Cascading Style sheet Key

Styles can be specified:

- inside an HTML element (*inline styles*)
- inside the head section of an HTML page (*internal style sheets*)
- in an external CSS file (*external style sheets*)

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

Cascading order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).

 Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

All CSS Background Properties

Property	Description
background	Sets all the background properties in one declaration
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-position	Sets the starting position of a background image
background-repeat	Sets how a background image will be repeated

All CSS Text Properties

Property	Description
color	Sets the color of text
direction	Specifies the text direction/writing direction
letter-spacing	Increases or decreases the space between characters in a text
line-height	Sets the line height
text-align	Specifies the horizontal alignment of text
text-decoration	Specifies the decoration added to text
text-indent	Specifies the indentation of the first line in a text-block
text-shadow	Specifies the shadow effect added to text
text-transform	Controls the capitalization of text
vertical-align	Sets the vertical alignment of an element
white-space	Specifies how white-space inside an element is handled
word-spacing	Increases or decreases the space between words in a text

All CSS Font Properties

Property	Description
font	Sets all the font properties in one declaration
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-weight	Specifies the weight of a font

Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

Special for links are that they can be styled differently depending on what state they are in.

The four links states are:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

When setting the style for several link states, there are some order rules:

- a:hover MUST come after a:link and a:visited
- a:active MUST come after a:hover

List

In HTML, there are two types of lists:

- unordered lists - the list items are marked with bullets
- ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

All CSS List Properties

Property	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies if the list-item markers should appear inside or outside the content flow
list-style-type	Specifies the type of list-item marker

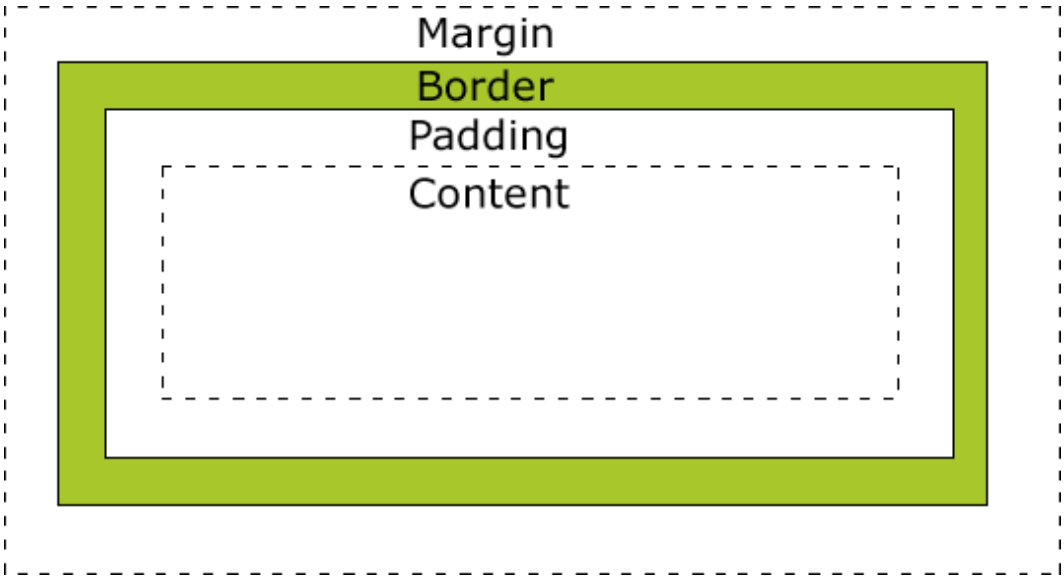
The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.

The image below illustrates the box model:



Explanation of the different parts:

- Margin - Clears an area around the border. The margin does not have a background color, it is completely transparent
- Border - A border that goes around the padding and content. The border is affected by the background color of the box
- Padding - Clears an area around the content. The padding is affected by the background color of the box
- Content - The content of the box, where text and images appear

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

Width and Height of an Element

Important: When you specify the width and height properties of an element with CSS, you are just setting the width and height of the content area. To know the full size of the element, you must also add the padding, border and margin.

The total width of an element should always be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should always be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Border Style

The border-style property can have from one to four values.

- border-style:dotted solid double dashed;
  - top border is dotted
  - right border is solid
  - bottom border is double
  - left border is dashed
- border-style:dotted solid double;
  - top border is dotted
  - right and left borders are solid
  - bottom border is double
- border-style:dotted solid;
  - top and bottom borders are dotted
  - right and left borders are solid
- border-style:dotted;
  - all four borders are dotted

All CSS Border Properties

Property	Description
border	Sets all the border properties in one declaration
border-bottom	Sets all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-color	Sets the color of the four borders
border-left	Sets all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-right	Sets all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-style	Sets the style of the four borders
border-top	Sets all the top border properties in one declaration
border-top-color	Sets the color of the top border
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders

Margin

The CSS margin properties define the space around elements.

