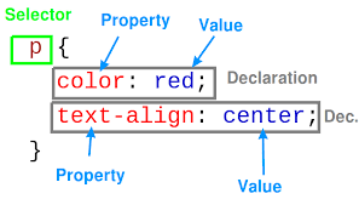
CSS

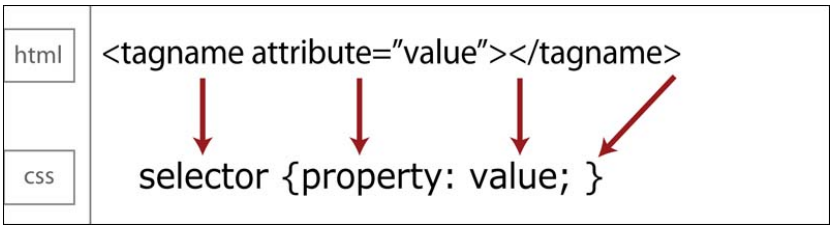
Table of Contents

Intro

- HTML was never intended to contain tags for formatting a document it was intended to define the content of a document.
- Formating tags like <font> & color attributes were added to HTML 3.2
- XHTML necessarily mixes style and content, but css help the separation.
- CSS : Cascaded Style Sheets
- CSS provides the means to control and change presentation of HTML documents separately from declaring content.
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents.
- Current Version → CSS3

CSS Structure





Levels of CSS

1. Inline

- Apply only to the specified tag
- Appears in the tag itself
- Not recommended
- Inline styling is useful for applying a unique style to a single HTML element

```
<p style="font-family:courier"> This is a paragraph. </p>
```

2. Document-Level

- Apply to the whole document
- Appears in the `<style>` element (in `<head>` of document)
- The `<style>` tag can include the type attribute, set to "text/css“,not required for HTML5

```
<head>
  <title> Font properties </title>
  <style type = "text/css">
    h2 {font-family: 'Times New Roman'; font-size: 24pt; font-weight: bold}
    h3 {font-family: 'Courier New'; font-size: 18pt}
  </style>
</head>
```

3. External Style Sheets (.css)

- Can ne applied to any number of documents.
- Saved as separate file.
- The link to an external style sheet MUST appear (in `<head>` of document)

CSS Hierarchy

From less to more dominante →

Browser → CSS File → Web Page → Individual Element

> Element, Pseudo element → Class, Pseudo classes, attributes → Id → Inline styles

- The closest level style sheet has precedence.

Example:

```
<style>
  h3 {background-color: red;}
  .spec {background-color:orange;}
  #new {background-color:lightskyblue; width:75%; line-height: 50px}
</style>
```

in the code, the output is lightskyblue background color, because the ID is more dominants in CSS Hierarchy.

Document Style Specification

1. Simple Selector

- A CSS rule has two main parts: a selector, and one or more declarations.
- The selector is a tag name or a list of tag names, separated by commas (All occurrences of this tag will be affected.)

```
p {
color: red;
}
```

2. Tag Class Selector

- Used to allow different occurrences of the **same tag** to have **different style** specifications according to a certain class.
- A style class has a name, which is attached to a tag name.

```
<p class = "className"> </p>
```

```
/*every p with class name=className*/
p.className {
```

```
color: red;
}
```

3. Generic Class Selector

- if you want the same style to apply to **more than one** kind of tag.

```
<p class = "className"> </p>
<h1 class = "className"> </h1>
```

```
/*every element with class name=className*/
.className {
color: red;
}
```

4. ID Selector

- it allows the application of a style to **one specific element**.
- Your code will not pass validation if you use the same id on more than one element.

```
<p id = "className"> </p>
```

```
#className {
color: red;
}
/*or*/
p#className {}
```

5. Selector with Pseudo Classes

- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists

```
a:link {}
a:visited {}
a:hover {}
a:focus {}
a:first-line {}
a:first-letter {}
```

6. Contextual Selectors

- List element hierarchy and this will apply style only to child elements in specified position in body of document.

```
<p>
Hello
<em> there </em>
<b>student </b>
</p>
```

```
p em {color:red}
p b {color:blue}
```

7. Universal Selectors

- The * selector selects all elements.
- The * selector can also select all elements inside another element

```
* {
color: red;
}
```

External style sheets

- The `<link>` tag goes inside the `<head>`
- You can have more than link tag in the head tag which means you can have more than one style sheets

```
<link rel = "stylesheet" type = "text/css" href = "http://www.wherever.org/example.css" >
```

`<div>` vs. `<span>`

div

- block-level element
- Support align attribute
- Used mainly with CSS in layouts
- div tags can not be contained within any other element

span

- inline element
- No align attribute
- Used to stylise text

Conflict Resolution

📌 A conflict occurs when there are two or more values for the same property on the same element

Sources:

1. Conflicting values between levels of style sheets
2. Within one style sheet
3. Inheritance
4. Property values can come from style sheets written by the document author, the browser user, and the browser defaults

Resolution mechanisms:

1. Precedence rules for the different levels of style sheets
2. Source of the property value
3. The specificity of the selector used to set the property value
4. Property value specifications can be marked to indicate their weight (importance)

```
p {
color: red !important;
}
```

A rule that has the !important directive will always be applied no matter where that rule appears.

