**3. Page designing with CSS**

3.1 Introduction to designing approaches

**3.1.1 Table based design**

**T**he design strategy for web structure where table is used for building a web pages is called Table base design. But because of the complexity of the web documents the developers are compelled to move to div based web structure. The maintainability of the table based design is difficult for the table based web design, search engine optimization is also difficult task for the table based design so anther new technique for web design CSS was introduced in 1996 for building website using div structure called table-less design.

**Example**

<table c ellpadding="0" cellspacing="0" border="0">

<tr>

<td colspan="3" height="120px">....</td>

</tr>

<tr>

<td class="menu" valign="top">...</td>

<td class="content" valign="top">...</td>

<td class="aSide" valign="top">...</td>

</tr>

<tr>

<td colspan="3">...</td>

</tr>

</table>

**3.1.2 Table-less design**

**Table-less web design** (or **table-less web layout** or **div based web design**) is a [web design](http://en.wikipedia.org/wiki/Web_design) philosophy eschewing(avoiding) the use of [HTML tables](http://en.wikipedia.org/wiki/HTML_table) for [page layout](http://en.wikipedia.org/wiki/Page_layout) control purposes. Instead of HTML tables, [style sheet](http://en.wikipedia.org/wiki/Style_sheet_(web_development)) languages such as [Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) are used to arrange elements and text on a [web page](http://en.wikipedia.org/wiki/Web_page).CSS was introduced in December 1996 by the [W3C](http://en.wikipedia.org/wiki/World_Wide_Web_Consortium) to improve [web accessibility](http://en.wikipedia.org/wiki/Web_accessibility) and to make [HTML](http://en.wikipedia.org/wiki/HTML) code purely [semantic](http://en.wikipedia.org/wiki/Semantics#Computer_science) rather than presentational. Around the same time, in the late 1990s, as the [dot-com boom](http://en.wikipedia.org/wiki/Dot-com_bubble) led to a rapid growth in the "[new media](http://en.wikipedia.org/wiki/New_media)" of web page creation and design, there began a trend of using HTML tables, and their rows, columns and cells, to control the layout of whole web pages. This was due to several reasons:

* the limitations at the time of CSS support in browsers;
* the new web designers' lack of familiarity with CSS;
* the lack of knowledge of, or concern for the reasons (including HTML [semantics](http://en.wikipedia.org/wiki/Semantic_Web) and [web accessibility](http://en.wikipedia.org/wiki/Web_accessibility)) to use CSS instead of what was perceived as an easier way to quickly achieve the intended layouts, and

The advantages of restricting the use of HTML tables to their intended and semantic purpose include improved accessibility of the information to a wider variety of users, using a wide variety of [user agents](http://en.wikipedia.org/wiki/User_agent). There are bandwidth savings as large numbers of semantically meaningless <table>, <tr> and <td> tags are removed from dozens of pages leaving fewer, but more meaningful headings, paragraphs and lists. Layout instructions are transferred into site-wide CSS style sheets, which can be [downloaded](http://en.wikipedia.org/wiki/HTTP) once and [cached](http://en.wikipedia.org/wiki/Web_cache) for reuse while each visitor navigates the site. Sites may become more maintainable as the whole site can be restyled or re-branded in a single pass merely by altering the mark-up of the specific CSS, affecting every page which relies on that style sheet. New HTML content can be added in such a way that consistent layout rules are immediately applied to it by the existing CSS without any further effort.

Some developers are now afraid to introduce a simple HTML table even where it makes good sense, some erring by the overuse of [span and div](http://en.wikipedia.org/wiki/Span_and_div) elements, perhaps even with table-like rules applied to them using CSS.

Advantages

**Accessibility**

Because of the Internet's rapid growth, expanding [disability discrimination legislation](http://en.wikipedia.org/wiki/Disability_discrimination_act), and the increasing use of [mobile phones](http://en.wikipedia.org/wiki/Mobile_phone) and [PDAs](http://en.wikipedia.org/wiki/Personal_digital_assistant), it is necessary for Web content to be made accessible to users operating a wide variety of devices. Table-less Web design considerably improves [Web accessibility](http://en.wikipedia.org/wiki/Web_accessibility) in this respect. [Screen readers](http://en.wikipedia.org/wiki/Screen_reader) and [braille](http://en.wikipedia.org/wiki/Braille) devices have fewer problems with table-lessdesigns because they follow a logical structure. The same is true for [search engine](http://en.wikipedia.org/wiki/Web_search_engine) [Web crawlers](http://en.wikipedia.org/wiki/Web_crawler), the software agents that most web site publishers hope will find their pages, classify them accurately and so enable potential users to find them easily in appropriate searches. As a result of the separation of design (CSS) and structure (HTML), it is also possible to provide different layouts for different devices, e.g. [handhelds](http://en.wikipedia.org/wiki/Handheld_device), mobile phones, etc. It is also possible to specify a different style sheet for print, e.g. to hide or modify the appearance of advertisements or navigation elements that are irrelevant and a nuisance in the printable version of the page.

