

This chapter will cover the following important @ rules –

- The **@import**: rule imports another style sheet into the current style sheet.
- The **@charset** rule indicates the character set the style sheet uses.
- The **@font-face** rule is used to exhaustively describe a font face for use in a document.
- The **!important** rule indicates that a user-defined rule should take precedence over the author's style sheets.

NOTE – There are other @ rules which we will cover in subsequent chapters.

The @import rule

The @import rule allows you to import styles from another style sheet. It should appear right at the start of the style sheet before any of the rules, and its value is a URL.

It can be written in one of the two following ways –

```
<style tyle="text/css">
  <!--
  @import "mystyle.css";
  or
  @import url("mystyle.css");
  .....other CSS rules .....
  -->
</style>
```

The significance of the @import rule is that it allows you to develop your style sheets with a modular approach. You can create various style sheets and then include them wherever you need them.

The @charset Rule

If you are writing your document using a character set other than ASCII or ISO-8859-1 you might want to set the @charset rule at the top of your style sheet to indicate what character set the style sheet is written in.

The @charset rule must be written right at the beginning of the style sheet without even a space before it. The value is held in quotes and should be one of the standard character-sets. For example –

```
<style tyle="text/css">
  <!--
  @charset "iso-8859-1"
  .....other CSS rules .....
  -->
</style>
```

The @font-face Rule

The @font-face rule is used to exhaustively describe a font face for use in a document. @font-face may also be used to define the location of a font for download, although this may run into implementation-specific limits.

In general, @font-face is extremely complicated, and its use is not recommended for any except those who are expert in font metrics.

Here is an example –

```

<style tyle="text/css">
  <!--
  @font-face {
    font-family: "Scarborough Light";
    src: url("http://www.font.site/s/scarbo-lt");
  }
  @font-face {
    font-family: Santiago;
    src: local ("Santiago"),
    url("http://www.font.site/s/santiago.tt")
    format("truetype");
    unicode-range: U+??, U+100-220;
    font-size: all;
    font-family: sans-serif;
  }
  -->
</style>

```

The !important Rule

Cascading Style Sheets cascade. It means that the styles are applied in the same order as they are read by the browser. The first style is applied and then the second and so on.

The !important rule provides a way to make your CSS cascade. It also includes the rules that are to be applied always. A rule having a !important property will always be applied, no matter where that rule appears in the CSS document.

For example, in the following style sheet, the paragraph text will be black, even though the first style property applied is red:

```

<style tyle="text/css">
  <!--
  p { color: #ff0000; }
  p { color: #000000; }
  -->
</style>

```

So, if you wanted to make sure that a property always applied, you would add the !important property to the tag. So, to make the paragraph text always red, you should write it as follows –

```

<html>
  <head>

    <style tyle="text/css">
      p { color: #ff0000 !important; }
      p { color: #000000; }
    </style>

  </head>
  <body>
    <p>Tutorialspoint.com</>
  </body>
</html>

```

Here you have made `p { color: #ff0000 !important; }` mandatory, now this rule will always apply even you have defined another rule `p { color: #000000; }`

It will produce the following result –

