

[Type text]

CASCADING STYLE SHEETS

INTRODUCTION

XHTML style sheets are called *cascading* style sheets because they can be defined at three different levels to specify the style of a document. Lower level style sheets can override higher level style sheets, so the style of the content of a tag is determined, in effect, through a cascade of style-sheet applications.

LEVELS OF STYLE SHEETS

- The three levels of style sheets, in order from lowest level to highest level, are inline, document level, and external.
- **Inline style sheets** apply to the content of a single XHTML element.
- **Document-level style sheets** apply to the whole body of a document.
- **External style sheets** can apply to the bodies of any number of documents.
- Inline style sheets have precedence over document style sheets, which have precedence over external style sheets.
- Inline style specifications appear within the opening tag and apply only to the content of that tag.
- Document-level style specifications appear in the document head section and apply to the entire body of the document.
- External style sheets stored separately and are referenced in all documents that use them.
- External style sheets are written as text files with the MIME type text/css.
- They can be stored on any computer on the Web. The browser fetches external style sheets just as it fetches documents.

The <link> tag is used to specify external style sheets.

Within <link>, the rel attribute is used to specify the relationship of the linked-to document to the document in which the link appears. The href attribute of <link> is used to specify the URL of the style sheet document. **EXAMPLE**

WHICH USES EXTERNAL STYLE SHEET

```
<html>
  <head>
    <title>Sample CSS</title>
    <link rel = "stylesheet" type = "text/css"
          href = "Style1.css" />
  </head>
  <h1>Puneeth Rajkumar</h1>
</html>
```

Style1.css

```
h1
{
  font-family: 'Lucida Handwriting';
  font-size: 50pt;
  color: Red;
}
```

EXAMPLE WHICH USES DOCUMENT LEVEL STYLE SHEET

```
<html>
  <head>
    <title>Sample CSS</title>

    <style type = "text/css">
      h1
      {
        font-family: 'Lucida Handwriting';
        font-size: 50pt;
      }
    </style>
  </head>
  <h1>Puneeth Rajkumar</h1>
</html>
```

[Type text]

```
        color: Red;  
    }  
</style>  
</head>  
<h1>Puneeth Rajkumar</h1>  
</html>
```

EXAMPLE WHICH USES INLINE STYLE SHEET

```
<html>  
<head>  
<title>Sample CSS</title>  
</head>  
<h1 style ="font-family: 'Lucida Handwriting'; font-size: 50pt; color: Red;">  
    Puneeth Rajkumar </h1>  
</html>
```

STYLE SPECIFICATION FORMATS

Inline Style Specification:

Style = "Property1 : Value1; Property2 : Value2; Property3 : Value3; ... Property_n : Value_n;"

Document Style Specification:

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

Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property–value pairs. The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

Selector { Property1 : Value1; Property2 : Value2; Property3 : Value3; ... Property_n : Value_n; }

[For examples on all three levels of style sheets along with specifications, Please refer the previous examples] .

SELECTOR FORMS

Simple Selector Forms:

In case of simple selector, a tag is used. If the properties of the tag are changed, then it reflects at all the places when used in the program. The selector can be any tag. If the new properties for a tag are not mentioned within the rule list, then the browser uses default behaviour of a tag.

```
<html>  
<head>  
<title>Sample CSS</title>  
<style type = "text/css">  
    p  
    {  
        font-family: 'Lucida Handwriting';  
        font-size: 50pt;  
        color: Red;  
    }  
</style>  
</head>
```

[Type text]

```
<body>
  <p>Puneeth Rajkumar</p>
  <p>Mahesh Babu</p>
  <p>Suriya</p>
</body>
</html>
```

Class Selectors:

In class selector, it is possible to give different properties for different elements

```
<html>
  <head>
    <title>Sample CSS</title>
    <style type = "text/css">
      p.one
      {
        font-family: 'Lucida Handwriting';
        font-size: 25pt;
        color: Red;
      }
      p.two
      {
        font-family: 'Monotype Corsiva';
        font-size: 50pt;
        color: green;
      }
    </style>
  </head>
  <body>
    <p class = "one">Puneeth Rajkumar</p>
    <p class = "two">Puneeth Rajkumar</p>
  </body>
</html>
```

