

Cascading Style Sheet

CSS

What are Style Sheets?

- HTML contained tags to indicate how to render pages: tags for structure and tags for style.
- CSS is an excellent addition to plain HTML.
- The idea of style sheets is to separate page content and page style (text vs. colour, font, format, etc.).
- Cascading Style Sheets (CSS) is a technology used as an addition to HTML and gives a sophisticated way to specify how content should be rendered (in style).

Cascade of Rules

- CSS are sets of style rules to customize HTML elements.
- Style rules can be: external (imported), inline (within an HTML element), or embedded (declared in a document).
- They could be different styles applied for the same text section.
- There are general rules for determining the precedence (cascading) of the styles.
the rule for precedence is the nearest works
1-HTML tags have the highest precedence
2-inline style
3-embedded style
4-external style sheet

Specifying Style Rules

- The syntax for specifying style properties is:

selector {property : value}

the last one shouldn't have ;

or

selector {property1 : value1;

property2: value2;

...

propertyN : valueN}

Style Properties

- There are many properties pertaining to: font, size, colour, background, margins, borders, width, height, alignment, text appearance, etc. (and even position as we shall see later)
- Not all properties are recognized by both major browsers Netscape and IE.
- Conform to W3C [CSS-level1](#) and [CSS-level2](#).

Style Sheets Advantages

- Separation of text content and displaying style
 - Possibility to create external style templates
 - Consistent rendering of style throughout site
 - can be written so the user will only need to download it once - in the external style sheet document. When surfing the rest of your site the CSS will be cached on the users computer, and therefore speed up the loading time.
- it means in any page we use css file is downloaded for first time so all other times we work in this webpage it works rapidly
- No need for new HTML tags for new styles
 - offers much more detailed attributes than plain HTML for defining the look and feel of your site.

Defining Rules

- The tag <STYLE> allows the definition of formatting rules.

<STYLE>

<!--

Style rules

-->

</STYLE>

```
<STYLE>
```

```
<!--
```

```
BODY {font: 12pt Helvetica; color:blue;  
margin-left: 0.5in}
```

```
H1 {font: 18pt Palatino; color: red}
```

```
H2 {font-family: MeppDisplayShadow}
```

```
KBD {text-decoration: underline}
```

```
-->
```

```
</STYLE>
```

Selectors

- Selectors are the names that you give to your different styles.
- In the style definition you:
 - define how each selector should work (font, color etc.).
 - Then, in the body of your pages, you refer to these selectors to activate the styles. For example:

Selectors

```
<html>
```

```
  <head>
```

```
    <style type="text/css">
```

```
      b.headline {color:red; font-size:22px; font-family:arial;  
      text-decoration:underline}
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <b>this is normal bold</b><br>
```

```
    <b class="headline">this is headline style bold</b>
```

```
  </body>
```

view page

Selectors

- Selectors are the names that you give to your different styles.
- **HTML selectors**
 - Used to define styles associated to HTML tags. (A way to redefine the look of tags)
- **Class selectors**
 - Used to define styles that can be used without redefining plain HTML tags.
- **ID selectors**
 - Used to define styles relating to objects with a

the great difference between class and ID that ID should be used only for one time but class can be used more than one time

HTML (tag) selectors

- HTMLSelector {Property:Value;}

```
<html>
<head>
<style type="text/css">
B {font-family:arial; font-size:14px; color:red}
</style>

</head>
<body>
<b>This is a customized headline style bold</b>
</body>
</html>
```

[view page](#)

Class selectors

- **ClassSelector {Property:Value;}.**

```
<html>
<head>
<style type="text/css">
.headline {font-family:arial; font-size:14px; color:red}
</style>

</head>

<body>
<b class="headline">This is a bold tag carrying the headline
class</b>
<br>
<i class="headline">This is an italics tag carrying the headline
class</i>
</body>
</html>
```

view page

SPAN and DIV as carriers

- The `` tag provides no visual change by itself.
- The `` tag provides a way to add a hook to a part of a text or a part of a document.
- When the text is hooked in a span element you can add styles to the content, or manipulate the content with for example

SPAN and DIV as carriers

- The <div> tag defines a division or a section in an HTML document.

