

CSS Vocabulary and Grammar Basics

- CSS Stands for Cascading Style Sheet
- CSS is a language system that allows you to define the presentation of elements in HTML
- The Presentation of HTML elements has two aspects to it:
 - Formatting – the way an element appears. Examples of this would be the font a piece a text is displayed in or a border placed around an image.
 - Positioning – the way an element fits into a document. For example how a sidebar is positioned in a document
- There are three types of styles that can be used in HTML documents:
 - Tag styles – this type of style controls the presentation of HTML elements formatted with a particular tag. For example a “p” tag style controls the formatting of a paragraph.
 - Class styles – A class is a style that can be applied to any HTML element. It is applied by adding the “class” attribute to an HTML element. For example you could create a class style called “blue-text” that formats the element it is applied to with a blue font color regardless of the tag it is applied to `<h1 class= “blue-text”>` and `<p class=“blue-text”>` will format both elements as blue text.
 - ID styles – An ID style is a style that is applied to an element on an HTML page. Since an ID can only be applied to one element on a page it is a unique style for the element. For example `<div id=“sidebar”>` would apply the “sidebar” style to the div that references it.
 - CSS Style names are case sensitive.
`class=“bluetext”` is different from `class=“blueText”`
- There are three different locations styles can be defined in:
 - Inline – this is when a style is defined within the element it affects. You can only define class styles in this way not tag or id styles. Inline styles are almost never used because of various limitations and problem associated with them.
 - Internal – this is when a style is defined in the `<head>` section of the HTML document and then applied to elements in the `<body>` section. You can define tag, id and class styles as internal styles. Internal styles avoid some of the issues related to inline styles but still have significant limitations are not considered best practice when working with styles.
 - External – External styles are styles defined in a separate document and then HTML pages are linked to this separate files. External style sheets allow for the full benefit of styles and are the best way to work with styles.

- When you define a style you name them with the following rules:
 - Class styles prefix the style with a “.” period. So the class style “bluetext” would be named as .bluetext
 - ID styles are prefixed with the “#” number symbol. So the id style “sidebar” would be defined as #sidebar
 - Tag styles do not have a prefix and are written with the plain tag name such as p or h1
- CSS styles are written in a specific format:

```
stylename {  
    property:value;  
    property:value;  
    property:value;  
}
```

- The stylename is followed by a space and then an opening curly bracket “{”
- On the next line you will enter in a property-value statement.
- The property is separated from the value by a colon “:”
- The statement ends after the value with a semicolon “;”
- On the final line you will place a closing curly bracket “}”
- You can have as many or few property values statements as you need in a single style. Remember to end each property value statement with a semicolon “;”
- You can group multiple selectors together to format them with the same style by separating the different selectors with commas

```
h1, h2, h3 {  
    color:blue;  
    font-weight:bold;  
}
```

This would make the tags `<h1>`, `<h2>` and `<h3>` all blue and bold.

- You can specify a parent – child relationship to select particular tags and format them with a style. For example if you wanted to make `` blue and bold but only list items contained in `` ordered lists not `` unordered list you would use the following format:

```
ol li {  
    color:blue;  
    font-weight:bold;  
}
```

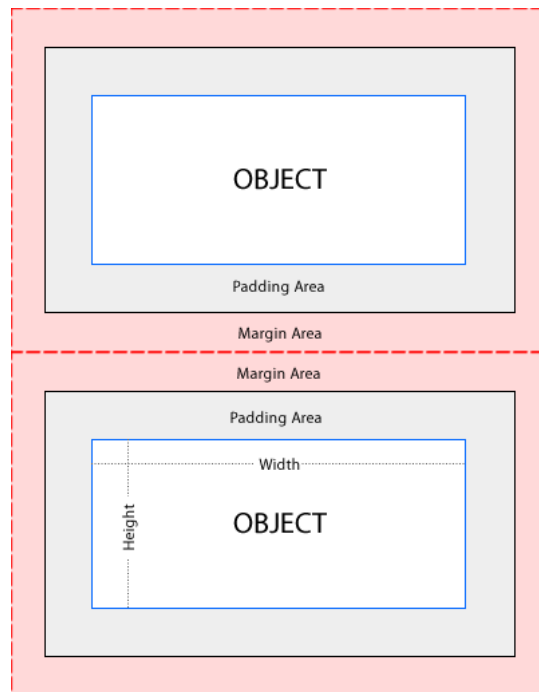
This targets any `` tags that are contained or are children of `` tags.

- Properties are characteristics of an element. For example text can have a font-family property, a font-size property and a color property. Every HTML element has many properties.
- The value is the setting for the property. For example the text property “font-size” could have the value “24px” and the text property “color” could have the value “blue”.

