgb (red,green,blue) values in **percentage** units with the **`rgb()`** command. Unit values between 0% and 100%.
	`em {color: rgb(100%,0%,100%); } /* purple emphasised text */`

Topics

- CSS properties define which aspect of the *selected* HTML element will be changed or styled
  - □ Size
  - □ Color
  - □ Typography
  - □ List
  - □ Display and Alignment

CSS Font and Text Example

```
...
<body>
<h1>CSS Text &amp; Font Demo</h1>
<p class="intro">The first paragraph of rambling text that we can use
   to show the effect of <abbr title="Cascading Style
   Sheets">CSS</abbr> font and text properties. There are lots of
   really cool things about CSS properties that are much better than
   the &quot; old way &quot; of doing things with <em>physical
   markup</em> and <strong>images</strong>!</p>
<p id="tag">Using CSS will be even better when commonly used browsers
   support CSS Level 3. </p>
<hr />
</body>
</html>
```

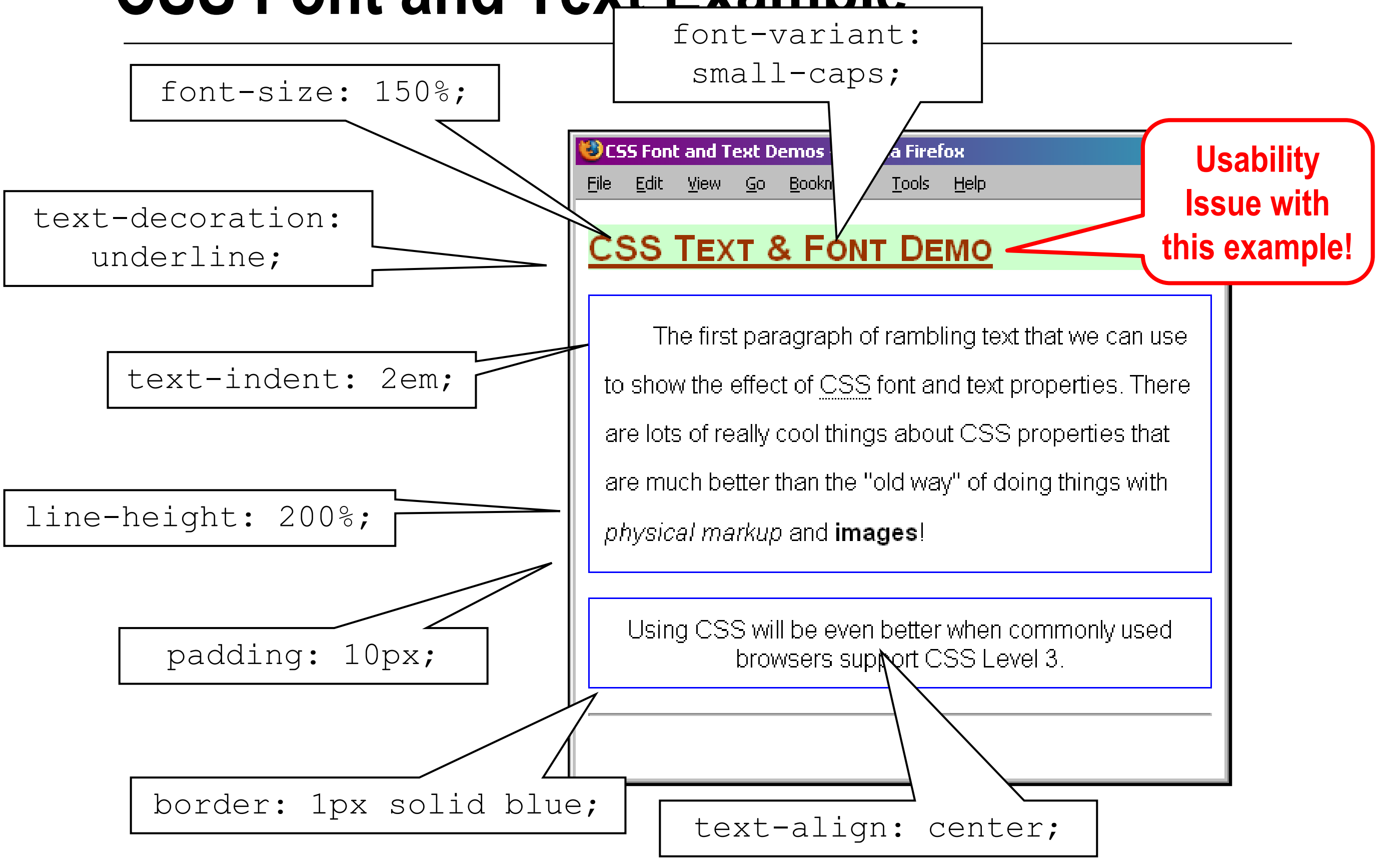


CSS Text & Font Demo

The first paragraph of rambling text that we can use to show the effect of CSS font and text properties. There are lots of really cool things about CSS properties that are much better than the "old way" of doing things with *physical markup* and **images**!

Using CSS will be even better when commonly used browsers support CSS Level 3.

CSS Font and Text Example



CSS Font and Text Demos - Mozilla Firefox

CSS TEXT & FONT DEMO