Conflict Resolution (Multistage process called the cascade):

1. Gather all of the style specs from the different levels of style sheets
2. All available specs, from all sources, are sorted by origin and weight, using the following rules, which are given in precedence order: (Type of declaration, with which origin)

> 📌 Important Declaration, user origin → Important, author → Normal, author → Normal, user → any, browser

3. If any conflict remain, sort by Specificity: (Selectors)

> 📌 ID → Class & pseudo-class → Contextual → Universal

4. If any conflict remain, resolve by precedence to the most recent. (Example: the paragraph text will be black)

```
p { color: red; }
p { color: black; }
```

Inheritance

```
<h1 style="color:red;">
How <em>are</em> you ?
</h1>
```

in this code, the color property in inherited to “are” if no color has been assigned to the em element.

The value of any property will be calculated by a browser in the following order:-

1. Cascaded style value of the property use it
2. The value in inherited is use it
3. the default value of the browser

```
p {
color: #000;
}
p a:link {
color: inherit;
}
```

Notice the value inherit is applied to change the link color from blue (default) to #000 as parent value.

Note: When you're using shorthand notation such as background → you can't inherit other values

---

Values of CSS Properties

1. Length

Numeric value + unit

Example of units:

1. in (inch), cm, mm
2. em: used in font size (1em = current font size, default = 16px)
3. ex: height of x letter
4. pt (point): 1/72 inch
5. px (pixels)

2. Color

1. Color name: white, ..
2. Hex form: #xxxxxx
3. rgb(n1, n2, n3): maximum value of each is 255

3. URL

📌 url(protocol://server/pathname)

4. Keyword

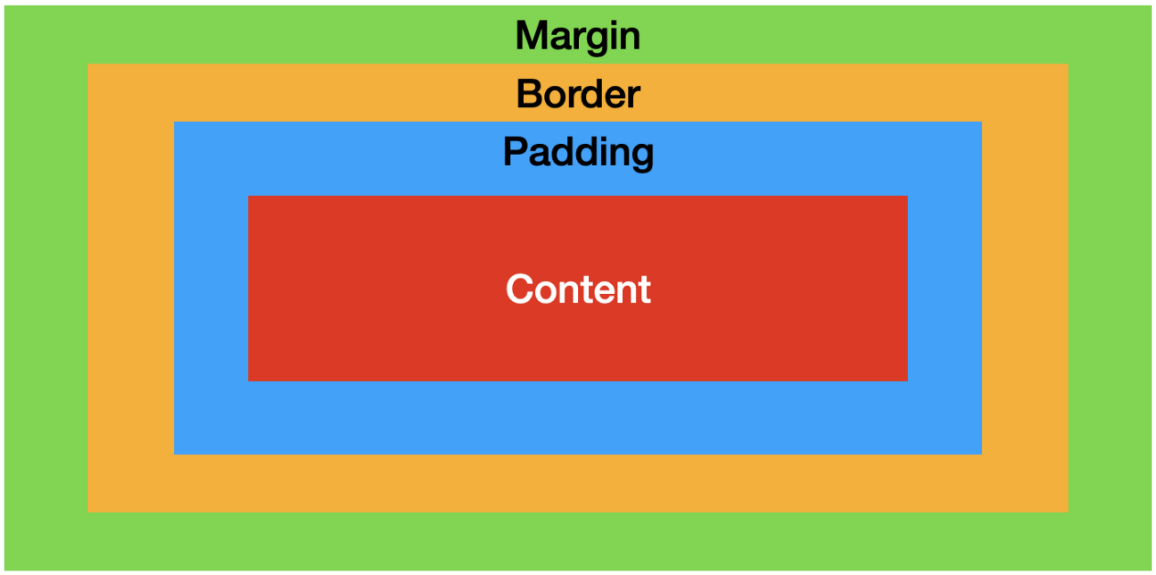
A keyword value is usually a single word that is translated into a numerical value by a browser.

5. Percentage

Instead of using a unit of measurement many properties can also take percentage values.

The Box Model

Each element in a document is considered a box.



Content: where text and images appear.

Padding: clear area around the content. The padding is affected by the background color of the box.

Border: it’s affected by the background color of the box.

Margin: clear area, it does not have a background color, it is completely transparent.