Generic Selectors:

In case of generic selector, when the class is created, it would not be associated to any particular tag. In other words, it is generic in nature.

```
<html>
  <head>
    <title>Sample CSS</title>
    <style type = "text/css">
      .one
      {
        font-family: 'Monotype Corsiva';
        color: green;
      }
    </style>
  </head>
  <body>
    <p class = "one">Puneeth Rajkumar</p>
```

[Type text]

```
<h1 class = "one">Puneeth Rajkumar</h1>
<h6 class = "one">Puneeth Rajkumar</h6>
</body>
</html>
```

id Selectors:

An id selector allows the application of a style to one specific element.

```
<html>
<head>
<title>Sample CSS</title>
<style type = "text/css">
#one
{
    font-family: 'lucida calligraphy';
    color: purple;
}

#two
{
    font-family: 'comic sans ms';
    color: orange;
}
</style>
</head>
<body>
<p id = "two">Puneeth Rajkumar</p>
<h1 id = "one">Puneeth Rajkumar</h1>
</body>
</html>
```

Universal Selectors:

The universal selector, denoted by an asterisk (*), applies its style to all elements in a document.

```
<html>
<head>
<title>Sample CSS</title>
<style type = "text/css">
*
{
    font-family: 'lucida calligraphy';
    color: purple;
}
</style>
</head>
<body>
    <h1>Puneeth Rajkumar</h1>
    <h2>Puneeth Rajkumar</h2>
    <h3>Puneeth Rajkumar</h3>
    <p>Puneeth Rajkumar</p>
</body>
```

[Type text]

```
</html>
```

Pseudo Classes:

Pseudo class selectors are used if the properties are to be changed dynamically. For example: when mouse movement happens, in other words, hover happens or focus happens.

```
<html>
<head>
<title>Sample CSS</title>
<style type = "text/css">
    input:focus
    {
        font-family: 'lucida calligraphy';
        color: purple;
        font-size:100;
    }
    input:hover
    {
        font-family: 'lucida handwriting';
        color: violet;
        font-size:40;
    }
</style>
</head>
<body>
    <form action = "">
        <p>
            <label>
                NAME:
                <input type = "text" />
            </label>
        </p>

    </form>
</body>
</html>
```

PROPERTY VALUE FORMS

CSS includes 60 different properties in seven categories: fonts, lists, alignment of text, margins, colours, backgrounds, and borders. Property values can appear in a variety of forms.

- ④ Keyword property values are used when there are only a few possible values and they are predefined.
- ④ A number value can be either an integer or a sequence of digits with a decimal point and can be preceded by a sign (+ or -).
- ④ Length values are specified as number values that are followed immediately by a two-character abbreviation of a unit name. The possible unit names are px, for pixels; in, for inches; cm, for centimeters; mm, for millimeters; pt, for points.
- ④ Percentage values are used to provide a measure that is relative to the previously used measure for a property value. Percentage values are numbers that are followed immediately by a percent sign (%). Percentage values can be signed. If preceded by a plus sign, the percentage is added to the previous value; if negative, the percentage is subtracted.
- ④ There can be no space between url and the left parenthesis.
- ④ Color property values can be specified as color names, as six-digit hexadecimal numbers, or in RGB form. RGB

[Type text]

form is just the word `rgb` followed by a parenthesized list of three numbers that specify the levels of red, green, and blue, respectively. The RGB values can be given either as decimal numbers between 0 and 255 or as percentages. Hexadecimal numbers must be preceded with pound signs (#), as in `#43AF00`.

FONT PROPERTIES

Font Families:

The `font-family` property is used to specify a list of font names. The browser uses the first font in the list that it supports. For example, the property:

font-family: Arial, Helvetica,

tells the browser to use Arial if it supports that font. If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can. If the browser does not support any of the specified fonts, it will use an alternative of its choosing.

If a font name has more than one word, the whole name should be delimited by single quotes, as in the following example:

font-family: 'Times New

Font Sizes:

The `font-size` property does what its name implies. For example, the following property specification sets the font size for text to 10 points:

font-size:

Many relative font-size values are defined, including `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, and `xx-large`. In addition, `smaller` or `larger` can be specified. Furthermore, the value can be a percentage relative to the current font size.

