**Quadrant II – Transcript and Related Materials**

**Programme: B.Sc.**

**Subject: Computer Science**

**Course Code: CSS 104/105/106**

**Course Title: Web Application Development Using FLASK/ASP.NET/DJANGO**

**Unit: III (CSS - Cascading Style Sheets)**

**Module Name: Fundamental text and font styling, values, units, colors, media queries**

**Module No: 12**

**Name of the Presenter: Mrs. Liana da Costa**

---

**Notes**

# Fundamental text and font styling

The CSS properties used to style text generally fall into two categories:

- **Font styles**: Properties that affect the font that is applied to the text, affecting what font is applied, how big it is, whether it is bold, italic, etc.
- **Text layout styles**: Properties that affect the spacing and other layout features of the text, allowing manipulation of, for example, the space between lines and letters, and how the text is aligned within the content box.

## Font and Text layout styling

CSS provides some properties to emphasize the text:

- **font-style**
- **font-weight**
- **text-transform**
- **text-decoration**

**font-style: Used to turn italic text on and off (normal|italic|oblique).**

- **normal:** Sets the text to the normal font (turns existing italics off.)
- **italic:** Sets the text to use the *italic version of the font* if available; if not available, it will simulate italics with oblique instead.
- **oblique:** Sets the text to use a simulated version of an italic font, created by *slanting the normal version*.

**font-weight: Sets how bold the text is**
**(normal|bold|bolder|lighter|100|200|...|800|900).**

- **normal, bold:** Normal and bold font weight
- **lighter, bolder:** Sets the current element's boldness to be one step lighter or heavier than its parent element's boldness.
- **100–900:** Numeric boldness values that provide finer grained control than the above keywords, if needed. The value of 400 is the normal weight; while 700 is bold.

**text-transform: Allows you to set your font to be transformed (none|uppercase|lowercase|capitalize).**

- **none:** Prevents any transformation.
- **uppercase:** Transforms ALL TEXT TO CAPITALS.
- **lowercase:** Transforms all text to lower case.
- **capitalize:** Transforms all words to Have The First Letter Capitalized.
- **full-width:** Transforms all glyphs to be written inside a fixed-width square, like a monospace font, allowing aligning of e.g., Latin characters along with Asian language glyphs (like Chinese, Japanese, Korean).
-

**Example:**


**text-decoration: Sets/un sets text decorations on fonts (none|underline|overline|line-through|blink)**

- **none:** Un sets any text decorations already present.
- **underline:** Underlines the text.
- **overline:** Gives the text an overline.
- **line-through:** Puts a ~~strikethrough over the text~~.

**Note: text-decoration can accept multiple values at once**

## Examples:

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

```css
p {
  text-decoration: underline dashed yellow;
}
```

```css
p {
  text-decoration-line: underline;
  text-decoration-color: yellow;
  text-decoration-style: dashed;
}
```

# Font & Text Layout styling

**Source File**

```html
<html>
  <head>
    <title>Fundamental Text and Font Styling</title>
    <style>
      html {
        font-size: 10px;
      }

      h1 {
        font-size: 50px;
        color: blue;
        text-align: center;
        letter-spacing: 2px;
      }

      h1,p {
        font-weight: bold;
      }

      p::first-line {
        letter-spacing: 4px;
        word-spacing: 4px;
        color: green;
      }

      ol,p {
        font-size: 20px;
        color: red;
        font-family: Helvetica, Arial, sans-serif;
        line-height: 1.6;
        letter-spacing: 1px;
      }
    </style>
  </head>
  <body>
    <h1>Fundamental Text and Font Styling</h1>
    <p>The CSS properties used to style text generally fall into two categories:</p>
    <p>
        <ol>
            <li>Font styles: Properties that affect the font that is applied to the text, affecting what font is applied, how big it is, whether it is bold, italic, etc.</li>
            <li>Text layout styles: Properties that affect the spacing and other layout features of the text, allowing manipulation of, for example, the space between lines and letters, and how the text is aligned within the content box</li>
        </ol>
    </p>

  </body>
</html>
```

**Output**

**Additional Font styles**

- **font-family: *font-name*|*generic-family-name***

  ***A*** *prioritized* list of fonts to be used. The browser will try to use the first font if it is available and goes down the list.

