# **Introduction to CSS**

# **Class: BSCS 6A/6B and BSSE 5B**

# **Instructor: Syda Fatima**

# Cascading style sheets

* All web pages can be broken down into bucketed content areas
* These areas can be updated by changing the code on every page/ By using cascading style sheets!
* Created by Hakon Lie of MIT in 1994
* Has become the W3C standard for controlling visual presentation of web pages
* Cascading style-sheets are powerful mechanism to add style to web document
* Enforce standards and uniformity
* Create dynamic effects
* Works by allowing you to specify rules

# Advantages of style sheets

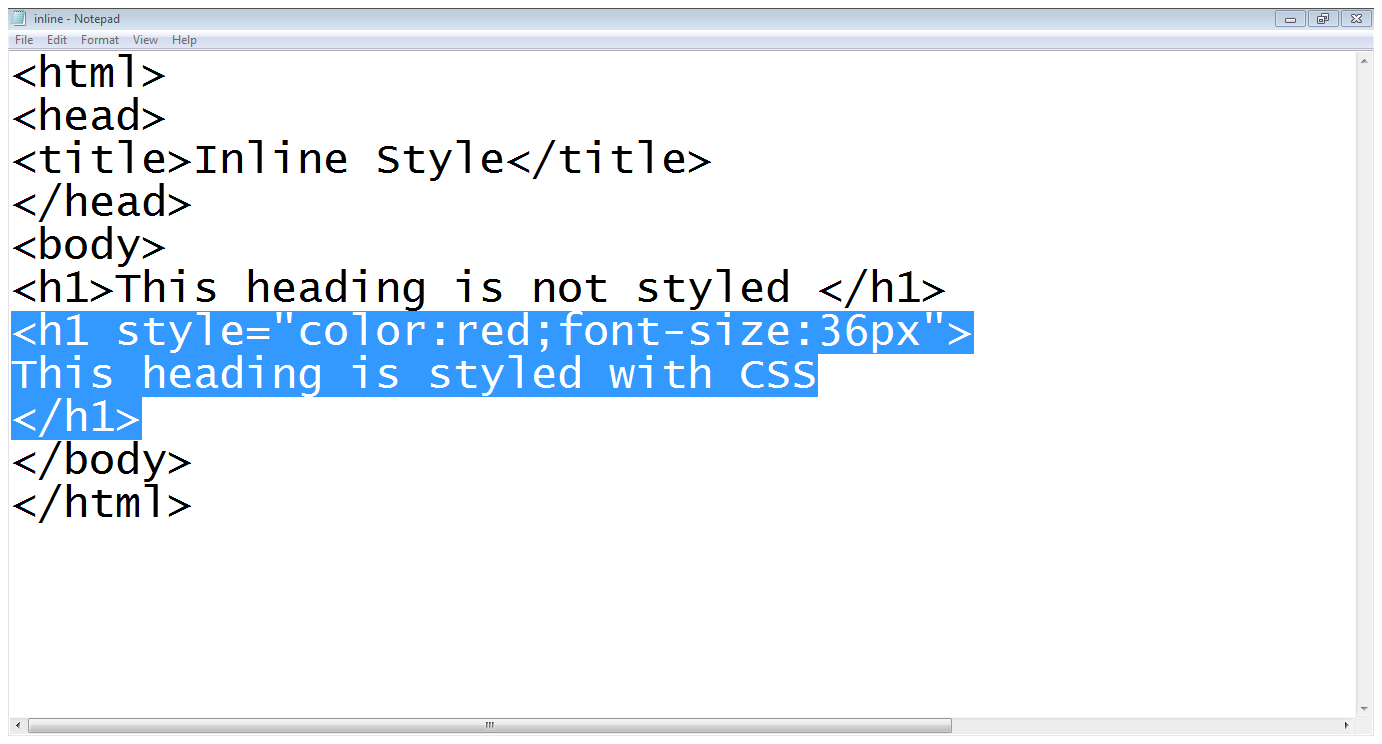
* Saves time
* Easy to change
* Keep consistency
* Give you more control over layout
* Use styles with JavaScript
* Make it easy to create a common format for all the Web pages

# Writing style sheets

* In-line styles
* Embedded/internal styles
* External style sheet

# Inline style sheets

* Inline styles
  + Add styles to each tag within the HTML file
  + Use it when you need to format just a single section in a web page
  + Style attribute is used to add style
* Example
  + <h1 style=“color:red; font-family: sans-sarif” > IU </h1>