Font Variants:

The default value of the `font-variant` property is `normal`, which specifies the usual character font. This property can be set to `small-caps` to specify small capital characters. These characters are all uppercase, but the letters that are normally uppercase are somewhat larger than those that are normally lowercase.

Font Styles:

The `font-style` property is most commonly used to specify `italic`, as in

font-style:

Font Weights:

The `font-weight` property is used to specify the degree of boldness, as in

font-weight:

Besides `bold`, the values `normal`, `bolder`, and `lighter` can be specified. Specific numbers also can be given in multiples of 100 from 100 to 900, where 400 is the same as `normal` and 700 is the same as `bold`.

Font Shorthands:

If more than one font property must be specified, the values can be stated in a list as the value of the `font` property. The order in which the property values are given in a font value list is important. The order must be as follows: The font names must be last, the font size must be second to last, and the font style, font variant, and font weight, when they are included, can be in any order but must precede the font size and font names.

font: bold 14pt 'Times New

```
<html>
<head>
```

[Type text]

```
<title>Font Properties</title>
<style type = "text/css">
p.one
{
    font-family: 'lucida calligraphy';
    font-weight:bold;
    font-size:75pt;
    color: purple;
}
h1.two
{
    font-family: 'cambria';
    color: violet;
    font-style:italics;
}
p.three
{
    font: small-caps italic bold 50pt 'times new roman'
}
</style>
</head>
<body>
    <p class = "one">Puneeth Rajkumar</p>
    <h1 class = "two">Puneeth Rajkumar</h1>
    <p class = "three">Puneeth Rajkumar</p>
</body>
</html>
```

Text Decoration:

The text-decoration property is used to specify some special features of text. The available values are line-through, overline, underline, and none, which is the default.

```
<html>
<head>
<title>Text Decoration</title>
<style type = "text/css">
h1.one
{text-decoration: line-through;}
h1.two
{text-decoration: overline;}
h1.three
{text-decoration: underline;}
</style>
</head>
<body>
    <h1 class = "one">Puneeth Rajkumar</h1> <p>[This is line-through]</p><br/>
    <h1 class = "two">Puneeth Rajkumar</h1> <p>[This is overline]</p><br/>
    <h1 class = "three">Puneeth Rajkumar</h1><p>[This is underline]</p><br/>
</body>
</html>
```

[Type text]

LIST PROPERTIES

Two presentation details of lists can be specified in XHTML documents: the shape of the bullets that precede the items in an unordered list and the sequencing values that precede the items in an ordered list. The list-style-type property is used to specify both of these.

The **list-style-type** property of an unordered list can be set to disc, circle, square, or none.

```
<html>
<head>
<title>CSS Bullets</title>
<style type = "text/css">
li.one {list-style-type:disc}
li.two{list-style-type:square}
li.three{list-style-type:circle}
</style>
</head>
<body>
<h3>South Indian Kings</h3>

<ul>
<li class = "one"> Puneeth Rajkumar</li>
<li class = "two"> Mahesh Babu</li>
<li class = "three"> Suriya</li>
</ul>
</body>
</html>
```

Bullets in unordered lists are not limited to discs, squares, and circles. Any image can be used in a list item bullet. Such a bullet is specified with the list-style-image property, whose value is specified with the url form.

```
<html>
<head>
<title>CSS Bullets-Image</title>
<style type = "text/css">
li.image {list-style-image: url(bullet.png); font-size:25pt;}
</style>
</head>
<body>
<h1>South Indian Kings</h1>
<ul>
<li class = "image"> Puneeth Rajkumar</li>
<li class = "image"> Mahesh Babu</li>
<li class = "image"> Suriya</li>
</ul>
</body>
</html>
```

The following example illustrates the use of different sequence value types in nested lists:

```
<html>
<head>
<title> CSS nested lists </title>
<style type = "text/css">
ol {list-style-type:upper-roman;}
```

