# HTML

## Getting Connected

<http://www.sphotonkhan.com/index.html>

🡨🡪 🡨…………………....🡪🡨…….🡪

The first part of the URL tells us the protocol that needs to be used to retrieve the resource.

The second part is the website name

The third part is the absolute path to the resource from the root path

## Uniform Resource Locator

A **Uniform Resource Locator (URL)** is a global address that can be used to locate anything on the Web, including HTML pages, audio, video, and many other forms of web content. In addition to specifying the location of the resource, a URL also names the protocol that we can use to retrieve the resource

## HTTP

**Hyper Text Transfer Protocol**, it is a simple request and response protocol. It’s an agreed-upon method (a protocol) for transferring hypertext documents around the Web.

## Absolute Path

The path with reference to root directory is called absolute.

<http://www.sphotonkhan.com/index.html>

## Relative Path

The path with reference to current directory is called relative.

<a href = **“../**lounge.html”> To the lounge </a>

“/” means Separate all parts of the path

“..” means parent folder

“../../” means move to parent folders up

## Element

**Element = Openting Tag + Content + Closing Tag**

<h1> Coffee Shop </h1>

Tag consist of the tag name surrounded by angle brackets that is, the < and > characters

## Basic Format

<!DOCTYPE html>

<html>

<head>

<title> Basic Format</title>

<meta charset = "UTF-8"/>

</head>

<body>

<h1> Heading One </h1>

<p> Paragraph </p>

</body>

</html>

**Code 1: Basic.html**

## Comment Tag

**<!-- -->**

The comment tag is used to insert comments in the source code. Comments are not displayed in the browsers.

## Document Type

The **<!DOCTYPE>** declaration must be the very first thing in your HTML document, before the <html> tag.

## Whole Document

<html> element defines the whole document. Standard Markup Language

## Head

**<head>**

<title>Basic Tutorial</title>

<meta charset = “UTF-8”/>

<link rel = “stylesheet” type = “text/css” href = “location.css”/>

<script src = “interaction.js”></script>

**</head>**

## Title

**<title>Basic Tutorial</title>**

The <title> tag is required in all HTML documents and it defines the title of the document.

It defines a title in the browser toolbar

It provides a title for the page when it is added to favorites

It displays a title for the page in search-engine results

## Character Encoding

**<meta charset = “UTF-8”/>**

Meta means we going to tell the browser something about the page

The charset attribute is where we specify the character encoding

UTF-8 is an encoding in the Unicode family of encodings (one of several). “UTF-8” is the version we for the web page

## Link

**<link rel = “stylesheet” type = “text/css” href = “location.css”/>**

Linking the html file to a css file

## Script

**<script src = “interaction.js”></script>**

## Body

The <body> element defines the document body.

## Heading

<h1> Hello </h1> It has in total 6 headings <h1><h2><h3><h4><h5><h6>

## Paragraph

<p> Paragraph </p>

## W3C Validator

http://validator.w3.org/

**Checks the format of the html page**

## Quote

It is a child of paragraph <p> node

**<q> To be or not to be </q>**

## Block Quote

Alternative way to write quotes

**<blockquote> Mr. Mobile have fun today </blockquote>**

|  |  |
| --- | --- |
| **Blockquote <blockquote>** | **Quote <q>** |
| Block quote stands on it own | It is a inline element |

## Block Element vs Inline Elements

|  |  |
| --- | --- |
| **Block Elements** | **Inline Elements** |
| <h1> <h2> <h6> <blockquote> | <q> <a> <em> |
| They can stand on its own | They need to be in block element |

## Class vs Id

|  |  |
| --- | --- |
| **Class** | **Id** |
| Class are not unique | Id are unique |
| You can use the same class on multiple elements. | Each element can have only one id |
| You can use multiple classes on the same element. | You cannot use multiple id in the same class |
| Each page can have multiple element with that class | Each page can have only one element with that id |

## Break

**<br/>**

Breaks the flow and insert a “line break”

## Void Element

Elements that don’t have any content by design are called void elements. When we need to use a void element, like <br> or <img>, we only use an opening tag. This is a convenient shorthand that reduces the amount of markup in our HTML

## Hyperlink

The **<a>** tag defines a hyperlink, which is used to link from one page to another.