**Bandwidth savings**

Table-less design produces web pages with fewer HTML tags used purely to position content. This normally means that the pages themselves become smaller to download. The philosophy implies that all the instructions regarding layout and positioning be moved into external style sheets. According to the basic capabilities of [HTTP](http://en.wikipedia.org/wiki/HTTP), as these rarely change and they apply in common to many web pages, they will be cached and reused after the first download. This further reduces bandwidth and download times across the site.

**Maintainability**

Maintaining a website may require frequent changes, both small and large, to the visual style of a website, depending on the purpose of the site. Under table-based layout, the layout is part of the HTML itself. As such, without the aid of template-based visual editors such as [HTML editors](http://en.wikipedia.org/wiki/HTML_editor), changing the positional layout of elements on a whole site may require a great deal of effort, depending on the amount of repetitive changes required. Even employing [sed](http://en.wikipedia.org/wiki/Sed" \o "Sed) or similar global find-and-replace utilities cannot alleviate the problem entirely. In table-less layout using CSS, the layout information may reside in a CSS document. Because the layout information may be centralized, it is possible that these changes can be made quickly and globally by default.

**Example**

<div id="header">...</div>

<div id="menu">...</div>

<div id="content">...</div>

<div id="aSide">...</div>

<div id="footer">...</div>

**3.2 Cascading Style Sheets** (**CSS**)

3.2.1 **Introduction** It is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [look and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language). While most often used to change the style of [web pages](http://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](http://en.wikipedia.org/wiki/HTML) and [XHTML](http://en.wikipedia.org/wiki/XHTML), the language can be applied to any kind of [XML](http://en.wikipedia.org/wiki/XML) document, including [plain XML](http://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](http://en.wikipedia.org/wiki/XUL). Along with HTML and [JavaScript](http://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](http://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.CSS is designed primarily to enable [the separation of document content from document presentation](http://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](http://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers.

The CSS specifications are maintained by the [World Wide Web Consortium](http://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](http://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by [RFC 2318](http://tools.ietf.org/html/rfc2318) (March 1998).

3.2.2 CSS vs CSS3

We can produce Pixel-perfect Web page with CSS3 quicker than with older based CSS methods, and that the CSS3 page will load faster, with a smaller overall file size and fewer HTTP requests. We can design and code a Web page and add visual enhancements twice: once with CSS3, and a second time using background images sliced directly from the PSD.  [Pingdom](http://tools.pingdom.com/fpt/) can be used to measure the loading times but not in **css**.

**Difference between CSS and CSS3**

## The original Cascading Style Sheets specification, as well as its much more recent CSS3

## revision, are technologies used to enhance and format HTML Web pages. CSS was initially designed to handle the presentation layer of a Web page in a more efficient manner than formatting with HTML, which was never intended to handle the complex construction of the contemporary Web. Because CSS2 is a a universally adopted extension of CSS1, the term CSS without a number implies the inclusion of CSS2. The CSS3 spec is still under development as of July 2014, so Web developers need to consider feature support varies between browsers. CSS3 Is Backwards Compatible With CSS

* CSS3 is an update to CSS2 that maintains compatibility with all of CSS's features -- CSS3 doesn't deprecate any of the CSS code. The CSS3 code is designed to make Web pages look better and load faster as well as reduce development time to build pages in a user's browser. CSS3 makes Web design less reliant on image files for page design elements and reduces file transfer requests and download time by using fewer images. CSS1 focused on appearance formatting, whereas CSS2 added positioning capabilities for text and objects. Older browser versions like IE 9 and prior do not widely support CSS3's added features, which can require extra development time to maintain CSS fall-back code.

## CSS3 Adds Rounded Corners and Gradients

* Prior to CSS3, Web developers needed to design image files to add things like rounded corners for structural borders and background image gradients; CSS3 includes comparable features that can be added with a few lines of code. In CSS, a Web developer has to design a border or gradient, upload it to the image server, place the image on the page and use CSS to correctly position the border. In CSS3, the Web developer can accomplish something similar with code like ".roundBorder{border-radius:10px;}". Background gradients are a bit more complicated and require different code for different browsers. For example, a white-to-black gradient for Chrome and IE uses code like ".gradBG{background: liner-gradient(white,black);}".