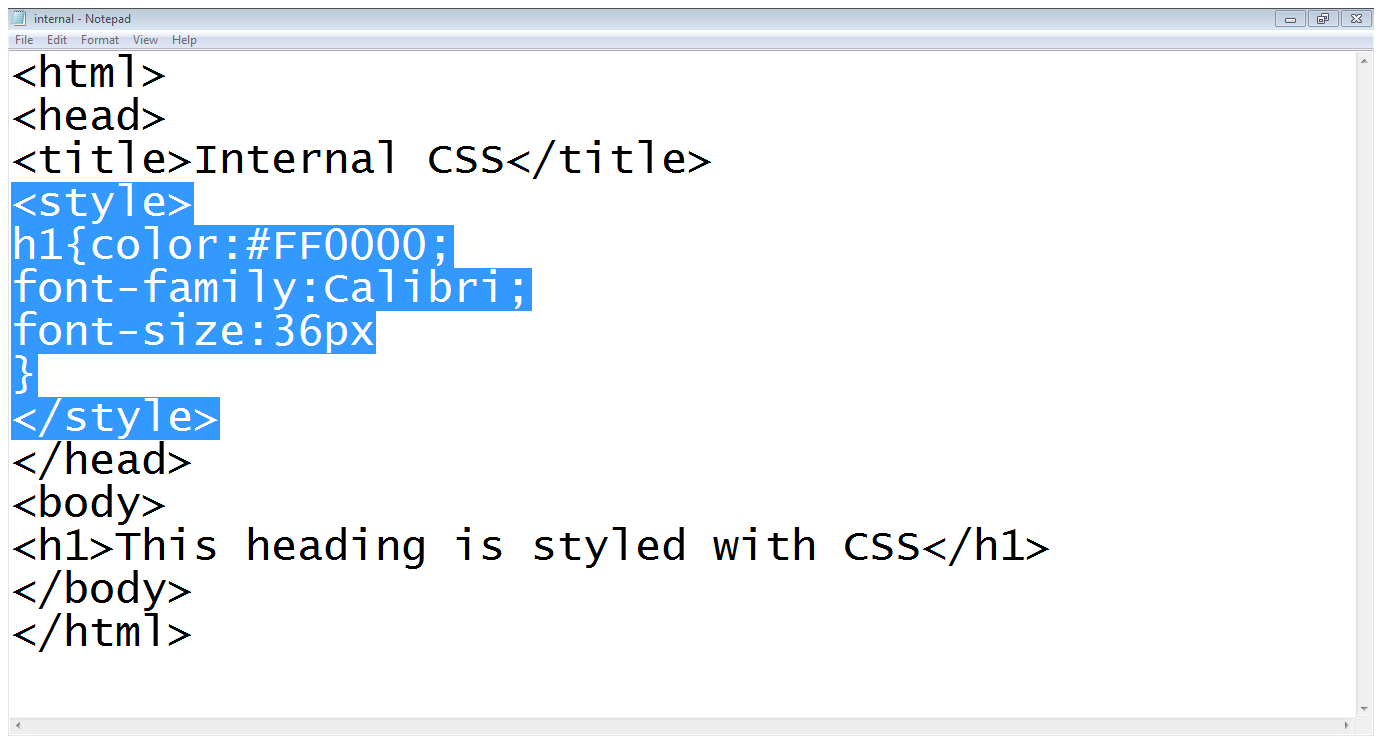


# Internal style

* + A style is applied to the entire HTML file
  + Use it when you need to modify all instances of particular element (e.g., h1) in a web page
* Example
* <style>