The first paragraph of rambling text that we can use to show the effect of CSS font and text properties. There are lots of really cool things about CSS properties that are much better than the "old way" of doing things with *physical markup* and **images**!

Using CSS will be even better when commonly used browsers support CSS Level 3.

font-size: 150%;

font-variant: small-caps;

text-decoration: underline;

Usability Issue with this example!

text-indent: 2em;

line-height: 200%;

padding: 10px;

border: 1px solid blue;

text-align: center;

CSS Font and Text Example – Family, Colour, Spacing

grouping selector **h1, p**

```
h1, p { font-family: Arial, Helvetica, sans-serif; }

/* shows the "block" in a background color */
h1 { background-color: #CCFFCC; color: #993300; }
/* percentage of the "normal" text size */
h1 { font-size: 150%; }
/* note that the h1 content is NOT in CAPITALS! Cool!*/
h1 { font-variant: small-caps; }
/* not good - confuses users - they think it's a hyperlink! */
h1 { text-decoration: underline; }

p.intro { line-height: 200%; }
/* "em" units will scale nicely with font size! */
p.intro { text-indent: 2em; }
/* note border values. padding between text and border */
p { border: 1px solid blue; padding: 10px; }

/* only effects the #tag element */
#tag { text-align: center;}
```

element selector **h1**

It would be better if these rules were grouped into one rule.