## CSS3 Adds Animation Features and Text Effects

* CSS3 has a handful of features not present in CSS to improve how your page elements look. With CSS3, Web developers can add a text-shadow to text to make it easier to read or add visual flair and can force line breaks within longer words to make them fit inside columns with word wrap. Before CSS3, Web developers needed to code animation in scripting languages like JavaScript and jQuery; CSS3 adds several animation features to the design layer. The W3C is still working on a Web animations standard to work out compatibility issues between the different animation techniques.

## CSS3 Defines Text Columns

* CSS3 adds the capability to split text sections into multiple columns to read like a newspaper. In the CSS2 spec, Web developers have a difficult time constructing multi-column text sections because the standard is not equipped to automatically split text. Particularly with sites that feature responsive design, a development technique that rearranges and resizes content based on the viewing device's screen resolution, people with larger monitors may have a difficult time reading text rows that sprawl across the screen.

Both pages are pretty fast, but CSS3 prevailed, with 10 fewer requests and a file size that was lighter by 81.3 KB. While loading times were close, the larger PNG files used on both pages accounted for most of the heft, which amounted to a .75 second difference on average. And when we’re talking 3 to 6 second loading times, those differences sure can add up.

|  | **CSS3** | **CSS** | **Difference** |
| --- | --- | --- | --- |
| **Size** | 767.9 KB | 849.2 KB | 81.3 KB |
| **Requests** | 12 | 22 | 10 |

For argument’s sake, I created yet another version of the image-based CSS version, with a sprite containing all four images used in the original version, and then I measured loading times. This [CSS Sprited version](http://trentwalton.com/examples/CSS3_Speed_Test/css_sprited.html) *did* improve things, taking HTTP requests from 22 to 19 and the overall size from 849.2 KB down to 846.7 KB. The way I see it, these differences are minimal and would have added to the development time, so it’s all relative.

**Various style sheets are as:**

1. Inline Style (inside (X)HTML element)

2. Internal Style Sheet (inside the <head> tag)

3. External Style Sheet

**Inline Styles**

Inline styles are defined right in

the (X)HTML file along side the element we want to style. example

below.

<p style="color: #ff0000;">*Some red text*</p>

**Internal Stylesheet**

First we will explore the internal method. This way you are simply placing

the CSS code within the <head></head> tags of each (X)HTML file you

want to style with the CSS. The format for this is shown in the example

below.

<head>

<title><title>

<style type="text/css">

*CSS Content Goes Here*

</style>

</head>

<body>

With this method each (X)HTML file contains the CSS code needed to

style the page. Meaning that any changes you want to make to one

page, will have to be made to all. This method can be good if you need

to style only one page, or if you want different pages to have varying

styles.

**External Stylesheet**

Next we will explore the external method. An external CSS file can be

created with any text or HTML editor such as "Notepad" or

"Dreamweaver". A CSS file contains no (X)HTML, only CSS. You simply

save it with the .css file extension. You can link to the file externally by

placing one of the following links in the head section of every (X)HTML

file you want to style with the CSS file.

<link rel="stylesheet" type="text/css" href=*"Path To*

*stylesheet.css"* /> or

<style type="text/css">@import url(*Path To*

*stylesheet.css*)</style>

Either of these methods are achieved by placing one or the other in the

head section as shown in example below.

<head>

<title><title>

*<link rel="stylesheet" type="text/css"href="style.css" />*

</head>

<body>

or

<head>

<title><title>

<style type="text/css"> *@import url(Path To stylesheet.css)*

</style>

</head>

By using an external style sheet, all of your (X)HTML files link to one

CSS file in order to style the pages. This means, that if we need to alter

the design of all your pages, you only need to edit one .css file to make

global changes to your entire website.

Here are a few reasons this is better.

* **Easier Maintenance**
* **Reduced File Size**
* **Reduced Bandwidth**
* **Improved Flexibility**

**3.2.3 CSS properties**

## CSS Syntax

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



Font properties

p {  
    font: 15px arial, sans-serif;  
}  
  
p{  
    font: italic bold 12px/30px Georgia, serif;  
}

**Font -Family**

You can set what font will be displayed in an element with the font-family

property.

font-family: Verdana, sans-serif;

**Font Size**

You can set the size of the text used in an element by using the font size

property.

font-size: value;

There are a lot of choices for values:

xx-large

x-large

larger

large

medium

% (percent)

There is quite a bit to learn about font sizes with CSS so, I am not even

going to try to explain it. Actually there are already some great resources

on how to size your text. (see below)

font: italic bold normal small/1.4em Verdana, sans-serif;

font-family: Verd

**Text Color**

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

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

Look at [CSS Color Values](http://www.w3schools.com/cssref/css_colors_legal.asp) for a complete list of possible color values.

