

# Examen final Itec/modr 2635

## Topic 1: Glendon cloud computing & html5 & Web

### Basics:

#### Cloud computing

Client-server model at oldtown.glendon.yorku.ca & web.glendon.yorku.ca & Google drives and apps.  
URLs: <http://web.glendon.yorku.ca/~lastname> and <http://oldtown.glendon.yorku.ca/~lastname>

Command line (CLI) for synchronization Glendon-Google: “**glendonsync\_start**” & “**glendonsync**”.

/home/lastname is **home** directory for an user

/home/lastname/www is the **web** directory. This is a **symbolic** link (shortcut) to

/home/lastname/mycloud/Dropbox/myweb

The directory **/home/lastname/mycloud** is synchronized to the Google-drive of an user

Practice web pages (examples and videos) at:

<http://web.glendon.yorku.ca/~gl/examen/practice/>  
<http://web.glendon.yorku.ca/~gl/examen/videos>

#### HTML5 language

Comme dans tous les langages il y a un minimum d'informations à donner dans le fichier HTML. Le fichier HTML minimum ressemble à ceci :

```
<!doctype html>
<HTML>
  <HEAD>
    <TITLE> Document bonjour </TITLE>
  </HEAD>
  <BODY>
    Bonjour.
  </BODY>
</HTML>
```

Titre du document/Title

En-tête/Header

Corps/Body

Les balises <HTML> et </HTML> stipulent que ce fichier texte est formaté selon le langage HTML et délimitent le contenu à interpréter. Les balises <HEAD> et </HEAD> viennent du mot HEADER (Entête) et délimitent l'en-tête du document contenant son titre et des informations sur son contenu. Les balises <BODY> et </BODY> délimitent le corps du document contenant le texte, son formatage les objets et les liens qu'il inclut

To publish information for widespread distribution, you must have a language understood by all computers. Such language, used on the World Wide Web, is called HTML (Hyper Text Markup Language). HTML is a set of fairly simple commands, which are inserted into the source text of the

document and allow control the representation of the document. Thus, the text created with a text editor, and then saved in HTML format becomes a Web-page/sites (HTML-document) after adding HTML language commands. Current version – HTML 5.

### **CSS language.**

Cascading Style Sheets - a language for describing the appearance of HTML-documents. Simply put, CSS language is designed to give the required appearance of HTML-documents. The purpose of CSS was to separate the description of the logical structure of the web page from her appearance. By using CSS web developer may specify the pages to separate its components and different typefaces and font sizes, colors elements padding elements from each other, arrangement of the individual blocks on the page, etc. Current version – CSS 3.

### **HTML main elements:**

Descriptor (tag) - the main element of coding adopted in the HTML standard. Almost everything depends on the tags, they serve to draw attention to particular commands in a document, thus making it clear for the browser what you want to select, where to display graphics and which hyperlinks are. Tags are allocated with angle brackets (<>). Tags can be single or paired. For the paired one it is necessary to have a an opening tag <tag> (most often with the option and its value) and a closing tag </ tag>.

**NOTE:** Tags can be written in any case (uppercase or lowercase letters).

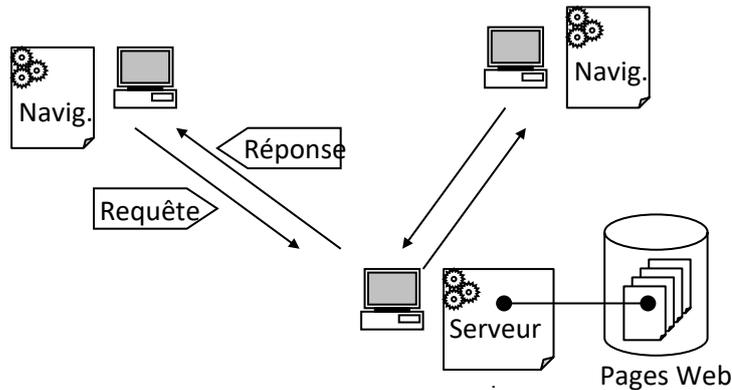
Container - descriptor pair consisting of opening and closing tags <tag> ... </ tag>. Containers are designed to store certain information, such as text or other HTML-tags. HTML-document element - a container along with its contents.