The margin clears an area around an element (outside the border). The margin does not have a background color, and is completely transparent.

The top, right, bottom, and left margin can be changed independently using separate properties. A shorthand margin property can also be used, to change all margins at once.

Possible Values

Value	Description
auto	The browser sets the margin. The result of this is dependant of the browser
length	Defines a fixed margin (in pixels, pt, em, etc.)
%	Defines a margin in % of the containing element

 It is possible to use negative values, to overlap content.

The margin property can have from one to four values.

- margin:25px 50px 75px 100px;
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px
- margin:25px 50px 75px;
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px
- margin:25px 50px;
  - top and bottom margins are 25px
  - right and left margins are 50px
- margin:25px;
  - all four margins are 25px

All CSS Margin Properties

Property	Description
margin	A shorthand property for setting the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element

margin-top	Sets the top margin of an element

Padding

The CSS padding properties define the space between the element border and the element content.

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element.

The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property can also be used, to change all paddings at once.

Possible Values

Value	Description
length	Defines a fixed padding (in pixels, pt, em, etc.)
%	Defines a padding in % of the containing element

The padding property can have from one to four values.

- padding:25px 50px 75px 100px;
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px
- padding:25px 50px 75px;
  - top padding is 25px
  - right and left paddings are 50px
  - bottom padding is 75px
- padding:25px 50px;
  - top and bottom paddings are 25px
  - right and left paddings are 50px
- padding:25px;
  - all four paddings are 25px

All CSS Padding Properties

Property	Description
padding	A shorthand property for setting all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element

All CSS Dimension Properties

The CSS dimension properties allow you to control the height and width of an element.

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
height	Sets the height of an element	auto length % inherit	1
max-height	Sets the maximum height of an element	none length % inherit	2
max-width	Sets the maximum width of an element	none length % inherit	2
min-height	Sets the minimum height of an element	length % inherit	2
min-width	Sets the minimum width of an element	length % inherit	2
width	Sets the width of an element	auto length % inherit	1

Display and Visability

Hiding an Element - display:none or visibility:hidden

Hiding an element can be done by setting the display property to "none" or the visibility property to "hidden". However, notice that these two methods produce different results:

visibility:hidden hides an element, but it will still take up the same space as before. The element will be hidden, but still affect the layout.

display:none hides an element, and it will not take up any space. The element will be hidden, and the page will be displayed as the element is not there:

CSS Display - Block and Inline Elements

A block element is an element that takes up the full width available, and has a line break before and after it.

Examples of block elements:

- <h1>
- <p>
-

An inline element only takes up as much width as necessary, and does not force line breaks.

Examples of inline elements:

-
- <a>

Changing How an Element is Displayed

Changing an inline element to a block element, or vice versa, can be useful for making the page look a specific way, and still follow web standards.

Note: Changing the display type of an element changes only how the element is displayed, NOT what kind of element it is. For example: An inline element set to display:block is not allowed to have a block element nested inside of it.

Positioning

The CSS positioning properties allow you to position an element. It can also place an element behind another, and specify what should happen when an element's content is too big.

Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

There are four different positioning methods.

Static Positioning

HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page.

Static positioned elements are not affected by the top, bottom, left, and right properties.

Fixed Positioning

An element with fixed position is positioned relative to the browser window.

Note: IE7 and IE8 support the fixed value only if a !DOCTYPE is specified.

Fixed positioned elements are removed from the normal flow. The document and other elements behave like the fixed positioned element does not exist.

Fixed positioned elements can overlap other elements.

Relative Positioning

A relative positioned element is positioned relative to its normal position.

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

Relatively positioned elements are often used as container blocks for absolutely positioned elements.

Absolute Positioning

An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>

Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

Absolutely positioned elements can overlap other elements.

Overlapping Elements

When elements are positioned outside the normal flow, they can overlap other elements.

The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order

An element with greater stack order is always in front of an element with a lower stack order.

Note: If two positioned elements overlap, without a z-index specified, the element positioned last in the HTML code will be shown on top.

All CSS Positioning Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
bottom	Sets the bottom margin edge for a positioned box	auto length % inherit	2
clip	Clips an absolutely positioned element	shape auto inherit	2
cursor	Specifies the type of cursor to be displayed	url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	2
left	Sets the left margin edge for a positioned box	auto length %	2

		inherit	
overflow	Specifies what happens if content overflows an element's box	auto hidden scroll visible inherit	2
position	Specifies the type of positioning for an element	absolute fixed relative static inherit	2
right	Sets the right margin edge for a positioned box	auto length % inherit	2
top	Sets the top margin edge for a positioned box	auto length % inherit	2
z-index	Sets the stack order of an element	number auto inherit	2

What is CSS Float?

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is very often used for images, but it is also useful when working with layouts.