  The ***generic*** *font family* names include: **serif** (with small tails), **sans-serif** (without small tails), **monospace**, **cursive**, **fantasy**. Use monospace for program listing. Use sans-serif for computer display. serif are mainly used in prints (such as "Times" for newspapers and books).
- **font-size: *n*|*n*%|xx-small|x-small|small|medium|large|x-large|xx-large|smaller|larger**
- **font-variant: (normal|small-caps)** Switch between small caps and normal font alternatives. small-caps is smaller than the uppercase.
- **font-kerning:** Switch font kerning options on and off.
- **font-feature-settings:** Switch various OpenType font features on and off.
- **font-variant-alternatives:** Control the use of alternate glyphs for a given font-face.
- **font-variant-caps:** Control the use of alternate capital glyphs.
- **font-variant-east-asian:** Control the usage of alternate glyphs for East Asian scripts, like Japanese and Chinese.
- **font-variant-ligatures:** Control which ligatures, and contextual forms are used in text.
- **font-variant-numeric:** Control the usage of alternate glyphs for numbers, fractions, and ordinal markers.
- **font-variant-position:** Control the usage of alternate glyphs of smaller sizes positioned as superscript or subscript.

- **font-size-adjust:** Adjust the visual size of the font independently of its actual font size.
- **font-stretch:** Switch between possible alternative stretched versions of a given font.
- **text-underline-position:** Specify the position of underlines set using the text-decoration-line property underline value.

**Examples:**

```
p { font: bold 14px/140% Arial, sans-serif; }
```

```
body {
  font: 20px Helvetica;
}
```

```
p {
  font-size: 14px;
  font-weight: bold;
  line-height: 140%;
  font-family: Arial, sans-serif;
}
```

```
font: <font-stretch> <font-style> <font-variant>
```

### Text layout styles

- **text-indent: (*n*|*n*%)** Specify how much horizontal space should be left before the beginning of the first line of the text content. Indent the first-line of the paragraph. To indent all the lines of a paragraph (i.e., the whole block), use padding or margin.
- **text-overflow:** Define how overflowed content that is not displayed is signaled to users.
- **word-break:** Specify whether to break lines within words.
    - **normal** (default) means the text is only broken between words, not inside a word

- **break-all** the browser can break a word (but no hyphens are added)
- **keep-all** suppress soft wrapping. Mostly used for CJK (Chinese/Japanese/Korean) text.

- **word-spacing:** Modifies the spacing between each word. You can use the normal keyword, to reset inherited values, or use a length value

**Examples:**

```
p {
  word-spacing: 2px;
}
```

```
span {
  word-spacing: -0.2em;
}
```

```
p {
  text-indent: -10px;
}
```

- **line-break:** Relax or strengthen line breaking for Asian languages.
- **text-align-last: (left|right|center|justify)** Define how the last line of a block or a line, right before a forced line break, is aligned. By default, the last line of a paragraph is aligned following the text-align value.
- **text-align: (start, end, left, right, center, justify)**
- **vertical-align: (baseline, sub, super, top , text-top, middle, bottom , text-bottom)** Determines how inline elements are vertically aligned.
- **overflow-wrap:** Specify whether the browser may break lines within words in order to prevent overflow. Default **(overflow-wrap: normal)**

**Examples:**

```
p {
  overflow-wrap: break-word;
}
```

```
p {
  overflow-wrap: anywhere;
}
```

```
p {
  text-align-last: right;
}
```

```
p {
  text-align: right;
}
```