### **Types of web sites:**

Initially, the only technology that allows create websites and web pages, was html technology. However, such pages and sites were poor looking and lacked in interactivity, so in a short time that has elapsed from the start of the mass distribution of the Internet, began to be used numerous complementary technologies and languages.

- *Static websites and web pages.* Static sites and Web pages - are sites made by the classical html technology. Web pages of such sites are written entirely in the language of html and have the extension \*.htm or \*.html. They are called static because, gaining a certain address (URL), corresponding site or a particular page, you will always see the same content. This is because the static html-pages uploaded on the server, and at your request the server simply picks them up and sends you to the browser in without any changing.

Static sites have their both pros and cons. Advantages - the ease of creating and undemanding to the hardware resources. Static sites do not require the support of scripts and databases, and can be placed on any hosting absolutely, and they will not cause almost no load on the server, and they do not have a problem when the site slows down or produces a variety of failures for a large influx of visitors. Static sites can include graphics, animation, CSS and javascript, they are great for providing information to users, which should not be vary considerably over time. The main disadvantage of static websites is the lack of interactivity. In addition, such sites are difficult in changing information, page appearance, and design.



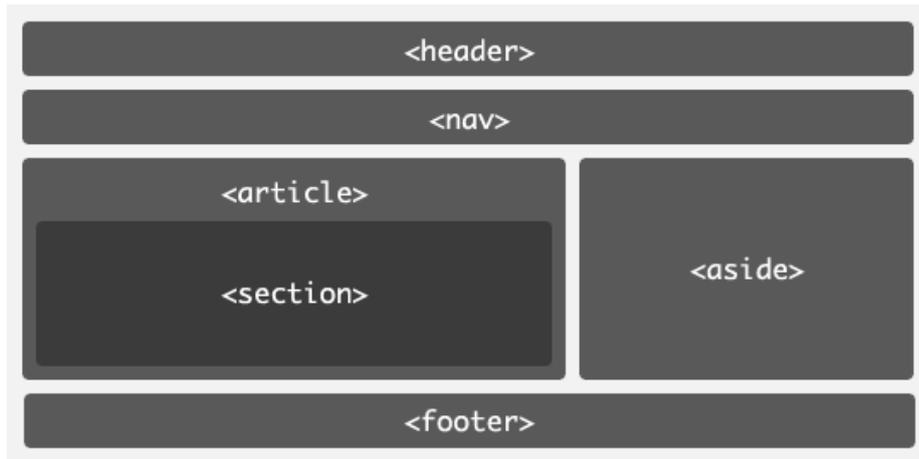
- *Dynamic websites and web pages.* Dynamic pages do not exist in a constant kind in the server and generated by using scripts. The content and form of the dynamic pages with the same address may be completely different depending on various conditions - for example, depending on particular user's time, current requests, etc. Web pages scripts can be written in different languages. Frequently used Web programming languages are, for example, php, perl, asp, etc. These sites can provide users with interactivity, to introduce different requests and search information within the sites, send and maintain their own information and carry out communication with other users (such as guest books, forums, etc.) and many other features. In addition, maintenance and updating of these sites much are easier. Cons of dynamic sites is that they require support for additional technologies, creating an increased load on the server, they are more difficult to optimize, in addition, scripts usage on the site is a potential threat to security.

**How to create html document:**

1. Create file using any text editor (except Microsoft Word), notepad for instance.
2. Fulfill your document with content and implementation of html tags.
3. Save your text document as “\*.html” or you can save it as txt file and simply change extension in your file browser, to do that follow those steps:
  - a) Click the Start button.
  - b) Open the Control Panel.
  - c) Find Appearance and Personalization.
  - d) Choose Folder Options.
  - e) Find Advanced Settings.
  - f) Display file extensions.
  - g) By default, the Hide Extensions for Known File Types check box is selected. Deselect this check box to display file extensions.