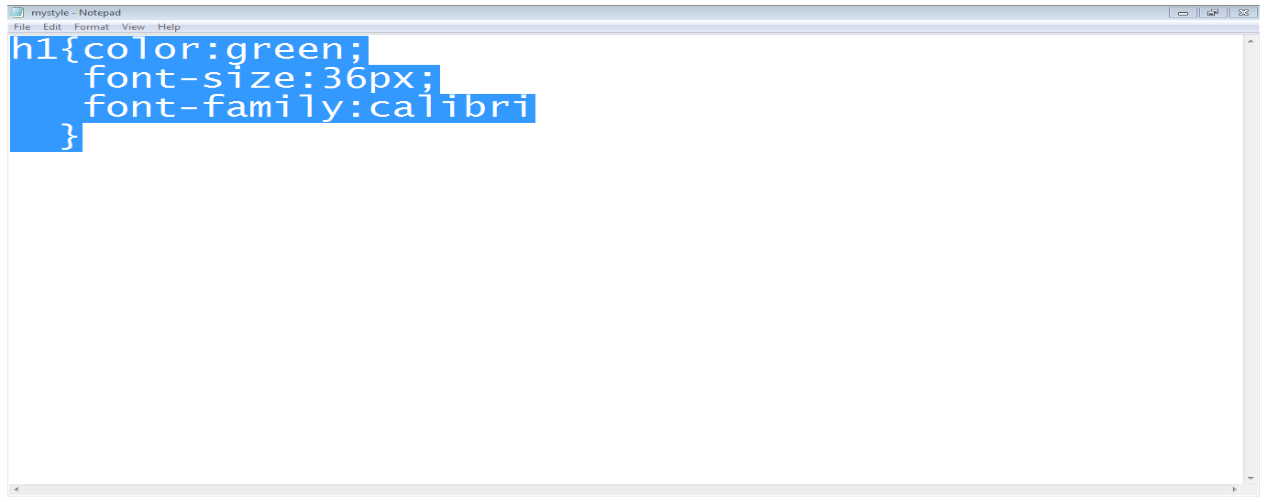
h1 {color:red; font-family:sans-serif}

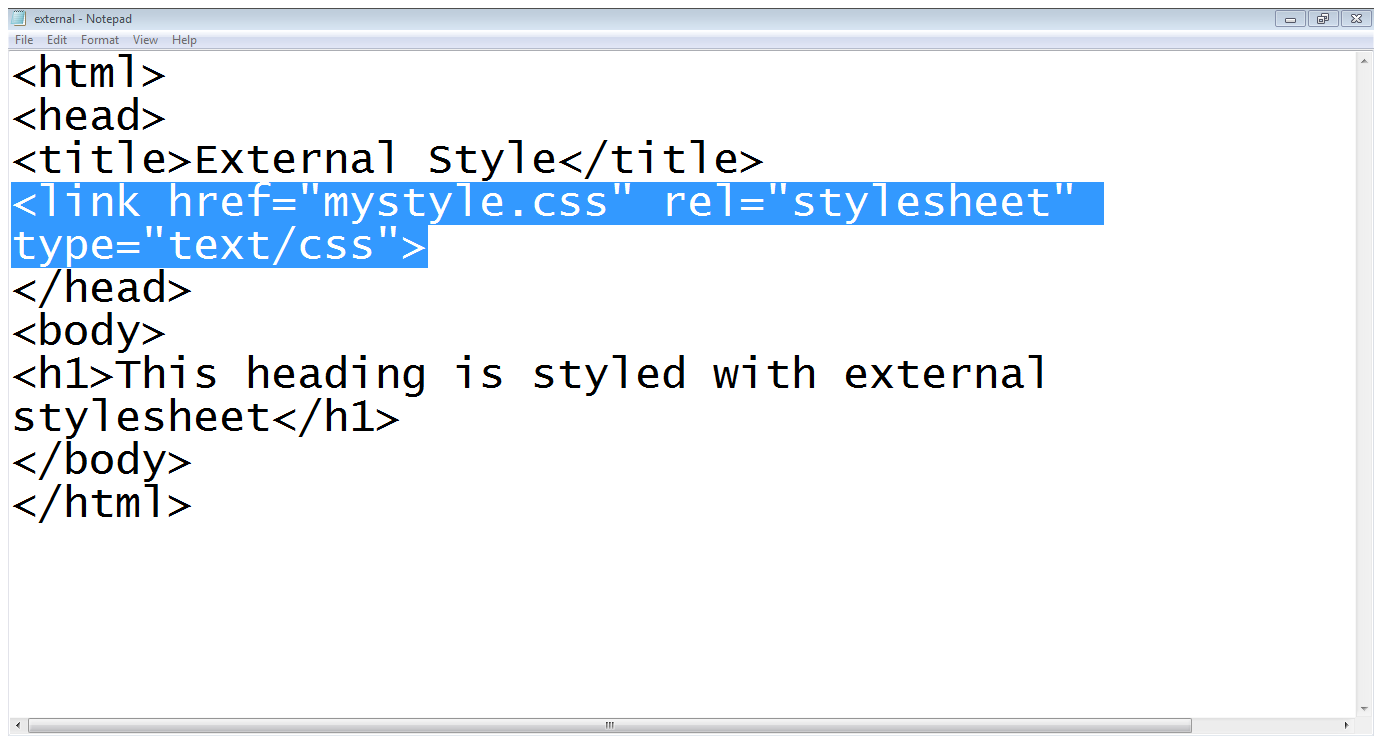
* </style>



# External style

* An external style sheet is a text file containing the style definition (declaration)
* Use it when you need to control the style for an entire web site
* Open a new blank document in Notepad
* Type style declarations
  + h1 {color:red; font-family:calibri;}
* Do not include <style> tags
* Save the document as filename.css
* Open an HTML file
* Between <head> and </head> add
  + <link href=URL rel=“relation\_type” type=“link\_type”>
    - URL is the file.css
    - Relation\_type=“stylesheet”
    - Link\_type=“text/css”
* Save this file and the .css file in the same web server directory



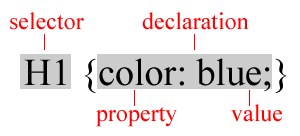


# Style sheet strategies

* Wherever possible, place your styles in external style sheets
* At the top level of your web site: define a default global.css style sheet
* Refine styles at sublevels with a section.css style sheet
* Try to avoid using styles in tags

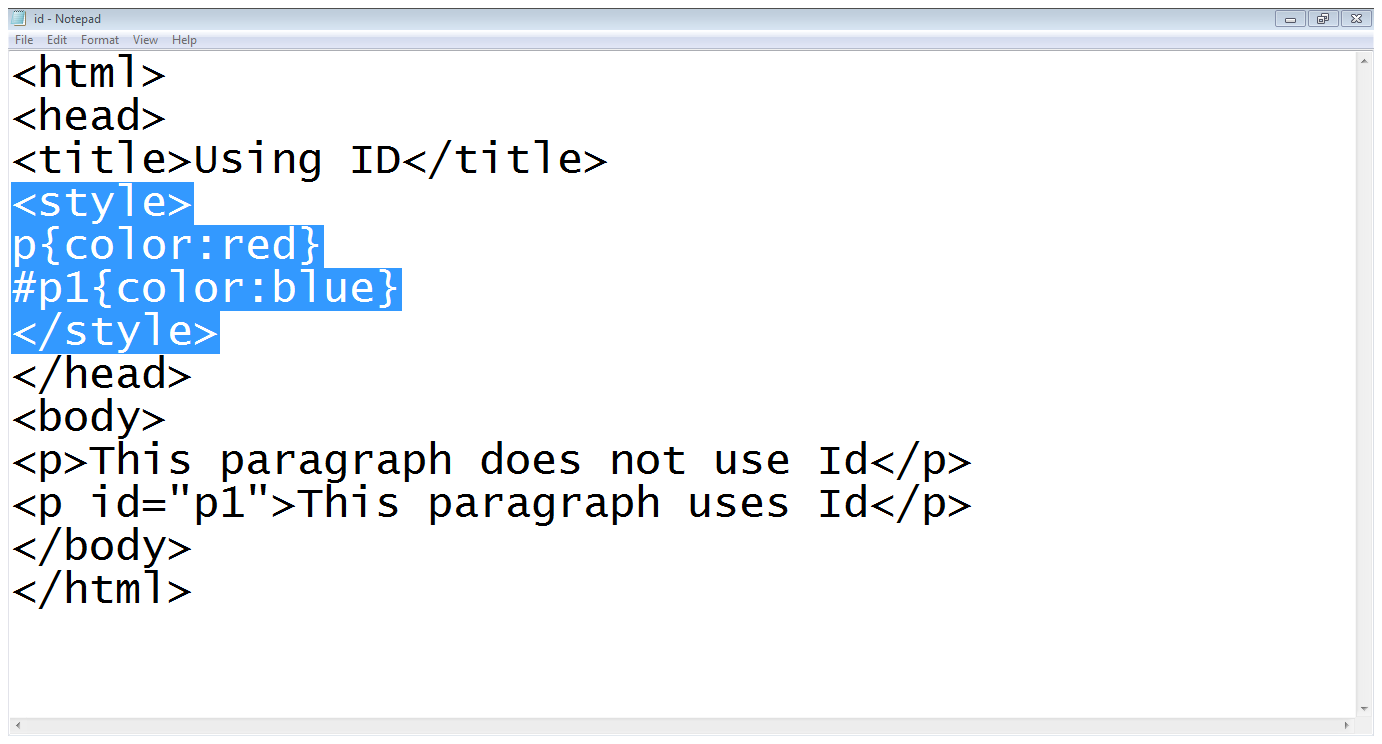
# Defining CSS rules

* Wherever possible, place your styles in external style sheets
* At the top level of your web site: define a default global.css style sheet
* Refine styles at sublevels with a section.css style sheet
* Try to avoid using styles in tags
* A rule consists of
  + A selector: element or elements the declaration applies to
  + Declaration: how the elements referred to in the selector should be styled
    - property: which is the property of the selected element
    - value: which is a specification for this property