- The <div> tag is often used to group block-elements to format them with styles.

the great difference between span and div that
span: can change style but doesn't break line before or after the block
div: can change style but it breaks line before and after the block

ID Selectors

the way to define
ID in style

#IDSelector {Property:Value;}.

```
<html>
  <head>
    <style type="text/css">
      #layer1 {position:absolute; left:100;top:100; z-Index:0}
      #layer2 {position:absolute; left:140;top:140; z-Index:1}
    </style>
  </head>

  <body>
    <div ID="layer1">
      <table border="1" bgcolor="#FFCC00"><tr><td>THIS IS
      LAYER 1 <br>POSITIONED AT 100,100</td></tr></table>
    </div>

    <div ID="layer2">
      <table border="1" bgcolor="#00CCFF"><tr><td>THIS IS
      LAYER2 <br>POSITIONED AT 140,140</td></tr></table>
    </div>
```

Grouped selectors

- Most often selectors will share some of the same style
- for example, being based on the same font.
- In these cases, we can assign the font to all the selectors

```
h1, h2, h3, h4, h5, h6 {  
  font-family:arial; color:black; background:yellow;  
  font-size:14pt;  
}
```

```
.sublines {  
  font-family:arial; color:black; background:yellow;  
  font-size:12pt;  
}
```

```
.infotext {  
  font-family:arial; color:black; background:yellow;
```


Grouped selectors

```
.headlines, .sublines, .infotext {  
font-family:arial; color:black; background:yellow;  
}  
.headlines {font-size:14pt;}  
.sublines {font-size:12pt;}  
.infotext {font-size: 10pt;}
```

Context dependant selectors

- `<i>example</i>`
- ~~`I B {font-size:16px;color:red}`~~

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
I B {font-size:16px;color:red}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<b>example</b><br>
```

```
<i>example</i><br>
```

```
<i><b>example</b></i>
```

it means that this style rule only when there is a text has tag B enclosed by tag I

for example :

p .general

because we left a space

it means that this style will only work when we have text has tag with class general and it's enclosed with

paragraph tag but

p.general

means we define a special class for p we in definition didn't leave space between p and general

Using grouped and context dependent selectors at the same time:

- `l B, .headlines, B .sublines {font-size:16px;}`

In the example the font-size of 16 pixels is in effect on:

- All `` tags enclosed by `<l>` tags
- All headlines classes.
- sublines classes enclosed by `` tags.

Where to place CSS

- 1) Inline Style Sheets**
- 2) Embedded Style Sheets**
- 3) External Style Sheets**

Inline Style Sheets (single tag)

- We can create style rules within a document directly inside an HTML element tag.
- The formatting rules apply to the section.

```
<P STYLE="font: 10 pt Arial;  
line-height: 12 pt;  
margin-left: 0.5in;  
margin-right: 0.5cm;  
color: green;  
font-weight: bold">
```

This paragraph will be displayed as specified</P>

[view page](#)

Inline Style Sheets (single tag)

- You should limit your use of single tag CSS
- If you define your styles for each and every tag they're used on, you will lose much of the power associated with CSS.
- Furthermore, if you wanted to change a certain style, you'd have to change it all over in your document, rather than in one place.

Embedded Style Sheets (single page)

- We can add style information in the document HEAD.
- The formatting rules apply for the whole document.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Style Sheets Demo 1</TITLE>
```

```
<STYLE>
```

```
H1 {text-align:center; color:blue; font-family:Arial}
```

```
H2, H3 {text-decoration:underline; font-style:italic}
```

```
</STYLE>
```

```
</HEAD>
```

[view page](#)

```

<html>
  <head>
    <title>MY CSS PAGE</title>
    <style type="text/css">
      .headlines, .sublines, infotext {font-face:arial; color:black;
      background:yellow; font-weight:bold;}
    .headlines {font-size:14pt;}
    .sublines {font-size:12pt;}
    .infotext {font-size: 10pt;}
    </style>
  </head>
  <body>
    <span class="headlines">Welcome</span><br>
    <div class="sublines">
      This is an example page using CSS.<br>
      The example is really simple,<br>
      and doesn't even look good,<br>
      but it shows the technique.
    </div>
    <table border="2"><tr><td class="sublines">
      As you can see:<br>
      The styles even work on tables.
    </td></tr></table>
    <hr>
    <div class="infotext">
      Example from yahoo.Com
    </div>
  </body>
</html>

```

important properties values

view page

External Style Sheets (Entire site)

- Style sheets can be kept separately from the HTML document.
- Possible re-use of the same style sheets with different HTML documents.
- Use the <LINK ...> tag in the document HEAD.
- `<LINK REL=STYLESHEET HREF="mystyle.css" TYPE="text/CSS">` important definition of external style sheet

External Style Sheets (Entire site)

- ```
<html>
<head>
<title>MY CSS PAGE</title>
<link rel=stylesheet href="whatever.css" type="text/css">
</head>
<body>
Welcome

<div class="sublines">
This is an example of a page using CSS.

The example is really simple,

and doesn't even look good,

but it shows the technique.
</div>

<table border="2"><tr><td class="sublines">
As you can see:

The styles even work on tables.
</td></tr></table>

<hr>

<div class="infotext">Example from Yahoo.Com.</div>
```