**NOTE:** To modify your html document you can open \*.html file in any text editor. At server oldtown, the text editors are “notepad” or “kate”.

## Html5 document standard scheme



### Listing of the example given above:

```
<!doctype html>
<html>
  <head>
    <title>Page title</title>
    <meta charset="utf-8">
    <meta name="description" content="The HTML5 template">
    <meta name="author" content="author name">
  </head>
  <body>
    <header>
      <h1>Page title</h1>
    </header>
    <nav>
      <!-- Navigation -->
    </nav>
    <main>
      <section>
        <!-- Main content area -->
        <article>
          <!-- Specific content area -->
        </article>
      </section>
    </main>
    <aside>
      <!-- Sidebar -->
    </aside>
    <footer>
      <!-- Footer -->
    </footer>
  </body>
</html>
```

See: [http://web.glendon.yorku.ca/~gl/examen/practice/html5\\_elements.html](http://web.glendon.yorku.ca/~gl/examen/practice/html5_elements.html)

### Scheme elements description:

#### <!doctype html>

First, we have the Document Type Declaration, or doctype. This is simply a way to tell the browser—or any other parser—what type of document it's looking at. In the case of HTML files, it means the

specific version and flavor of HTML. The doctype should always be the first item at the top of any HTML file.

### **<meta>**

Metadata is data (information) about data. The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata. The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

### **<body>**

The <body> tag defines the document's body. The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.

### **<main>**

The <main> tag specifies the main content of a document. The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.

**NOTE:** There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

### **<header>**

The <header> element specifies a header for a document or section. <header> represents a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, wrapped section's header, a search form, and so on.

### **<nav>**

The <nav> element represents a section of a page that links to other pages or to parts within the page: a section with navigation links.

### **<article>**

The <article> element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). This could be a forum post, a magazine or newspaper article, a blog entry, an object, or any other independent item of content.

### **<section>**

The <section> element represents a generic section of a document, i.e., a thematic grouping of content, typically with a heading. Each <section> should be identified, typically by including a heading (<h1>-<h6> element)

**NOTE:** Nesting <article> in <section> or Vice Versa? <Section> element should be used for a thematic grouping of content - it can encompass, for example, a chapter of a larger article or, alternatively, include several on the same page of relatively short articles, combined with some common features. <Article> element has a narrower mission than <section>. This tag makes sense to mark up some self-contained, independent piece of content, which in principle can be used on their own without reference to the rest of the page content. Nevertheless depending on the content, we can either nest element <article> in the <section> element, and vice versa.

### **<aside>**

The <aside> element represents a section of the page with content connected tangentially to the rest, which could be considered separate from that content. These sections are often represented as sidebars or inserts. They often contain the definitions on the sidebars, such as definitions from the glossary; there may also be other types of information, such as related advertisements; the biography of the author; web applications; profile information or related links on the blog.

### **<footer>**

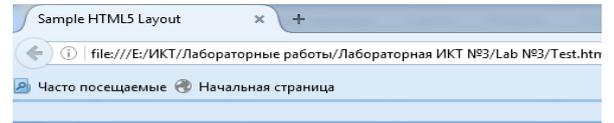
The <footer> element represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.

## Example of simply HTML document and corresponding web page:

```
<!DOCTYPE html>

<html>
<head>
  <title>Sample HTML5 Layout</title>
  <link href="StyleSheet.css" rel="stylesheet"/>
</head>
<body>
  <header>
    <h1>This is page heading</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About Us</a></li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </nav>
  <article>
    <h1>This is article heading</h1>
    <p>
      Hello world! Hello world! Hello world!
    </p>
  </article>
  <aside>
    <figure>
      
      <figcaption>Figure caption goes here</figcaption>
    </figure>
    <p>
      <small>Text in aside section of web page. </small>
    </p>
  </aside>
  <section>
    <h1>This is a section heading</h1>
    <p>
      You can fill this section with any content. The
      same is true for all other sections
    </p>
  </section>
  <footer>
    <hr />
    Copyright (C) 2016. All rights reserved.
  </footer>
</body>
</html>
```

## Web page in web browser



### This is page heading

- [Home](#)
- [About Us](#)
- [Contact Us](#)

### This is article heading

Hello world! Hello world! Hello world!