Use the <a> element to create a hypertext link to another web page. The content of the <a> element becomes clickable in the web page. The

The href attribute tells the browser the destination of the link

The title attribute gives the textual description of the page

**<a href = “elixir.html” title = “Description of the page”>Elixirs</a>**

href is the attribute

### Create Destination

The id attribute allows us to uniquely identify an element

# on the end of our lonk follow the destination identifier

<h2 **id = “books”**> Books </h2>

<a href = “**index.html#books**”> See Books </a>

### New window using target

<a **target = “\_blank”** href = <http://sphotonkhan.com>

title = Read all about caffeine on the site> Photon Khan</a>

The target attribute tells the browser where to open the page that is at the link the href attribute. If there is not target, then the browser opens the link in the same window. If the target is “\_blank”, then the browser opens the link in a new window.

## Image

**<img src = “drinks.gif” alt = “Drinks Page”/>**

The src attribute is the location of the image

The alt attribute is the alternative description of the image

|  |  |  |
| --- | --- | --- |
| JPEG | PNG | GIF |
| Works best for continuous tone images, like photographs | It works best for images with a few solid colors, and image with lines, like logos, clip art, and small text in images | GIF works best for few solid colors and images with lines, logos, clip art and small text in images |
| It represents 16 million different colors | It represents images with millions of different colors. PNG comes in three different formats  PNG-8, PNG-24, PNG-32  Depending on color representation | It represents 256 different colors |
| It is a “lossy” format because to reduce the file size, it throws away some information about the image | PNG compresses the file to reduce its size, it is not a lossy format | It is a lossless format |
| It does not support transparency | It supports transparency | It supports transparency |
| Files are smaller for more efficient web pages | Files tend to be larger than JPEG but can be smaller or larger than GIF depending on the colors used | GIF size larger than JPEG |
| No support for animation |  | Support animation |

### Sizing Image

<img src = “tea.gif” alt = “Drinks Page” **width = “30” height = “23”**/>

### Pixel

A good rule of thumb is **96 pixels** to every inch. Browser fits **1200 x 800 pixels (Average)**

## Link out of images

<**a** href = "http://sphotonkhan.com" target = "\_blank" title = "Cafe">

<**img** src = "images/computer.ico" alt = "computer icon"/>

</**a**>

If we are placing transparent image in our web page, make sure the matte color of the image matches the background color of our web page. We can use PNG or GIF format for our transparent image.

## Ordered List

The **<ol>** tag defines an ordered list. An ordered list can be **numerical or alphabetical.**

<ol>

<li> Bread </li>

<li> Egg </li>

<li> Jam </li>

</ol>

## Unordered List

The **<ul>** tag defines an unordered **(bulleted)** list.

Use the <ul> tag together with the <li> tag to create unordered lists.

<ul>

<li> Bread </li>

<li> Egg </li>

<li> Jam </li>

</ul>

## Nesting

Putting one element inside another

<**ol**>

<**li**> Charge Segway </**li**>

<**ul** Pack for trip>

<**li**>Cell Number</**li**>

<**li**>iPod</**li**>

<**li**>Digital Camera</**li**>

<**li**>Protein Bar</**li**>

</**ul**>

</**ol**>

## Definition List

<**dl**>

<**dt**>Byrna Shave Signs</**dt**>

<**dd**> Road Signs common</**dd**>

<**dt**>Route 6666</**dt**>

<**dd**>Most Famous Guys</**dd**>

</**dl**>

The <dd> tag is used to describe a term/name in a description list.

The <dd> tag is used in conjunction with <dl> (defines a description list) and <dt> (defines terms/names).

Inside a <dd> tag you can put paragraphs, line breaks, images, links, lists, etc.

## Emphasize

The <em> tag is a phrase tag. It renders as emphasized text.