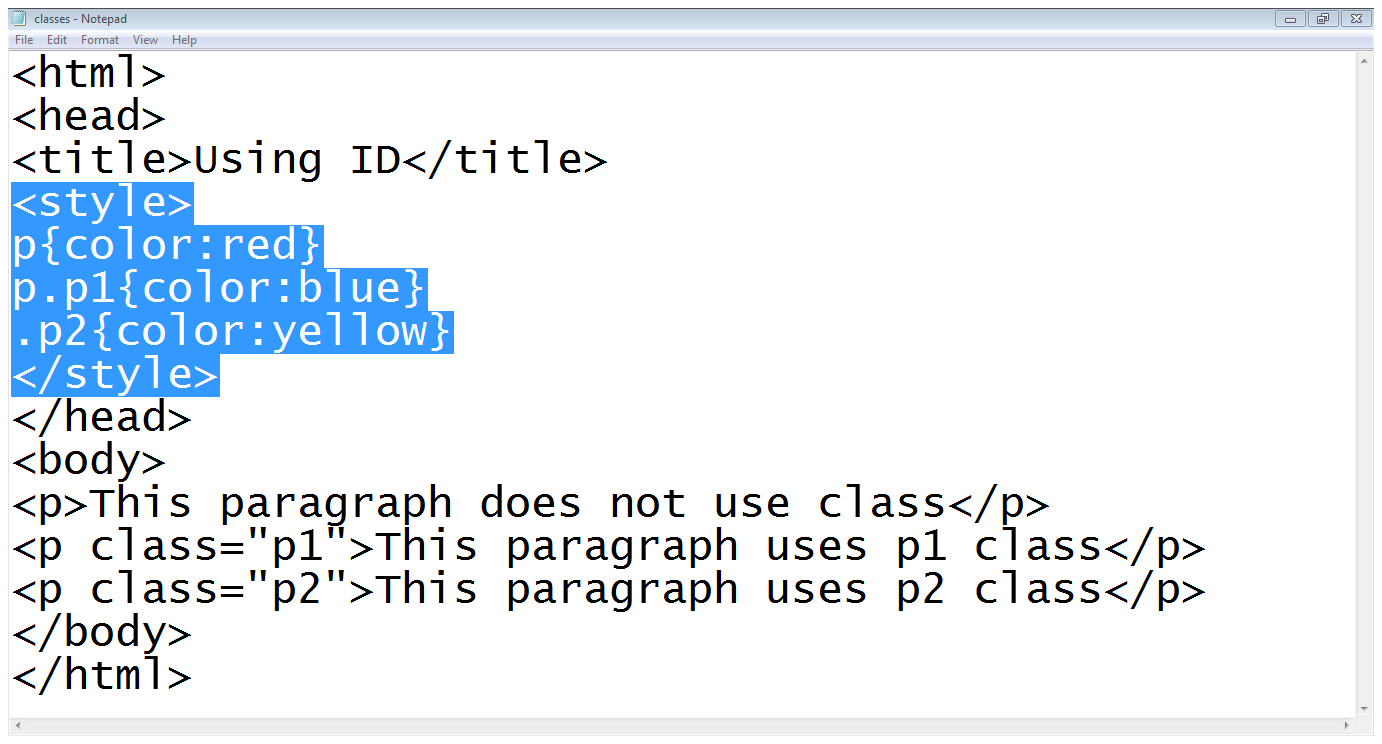
# Using ID’s

* Use an id to distinguish something, like a paragraph, from the others in a document
* The id selector is used to specify a style for a single, unique element
* Create a style Id:
  + #id\_name {style attributes and values}
* Use a style Id:
  + <tag ID=id\_name>



# Using classes

* HTML and XHTML require each id be unique– therefore an id value can only be used once in a document
* You can mark a group of elements with a common identifier using the class attribute
* To create a class
  + tag.class\_name {style attributes} or
  + .class\_name {style attributes}
* To apply a style
  + <tag CLASS=class\_name>
    - <h1 CLASS=FirstHeader>IU</h1>



# Difference between Id’s and classes

* you can’t have more than one tag with the same ID value
* You can apply the same Class value to multiple document tags

Classes or Id?

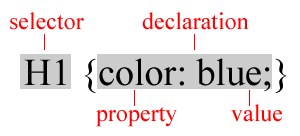
* use ID's for any elements that are simply used once on a page

OR

* only use classes to style websites, but, when you have to use an element in JavaScript, use an identifier

# CSS properties

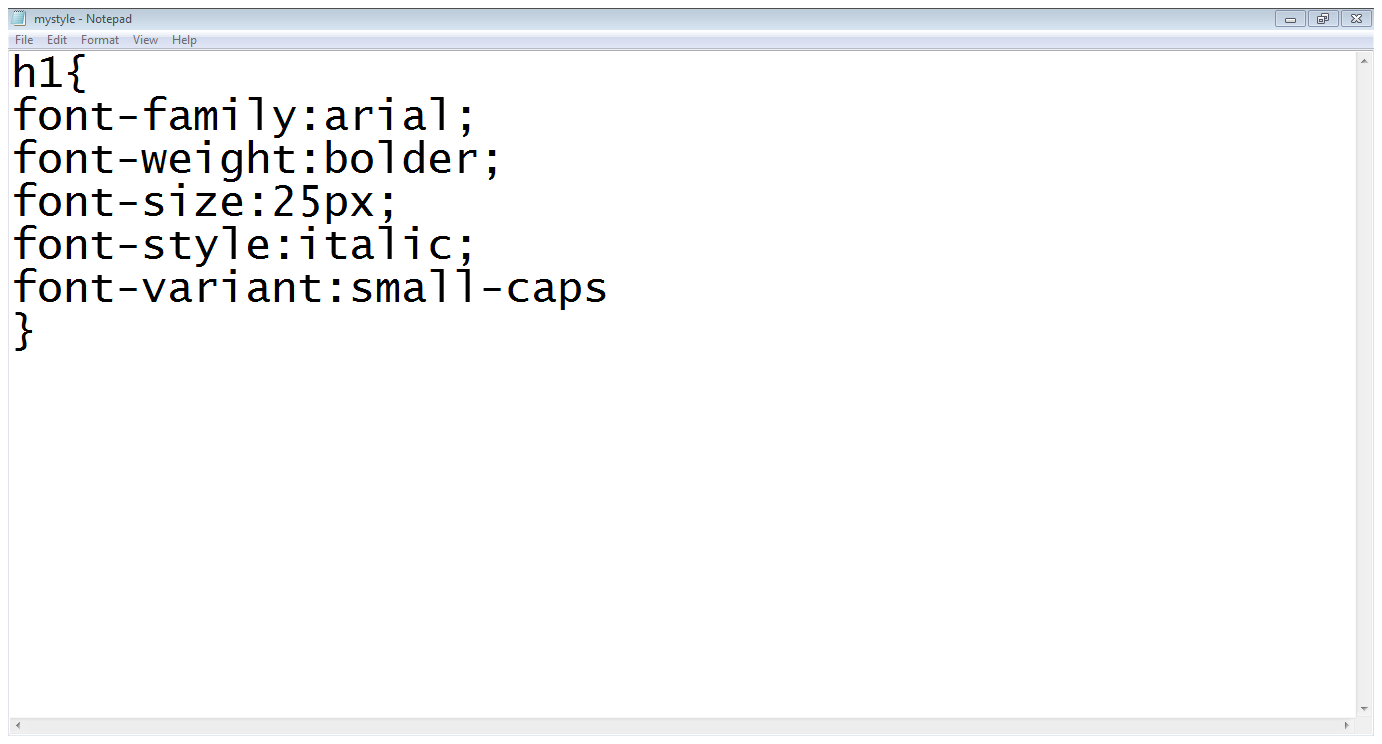
* CSS works by allowing you to associate rules with the elements that appear in a web page
* These rules govern how the content of those elements should be rendered
* A rule consists of
  + A selector: element or elements the declaration applies to
  + Declaration: how the elements referred to in the selector should be styled
    - property: which is the property of the selected element
    - value: which is a specification for this property