Content & Overflow

```
width:  ;
height:  ;
/*if the 2 dimensions made
the content exceed boundaries,
the element is larger if we not
defiened overflow property*/

overflow: ;
/*values: visible,
hidden,
scroll,
auto*/
```

Padding

```
padding: ;
/*applies to all 4 direction*/
padding-top: ;
/*and the other directions*/
```

Note: if padding has 4 values, order is:
top → right → bottom → left

if 3 values,
top → left & right → bottom

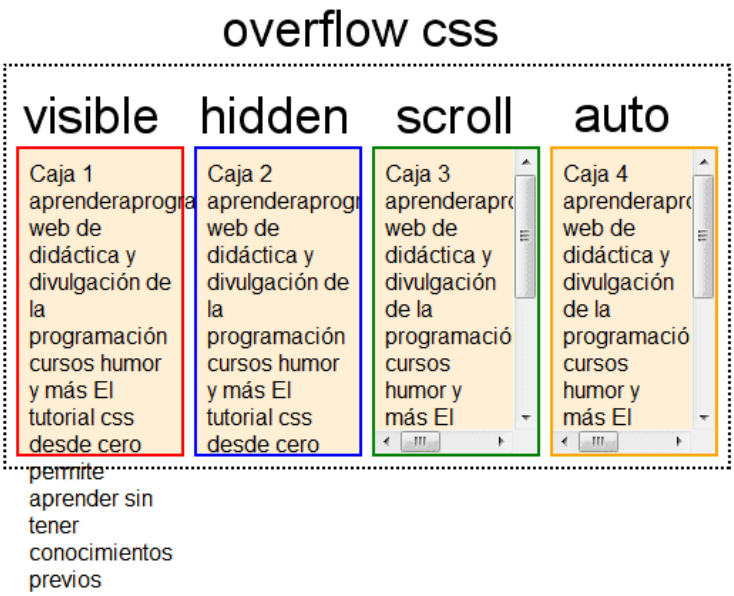
Borders

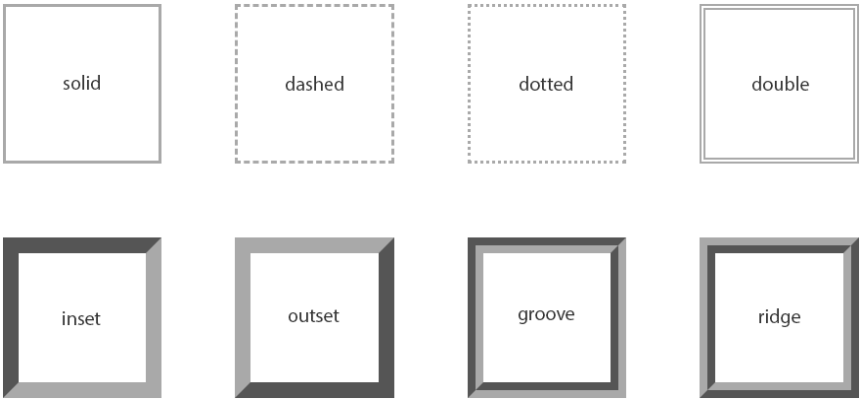
```
border-style:  ;
/*values: none, dotted,
dashed, double, groove, solid*/

border-width:  ;
/*values; thin, medium (default)
thick or a length value in px*/

border-color:  ;
border: width style color;
/*SHORTHAND*/
```

if 2 values,
top & bottom → right & left

Margins

```
margin:  ;
/*applies to all 4 direction*/
margin-left:  ;
/*and the other directions*/
/*it takes length value*/
```



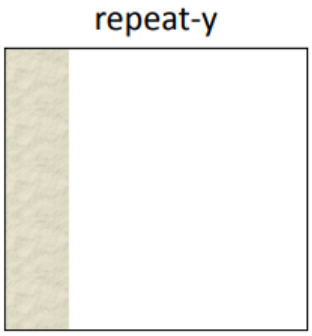


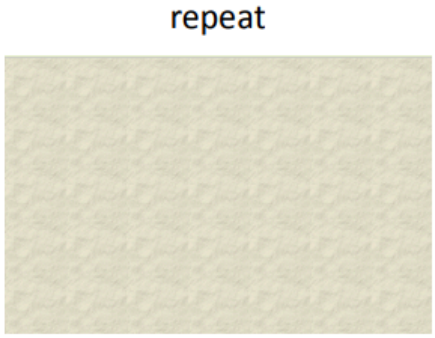
CSS Properties

Note: Shorthand property is a property where you can define more than one property for an category in one line.