**<p><em>** Paragraph **</em></p>**

## Strong

This element is used to mark up text we want emphasized with extra strength

<**li**><**strong**> Charge Segway </**strong**></**li**>

## Preformatted text

The **<pre>** tag defines preformatted text.

Text in a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

<pre>

Text in a pre element

is displayed in a fixed-width

font, and it preserves

both spaces and

line breaks

</pre>

## Formatting elements

**<i>** - Italic text

**<mark>** - Marked text

**<small>** - Small text

**<del>** - Deleted text

**<ins>** - Inserted text

**<sub>** - Subscript text

**<sup>** - Superscript text

## Time

The element tells the browser that the context is a date or time, or both

<p> We open at **<time>10:00</time>** every morning. </p>

## Code

The <code> tag is a phrase tag. It defines a piece of computer code.

**<code>A piece of computer code</code>**

## Center

**<center>**

<font size = “small”>You got to be kidding me</font>

**</center>**

## Font

**<font size = “small”>You got to be kidding me</font>**

**<font face = “arial”>You got to be kidding me</font>**

## Special Attributes

### Text Align Attribute

<p **align=right**>Come over here</p>

No double apostrophe for the attribute

### Color Attribute

<body **bgcolor = “tan” text = “black”**>

## Division

This helps to organize the elements belong to the same group. This structure helps us separate a page into a logical sections for clarity and styling.

The <div> tag defines a division or a section in an HTML document.

The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

<div style="background-color:lightblue">  
  <h3>This is a heading</h3>  
  <p>This is a paragraph.</p>  
</div>

## Span

The <span> tag is used to group inline-elements in a document.

The <span> tag provides no visual change by itself.

The <span> tag provides a way to add a hook to a part of a text or a part of a document.

<p>My mother has <span style="color:blue">blue</span> eyes.</p>

## Header

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

* One or more heading elements (<h1> - <h6>)
* Logo or icon authorship information
* We can have several <header> elements in one document.

<header>  
    <h1>Most important heading here</h1>  
    <h3>Less important heading here</h3>  
    <p>Some additional information here</p>  
</header>

## Footer

The <footer> tag defines a footer for a document or section.

A <footer> element should contain information about its containing element.

A <footer> element typically contains:

* Authorship information
* Copyright information
* Contact information
* Sitemap
* Back to top links
* Related documents

You can have several <footer> elements in one document.

<footer>  
  <p>Posted by: Hege Refsnes</p>  
  <p>Contact information: <a href="mailto:someone@example.com">  
  someone@example.com</a>.</p>  
</footer>

## Aside

The <aside> tag defines some content aside from the content it is placed in.

The aside content should be related to the surrounding content.

<aside>  
  <h4>Epcot Center</h4>  
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>  
</aside>

## Section

It defines sections in a document, such as chapters, headers, footers, or any other sections of the document.

<section>  
  <h1>WWF</h1>  
  <p>The World Wide Fund for Nature (WWF) is....</p>  
</section>

## Navigation

It defines group of navigation links

<nav>  
  <a href="/html/">HTML</a> |  
  <a href="/css/">CSS</a> |  
  <a href="/js/">JavaScript</a> |  
  <a href="/jquery/">jQuery</a>  
</nav>

## Video

<video controls autoplay width=”512” height=”288”

src=”video/tweetsip.mp4”

poster=”images/poster.png”>

</video>

**autoplay** Specifies that the video will start playing as soon as it is ready

**controls** Specifies that video controls should be displayed (such as a play/pause button etc).

**height** Sets the height of the video player

**loop** Specifies that the video will start over again, every time it is finished

**muted** Specifies that the audio output of the video should be muted

**poster** Specifies an image to be shown while the video is downloading, or until the user hits the play button

**preload** Specifies if and how the author thinks the video should be loaded when the page loads

**src** Specifies the URL of the video file

**width** Sets the width of the video player

<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Your browser does not support the video tag.  
</video>

The HTML5 specification allows for any video format. It is the browser implementation that determines what formats are actually supported