Figure caption goes here

Text in aside section of web page.

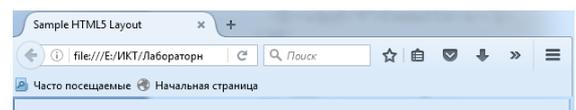
### This is a section heading

You can fill this section with any content. The same is true for all other sections

---

Copyright (C) 2016. All rights reserved.

## The same page after applying StyleSheet.css



### This is page heading

- [Home](#)
- [About Us](#)
- [Contact Us](#)

### This is article heading

Hello world! Hello world! Hello world!



Figure caption goes here

Text in aside section of web page.

### This is a section heading

You can fill this section with any content. The same is true for all other sections

---

Copyright (C) 2016. All rights reserved.

## Description of basic html elements (tags)

Formatting paragraphs		
Inner headers of different levels	<h№> text </h№>	Where № - heading level number from biggest to lowest (1 to 6). For example, <H1> ... </ H1> - 1st level header.
Create a paragraph	<p> text </p>	Paragraphs are separated by a double line spacing
Transfer to a new line within a paragraph	 	Single tag
Separating horizontal line between paragraphs	<hr size=«?»>	Single tag. «?» - the thickness of the line in pixels. The thickness of the line may be left blank.
Formatting text		
Bold	<b> text </b>	<b> <b>Bold</b> </b>
Italics	<i> text </i>	<I> <i>Italics</i> </I>
Underlined	<u> text </u>	<U> <u>Underlined</u> </U>
Slashed	<s> text </s>	<S> <del>Slashed</del> </S>
Reduced size	<small> text </small>	<SMALL> <small>SMALL</small> </SMALL>
Superscript	<sup> text </sup>	<SUP> <sup>Superscript</sup> </SUP>
Subscript	<sub> text </sub>	<SUB> <sub>Subscript</sub> </SUB>
Generating lists		
Numbered	<ol> <li> Element </li> </ol>	<OL> <LI> Element 1 </LI> <LI> Element 2</LI> <LI> Element 3</LI> </OL>
Marked	<ul> <li> Element </li> </ul>	
Element of the list	<li> Element </li>	

### HTML attributes

Elements may contain attributes that are used to set various properties of an element. Some attributes are defined globally and can be used on any element, while others are defined for specific elements only. All attributes have a name and a value and look like as shown below in the example:

<h1 id="myHeader" style = "color:blue; margin-left:50%;"> Attribute "id" indicates specific name: "myHeader" of the element "h1". Attribute "style" states that this text is blue and is in the center of the screen. </h1>

Attribute	Belongs to tag	Description
alt	<area>, <img>, <input>	Specifies an alternate text when the original element fails to display.
height	<canvas>, <embed>,<iframe>, <img>, <input>, <video>	Specifies the height of the element
href	<a>, <area>, <base>, <link>	Specifies the URL of the page the link goes to.
id	Global Attributes	Specifies a unique id for an element.
src	<audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video>	Specifies the URL of the media file.
style	Global Attributes	Specifies an inline CSS style for an element.

## Inserting images

To insert image you should use <img> tag. It contains attributes only (src, alt, width, height), and does not have a closing tag. The src attribute specifies the URL (web address) of the image. The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). If a browser cannot find an image, it will display the value of the alt attribute.

```
<img src = "image name.jpg" height = "480" width = "640" alt = "Your browser don't support images"/>
```

**NOTE:** Always specify the width and height of an image. If width and height are not specified, the page will flicker while the image loads. Width and height parameters can be indicate either in pixels (i.e. 100px) or percents of the current canvas screen (10%).

**NOTE:** If not specified, the browser expects to find the image in the same folder as the web page. However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

```
<img src = "/Test images/image name.jpg" height = "480" width = "640" alt = "Your browser don't support images"/>
```

## HTML Links

Types of links: Hyperlinks (absolute links); Local links (relative links); Bookmarks.