- **text-orientation:** Define the orientation of the text in a line.
- Valid values are
    - **mixed** is the default, and if a language is vertical (like Japanese) it preserves that orientation, while rotating text written in western languages
    - **upright** makes all text be vertically oriented
    - **sideways** makes all text horizontally oriented
- **writing-mode:** Define whether lines of text are laid out horizontally or vertically and the direction in which subsequent lines flow.
    - **horizontal-tb** (default)
    - **vertical-rl** content is laid out vertically. New lines are put on the left of the previous
    - **vertical-lr** content is laid out vertically. New lines are put on the right of the previous
- **direction:** Sets the direction of the text. Valid values are **ltr** and **rtl.**

  (This depends on the language )

**Example:**

```
p {
  direction: rtl;
}
```

- **text-shadow:** Apply a shadow to the text. By default, the text has no shadow. This property accepts an optional color, and a set of values that set

- the X offset of the shadow from the text
- the Y offset of the shadow from the text
- the blur radius
- If the color is not specified, the shadow will use the text color.

- **tab-size:** Sets the width of the tab character.
- **hyphens:** Switch on and off hyphenation for supported languages.

## Examples:

```
p {
  text-shadow: 0.2px 2px;
}
```

```
p {
  tab-size: 2;
}
```

```
span {
  tab-size: 4px;
}
```

```
span {
  text-shadow: yellow 0.2px 2px 3px;
}
```

- **letter-spacing:** **normal|*n*** ***word-spacing:*** ***normal|n***

  ***Additional spacing to be applied to letters or words.*** Modifies the spacing between each letter.

- **white-space:** **normal|pre|nowrap**

  **Specify how** white spaces inside the element is to be handled. For "pre" (pre-formatted), preserve the white-spaces.

- **line-height:** **normal|*n*|*n*%|*factor***

  *Set the height of the line. The factor* gives the factor to be multiplied by the current font-size. E.g., factor of 1.5 means 1.5 times of the current font.

**Examples:**

```
p {
  line-height: 0.9rem;
}
```

```
p {
  letter-spacing: 0.2px;
}
```

```
span {
  letter-spacing: -0.2em;
}
```

**CSS values**

**CSS values** are set against CSS Properties and reside within CSS declaration block, which is a part of the CSS rule / statement.

**CSS** 2.1 allows following types of values :

- **Integers and real numbers**
- **Lengths**
- **Percentages**
- **URLs and URIs**
- **Counters**
- **Colors**
- **Strings**
- **Unsupported Values**

# Integers and real numbers

- they can be preceded by + or - sign.
- Decimal numbers are also allowed.

**Used in following CSS properties**

- z-index
- line-height

- counter-increment
- can be used as CSS color values.

## Lengths

- can be used to specify horizontal or vertical measurements.
- <length> is followed immediately by a unit.
- there should not be any space between the number and the unit.
- for certain properties, negative lengths are allowed , but for certain properties using a negative value results in an error.

**Used in following CSS properties**

- Width
- Margin
- Padding
- font-size
- border-width
- text-shadow etc.

## Units

Two types of units are used in CSS

- **Relative**
- **Absolute**.

**Relative units** are relative to another length. If used, they can generate a flexible output across various media ( e.g., screen, print ).

**Absolute units** are used to specify a fixed width irrespective of viewing area. Useful when the length of the output medium is known.

## Relative units used in CSS

| Unit | Sign used | Description |
|---|---|---|
| Pixel | px | Pixel of the viewing device. |
| em | em | The relevant font size. |
| ex | ex | The 'x-height' of the relevant font. |

## Absolute units used in CSS

| Unit | Sign used | Description |
|---|---|---|
| Inches | in | 1 inch is equal to 2.54 centimeters and approximately 96 pixels. |
| centimeters | cm | Known measurement unit.100 cm = 1 meter. |
| Millimeters | mm | Known measurement unit.1000 mm = 1 meter |
| Points | pt | 1 pt = 1/72 nd pf an inch. |
| Picas | pc | 1 pica = 12 points. |

## Percentages

**Percentages** can be used by specifying a number followed by a "**%**".

**Used in following CSS properties**

- width
- margin-top
- font-size etc.