[Type text]

```
ol ol {list-style-type:upper-alpha;}
ol ol ol {list-style-type:decimal;}
</style>
</head>

<ol>
<li> Information Science </li>
<ol>
<li>OOMD</li>
<li>Java & J2ee</li>
<ol>
<li>classes and methods</li>
<li>exceptions</li>
<li>applets</li>
<li>servelets</li>
</ol>
<li>Computer Networks</li>
<ol>
<li>Part 1</li>
<li>Part 2</li>
</ol>
<li>DBMS</li>
<li>Operations Research</li>
</ol>
<li> Computer Science</li>
<ol>
<li>Compiler Design</li>
<li>FLAT</li>
<ol>
<li>NFA</li>
<li>DFA</li>
<li>CFG</li>
</ol>
<li>Computer Graphics</li>
<li>Artificial Intelligence</li>
</ol>
</ol>
</html>
```

COLOR

Color Groups:

Three levels of collections of colours might be used by an XHTML document. The smallest useful set of colours includes only those that have standard names and are guaranteed to be correctly displayable by all browsers on all color monitors. This collection of 17 colours is called the *named colours*.

Larger set of colors, called the Web palette, consists of 216 colors. The colors of the Web palette can be viewed at http://www.web-source.net/216_color_chart.htm

[Type text]

Color Properties:

The color property is used to specify the foreground color of XHTML elements.

```
<html>
<head>
<title>Colours</title>
<style type = "text/css">
  p.one
  {color: pink; }
  p.two
  {color: # 9900FF; }
  p.three
  {background-color:#99FF00; }
</style>
</head>
<body>
  <p class = "one">Puneeth Rajkumar</p>
  <p class = "two">Puneeth Rajkumar</p>
  <p class = "three">Puneeth Rajkumar</p>
</body>
</html>
```

ALIGNMENT OF TEXT

- ④ The text-indent property can be used to indent the first line of a paragraph. This property takes either a length or a percentage value. The text-align property, for which the possible keyword values are left, center, right, and justify, is used to arrange text horizontally.
- ④ The float property is used to specify that text should flow around some element, often an image or a table. The possible values for float are left, right, and none, which is the default.

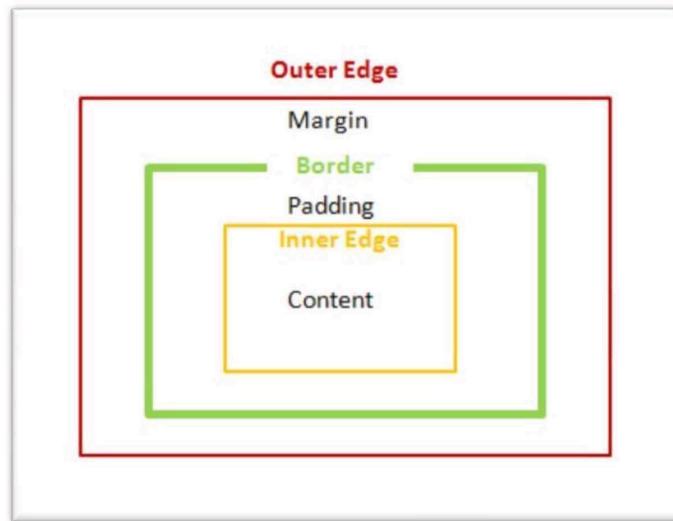
```
<html>
<head>
<title>Text Alignment</title>
<style type = "text/css">
  h1.one
  {text-align: center}
  p.two
  {text-indent: 0.5in; text-align: justify;}
  img{float:right}
</style>
</head>
<body>
  <h1 class = "one">Kannadada Kotyadhipathi</h1>
  <p>
    <img src = "kk.jpg" alt="error"/>
  </p>
  <p class = "two">Kannadada Kotyadhipathi is a Kannada primetime quiz show hosted by the power star of Kannada cinema Mr. Puneet Rajkumar. This is the biggest game show ever on Kannada Television. This show will be aired on Suvarna TV. This show gives the common man an opportunity to win Rs 1 crore. Kannadada Kotyadipathi is a Kannada primetime quiz and human drama show hosted by matinee idol Puneeth Rajkumar on Suvarna TV. Contestants participate in a game that allows them to win up to Rs. 1 crore. Short-listed contestants play a 'Fastest Finger First' round to make it to the main game. From there on, they play rounds with increasing levels of difficulty, and winning higher amounts of money, culminating in the Rs. 1 crore prize. Contestants can stop at any time having viewed the next question. Or they can avail of a 'Lifeline' and play on.
```