### Video Format Contenders

|  |  |  |
| --- | --- | --- |
| **MP4 container with H.264 and AAC audio** | **WebM container with VP8 video and Vorbis audio** | **Ogg container with Theora video and Vorbis audio** |
| H.264 is licensed by the MPEG-LA group | WebM was designed by Google to work with VP8-encoded videos. | Theora is a open source codec |
| There is more than one kind of H.264, each is known as profile | WebM/VP8 is supported by Firefox, Chrome, and Opera | Video encoded with Theora is usually contained in an Ogg file, with the .ogv file extension |
| MP4/H.274 is supported by Safari and IE9+. We can find support in some version of chrome | WebM-formatted videos with the.webm extension | Ogg/Theora is supported by Firefox, Chrome and Opera |

**Video with specific format**

<video src = “video/tweetsip.ogv” type = “video/ogg; codecs=’theora, vorbis’”>

## Audio

It defines sound, such as music or other audio streams.

<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
  Your browser does not support the audio tag.  
</audio>

**autoplay** Specifies that the audio will start playing as soon as it is ready

**control**  Specifies that audio controls should be displayed (such as a play/pause button etc)

**loop**  Specifies that the audio will start over again, every time it is finished

**muted**  Specifies that the audio output should be muted

**preload** Specifies if and how the author thinks the audio should be loaded when the page loads

**src** Specifies the URL of the audio file

## Object

The defines an embedded object within an HTML document. Use this element to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in your web pages.

<object width="400" height="400" data="helloworld.swf"></object>

It is also embedded to another webpage into your HTML document.

We can use the <param> tag to pass parameters to plugins that have been embedded it

## Meter

It defines a scalar measurement within a known range, or a fractional value. This is also known as a gauge.

<meter value="2" min="0" max="10">2 out of 10</meter><br>  
<meter value="0.6">60%</meter>

**form** Specifies one or more forms the <meter> element belongs to

**high** Specifies the range that is considered to be a high value

**low** Specifies the range that is considered to be a low value

**max** Specifies the maximum value of the range

**min** Specifies the minimum value of the range

**optimum** Specifies what value is the optimal value for the gauge

**value** Required. Specifies the current value of the gauge

## Canvas

It is used to draw graphics, on the fly, via scripting (usually JavaScript).

It is only a container for graphics, you must use a script to actually draw the graphics.

<canvas id="myCanvas"></canvas>

## Figure

It specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

<figure>  
  <img src="img\_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">  
</figure>

## Progress

It represents the progress of a task.

<progress value="22" max="100"></progress>

## Table

It defines an HTML Table

<table>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>$100</td>  
  </tr>  
</table>

<**html**>

<**head**>

<**title**> Basic Format</**title**>

<**meta** charset = "UTF-8"/>

<**style** type = "text/css">

**table**, **td**, **th**{

**border**: **thin** **solid** black;

**border-collapse**: **collapse**;

}

</**style**>

</**head**>

<**body**>

<**table**>

<**tr**>

<**th**> City </**th**>

<**th**> Date </**th**>

<**th**> Temperature </**th**>

<**th**> Altitude </**th**>

<**th**> Population </**th**>

<**th**> Diner Rating </**th**>

</**tr**>

<**tr**>

<**td**> Walla Walla</**td**>

<**td**> June 4th</**td**>

<**td**> 75</**td**>

<**td**> 1,204 </**td**>

<**td**> 29.686 </**td**>

<**td**> 4/5 </**td**>

</**tr**>

<**tr**>

<**td** rowspan = "2"> Truth or Consequence </**td**>

<**td**> August 9th </**td**>

<**td** colspan = "2"> 93 </**td**>

<**td** rowspan = "2"> 7.289 </**td**>

<**td**> 5/5 </**td**>

</**tr**>

<**tr**>

<**td**> August 27th </**td**>

<**td**> 98 </**td**>

<**td**> 4,242 </**td**>

<**td**> 4/5 </**td**>

</**tr**>

<**tr**>

<**td**> Why </**td**>

<**td**> August 18th </**td**>

<**td**> 104 </**td**>

<**td**> 860 </**td**>

<**td**> 480 </**td**>

<**td**> 3/5 </**td**>

</**tr**>

</**table**>

</**body**>

</**html**>

## Form

<**form** enctype = "multipart/form-data" method = "post" action = "">

<**label** for = "fullname"> Full Name </**label**><**br**/>

<**input** type = "text" name = "fullname" placeholder = "Your Name" required/><**br**/><**br**/>