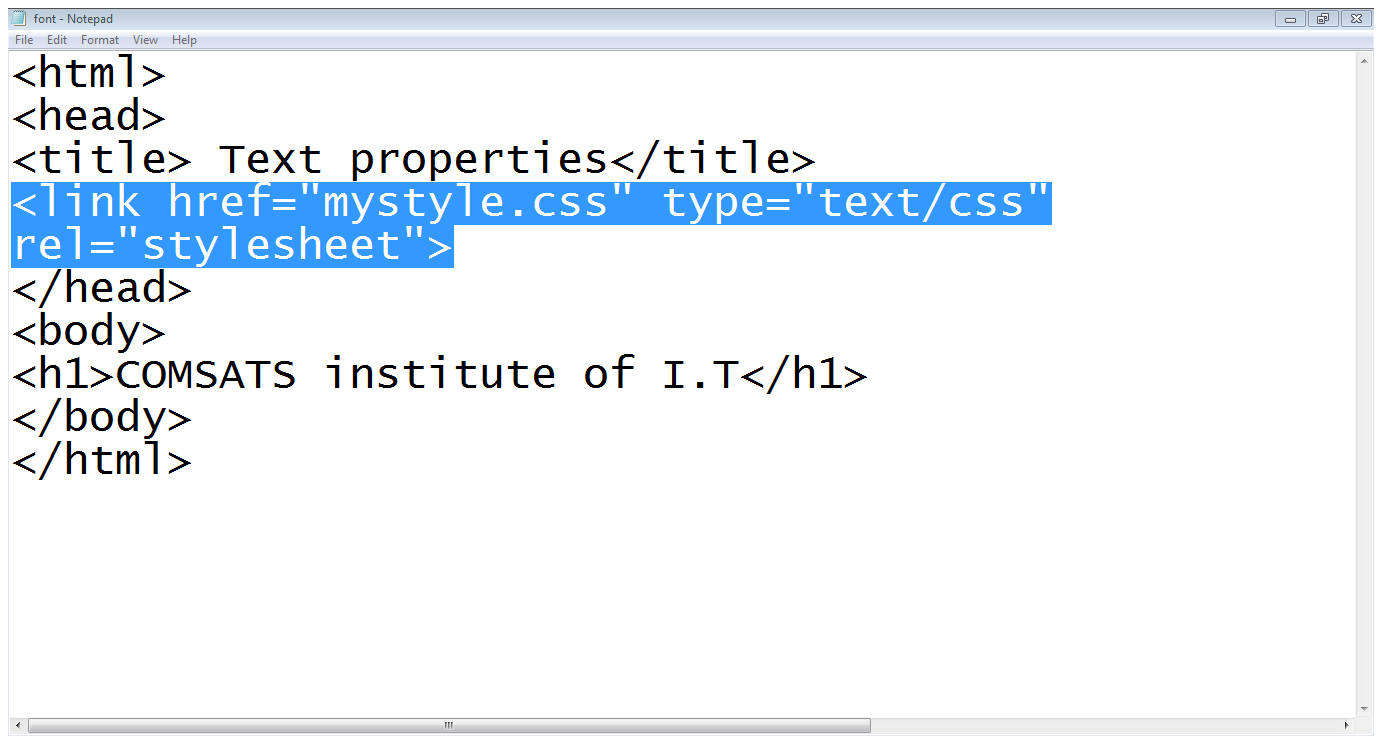


* To control the presentation of an element, you need to know the corresponding properties
* Properties with related functionalities are grouped together
  + Example:
    - Properties to control the presentation of text