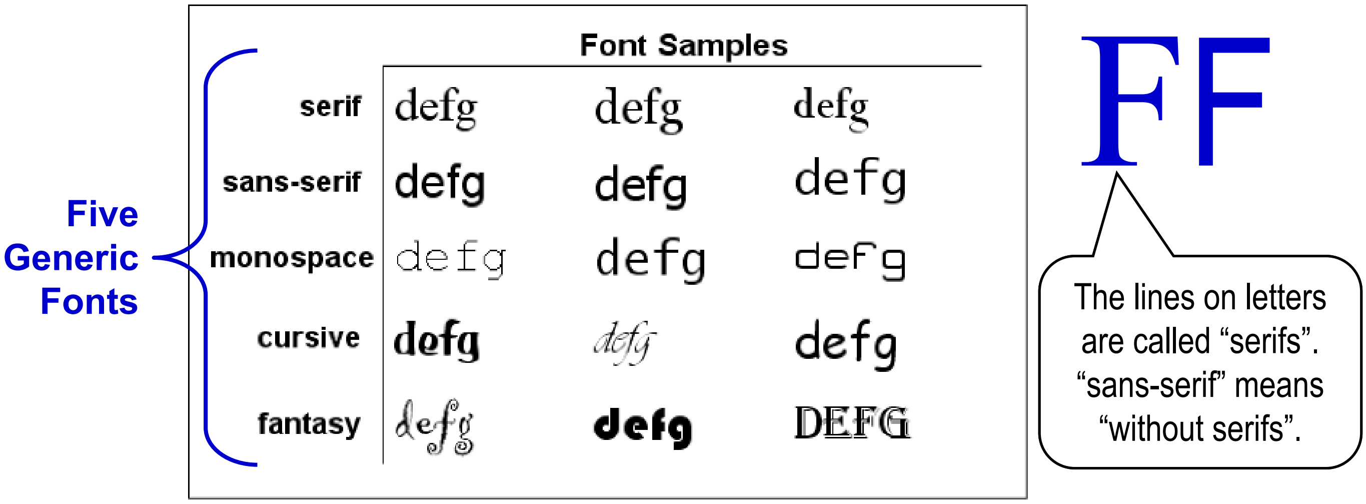
class selector **.intro**

element selector **p**

id selector **#tag**

CSS: Generic & Specific Fonts

- A ***specific font*** is a font such as “Times New Roman”, “Arial”, or “Garamond”. *Specific fonts are installed on a user’s computer, so availability depends on the user’s machine.*
- A ***generic font*** refers to the font’s general appearance such as: “serif”, “sans-serif”, “monospace”, “cursive” or “fantasy”.



The lines on letters are called “serifs”. “sans-serif” means “without serifs”.

CSS Font Family

- To specify the font, we use the `font-family` property. Example:

```
p {
   font-family: Verdana; }
```

Any font names containing characters such as whitespace, font must be quoted. eg. “Times New Roman”

- If you specify a **“specific font”** a user might not have it on their device, so you ***may*** list alternatives, and ***should*** include a final **“generic font”**

```
h1 {
   font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

The preferred specific font

Alternative specific fonts

A “generic font” alternative

Validation Warning if no generic font

- Fonts can also be downloaded using @font

```
@font-face {
   font-family: myfont;
   src : url("http://www.allfont.com/myfont.ttf");}
```

CSS Font Properties

- **font-size:**

  xx-small, x-small, small, **medium**, large, x-large, xx-large, smaller, larger, [length], [%]

- **font-style:**

  normal, italic, oblique

- **font-weight:**

  normal, bold, bolder, lighter, [100,200, **400**, … , 900]

- **font-variant:**

  normal, small-caps

- **font-stretch:**

  normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded

- **line-height:**

  normal, [number], [length], [%]

red = default values

CSS Font Property

- We can write several font-properties in a shorthand, single declaration, format:

  font: [style] [variant] [weight] **size** [/line-height] **family**

  □ **size** and **family** values ***are required***

  □ The values in square brackets [ ] are optional.

  □ The first three values can be specified in any order

  □ /line-height, if used, *must* come straight after size

- Example:

```
p {
    font: italic normal bold 10pt/14pt Helvetica, sans-serif;
}
```

Note: Be aware of ***default*** and ***required*** values for style properties.

Text Shadow

- text-shadow: adds shadow to text

Horizontal Shadow

Vertical Shadow

Blur Radius

Shadow Color