Background

```
background-image:  ;
/*url or none*/

background-color:  ;
background-size:  ;
/*could be in pixels or percentages*/

background-repeat:  ;
/*values: repeat (default), no repeat, repeat-x, repeat-y*/

background-attachment:  ;
/*Sets whether a background image is fixed or scrolls with the rest of
the page [scroll-fixed]*/

background-position:  ;
/*Sets the starting position of a background image*/
/*values: left, top-left, center-left, bottom-right, ....*/

background: color image repeat attachment position;
/*SHORTHAND*/
```





Font

```
font-family:  ;
/*font-family: Arial, Helvetica, Futura*/
/*browser uses the first in the list*/
/*If the font name has more than one word, the whole
name should be delimited by single quotes*/

font-size:  ;
/*it can be length number in (pt) or a keyword*/

font-style:  ;
/*italic or normal*/

font-weight:  ;
/*it's the degree of boldness*/
/*bolder, lighter, bold, normal*/

font-variant:  ;
/*small-caps, normal*/

font: style variant weight size font name(s) ;
/*SHORTHAND*/
/*size and font names are REQUIRED*/
```

List

```
list-style-type:  ;
list-style-image:  ;
list-style: ;
```

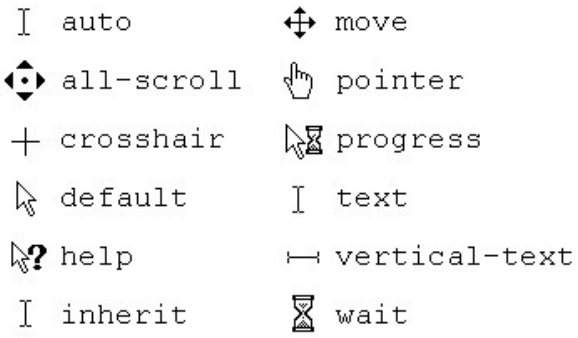
Text

```
color:  ;

text-align:  ;
/*the horizontal alignment of text*/
/*values: center, left or right*/

text-shadow:  ;

text-indent:  ;
/*indentation of the first line in px*/

text-decoration:  ;
/*values: line-through, overline, underline, blink or none*/

vertical-align:  ;
/*the vertical alignment of text*/
/*values: sub (for subscript), super (for superscript) or
baseline (default)*/
```

Cursor

Used to change the mouse cursor icon

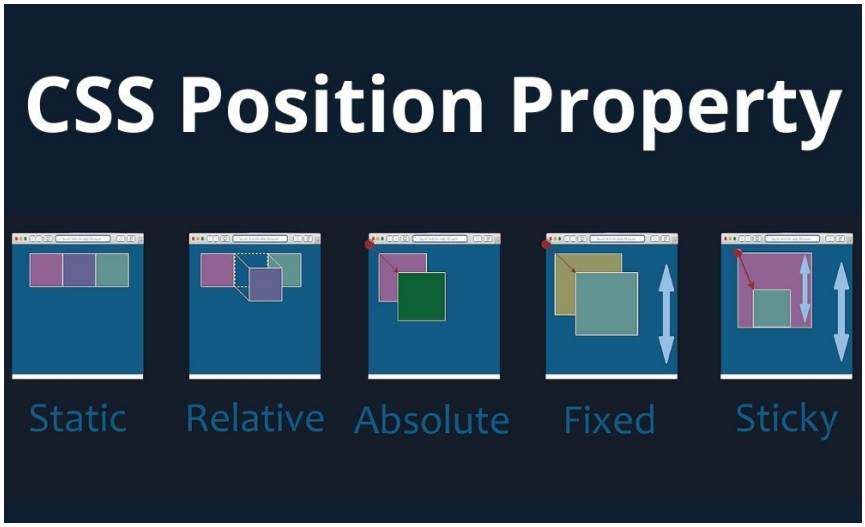
```
p {
  cursor:  ;
/*url(cursorPic.cur)*/
}
```



Position

You can control position of element through the following properties:

```
position:  ;
/*values: static, absolute, relative or fixed*/
/*static -> elements are in order as document flow (default)
absolute -> sets the left, top margin edge for a positioned box
relative -> -If no offsets properties (left, top, right, bottom)
are specified, the element is placed exactly where it would have been
placed if no position property were given
fixed -> not affected with scrolling the page*/
left:  ;
top:  ;
z-index:  ;
/*sets the stack order of an element, elements with higher z-index are
in front of elements with lower z-index*/
/*value could be in negative*/
```



Superscript vs. Suberscript effect

```
position: relative;
top: -1ex;
```

```
position: relative;
bottom: -1ex;
```

Float

Used when we want text to flow around another element

Clear

No floating elements allowed on the left or the right side of a specified element

```
float: ;
/*values: left, right, none (default)*/
```

```
clear:  ;
/*values: left, right, both,
none (default)*/
```

Direction

text direction/writing direction

```
direction:  ;
/*ltr: left to right
or
rtl: right to left*/
```

Display

Changing How an Element is Displayed

```
display:  ;
/*inline, block or
none (element is not displayed at all*/
```