<**label** for = "email"> Email </**label**><**br**/>

<**input** type = "email" name = "email" placeholder = "Your Email"required/><**br**/><**br**/>

<**label** for = "password"> Password </**label**><**br**/>

<**input** type = "password" name = "password" required><**br**/><**br**/>

<**label** for = "age"> Age </**label**><**br**/>

<**input** type = "number" name = "age" required/><**br**/><**br**/>

<**label** for = "gender"> Gender </**label**><**br**/>

Male <**input** type = "radio" name = "gender" value = "male">

Female <**input** type = "radio" name = "gender" value = "female"><**br**/><**br**/>

<**fieldset** height = "100px" width = "100px">

<**legend** for = "spice"> Spice </**legend**><**br**/>

Salt <**input** type = "checkbox" name = "spice[]" value = "salt"><**br**/>

Pepper <**input** type = "checkbox" name = "spice[]" value = "pepper"><**br**/>

Garlic <**input** type = "checkbox" name = "spice[]" value = "garlic"><**br**/>

</**fieldset**><**br**/><**br**/>

<**label** for = "car"> Cars </**label**><**br**/>

<**select** name = "car" multiple>

<**option** value = "toyota">Toyota</**option**>

<**option** value = "mazda">Mazda</**option**>

<**option** value = "nissan">Nissan</**option**>

</**select**><**br**/><**br**/>

<**label** for = "distance">Distance Travelled</**label**><**br**/>

<**input** type = "range" min = "5" max = "30" step = "5"><**br**/><**br**/>

<**label** for = "date">Date Today </**label**><**br**/>

<**input** type = "date" name = "date"/><**br**/><**br**/>

<**label** for = "cellphone"> Cellphone </**label**><**br**/>

<**input** type = "tel" name = "cellphone"><**br**/><**br**/>

<**label** for = "fav\_color"> Favorite Color </**label**><**br**/>

<**input** type = "color" name = "fav\_color"/><**br**/><**br**/>

<**label** for = "website"> Website Name </**label**><**br**/>

<**input** type = "url" name = "website"><**br**/><**br**/>

<**label** for = "feedback"> Feedback </**label**><**br**/>

<**textarea** name = "feedback" rows = "10" cols = "48"></**textarea**><**br**/><**br**/>

<**input** type = "submit" name = "submit" value = "Confirm"/>

</**form**>

**<form>** Write a form

**<input>** Take different types of input

Types 🡪 email, password, range, color, tel, submit, checkbox, radio

**<textarea>**

## HTML TERMS

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Element** | **Description** |
| <!-- --> | Comment | <br/> | Break |
| <!DOCTYPE> | Doc Declaration | <ul> | Unordered List |
| <a> | Hyperlink | <ol> | Ordered List |
| <img> | Image | <li> | List |
| <q> | Quote | <em> | Emphasize |
| <blockquote> | Blockquote | <strong> | Strong Emphasize |
| <h1> 🡪 <h6> | Heading 1 🡪 6 | <dd> | Describe Data |
| <p> | Paragraph | <dl> | Description List |
| <pre> | Preformatted text | <dt> | Define Terms |
| <time> | Date or Time | <code> | Computer Code |
| <i> | Italic text | <mark> | Marked text |
| <small> | Small text | <del> | Deleted text |
| <ins> | Inserted text | <sub> | Subscript text |
| <sup> | Superscript text | <center> | Center script |
| <meta> | Character encoding | <font> | Font |
| <link> | Link Css File | <script> | Include JavaScript |
| <div> | Division | <title> | Title |
| <span> | Add a hook/Error | <header> | Header |
| <footer> | Footer | <aside> | Sidebar |
| <section> | Section | <nav> | Navigation Bar |
| <video> | Video Element | <audio> | Audio Element |
| <canvas> | Canvas Container | <figure> | Contains diagrams |
| <table> | Table | <progress> | Shows progress bar |
| <th> | Table Heading | <tr> | Table Row |
| <td> | Table Data | <input | User Input |
| <form> | User Form | <fieldset> | Creates a Field |
| <legend> | Subtopic Region | <caption> | Caption |