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

**Example**

## body {     color: blue; } 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 (like in magazines and newspapers).

## 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;  
}

It can also be used to decorate text:

## Example

h1 {  
    text-decoration: overline;  
}  
  
h2 {  
    text-decoration: line-through;  
}  
  
h3 {  
    text-decoration: underline;  
}

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

## Example

p {  
    text-indent: 50px;  
}

## Background Color

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

The background color of a page is defined in the body selector:

## Example

body {  
    background-color: #b0c4de;  
}

With CSS, a color is most often specified by:

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

In the example below, the h1, p, and div elements have different background colors:

**Example**

h1 {  
    background-color: #6495ed;  
}  
  
p {  
    background-color: #e0ffff;  
}  
  
div {  
    background-color: #b0c4de;  
}

## 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.

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

## Example

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

Below is an example of a bad combination of text and background image. The text is almost not readable:

## Example

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

## Background Image - Repeat Horizontally or Vertically

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("gradient\_bg.png");  
}

If the image is repeated only horizontally (repeat-x), the background will look better:

## Example

body {  
    background-image: url("gradient\_bg.png");  
    background-repeat: repeat-x;  
}

**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 add a border around elements, and to define space between elements.

The image below illustrates the box model:



Explanation of the different parts:

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent

**Example**

div {  
    width: 300px;  
    padding: 25px;  
    border: 25px solid navy;  
    margin: 25px;  
}

## Width and Height of an Element

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

Let's make a div element with a total width of 350px:

## Example

div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;   
}

**Let's do the mathematics:**  
320px (width)  
+ 20px (left + right padding)  
+ 10px (left + right border)  
+ 0px (left + right margin)  
= 350px

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

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

## 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 a fixed position is positioned relative to the browser window, and will not move even if the window is scrolled:

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.

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

## Relative Positioning

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

h2.pos\_left {  
    position: relative;  
    left: -20px;  
}  
  
h2.pos\_right {  
    position: relative;  
    left: 20px;  
}

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.

h2.pos\_top {  
    position: relative;  
    top: -50px;  
}

## 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>:

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

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.

**Display Property**

# CSS display Property

## Example

Display <p> elements as inline elements:

p.inline {  
    display: inline;  
}

## Example

A demonstration of how to use the inherit property value:

body {  
    display: inline  
}  
  
p {  
    display: inherit  
}

## Example

Set the direction of some flexible items inside a <div> element in reverse order:

div {  
    display: -webkit-flex; /\* Safari \*/  
    -webkit-flex-direction: row-reverse; /\* Safari 6.1+ \*/  
    display: flex;  
    flex-direction: row-reverse;  
}

# CSS Lists

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

## List

In HTML, there are two 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

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

## Different List Item Markers

The type of list item marker is specified with the list-style-type property:

## 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;  
}

Some of the values are for unordered lists, and some for ordered lists.

## An Image as The List Item Marker

To specify an image as the list item marker, use the list-style-image property:

## Example

ul {  
   list-style-image: url('sqpurple.gif');  
}

The example above does not display equally in all browsers. IE and Opera will display the image-marker a little bit higher than Firefox, Chrome, and Safari.

If you want the image-marker to be placed equally in all browsers, a crossbrowser solution is explained below.

## Crossbrowser Solution

The following example displays the image-marker equally in all browsers:

## Example

ul {  
    list-style-type: none;  
    padding: 0px;  
    margin: 0px;  
}  
  
ul li {  
    background-image: url(sqpurple.gif);  
    background-repeat: no-repeat;  
    background-position: 0px center;   
    padding-left: 15px;   
}

Example explained:

* For <ul>:
  + Set the list-style-type to none to remove the list item marker
  + Set both padding and margin to 0px (for cross-browser compatibility)
* For all <li> in <ul>:
  + Set the URL of the image, and show it only once (no-repeat)
  + Position the image where you want it (left 0px and vertical value: center)
  + Position the text in the list with padding-left

## List - Shorthand property

The list-style property is a shorthand property. It is used to set all the list properties in one declaration:

## Example

ul {  
    list-style: square inside url("sqpurple.gif");  
}

When using the shorthand property, the order of the property values are:

* **list-style-type** (if a list-style-image is specified, the value of this property will be displayed if the image for some reason cannot be displayed)
* **list-style-position** (specifies whether the list-item markers should appear inside or outside the content flow)
* **list-style-image** (specifies an image as the list item marker)

# CSS Tables

The look of an HTML table can be greatly improved with CSS:

|  |  |  |
| --- | --- | --- |
| **Company** | **Contact** | **Country** |
| Alfreds Futterkiste | Maria Anders | Germany |
| Berglunds snabbköp | Christina Berglund | Sweden |
| Centro comercial Moctezuma | Francisco Chang | Mexico |
| Ernst Handel | Roland Mendel | Austria |
| Island Trading | Helen Bennett | UK |
| Königlich Essen | Philip Cramer | Germany |
| Laughing Bacchus Winecellars | Yoshi Tannamuri | Canada |
| Magazzini Alimentari Riuniti | Giovanni Rovelli | Italy |
| North/South | Simon Crowther | UK |
| Paris spécialités | Marie Bertrand | France |
| The Big Cheese | Liz Nixon | USA |
| Vaffeljernet | Palle Ibsen | Denmark |

## 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;  
}

Notice that the table in the example above has double borders. This is because both the table and the <th>/<td> elements have separate borders.

To display a single border for the table, use the border-collapse property.

## Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

## Example

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

## Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the <th> elements to 50px:

## Example

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

## Horizontal Text Alignment

The text-align property sets the horizontal alignment, like left, right, or center.

By default, the text in <th> elements are center-aligned and the text in <td> elements are left-aligned.

The following example left-aligns the text in <th> elements:

## Example

th {  
    text-align: left;  
}

## Vertical Text Alignment

The vertical-align property sets the vertical alignment, like top, bottom, or middle.

By default, the vertical alignment of text in a table is middle (for both <th> and <td> elements).

The following example sets the vertical text alignment to bottom for <td> elements:

## Example

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

## Table Padding

To control the space between the border and content in a table, use the padding property on <td> and <th> elements:

## Example

td {  
    padding: 15px;  
}

## Table Color

The example below specifies the color of the borders, and the text and background color of <th> elements:

## Example

table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}

# CSS3 @media Rule

## Example

Change the background-color if the document is smaller than 300 pixels wide:

@media screen and (max-width: 300px) {  
    body {  
        background-color: lightblue;  
    }  
}

More "Try it Yourself" examples below.

## Definition and Usage

The @media rule is used to define different style rules for different media types/devices.

In CSS2 this was called media types, while in CSS3 it is called media queries.

Media queries look at the capability of the device, and can be used to check many things, such as:

* width and height of the browser window
* width and height of the device
* orientation (is the tablet/phone in landscape or portrait mode?)
* resolution
* and much more

**CSS Syntax**

@media not|only *mediatype*and*(media feature)* { *CSS-Code;*}

You can also have different stylesheets for different media:

<link rel="stylesheet" media="mediatype and|not|only (media feature)" href="mystylesheet.css">

## Example:Use the @media rule to make responsive design:

@media only screen and (max-width: 500px) {  
    .gridmenu {  
        width:100%;  
    }  
  
    .gridmain {  
        width:100%;  
    }  
  
    .gridright {  
        width:100%;  
    }  
}

**3.2.4 Converting Image Design to HTML (Slicing)**

It is typically part of the [client side](http://en.wikipedia.org/wiki/Client_side) development process of creating a web page and/or [web site](http://en.wikipedia.org/wiki/Web_site), but is also used in the [user interface design](http://en.wikipedia.org/wiki/User_interface_design) process of [software development](http://en.wikipedia.org/wiki/Software_development) and [game development](http://en.wikipedia.org/wiki/Game_development). Slicing is used in many cases where a [graphic design](http://en.wikipedia.org/wiki/Graphic_design) layout must be implemented as [interactive media](http://en.wikipedia.org/wiki/Interactive_media) content. Therefore, this is a very important skill set typically possessed by "[front end](http://en.wikipedia.org/wiki/Front_end_processor_(program))" developers; that is interactive media developers who specialize in user interface development .Slices may be produced and used in several different ways. Before [table-less web design](http://en.wikipedia.org/wiki/Tableless_web_design), sliced images were held together precisely with [html tables](http://en.wikipedia.org/wiki/Html_table). Modern interactive page layout includes extensive use of [Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheet) (CSS) and [semantic markup](http://en.wikipedia.org/wiki/Semantic_markup). Tables may be used for compatibility with rarer older [web browsers](http://en.wikipedia.org/wiki/Web_browser) that are incapable of processing modern table less coding accurately.