- There are a variety of acceptable units of measurement in CSS
 - Pixels: This is the standard fixed unit for measurement for stylesheets intended for screens. It is abbreviated “px”. For example a font size can be expressed as “24px”
 - Percentages: You can also use percentages as a proportional unit of measurement. For example you could express that the width of a layout take up 100% of the screen.
 - Ems: A em is a proportional unit of measurement that is relative to the base line of the element being measured. This is expressed in the form “10em”
 - Points: The standard unit of measurement for print targeted stylesheets. This is the same as the points used in word processors. For example if you create a print style sheet you could express the standard paragraph font size as “12pt”
- Colors can be expressed in a variety of ways.
 - Normally colors are expressed in hexadecimal format in a three part two digit format prefixed by a number sign “#”.
 - Each place in the color code can have a value between zero 0 and fifteen 15. Ten is represented by the letter “A”, Eleven by the letter “B” and so on.
 - For example the standard color blue link color could be expressed as “#0000FF”
 - This six digit code represents an RGB (red, green, blue) color value and is divided into three two digit parts. The first two digits represent Red, the second two digits Green, and the final two digits Blue.
 - In the RGB color model zero “0” is the absent of color. The value fifteen “15” represented by the letter “F” represents the color being turned up to maximum.
 - For example “#FF0000” is a bright red, “#00FF00” is a bright green, “#0000FF” is a bright blue. Black is the absent of all color so it is represented by the value “#000000” and white is represented by all the color values turned to maximum “#FFFFFF”
 - If a color value has the same value for each of the places in the two digits sets it can be abbreviated. So the value “#336699” can be expressed as “#369”. This is possible but not necessary.

The CSS Box Model

- The CSS Box model is a form that represents how spacing is added around objects. The model has three different values: Padding, Border, and Margin.



- The padding is the area between the object and any border around the object. The margin is the space between the border around the object and the objects surrounding the object.

CSS Properties Vocabulary List

Text Properties	
<code>font-family</code>	The is the font the text will be displayed in. Can have multiple values. Subsequent values will be used if the preceding font is not available on the visitor's computer
<code>font-size</code>	This is the font size text is displayed in.
<code>font-weight</code>	This is the property to set a font weight to either bold or normal
<code>font-style</code>	This is the property to set a font style to italics or normal
<code>color</code>	The color of the text
<code>line-height</code>	The line spacing in between lines of text.
<code>text-align</code>	The alignment of text
<code>text-decoration</code>	The underline and underline style on text and links
Box Model Properties	
<code>width</code>	The width of an object
<code>height</code>	The height of an object
<code>padding</code>	<p>The spacing between an object and the border (if any) around the object. You can specify the location of the padding for example you can use the property - <code>padding-left</code>, <code>padding-right</code>, <code>padding-top</code>, <code>padding-bottom</code>.</p> <p>If only one value is given for the padding it is applied to all sides of the object. For example "<code>padding:10px;</code>" would place ten pixels of padding on all sides of an object. If two values are given for the padding it is applied to the top and bottom and the left and right side of an object. For example "<code>padding:10px 20px;</code>" would apply 10px to both the top and bottom and 20px on the left and right of an object. If four values are given for the padding, it is applied to the top, right, bottom and left sides of an object (clockwise). For example: "<code>padding:10px 4px 8px 12px;</code>"</p>
<code>margin</code>	The spacing between the border around an object. Follows the same rules as the padding property.
<code>border</code>	<p>Applies a border around an object. Requires three values: width, color and style. For example "<code>border:1px #000000 solid;</code>" would apply a 1px black solid border around the object.</p> <p>If you wish to specify a particular side of an object to apply the border to you can use "<code>border-top:1px #000000 solid;</code>", etc just as you can specify a side of an object for padding and margin.</p>
Other Properties	
<code>list-style-type</code>	The bulleting or numbering style for an ordered or unordered list
<code>background-image</code>	The property that sets an image as a background for an object

<code>background-repeat</code>	The property that specifies if an image background repeats horizontally or vertically or both.
<code>background-color</code>	The background color for an object
<code>display</code>	Controls the way a browser displays an element. The two standard values are <code>"display:block;"</code> , <code>"display:inline;"</code> and <code>"display:none"</code> The value "none" will cause an element not to be displayed. The value "block" forces an element to display as a block level element and allows for CSS box model properties to be applied. The value "inline" causes an element to be displayed as an inline level element. Inline element do not have breaks after them and cannot have box model properties applied to them.
<code>float</code>	Removes an element from the normal flow of elements in a document causes it to "float" either "left" or "right". For example a left side navigation sidebar would be floated to the left to allow the content area to be displayed to the right.
<code>clear</code>	Returns an element to the normal flow of the document after a floated element