# Font properties

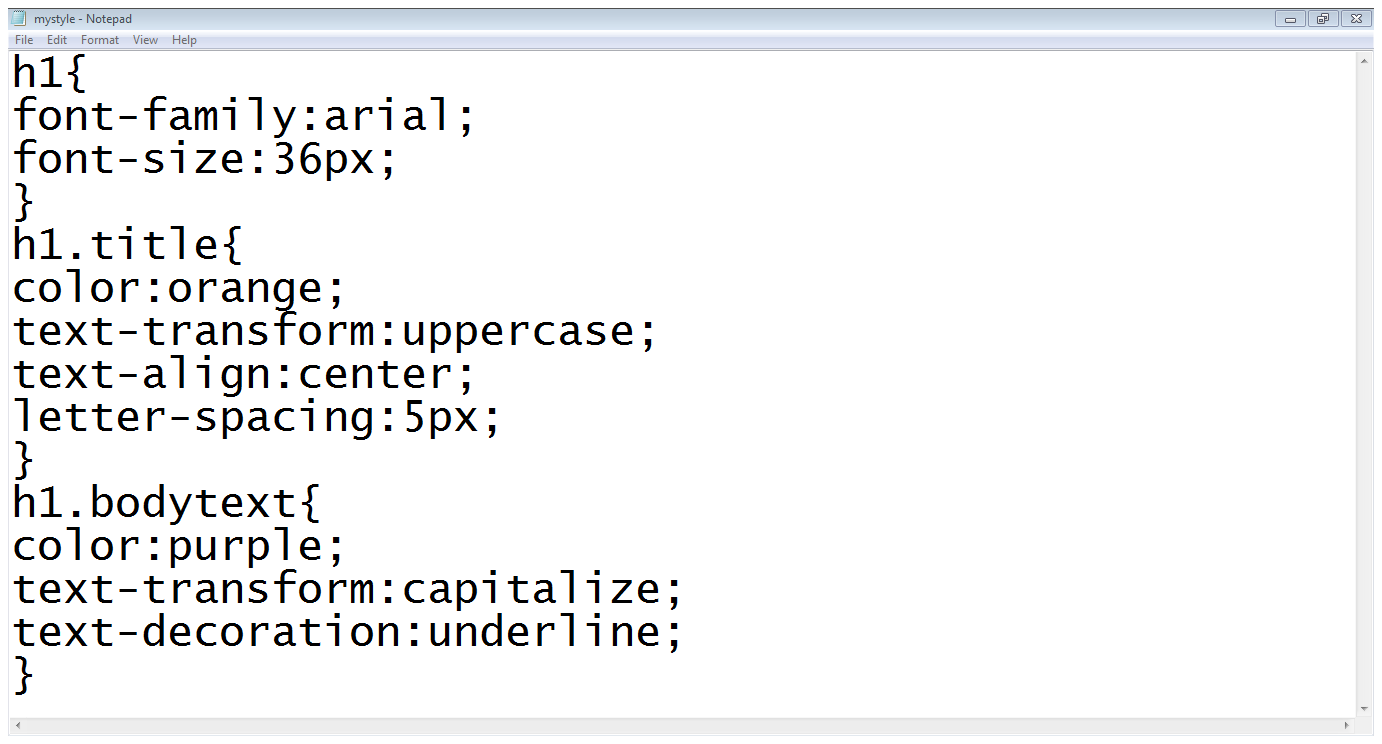
* Several properties allow you to control the appearance of text in your documents
* These properties directly affect the font and its appearance
* Common font properties:
  + font-family
  + font-size
  + font-weight
  + font-style
  + font-variant
* font-family:
  + Specifies the typeface or family of font that should be used
* Common values:
  + Arial, Courier/Courier New, Georgia, Times/Times New Roman and Verdana
* Example:
  + H1{font-family:arial}
* font-size:
  + Specifies the size of a font
* Common values:
  + In pixels (12px,20px etc.)
  + Absolut size (small, medium, large, x-large etc.)
* Example:
  + H1{font-size:20px}
* font-weight:
  + Specifies whether the font should be bold or normal
* Common values:
  + Normal, bold, bolder, lighter
  + 100, 200, 300, 400, 500, 600, 700, 800, 900
  + Example:
  + H1{font-weight:bold}
* font-style:
  + Specifies whether the font should be normal, italic or oblique
* Common values:
  + Normal, italic, oblique
  + Example:
  + H1{font-style:italic}
* font-variant:
  + Specifies whether the font should be normal or small-caps (smaller version of upper case)
* Common values:
  + Normal, small-caps
  + Example:
  + H1{font-variant:small-caps}