# CSS

## Basic Format

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



The selector points to the HTML element we 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.

## Jigsaw Validator

http://jigsaw.w3.org/css-validator/

Checks the format of css

## Overriding Inheritance

body {font-family: sans-serif;}

h1, h2 {color: gray;}

p {color: maroon;}

## Selector

Select more than one element

h1, h2{color: gray;}

### Class Selector

<p class = “greentea”>Green Tea<p>

p.greentea{color: green;}

#### Multiple Class Selector

We might have a **conflict** while using multiple classes.

<p class = “greentea raspberry blueberry”>Green Tea<p>

<p class = “raspberry”> Raspberry Pie</p>

p.greentea{color: green;}

### ID Selector

<p id = “greentea”>Green Tea<p>

p#greentea{color: green;}

## Attribute Selector

img[width] {border: black thin solid;}

Selects all images that have a width attribute in their HTML

img[height=”300”] {border: red thin solid;}

This selects all images that have a height attribute with a value of 300

image[alt~=”flower”]{border: #ccc thin solid}

This selects all images that have an alt attribute that includes the word “flowers”

## Selecting by Sibling

h1 + p {font-style: italic}

Selects all the paragraphs that comes immediately after an <h1> element

## Combining Selectors

.blueberry p {color: purple}

Select all paragraphs that are descendants of an element in the blue berry class

div#greentea > blockquote

Descendant selector with a <div> with am id “greentea” must be the parentt of the <blockquote>

div#greentea > blockquote p

🡨………………Context……………………🡪🡨Element🡪

<p> element must be a descendant of <blockquote> which must be a child of a <div> with an id of “greentea”

div#greentea > blockquote p: first-line{font-style: italic}

🡨………………Context……………………🡪 🡨E🡪🡨Pseudo Element🡪

<p> element must be a descendant of <blockquote> which must be a child of a <div> with an id of “greentea” and first line of the paragraph must be italic

## Color

The color property specifies the color of text.

h1 {color: #00ff00;}

List of fixed colors: aqua, black, blue, gray, green, lime, maroon, navy, olive, purple, red. silver, teal, white, yellow

### Hex Code Color Calculation

# CC 66 00

🡨Red🡪 🡨Green🡪 🡨Blue🡪

For Red,

12\*16 + 16 = 208

For Green,

6\*16 + 6 = 102

For Blue,

0\*16 + 0 = 0

|  |  |  |
| --- | --- | --- |
| **Decimal (base 10)** | **Binary** | **Hexa (Base 16)** |
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0111 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| **10** | **1010** | **A** |
| **11** | **1011** | **B** |
| **12** | **1100** | **C** |
| **13** | **1101** | **D** |
| **14** | **1110** | **E** |
| **15** | **1111** | **F** |

## Background

The background shorthand property sets all the background properties in one declaration.

body {background: lightblue url("img\_tree.gif") no-repeat fixed center}

### background-color

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

body {background-color: #92a8d1;}

body {background-color: rgb(201, 76, 76);}

body {background-color: rgba(201, 76, 76, 0.3);}

body {background-color: hsl(89, 43%, 51%);}

body {background-color: hsla(89, 43%, 51%, 0.3);}

### background-image

The background-image property sets one or more background images for an element.

By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.

body {background-image: url("img\_tree.gif"), url("paper.gif");}

### background-position

The background-position property sets the starting position of a background image.

body {background-position: center;}

left top, left center, left bottom, right top, right center, right bottom, center top, center center, center bottom

x%, y%

xpos, ypos

### background-size

The background-size property specifies the size of the background images.