**NOTE:**A link does not have to be text. It can be an image or any other HTML element. To use an image as a link, simply nest the <img> tag inside the <a> tag.

### Hyperlinks

Links are found in nearly all web pages. Links allow users to click their way from page to page. Any HTML links are hyperlinks. You can click on a link and jump to another document. The key here is an oddly named tag called the *anchor* tag. This tag is encased in an <a></a> set of tags and contains all the information needed to manage links between pages.

```
<a href = "http://www.wikipedia.org">wikipedia.</a>
```

The **href** attribute specifies the destination address (http://www.wikipedia.org) of the link. The **link text** (wikipedia.) is the visible part and displayed on the page.

### Local links

Relative references are used when your website includes more than one page. You might choose to have several pages and a link mechanism for moving among them. Example below will lead us from current page to the "About" page.

```
<a href = "about.html">About</a>
```

### Bookmarks

HTML bookmarks are used to allow readers to jump to specific parts of a Web page. Bookmarks can be useful if your webpage is very long. To make a bookmark, you must first create the bookmark, and then add a link to it. When the link is clicked, the page will scroll to the location with the bookmark.

1. Create bookmark with the id attribute:

```
<p id = "Section1"> Section 1 text </p>
```

2. Add a link to the bookmark ("Section 1 "), from within the same page:

```
<a href = "#Section1"> Jump to section 1 </a>
```

## HTML Symbol and Entities

Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML. Many mathematical, technical, and currency symbols, are not present on a normal keyboard. To add such symbols to an HTML page, you can use an HTML entity name. If no entity name exists, you can use an entity number, a decimal, or hexadecimal reference. A syntax of entity and symbols usage in html:

*&entity\_name; OR #entity\_number;*

Table of common html symbols

Symbol	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	double quotation mark	&quot;	&#34;
'	single quotation mark (apostrophe)	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;
€	euro sign	&euro;	&#8364;
\$	dollar sign	&dollar	&#36;

## **Topic 2: CSS. Giving the style to your site**

A CSS (cascading style sheet) file allows you to separate your web sites (X)HTML content from it's style. As always you use your (X)HTML file to arrange the content, but all of the presentation (fonts, colors, background, borders, text formatting, link effects & so on...) are accomplished within a CSS.

### **How to create CSS file:**

1. Create file using any text editor (except Microsoft Word), notepad for instance.
2. Fulfill your document with description of yours html page design and style using CSS format.
3. Save your text document as "\*.css" or you can save it as txt file and simply change extension in your file browser, to do that follow those steps:

**NOTE:** The file should not contain any html tags.

### **CSS Syntax**

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



The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

**NOTE:** Dimension values can be indicate in percents (13%) or pixels (10px).

**NOTE:** Do not add a space between the property value and the unit (such as margin-left: 20 px;). The correct way is: margin-left: 20px;

### CSS selectors

- The element selector

The element selector selects elements based on the element name. You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

```
p {
    text-align: center;
    color: red;
}
```

- The id Selector

The id selector uses the id attribute of an HTML element to select a specific element. To select an element with a specific id, write a hash (#) character, followed by the id of the element. The style rule below will be applied to the HTML element with id="sect1":

```
<head>
  <style>
    #sect1 {
      text-align: center;
      color: red;
    }
  </head>
  <section id = "sect1">
    Red text in the center of page
  </section>
```

**NOTE:** The id of an element should be unique within a page, so the id selector is used to select one unique element.

### Ways to Insert CSS:

There are three ways of inserting a style sheet:

#### 1. External style sheet

With an external style sheet, you can change the look of an entire website at once. Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

```
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

**NOTE:** always indicate correct file path to your \*.css file.

#### 2. Internal style sheet

An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```

<head>
  <style>
    body {
      background-color: linen;
    }
    h1 {
      color: red;
      margin-left: 40px;
    }
  </style>
</head>

```

### 3. Inline style

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue;margin-left:30px;">This is a heading.</h1>
```

**NOTE:** you can use several ways, i.e. create multiple style sheets but in this case if some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used. Example:

Assume that an external style sheet has the following style for the <h1> element:

```

h1 {
  color: navy;
}

```

then, assume that an internal style sheet also has the following style for the <h1> element:

```

<head>
  <style>
    h1 {
      color: orange;
    }
  </style>
</head>

```

If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":

```

<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  <style>
    h1 {
      color: orange;
    }
  </style>
</head>

```

However, if the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy".

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

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific 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 a browser default value.

## CSS3 common properties list:

### • Background Properties

Property	Description	Possible Values	Examples
background-color	Declares the background color.	Valid color names, RGB values, hexadecimal notation.	article { background-color:green; } article { color:#00FF00; }
background-image	Declares the background image of an element.	URL values.	div { background-image:url(images/img.jpg); } body { background-image:url(img.jpg); }
background	Used as a shorthand property to set all the background properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values): background-color background-image background-position	body { background:green; } body { background:url(image.jpg) fixed; }

### • Text Properties

Property	Description	Possible Values	Examples
color	Declares the color of the text.	Valid color names, RGB values, hexadecimal notation.	p { color:green; } h1 { color:rgb(0,255,0); } #elm4 { color:#00FF00; }
direction	Declares the reading direction of the text.	Ltr = left-to-right rtl = right-to-left	h1 { direction:ltr; }
text-align	Declares the horizontal alignment of inline content.	Left; right; center;justify If used on a set of table cells, this property can be given a string value to which the text of each row of the column will be aligned.	div { text-align:center; } h5 { text-align:right; } td { text-align:"."; }
text-decoration	Declares the text decoration.	None; underline; overline; line-through; blink	p { text-decoration:none; } p { text-decoration:underline; }

**NOTE:** Different styles of letters is achieved by using the tag <span>.

Let<span style = "color: red;">ers</span>with<span style = "color: green;">differ</span><span style = "color: blue;">ent</span> colors.

### • Font Properties

Property	Description	Possible Values	Examples
font-family	Declares the name of the font to be used.	Font family names can be separated by a comma in the same declaration to allow additional and/or generic family names to be used if the preferred font is unable to be displayed.	section { font-family:Arial; } footer { font-family:Arial, Helvetica, sans-serif; }
font-size	Declares the size of the font.	Lengths (number and unit type— i.e. <i>1em</i> , <i>12pt</i> , <i>10px</i> , <i>80%</i> )	p { font-size:70%; } section { font-size:0.85em; }

font-style	Declares the font style.	normal italic oblique	section { font-style:italic; } p { font-style:oblique; }
font	Used as a shorthand property to declare all of the font properties at once.	Separate values by a space in the following order (those that are not defined will use inherited or default initial values):	aside { font:italic small-caps bold 1em 1.2em Arial } div { font:bold 0.8em Verdana }

- **Border Properties**

Property	Description	Possible Values	Examples
border-style	Declares the border style of all four borders at once.	None ; hidden; dotted; dashed; solid; double; groove; ridge; inset; outset  If only one style value is declared, all four borders will use that style. If two styles are declared, the top and bottom borders will use the first style while the right and left borders will use the second style. If three styles are declared, the top border will use the first style, the right and left borders will use the second style, and the bottom border will use the third style.	aside { border-style:solid dotted dashed double; } p { border-style:solid; } article { border-style:solid dotted; } body { border-style:solid dotted dashed; }
border-color	Declares the border color of all four borders at once.	Valid color names, RGB values, hexadecimal notation, or the predefined value transparent.	article { border-color:green; } p { border-color:green red; }
border-width	Declares the width of all four borders at once.	Lengths in pixels or the following predefined values: thin; medium; thick	section { border-width:1px; } p { border-width:thin; }

