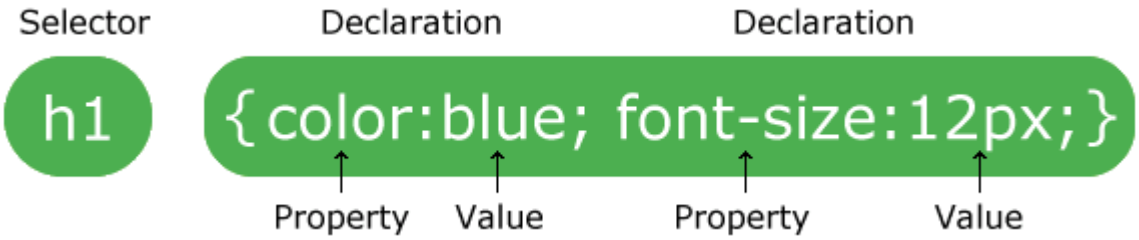
CSS

CSS stands for Cascading Style Sheets

- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p{
        color: red;
        text-align: center;
  }
```

Two Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS

- Internal CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Example

External styles are defined within the <link> element, inside the <head> section of an HTML page:

```
<!Doctype html>
<html>
<head>
    <link rel=”stylesheet” href=”mystyle.css”>
</head>
<body>
    <h1>This is a heading</h1>
    <p>This is a Paragraph</p>
</body>
</html>
```

An external style sheet can be written in any text editor, and must be saved with a .css extension.
The external .css file should not contain any HTML tags.

Here is how the "mystyle.css" file looks like:

"mystyle.css"

```
body{
    background-color:lightblue;
    }
```

```
h1{
        color:navy;
        margin-left:20px;
}
```

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

Example

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<!Doctype html>
<html>
<head>
        <style>
                body{
                        background-color:linen;
                    }
                h1{
                        color:maroon;
                        margin-left:40px;
                    }
        </style>
</head>
<body>
        <h1>This is a Heading</h1>
        <p>This is a Paragraph</p>
</body>
```

```
</html>
```

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p{
        text-align:center;
        color:red;
 }
```

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1{
                text-align:center;
                color:red;
        }
```

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center{
        text-align:center;
        color:red;
    }
```

You can also specify that only specific HTML elements should be affected by a class.

Example

In this example only <p> elements with class="center" will be center-aligned:

```
p.center{
    text-align:center;
    color:red;
    }
```

HTML elements can also refer to more than one class.

Example

In this example the <p> element will be styled according to class="center" and to class="large":

```
<p class="center large">This paragraph refers to two classes.</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
*{
        text-align:center;
        color:blue;
  }
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
h1{
        text-align:center;
        color:red;
   }
h2{
        text-align:center;
        color:red;
   }
p{
        text-align:center;
        color:red;
   }
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1,h2,p {
        text-align:center;
        color:red;
    }
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a l ater date.

Comments are ignored by browsers

A CSS comment is placed inside the <style> element, and starts with /* and ends with */:

Example

```
/* This is a single-line comment */
p{
    color:red;
 }
```

Example

```
/* This is a multi-line
comment */
p{
    color:red;
 }
```

CSS Background Color

Example

```
<h1 style="background-color : DodgerBlue;"> Hello World</h1>
<p style="background-color : Tomato;"> Lorem psum ...</p>
```

CSS Text Color

```
<h1 style="background-color : DodgerBlue;"> Hello World</h1>
<p style="background-color : Tomato;"> Lorem psum ...</p>
<p style="color : MediumSeaGreen;">Ut wisi enim...</p>
```

CSS Border Color

Example

```
<h1 style="border : 2px solid Tomato;"> Hello World</h1>
<h1 style=" border : 2px solid DodgerBlue;"> Hello World</h1>
<h1 style=" border : 2px solid Violet;"> Hello World</h1>
```

CSS background-color

The background-color property specifies the background color of an element.

Example

The background color of a page is set like this:

```
body {
        background-color : lightblue;
      }
```

Opacity / Transparency

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:

```
div {
            background-color : green;
```

```
        opacity : 0.3;
    }
```

CSS background-image

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

Example

The background image for a page can be set like this:

```
body {
        background-image : url("paper.jpg");
    }
```

CSS background-repeat

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Example

```
body {
        background-image : url("paper.jpg");
    }
```

CSS background-repeat: no-repeat

Showing the background image only once is also specified by the background- repeat property:

Example

Show the background image only once:

```
body {
```

```
        background-image : url("paper.jpg");
        background-repeat : no-repeat;
    }
```

CSS Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border

Example

Demonstration of the different border styles:

```
p.dotted { border-style : dotted;}
p. solid { border-style : solid;}
p. double { border-style: double;}
p. groove { border-style : groove;}
p. ridge { border-style : ridge;}
p. inset { border-style : inset;}
p. outset { border-style : outset;}
p. none { border-style : none;}
```

```
p. hidden { border-style : hidden;}
p. mix { border-style : dotted dashed solid double;}
```

CSS Border Width

```
p.one {
        border-style : solid;
        border-width : 5px;
    }
```

CSS Margins

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- margin-top
- margin-right
- margin-bottom
- margin-left

Example

Set different margins for all four sides of a <p> element:

```
p {
    margin-top : 100px;
    margin-bottom : 100px;
    margin-right : 150px;
    margin-left : 80px;
  }
```

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

adding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- padding-top
- padding-right
- padding-bottom
- padding-left

Example

Set different padding for all four sides of a element:

```
div {
        padding-top : 50px;
        padding-right : 30px;
        padding-bottom : 50px;
        padding-left : 80px;
    }
```

CSS Lists

HTML Lists and CSS List Properties

In HTML, there are two main types of lists:

- unordered lists (<ul>) - the list items are marked with bullets
- ordered lists (<ol>) - the list items are marked with numbers or letters

The CSS list properties allow you to:

- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

Different List Item Markers

The list-style-type property specifies the type of list item marker.

The following example shows some of the available list item markers:

Example

```
ul.a {
        list-style-type : circle;
    }
ul.b {
        list-style-type: square;
    }
ol.c {
        list-style-type: upper-roman;
    }
ol.d {
        list-style-type : lower-alpha;
    }
```

CSS Tables

Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for <table>, <th>, and <td> elements

Example

```
table ,  th , td {
```

```
        border : 1px solid black;
    }
```

Collapse Table Borders

The border-collapse property sets whether the table borders should be collapsed into a single border

Example

```
table  {
        border-collapse : collapse;
    }
table ,  th , td {
        border : 1px solid black;
    }
```

Table Width and Height

Example

```
 table {
        width : 100%
    }
th {
        height : 50px;
   }
```

Horizontal Alignment

The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.

Example

```
th {
    text-align : left;
   }
```

Vertical Alignment

The vertical-align property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

Example

```
td {
        height : 50px;
        vertical-align : bottom;
   }
```

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {
                        property: value;
                        }
```

Example

```
/* unvisited link */
a:link {
        color: #FF0000;
        }

/* visited link */
a:visited {
         color: #00FF00;
```

```
        }

/* mouse over link */
a:hover {
         color: #FF00FF;
        }

/* selected link */
a:active {
        color: #0000FF;
        }

p:first-child {
                 color: blue;
                }
```