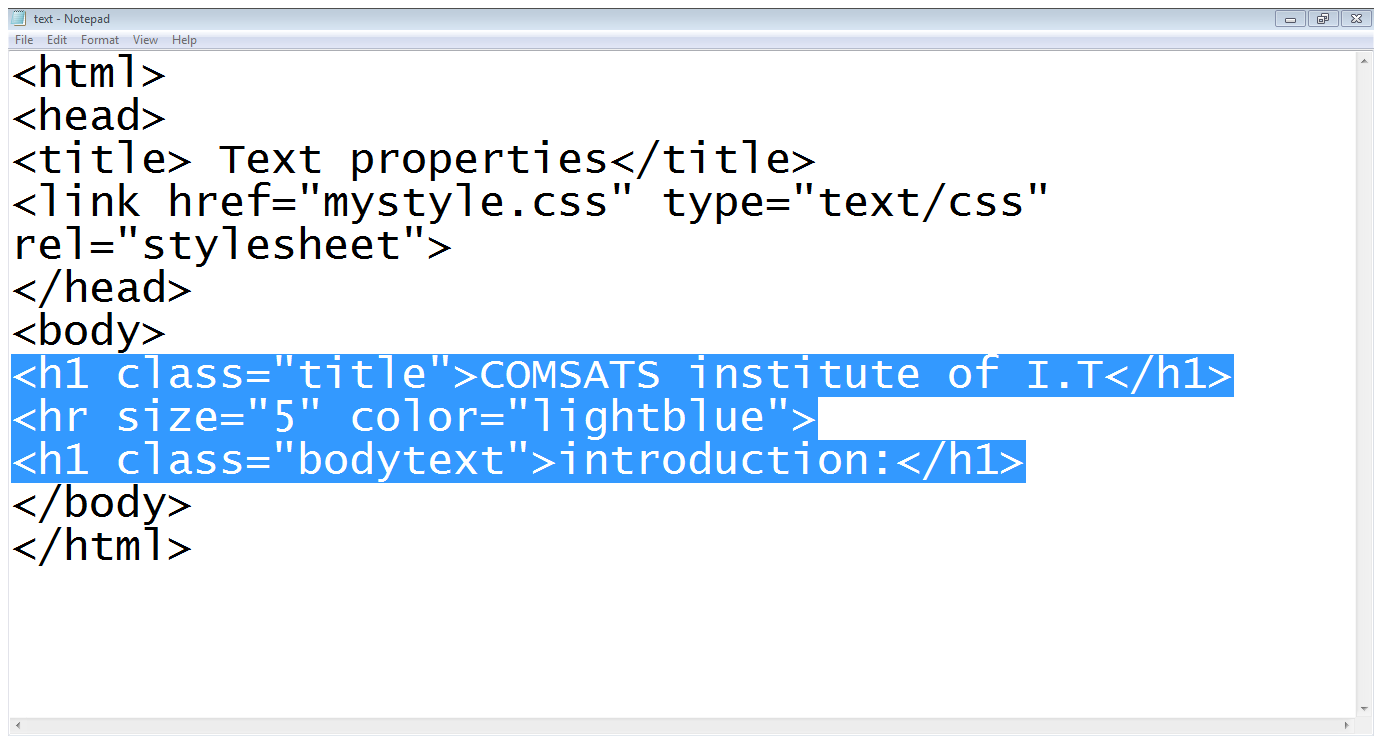




# Text properties

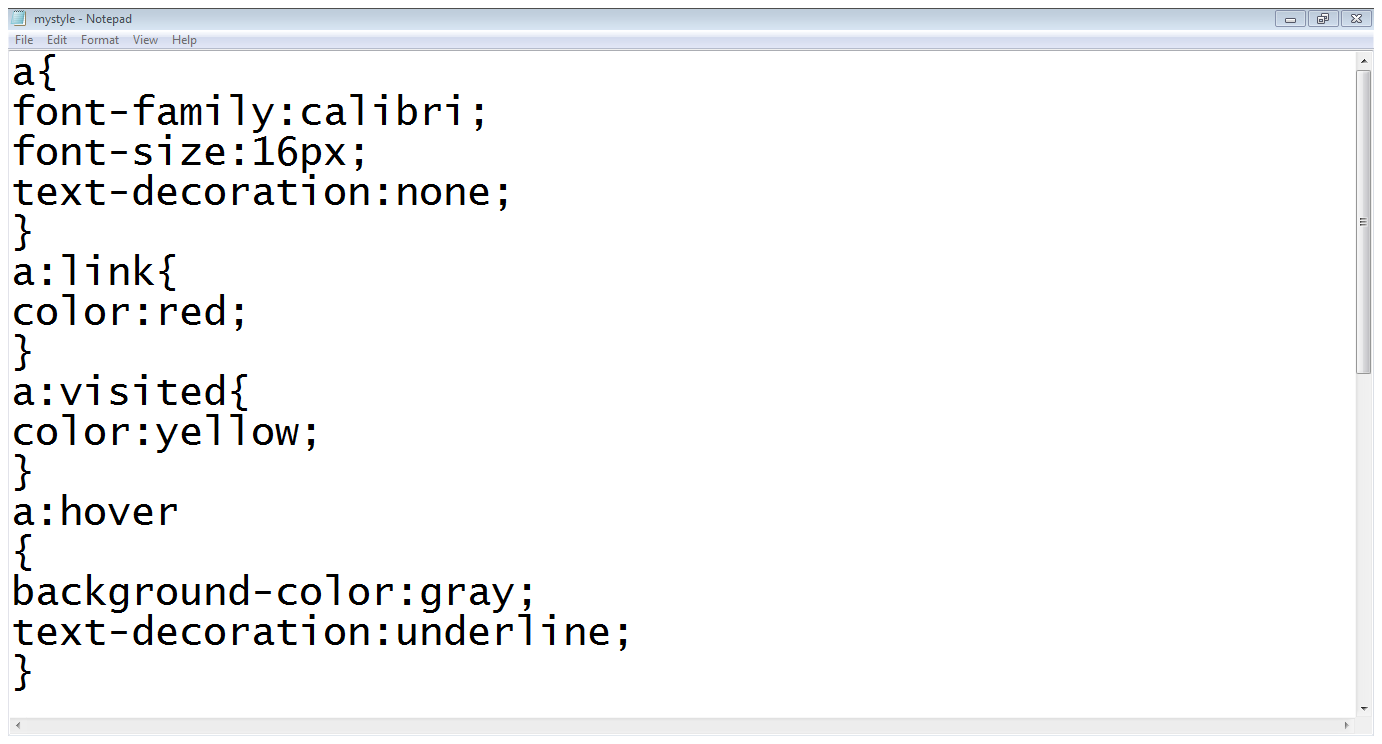
* color
* text-align
* vertical-align
* text-decoration
* text-transform
* word-spacing
* letter-spacing
* color:
  + specifies the color of the text
  + P{color : green}
* text-align:
  + horizontal alignment of the text
  + Left, right, center or justify
* vertical-align:
  + Vertical alignment of the text
  + Sub, super, top, middle, bottom
* text-decoration:
  + specifies the whether the text should be underline, overline, line-through or blinking
* text-transform:
  + text should be lowercase, uppercase or capitalized
* letter-spacing:
  + Specifies the space between letters
  + H1{letter-spacing:3px}
* word-spacing:
  + Specifies the space between words
  + H1{word-spacing:4px}