[Type text]

```
Welcome to the world of high stakes chills and thrills! Welcome to the world of the crorepati!</p>
</body>
</html>
```

THE BOX MODEL

- On a given web page or a document, all the elements can have borders.
- The borders have various styles, color and width.
- The amount of space between the content of the element and its border is known as *padding*.
- The space between border and adjacent element is known as *margin*.



Borders:

Border-style

It can be dotted, dashed, double

- ▶ Border-top-style
- ▶ Border-bottom-style
- ▶ Border-left-style
- ▶ Border-right-style

Border-width

It can be thin, medium, thick or any length value

- ▶ Border-top-width
- ▶ Border-bottom-width
- ▶ Border-left-width
- ▶ Border-right-width

Border-color

- ▶ Border-top-color
- ▶ Border-bottom-color
- ▶ Border-left-color
- ▶ Border-right-color

```
<html>
<head>
<title> Table with border effects </title>
<style type = "text/css">
table
{
    border-width:thick;
    border-top-color:red;
```

[Type text]

```
border-left-color:orange;
border-bottom-color:violet;
border-
border-
border-
border-bottom-style:double;
border-right-style:dotted;
}
</style>
</head>
<body>
<table border = "border">
<caption>PARAMATHMA </caption>
<tr>
<td> Puneeth Rajkumar </td>
<td> <img src = "puneeth.jpg" alt = "cant display"/></td>
</tr>
</table>
</body>
</html>
```

Margins and Padding:

The margin properties are named margin, which applies to all four sides of an element: margin-left, margin-right, margin-top, and margin-bottom. The padding properties are named padding, which applies to all four sides: padding-left, padding-right, padding-top, and padding-bottom.

```
<html>
<head>
<title> Margins and Padding </title>
<style type = "text/css">
p.one
{
    margin:0.1in;
    padding:0.5in;
    background-color:#FF33FF;
    border-style:dotted;
}
p.two
{
    margin:0.5in;
    padding:0.1in;
    background-color:#00FF33;
    border-style:dashed;
}
p.three
{
    margin:0.3in;
    background-color:#FFFF00;
}
p.four
{
    padding:0.3in;
```

[Type text]

```
background-color:#FF9900;
}
</style>
</head>
<body>
<p class = "one"> Puneeth Rajkumar is the Power Star of Sandalwood<br/>
[margin=0.1in, padding=0.5in]</p>
<p class = "two"> Puneeth Rajkumar is the Power Star of Sandalwood<br/>
[margin=0.5in, padding=0.1in]</p>
<p class = "three"> Puneeth Rajkumar is the Power Star of Sandalwood<br/>
[margin=0.3in, no padding, no border]</p>
<p class = "four"> Puneeth Rajkumar is the Power Star of Sandalwood<br/> [no
margin, padding=0.3in, no border]</p>
</body>
</html>
```

BACKGROUND IMAGES

The background-image property is used to place an image in the background of an element.

```
<html>
<head>
<title>Background Image</title>
<style type = "text/css">
body {background-image:url(bg3.jpg);} p
{text-align: justify; color:white;font-size:25pt;}
</style>
</head>
<body>
```

<p >Kannadada Kotyadhipathi is a Kannada primetime quiz show hosted by the power star of Kannada cinema Mr. Puneet Rajkumar. This is the biggest game show ever on Kannada Television. This show will be aired on Suvarna TV. This show gives the common man an opportunity to win Rs 1 crore. Kannadada Kotyadipathi is a Kannada primetime quiz and human drama show hosted by matinee idol Puneeth Rajkumar on Suvarna TV. Contestants participate in a game that allows them to win up to Rs. 1 crore. Short-listed

contestants play a 'Fastest Finger First' round to make it to the main game. From there on, they play rounds with increasing levels of difficulty, and winning higher amounts of money, culminating in the Rs. 1 crore prize. Contestants can stop at any time having viewed the next question. Or they can avail of a 'Lifeline' and play on. Welcome to the world of high stakes chills and thrills! Welcome to the world of the crorepati!

```
</body>
</html>
```