#example1 {background-size: auto;}

#example2 {background-size: 300px 100px;}

("auto", "cover" and "contain"), the one-value syntax (sets the width of the image (height becomes "auto"), the two-value syntax (first value: width of the image, second value: height), and the multiple background syntax (separated with comma).

**auto** Default value. The background image is displayed in its original size

**cover** Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges

**contain** Resize the background image to make sure the image is fully visible

### background-repeat

The background-repeat property sets if/how a background image will be repeated.

The background image is repeated both vertically and horizontally. The last image will be clipped if it does not fit. This is default

body {background-repeat: repeat-y}

body {background-repeat: no-repeat}

**repeat-x** The background image is repeated only horizontally

**repeat-y** The background image is repeated only vertically

**no-repeat** The background-image is not repeated.

**space** The background-image is repeated as much as possible without clipping.

**round** The background-image is repeated and squished or stretched to fill the space

### background-origin

The background-origin property specifies the origin position (the background positioning area) of a background image.

#example1 {background-origin: content-box;}

**padding-box** The background image starts from the upper left corner of the padding edge

**border-box** The background image starts from the upper left corner of the border

**content-box** The background image starts from the upper left corner of the content

### background-clip

The background-clip property defines how far the background (color or image) should extend within an element.

div {background-clip: padding-box;}

### background-attachment

The background-attachment property sets whether a background image scrolls with the rest of the page, or is fixed.

body {background-attachment: fixed;}

## Box Model

margin

Border

Padding

Content

## Border

Defines a border around the body that is dotted and the color black

h2 {border: 4px dotted blue;}

**border-width** Specifies the width of the border. Default value is "medium"

**border-style** Specifies the style of the border. Default value is "none"

**border-color** Specifies the color of the border. Default value is the color of the text

table{border-collapse: collapse} Collapses two lines into one

### Border Style

The border-style property sets the style of an element's four borders. This property can have from one to four values.

div {border-style: dotted;}

**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

**hidden**  The same as "none", except in border conflict resolution for table elements

**dotted** Specifies a dotted border

**dashed** Specifies a dashed border

**solid** Specifies a solid border

**double** Specifies a double border

**groove** Specifies a 3D grooved border. The effect depends on the border-color value

**ridge** Specifies a 3D ridged border. The effect depends on the border-color value

**inset** Specifies a 3D inset border. The effect depends on the border-color value

**outset** Specifies a 3D outset border. The effect depends on the border-color value

### Border Color

The border-color property sets the color of an element's four borders.

div {border-color: coral;}

### Border Width

The border-width property sets the width of an element's four borders.

div {border-width: thin;}

**medium** Specifies a medium border. This is default

**thin** Specifies a thin border

**thick** Specifies a thick border

### Specifying Border Sides

div {border-top-color: coral;}

div {border-top-style: dotted;}

div {border-top-width: thin;}

div {border-right-color: coral;}

div {border-right-style: dotted;}

div {border-right-width: thin;}

div {border-bottom-color: coral;}

div {border-bottom-style: dotted;}

div {border-bottom-width: thin;}

div {border-left-color: coral;}

div {border-left-style: dotted;}

div {border-left-width: thin;}

## Margin

The margin property sets the margins for an element.

p {margin: 35px;}

margin-top, margin-right, margin-bottom, margin-left

## Padding

An element's padding is the space between its content and its border.

p {padding: 35px 70px 50px 90px;}

35 pixels for top, 70 pixels for right, 50 pixels for bottom, and to 90 pixels for left:

p {padding: 35px 70px 50px}

35 pixels for top, 70 pixels for right and left, and to 50 pixels for bottom:

p {padding: 35px 70px}

35 pixels for top and bottom, and to 70 pixels for right and left:

p {padding: 35px}

Padding for all four sides of a <p> element to 35 pixels:

## Top

The top property affects the vertical position of a positioned element. This property has no effect on non-positioned elements.  
div {  
    position: absolute;  
    top: 50px;  
    border: 3px solid green;  
}

**auto** Lets the browser calculate the top edge position. This is default

**length** Sets the top edge position in px, cm, etc. Negative values are allowed.

**%** Sets the top edge position in % of containing element. Negative values are allowed

## Font-Family

The font-family property specifies the font for an element.