## URLs and URIs

- **URLs and URIs** are used as CSS values to point to a resource ( e.g., an image ).
- A UR**I** is an **identifier** of a specific resource. Like a page, or book, or a document.
- A UR**L** is special type of identifier **that also tells you how to access it**, such as https, ftp, etc https://www.google.com.
- If the protocol (https, ftp, etc.) is either present or implied for a domain, **you should call it a URL**—even though it's also a URI.
- You can also use Relative **URIs**

**Used in following CSS properties**

- background-image
- cursor
- list-style etc.

## Counters

- **Counters** are denoted by case-sensitive identifiers.

- **CSS counters** let you adjust the appearance of content based on its location in a document.
    - For example, you can use counters to automatically number the headings in a webpage.
- Counters are, in essence, variables maintained by CSS whose values may be incremented by CSS rules to track how many times they're used.

**Used in**

- Can be used with most of the ***element selectors*** and ***pseudo selectors.***

## Manipulating a counter's value

- To use a CSS counter, it must first be initialized to a value with the counter-reset property (0 by default).
- The same property can also be used to change its value to any specific number.
- Once initialized, a counter's value can be increased or decreased with counter-increment.

## Displaying a counter

- The value of a counter can be displayed using either the counter() or counters() function in a content property.
- The counter() function has two forms:
    - 'counter(name)'

- 'counter(name, style)'.

- The counters() function also has two forms:
  - 'counters(name, string)'
  - 'counters(name, string, style)'.

**CSS Code**

```
body {
  counter-reset: section;
}

h3::before {
  counter-increment: section;
  content: "Section " counter(section) ": ";


}
```

**HTML Code**

```
<h3>Introduction</h3>
<h3>Body</h3>
<h3>Conclusion</h3>
```

**Output**

**Section 1: Introduction**

**Section 2: Body**

**Section 3: Conclusion**

## Colors

- A **color** is either a keyword or a numerical RGB specification or a hexadecimal color value**.**
- It is used to set the color of the text written within an HTML document.

**Used in**

- Color
- Background
- border
- box-shadow  etc.

Most of the CSS properties accept color as values.

## Units

| Keywords | RGB | Hexa decimal |
|---|---|---|
| aqua | 0,25,255 | #00ffff |
| black | 0,0,0 | #000000 |
| blue | 0,0,255 | #0000ff |
| fuchsia | 255,0,255 | #ff00ff |
| gray | 128,128,128 | #808080 |
| green | 0,128,0 | #008000 |
| lime | 0,255,0 | #00ff00 |
| maroon | 128,0,0 | #800000 |
| navy | 0,0,128 | #000080 |

| Keywords | RGB | Hexa decimal |
|---|---|---|
| olive | 128,128,0 | #808000 |
| orange | 255,165,0 | #ffA500 |
| purple | 128,0,128 | #800080 |
| red | 255,0,0 | #ff0000 |
| silver | 192,192,192 | #c0c0c0 |
| teal | 0,128,128 | #008080 |
| white | 255,255,255 | #ffffff |
| yellow | 255,255,0 | #ffff00 |

## Strings

### Strings

- The format of writing strings may be either with double quotes or with single quotes.

**Used in**

- Most of the CSS properties accept strings as values.

## Unsupported Values

### Unsupported

- If a value is not supported by a user agent, it ignores that value as an illegal value.

# Media Queries

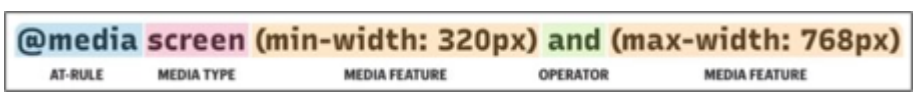
- enable you to format your documents to be presented correctly on different size of output devices
- allow you to customize the presentation of your web pages for a specific range of devices like cell phones, tablets, desktops, etc. without any change in markups.
- a popular technique for delivering a tailored style sheet (responsive web design) to desktops, laptops, tablets, and cell phones.
- used to specify that certain styles are only for printed documents or for screen readers (media type: print, screen, or speech).