In the example, notice that the background image is replicated as necessary to fill the area of the element. This replication is called *tiling*. Tiling can be controlled with the background-repeat property, which can take the value repeat (the default), no-repeat, repeat-x, or repeat-y. The no-repeat value specifies that just one copy of the image is to be displayed. The repeat-x value means that the image is to be repeated horizontally; repeat-y means that the image is to be repeated vertically. In addition, the position of a non-repeated background image can be specified with the background-position property, which can take a large number of different values. The keyword values are top, center, bottom, left, and right, all of which can be used in many different combinations.

THE AND <div> TAGS

In many situations, we want to apply special font properties to less than a whole paragraph of text. The

[Type text]

```
<span> tag is designed for just this purpose.  
<html>  
<head> <title>span</title>  
<style type = "text/css">  
  .spanviolet {font-size:25pt;font-family:'lucida calligraphy';color:violet;}  
 </style>  
</head>  
<body>  
  <p>Kannadada Kotyadhipathi is a Kannada primetime quiz show hosted by <span class = "spanviolet"> Puneeth  
 Rajkumar </span>, the power star of Kannada cinema </p>  
</body>  
</html>
```

It is more convenient, however, to be able to apply a style to a section of a document rather than to each paragraph. This can be done with the `<div>` tag. As with ``, there is no implied layout for the content of the `<div>` tag, so its primary use is to specify presentation details for a section or division of a document.

```
<html>  
<head>  
<title>div</title>  
<style type = "text/css">  
  .one  
    {font-size:20pt;font-family:'lucida calligraphy';color:violet;}  
  .two  
    {font-size:25pt;font-family:'comic sans ms';color:green;}  
</style>  
</head>  
<body>  
  <div class = "one">  
    <p>Paragragh 1 under division 1</p>  
    <p>Paragragh 2 under division 1</p>  
    <p>Paragragh 3 under division 1</p>  
  </div>  
  <div class = "two">  
    <p>Paragragh 1 under division 2</p>  
    <p>Paragragh 2 under division 2</p>  
    <p>Paragragh 3 under division 2</p>  
  </div>  
</body>  
</html>
```

CONFLICT RESOLUTION

- ② Sometimes on a web page, there can be two different values for the same property on the same element leading to conflict.
 - ② `h3 {color: blue;}` `body`
`h3 {color: red;}`
 - ② The browser has to resolve this conflict.
 - ② There can be one or more type of conflict: i.e. when style sheets at 2 or more levels specify different value for same property on some element.
 - ② This conflict is resolved by providing priority to the different levels of style sheets.

[Type text]

- ② The inline level gets the highest priority over the document level.
- ② The document level gets the higher priority over the external level
- ② But the browser must be able to resolve the conflict in the first example using same technique.
- ② There can be several different origins of the specification of property values.
- ② One of the value may come from a style sheet created by the author or it can be specified by the user using the options provided by the browser.
- ② The property values with different origin have different precedence.
- ② The precedence can also be set for a property by marking it as important.
- ② `p.special {font-style: italic !important; font-size: 14}`
- ② This means that font-style:italic is important [this is known as weight of specification]
- ② The process of conflict resolution is a multi-stage sorting process.
- ② The first step is to gather information about levels of style sheet.
- ② Next, all the origins and weights are sorted. The following rules are considered:
 1. Important declarations with user origin
 2. Important declarations with author origin
 3. Normal declarations with author origin
 4. Normal declarations with user origin
 5. Any declarations with browser (or other user agent) origin
- ② If there are other conflicts even after sorting, the next step is sorting by specificity. Rules are:
 1. id selectors
 2. Class and pseudo class selectors
 3. Contextual selectors (more element type names means that they are more specific)
 4. Universal selectors
- ② If there still conflicts, they are resolved by giving precedence to most recently seen specification.