```
h1 {
    text-shadow:  5px      2px      3px      red;
}
```

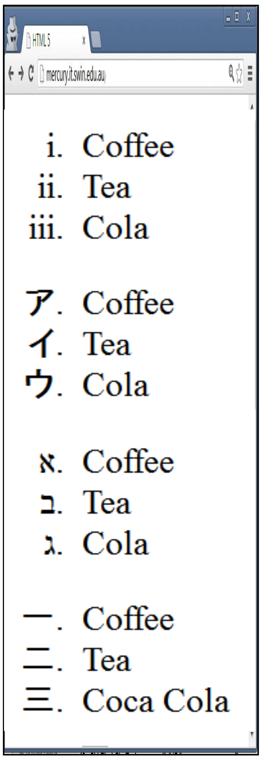
```
h1 {
    text-shadow:  10px      20px      5px      green;
}
```

Topics

- CSS properties define which aspect of the *selected* HTML element will be changed or styled
  - □Size
  - □Color
  - □Typography
  - □List
  - □Display and Alignment

Creating List styles with CSS

- list-style-type : none | **disc** | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower greek | lower-alpha | lower-latin | …
  - □ ul.a {list-style-type:lower-roman;}
  - □ ul.b {list-style-type:katakana ;}
  - □ ul.c {list-style-type:hebrew ;}
  - □ ul.d {list-style-type:cjk-ideographic ;}
- list-style-image: **none** | <url>
  - □ list-style-image : url("spade.gif");
- list-style-position : inside | **outside**;
  - □ list-style-position : inside;
- list-style : [type] [position] [image];
  - □ list-style : lower-alpha inside;



i. Coffee
ii. Tea
iii. Cola

ア. Coffee
イ. Tea
ウ. Cola

א. Coffee
ב. Tea
ג. Cola

一. Coffee
二. Tea
三. Coca Cola

Topics

- CSS properties define which aspect of the *selected* HTML element will be changed or styled
  - □ Size
  - □ Color
  - □ Typography
  - □ List
  - □ Display and Alignment

Display: block, inline, inline-block

- display : **inline** | block | list-item | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | none
  - □ **display: block**
    used to change an inline element to a block level element,
  - □ **display: inline**
    used to change a block level element to an inline element
  - □ **display: table values**
    used to create table-like displays using CSS (HTML tables are only for tabular data)
  - □ **display: none**
    value hides the element from display

Useful for aligning <li> menu items horizontally

e.g. you might want to hide some elements in a small mobile layout

http://css-tricks.com/almanac/properties/d/display/

CSS: Inline Text Alignment, and other properties

- The text-align CSS property describes how **inline** content like text is aligned ***within*** its parent block element. text-align does not control the alignment of block elements, only their **inline** content.
- **text-align** : **left** | right | center | justify;
  - □ text-align : center;
    - □ Justify is not supported by all browser

Default *values* shown in **red**

- **text-indent**: <value>; (indents first line of paragraph) default is **0**
  - □ text-indent : 2em;
  - □ text-indent : -2em; (for hanging indent)
- **line-height** : **normal** | <value>;
  - □ line:-height : 150%; (1.5 spacing assuming font size is normal)
- **text-decoration**: **none**, underline, overline, line-through

'underline' is default for 'a' element

- *Also see CSS3 text-decoration:*
  http://www.w3.org/TR/css-text-decor-3/

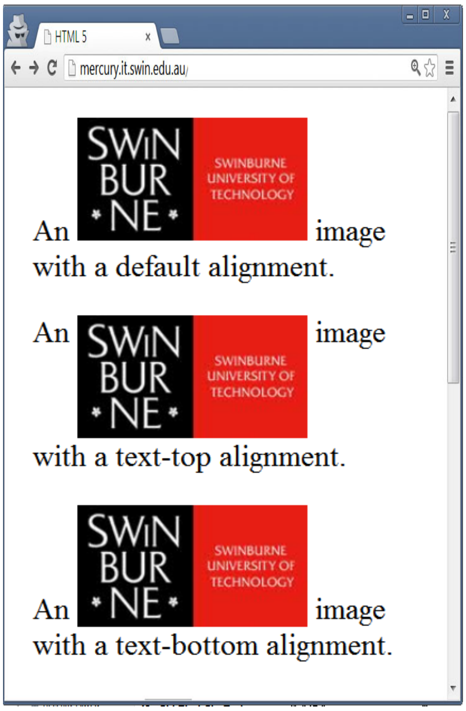
CSS: Inline Alignment of Graphics with Text

```
<p>An <img src="logo.jpg" alt="SUT" width="120"
height="64" />
image with a default alignment.</p>

<p>An <img class="top" src=" logo.jpg"
alt="SUT" width="120" height="64" />
image with a text-top alignment.</p>

<p>An <img class="bottom" src=" logo.jpg"
alt="SUT" width="120" height="64" />
image with a text-bottom alignment.</p>

</body>
```



An image with a default alignment.

An image with a text-top alignment.

An image with a text-bottom alignment.

```
CSS
   img.top     {
         vertical-align:text-top;}

   img.bottom {
         vertical-align:text-bottom;}
```

vertical-align : **baseline** | sub | super | top | text-top | middle | bottom | text-bottom