**NOTE:** The "border-width" property does not work if it is used alone. Always specify the "border-style" property to set the borders first.

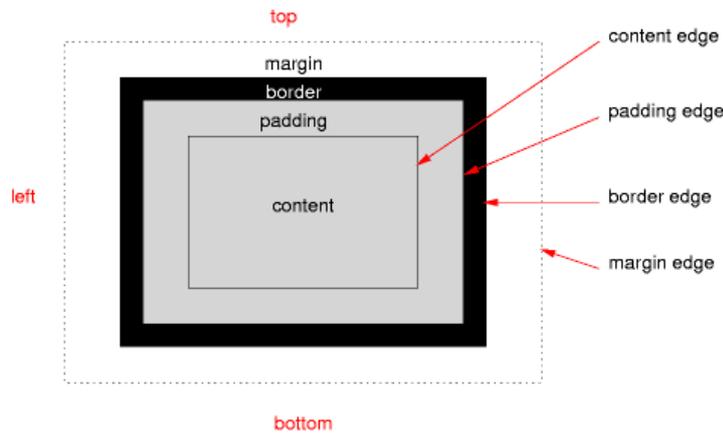
**NOTE:** you can define style, color, width of each single border line:

*border-bottom-style:solid; border-right-color:green; border-top-width:2px;*

- **Dimension Properties**

Property	Description	Possible Values	Examples
Height/width	Declares the height/width of the element.	Lengths, percentages, and the predefined value auto.	#img { height:200px; } article { width:50%; }
Max(min)-height/ width	Declares the maximum (minimum) height/width of the element.	Lengths, percentages, and the predefined value auto.	#img { max-height:200px; } p { min-width:75%; }

## Positioning of elements in CSS



The position property specifies the type of positioning method used for an element.

**position: static;**

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

**position: relative;**

An element with position: relative; is positioned relative to its normal position. Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.

**position: fixed;**

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element. A fixed element does not leave a gap in the page where it would normally have been located.

**position: absolute;**

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

• **Positioning Properties**

Property	Description	Possible Values	Examples
float	Declares whether a box should float to the left or right of other content, or whether it should not be floated at all.	left right none	div { float:left; } div { float:right; }
visibility	Declares the visibility of boxes generated by an element.	visible hidden collapse	div { visibility:visible; } div { isibility:hidden; }
position	Declares the type of positioning of an element.	static relative absolute fixed	div { position:absolute; } div { position:relative; }

• **Margin Properties.**

The CSS margin properties are used to generate space around elements. The margin properties set the size of the white space outside the border.

Property	Description	Possible Values	Examples
margin-top(right/ bottom/ left)	Declares the top (right/ bottom/ left)	Lengths, percentages, and the predefined value auto.	div { margin-top:5px; }

bottom/ left)	margin for the element.		section { margin- left:15%; }
margin	Shorthand property used to declare all the margin properties at once.	<p>Separate values by a space in the following order (those that are not defined will use inherited or default initial values):</p> <p>margin-top margin-right margin-bottom margin-left</p> <p>Undeclared values work as further shorthand notation. If only one length value is declared, all four margins will use that length. If two lengths are declared, the top and bottom margins will use the first length while the right and left margins will use the second length. If three lengths are declared, the top margin will use the first length, the right and left margins will use the second length, and the bottom margin will use the third length.</p>	<pre>div { margin:5px 12px 4px 7px; } div { margin:5px; } div { margin:5px 10px; } div { margin:5px 7px 4px; }</pre>

- **Padding Properties**

The CSS padding properties are used to generate space around content. The padding clears an area around the content (inside the border) of an element.

Property	Description	Possible Values	Examples
padding- top(right/ bottom/ left)	Declares the top(right/ bottom/ left) padding for the element.	Lengths, percentages, and the predefined value auto.	<pre>div { padding- top:5px; } div { padding- left:15%; }</pre>
padding	Shorthand property used to declare all the margin properties at once.	<p>Separate values by a space in the following order (those that are not defined will use inherited or default initial values):</p> <p>padding-top padding-right padding-bottom padding-left</p> <p>Undeclared values work as further shorthand notation. If only one length value is declared, all four sides will use that length. If two lengths are declared, the top and bottom sides will use the first length while the right and left sides will use the second length. If three lengths are declared, the top side will use the first length, the right and left sides will use the second length, and the bottom side will use the third length.</p>	<pre>div { padding:5px 12px 4px 7px; } div { padding:5px; } div { padding:5px 10px; } div { padding:5px 7px 4px; }</pre>