CSS

CONTENTS

- INTRODUCTION
- CSS COMMENTS
- CSS COLORS
- CSS TEXT
- CSS ID AND CLASS
- POSITIONING
- OVERLAPPING ELEMENTS
- CSS ALIGN
- CROSS BROWSERS COMPATIBLE ISSUES

INTRODUSTION

* CSS stands for Cascading Style Sheets

* Styles define how to display HTML elements
* Styles were added to HTML 4.0 to solve a problem
* External Style Sheets can save a lot of work
* External Style Sheets are stored in CSS files

CSS COMMENTS

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment begins with "/*", and ends with "*/", like this:

```
/*This is a comment*/
p
{
text-align:center;
/*This is another comment*/
color:black;
font-family:arial;
}
```

CSS Colors

Colors are displayed combining RED, GREEN, and BLUE light.

COLOR VALUES

CSS colors are defined using a hexadecimal (hex) notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one of the light sources is 0 (hex 00). The highest value is 255 (hex FF).

Hex values are written as 3 double digit numbers, starting with a # sign.

CSS TEXT

Text Color

The color property is used to set the color of the text. The color can be specified by:

* name - a color name, like "red"
* RGB - an RGB value, like "rgb(255,0,0)"
* Hex - a hex value, like "#ff0000"

The default color for a page is defined in the body selector.

Example

```
body {color:blue;}
h1 {color:#00ff00;}
h2 {color:rgb(255,0,0);}
```

Text Alignment

The text-align property is used to set the horizontal alignment of a text. Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight

Example

```
h1 {text-align:center;}
p.date {text-align:right;}
p.main {text-align:justify;}
```

Text Decoration

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

Example

```
a {text-decoration:none;}
```

Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

Example

```
p.uppercase {text-transform:uppercase;}
p.lowercase {text-transform:lowercase;}
p.capitalize {text-transform:capitalize;}
```

Text Indentation

The text-indentation property is used to specify the indentation of the first line of a text.

Example

```
p {text-indent:50px;}
```

CSS Example

```
body
{
background-color:#d0e4fe;
}
h1
{
color:orange;
text-align:center;
}
p
{
font-family:"Times New Roman";
font-size:20px;
}
```

CSS ID AND CLASS

The id Selector

The id selector is used to specify a style for a single, unique element.

The id selector uses the id attribute of the HTML element, and is defined with a "#".

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

Example

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

The c lass Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

This allows you to set a particular style for any HTML elements with the same class.

The class selector uses the HTML class attribute, and is defined with a "."

In the example below, all HTML elements with class="center" will be center-aligned:

Example

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

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.

It will not move even if the window is scrolled:

Example

```
p.pos_fixed
{
position:fixed;
top:30px;
right:5px;
}
```

Relative Positioning

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

Example

```
h2.pos_left
{
position:relative;
left:-20px;
}
h2.pos_right
{
position:relative;
left:20px;
}
```

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>:

Example

```
h2
{
position:absolute;
left:100px;
top:150px;
}
```

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:

Example

```
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1 }
```

CSS Align

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>
*

Center Aligning

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

Note: Using margin:auto will not work in Internet Explorer, 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:

Example

```
. center {
  margin-left:auto;
  margin-right:auto;
  width:70%;
  background-color:#b0e0e6; }
```

Left and Right Aligning

Using the position Property

Example

```
.right
{
position:absolute;
right:0px;
width:300px;
background-color:#b0e0e6;
}
```

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 also another problem with IE when using the position property. If a container element (in our case) has a specified width, and the !DOCTYPE declaration is missing, IE 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:

Example

```
body{
margin:0;
padding:0;}
.container{
position:relative;
width:100%;
}
.right
{
position:absolute;
right:0px;
width:300px;
background-color:#b0e0e6;
}
```

THANK YOU