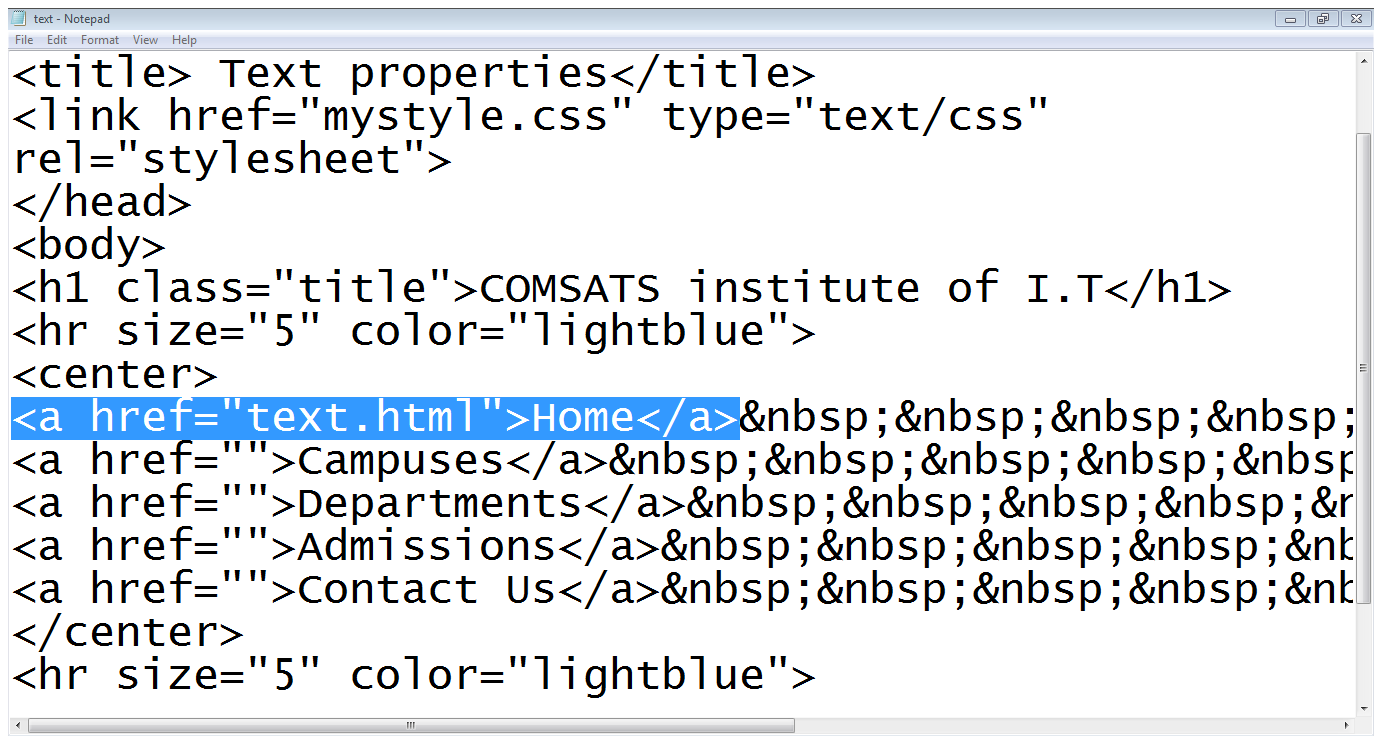




# Styling links

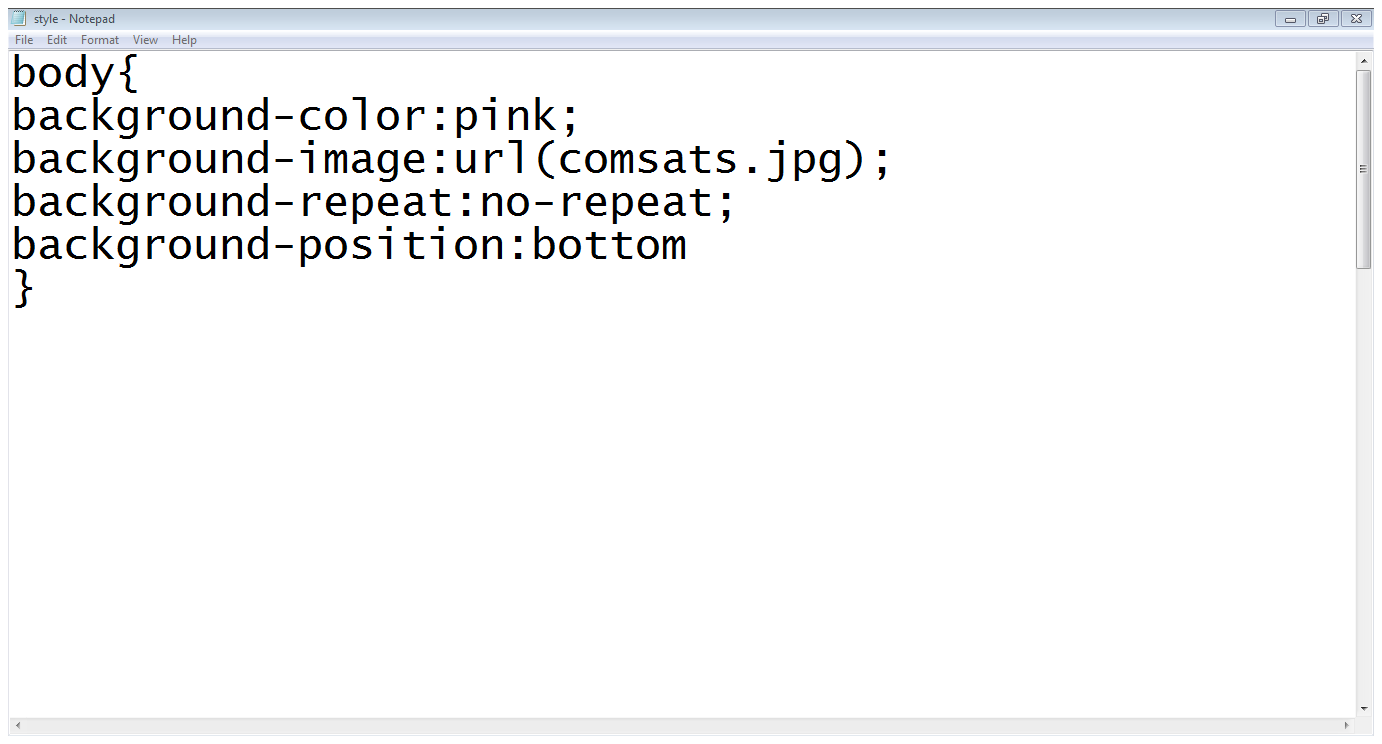
* color :
  + Changes the color of the links
* background-color :
  + Highlights the link, as if it had been highlighted with a highlighter pen
* text-decoration :
  + Underline, strike through, over-line, blink
* Pseudo-classes of links:
* Link:
  + Styles for links in general
* Visited:
  + Styles the links which are already visited
* Hover:
  + Styles when some on hovering over a link
* Active:
  + Styles the links when a links is being clicked





# Styling background

* background-color:
  + Specifies the background color
* background-image:
  + Specifies the background image
* background-repeat:
  + Specifies whether the image should repeat or not
* background-position:
  + Where an image should be positioned



Styling tables

* text and font
* vertical – align
* width
* height
* background - color
* background - image
* border :
  + border-style (solid, dashed, doted, double etc.)
  + border-color
  + border-bottom (solid, dashed, doted, double etc.)
* padding
  + Padding-left
  + Padding-right
  + Padding-top
  + padding-bottom