## Anatomy of a Media Query

- A media query consists of a media type and zero or more expressions that match the type and conditions of a particular media feature such as device width or screen resolution.

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {
  CSS-Code;
}
```

Example:



Media queries can be used to check

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- Resolution

## Media Query

- **CSS @media Rule**

  The @media rule is used in media queries to apply different styles for different media types/devices.

- **Media type**: print, screen, or speech.
- **Media features** provide more specific details to media queries, by allowing to test for a specific feature of the user agent or display device.

  For example, you can apply styles to only those screens that are greater, or smaller, than a certain width.

## Media Types

| Value | Description |
|---|---|
| all | Used for all media type devices |
| print | Used for printers |
| screen | Used for computer screens, tablets, smart-phones etc. |
| speech | Used for screen readers that "reads" the page out loud |

# Media Features

| Value | Description |
|---|---|
| inverted-colors | Is the browser or underlying OS inverting colors? (added in Media Queries Level 4) |
| light-level | Current ambient light level (added in Media Queries Level 4) |
| max-aspect-ratio | The maximum ratio between the width and the height of the display area |
| max-color | The maximum number of bits per color component for the output device |
| max-color-index | The maximum number of colors the device can display |
| max-height | The maximum height of the display area, such as a browser window |
| max-monochrome | The maximum number of bits per "color" on a monochrome (greyscale) device |
| max-resolution | The maximum resolution of the device, using dpi or dpcm |
| max-width | The maximum width of the display area, such as a browser window |

| Value | Description |
|---|---|
| any-hover | Does any available input mechanism allow the user to hover over elements? (added in Media Queries Level 4) |
| any-pointer | Is any available input mechanism a pointing device, and if so, how accurate is it? (added in Media Queries Level 4) |
| aspect-ratio | The ratio between the width and the height of the viewport |
| color | The number of bits per color component for the output device |
| color-gamut | The approximate range of colors that are supported by the user agent and output device (added in Media Queries Level 4) |
| color-index | The number of colors the device can display |
| grid | Whether the device is a grid or bitmap |
| height | The viewport height |
| hover | Does the primary input mechanism allow the user to hover over elements? (added in Media Queries Level 4) |

| Value | Description |
|---|---|
| overflow-block | How does the output device handle content that overflows the viewport along the block axis (added in Media Queries Level 4) |
| overflow-inline | Can content that overflows the viewport along the inline axis be scrolled (added in Media Queries Level 4) |
| pointer | Is the primary input mechanism a pointing device, and if so, how accurate is it? (added in Media Queries Level 4) |
| resolution | The resolution of the output device, using dpi or dpcm |
| scan | The scanning process of the output device |
| scripting | Is scripting (e.g., JavaScript) available? (added in Media Queries Level 4) |
| update | How quickly can the output device modify the appearance of the content (added in Media Queries Level 4) |
| width | The viewport width |

| Value | Description |
|---|---|
| min-aspect-ratio | The minimum ratio between the width and the height of the display area |
| min-color | The minimum number of bits per color component for the output device |
| min-color-index | The minimum number of colors the device can display |
| min-height | The minimum height of the display area, such as a browser window |
| min-monochrome | The minimum number of bits per "color" on a monochrome (greyscale) device |
| min-resolution | The minimum resolution of the device, using dpi or dpcm |
| min-width | The minimum width of the display area, such as a browser window |
| monochrome | The number of bits per "color" on a monochrome (greyscale) device |
| orientation | The orientation of the viewport (landscape or portrait mode) |

# Media Queries

## Source File

```
<html>
<head>
<style>
body {
  background-color: red;
}
@media screen and (min-width: 480px) {
  body {
    background-color: blue;
  }
}
</style>
</head>
<body>
<h1>Resize the browser window to see the effect!</h1>
<p>The media query will only apply if the media type is screen and the viewport is 480px wide or
wider.</p>
</body>
</html>
```

## Output