# External Style Sheets (Entire site)

- File: **whatever.css**

```
.headlines, .sublines, infotext {font-face:arial;
color:black; background:yellow; font-weight:bold;}
.headlines {font-size:14pt;}
.sublines {font-size:12pt;}
.infotext {font-size: 10pt;}
```

# applying styles for non tag contained text?

- When we want to apply a style to part of a document that is not contained between an opening and closing tag, we can use the

- `<SPAN> ... </SPAN>` tag.

```
<OL TYPE=A>
<SPAN STYLE="font-style:italic;
color:red">
 my first element
 my second element

 this element is normal

```

view page

# CSS Text

- Font properties

- B {font-family:arial, helvetica; font-size:12px; font-weight:bold;}

Which is equivalent to:

- B {font:arial, helvetica 12px bold}

With this word we specify that all coming values are for font so we don't need to use properties names

view page

- Text properties

- line-height :
    - 1.5 lines spacing (using the current font size).

- Text-transform:

- Capitalize sets the first letter of each word in uppercase.
    - Uppercase forces all letters to uppercase.

view page

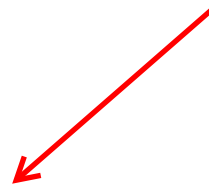
# CSS Text

- text-indent :
  - Use this to indent the first word of a paragraph.
- Color:
  - [B {font:arial, helvetica 12px bold; color:red}](#)
- white-space :
  - pre the browser will show all spaces in the text,
  - This is similar to the <pre> tag in plain HTML.

# CSS Colors

- With CSS, you can define an area to have a specific color without that area being part of a table.
- With CSS you can simply refer to a certain class in your **<TD>** tags.
- CSS Colors properties
  - Example
  - BODY {backgroundimage:url(myimage.gif); background-position: 75px 75px;}
  - BODY {background-image:url(myimage.gif); background-attachment: fixed;}

for putting image in the background by css external file



# CSS Colors

- basically you have three color options with CSS:

- Setting the foreground color for contents
- Setting the background color for an area
- Setting a background image to fill out an area

- In CSS colors could be defined as: three ways to define color in css style sheets

- .myclass {color:red; background-color:blue;}
- .myclass {color:#000000; background-color:#FFCC00;}
- .myclass {color:rgb(255,255,204); background-



# CSS Links

CSS has several options for redefining the style of links.

Property	Values
<b>A:link</b> <b>A:visited</b> <b>A:active</b> <b>A:hover</b>	<code>&lt;style&gt;</code> <code>&lt;style&gt;</code> <code>&lt;style&gt;</code> <code>&lt;style&gt;</code>

## **A:link**

Defines the style for normal unvisited links.

## **A:visited**

Defines the style for visited links.

## **A:active**

Defines the style for active links. A link becomes active once you click on it.

## **A:hover**

Defines the style for hovered links. A link is hovered when the mouse moves

view page

not covered

# CSS Links

- It is smart to Use **<span>** tag to define the context for two reasons:
  - The obvious, that it allows us to use different link styles on the same page, rather than being limited to using a single overall link style.
  - We can define entire areas where a certain link style works for all links within that area. Thus, we don't have to add a style definition to each and every link in that area.

# CSS Lists

- List Properties
- **list-style type:** Defines the look of the bullets used in your list.
- **list-style image:** Let's you use a custom graphic for bullets.
- **list-style position:** Often the text in a list is longer than one line.
  - outer lets the second line align with the first line. That is: the bullet is to the left of both lines.
  - inner lets the second line align with the bullet.

# CSS Cursors

not covered

- Cursor properties
  - **Adding a customized cursor:** The syntax for a customized cursor is this: (Position the mouse over each link to see the effect) **Selector {cursor:value}**.
  - Example
- **Redefining the cursor for entire pages** If you want to redefine the cursor so that it's not only showing up when moved over a link, you simply specify the desired cursor using the body-selector.
  - ENTIRE PAGE
- **Cursor for Areas on a page**

not covered

# CSS Layers

- With CSS, it is possible to work with layers.
- Layers are pieces of HTML that are placed on top of the regular page with pixel precision.
  - [Example](#)
- **Visible versus hidden Layers:**
  - `<div style="position:relative; visibility:hidden;">HELLO!!!</div>`
  - Visibility property takes either : visible or hidden

# Project 2

- [Sheet 2](#)
  - Deadline week of 16-4-2011
- Project2
  - Redo Project 1 using CSS.
  - Deadline Sunday 16-4-2011
- [Sheet 3](#)
  - Deadline week of 16-4-2011