How Elements Float

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

The elements after the floating element will flow around it.

The elements before the floating element will not be affected.

If an image is floated to the right, a following text flows around it, to the left

Floating Elements Next to Each Other

If you place several floating elements after each other, they will float next to each other if there is room.

Turning off Float - Using Clear

Elements after the floating element will flow around it. To avoid this, use the clear property.

The clear property specifies which sides of an element other floating elements are not allowed.

Add a text line into the image gallery, using the clear property

All CSS Float Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

Property	Description	Values	CSS
clear	Specifies which sides of an element where other floating elements are not allowed	left right both none inherit	1
float	Specifies whether or not a box should float	left right none inherit	1

Horizontal Align

In CSS, several properties are used to align elements horizontally.
Aligning Block Elements

A block element is an element that takes up the full width available, and has a line break before and after it.

Examples of block elements:

- <h1>
- <p>
-

For aligning text, see the CSS Text chapter.

In this chapter we will show you how to horizontally align block elements for layout purposes.

Center Aligning Using the margin Property

Block elements can be aligned by setting the left and right margins to "auto".

Note: Using margin:auto will not work in IE8 and earlier, unless a !DOCTYPE is declared.

Setting the left and right margins to auto specifies that they should split the available margin equally. The result is a centered element

Tip: Aligning has no effect if the width is 100%.

Note: In IE5 there is a margin handling bug for block elements. To make the example above work in IE5, add some extra code

Left and Right Aligning Using the position Property

One method of aligning elements is to use absolute positioning

Crossbrowser Compatibility Issues

When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is a problem with IE8 and earlier, when using the position property. If a container element (in our case) has a specified width, and the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the position property

Left and Right Aligning Using the float Property

One method of aligning elements is to use the float property

Crossbrowser Compatibility Issues

When aligning elements like this, it is always a good idea to predefine margin and padding for the <body> element. This is to avoid visual differences in different browsers.

There is a problem with IE8 and earlier when using the float property. If the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. Always set the !DOCTYPE declaration when using the float property

CSS pseudo-classes

CSS pseudo-classes are used to add special effects to some selectors.
Syntax

The syntax of pseudo-classes:

selector:pseudo-class {property:value;}

CSS classes can also be used with pseudo-classes:

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

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective!!

Note: a:active MUST come after a:hover in the CSS definition in order to be effective!!

Note: Pseudo-class names are not case-sensitive.

All CSS Pseudo Classes/Elements

Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
:first-letter	p:first-letter	Selects the first letter of every <p> element
:first-line	p:first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:before	p:before	Insert content before every <p> element
:after	p:after	Insert content after every <p> element
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

Navigation Bars

Having easy-to-use navigation is important for any web site.

With CSS you can transform boring HTML menus into good-looking navigation bars.

Navigation Bar = List of Links

A navigation bar needs standard HTML as a base.

In our examples we will build the navigation bar from a standard HTML list.

A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense

Example:

```
<ul>
<li><a href="default.asp">Home</a></li>
<li><a href="news.asp">News</a></li>
<li><a href="contact.asp">Contact</a></li>
<li><a href="about.asp">About</a></li>
</ul>
```

Now let's remove the bullets and the margins and padding from the list:

```
ul
{
list-style-type:none;
margin:0;
padding:0;
}
```

Example explained:

- list-style-type:none - Removes the bullets. A navigation bar does not need list markers
- Setting margins and padding to 0 to remove browser default settings

The code in the example above is the standard code used in both vertical, and horizontal navigation bars.

Vertical Navigation Bar

To build a vertical navigation bar we only need to style the <a> elements, in addition to the code above

```
a
{
display:block;
width:60px;
}
```

Example explained:

- display:block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width:60px - Block elements take up the full width available by default. We want to specify a 60 px width
- Note: Always specify the width for <a> elements in a vertical navigation bar. If you omit the width, IE6 can produce unexpected results.

Horizontal Navigation Bar

- There are two ways to create a horizontal navigation bar. Using inline or floating list items.
- Both methods work fine, but if you want the links to be the same size, you have to use the floating method.

Inline List Items

- One way to build a horizontal navigation bar is to specify the <li> elements as inline, in addition to the "standard" code above

```
li
{
display:inline;
}
```

Example explained:

- display:inline; - By default, <li> elements are block elements. Here, we remove the line breaks before and after each list item, to display them on one line

Floating List Items

In the example above the links have different widths.

For all the links to have an equal width, float the <li> elements and specify a width for the <a> elements:

```
li
{
float:left;
}
a
{
display:block;
width:60px;
}
```

Example explained:

- float:left - use float to get block elements to slide next to each other
- display:block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width:60px - Since block elements take up the full width available, they cannot float next to each other. We specify the width of the links to 60px

Visit the W3 School's website for more information and css coding
http://www.w3schools.com/CSS/default.asp