The font-family property can hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

There are two types of font family names: **family-name, generic-family**

p.a {font-family: "Times New Roman", Times, serif;}

There are five font family-name: **sans-serif, serif, monospace, cursive, and fantasy**

Sans-serif family: Verdana, Arial Black, Trbuchet MS, Arial, Geneva

Serif-family: Times, Times New Roman, Georgia

Monospace family: Courier, Courier New, Andale Mono

Cursive Family: Comic Sans, Apple Chancery

Fantasy family: Last Ninja, Impact

p.a {font-family: Verdana, Geneva, Arial, sans-serif;}

If Verdana is not available use Geneva, If Geneva not available use Arial or else sans-serif

**Shortcut**

font: font-style font-variant font-weight font-size/line0height font-family

.a {font: small/1.6em Verdana, Helvetica, Arial, sans-serif;}

## Font Face

With the @font-face rule, web designers do not have to use one of the "web-safe" fonts.

In the @font-face rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.

@font-face {  
    font-family: myFirstFont;  
    src: url(sansation\_bold.woff);  
    font-weight: bold;  
}

**Formats of Fonts**

TrueType fonts: **.ttf**

OpenType fonts: **otf**

Embedded OpenType fonts: **.eot**

SVG fonts: **.svg**

Web open font format: **.woff**

## Font Weight

The font-weight property sets how thick or thin characters in text should be displayed.

p.thick {font-weight: bold;}

**normal** Defines normal characters

**bold**  Defines thick characters

**bolder** Defines thicker characters

**lighter**  Defines lighter characters

**100, 200, 300, 400, 500, 600, 700, 800, 900**

## Font Style

The font-style property specifies the font style for a text.

p.a {font-style: normal;}

**normal** The browser displays a normal font style. This is default

**italic**  The browser displays an italic font style

**oblique** The browser displays an oblique font style

## Font Size

The font-size property sets the size of a font.

div.a {font-size: 15px;}  
div.b {font-size: large;}  
div.c {font-size: 150%;}

## Text Decoration

The text-decoration property specifies the decoration added to text, and is a shorthand property for: text-decoration-line (required), text-decoration-color, text-decoration-style

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

h1 {text-decoration: underline overline dotted red;}  
h2 {text-decoration: underline overline wavy blue;}

## Line Height

The line-height property specifies the height of a line.

div.a {line-height: normal;}  
div.b {line-height: 1.6;}  
div.c {line-height: 80%;}  
div.d {line-height: 200%;}

div.b {line-height: 1.6em;}

em 🡪 1.6 times more than the inherited font size

## List Style

The list-style property is a shorthand for the following properties:

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

list-style-type

list-style-position

list-style-image

### List Style Image

It replaces the list-item marker with an image

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

## Text Alignment

The text-align property specifies the horizontal alignment of text in an element.

div.a {text-align: center;}  
div.b {text-align: left}  
div.c {text-align: right;}  
div.c {text-align: justify;}

**justify** Stretches the lines so that each line has equal width (like in newspapers and magazines)

## Letter Spacing

The letter-spacing property increases or decreases the space between characters in a text.

h1 {letter-spacing: 3px;}

**normal** No extra space between characters. This is default

**length** Defines an extra space between characters (negative values are allowed).

## Media Queries

Media queries area an are of active development by the standards groups.

The @media rule is used in media queries to apply different styles for different media types/devices.

@media only screen and (max-width: 600px) {  
    body {  
        background-color: lightblue;  
    }  
}

@media only screen and (orientation: landscape) {  
    body {  
        background-color: lightblue;  
    }  
}

/\* When the width is between 600px and 900px OR above 1100px - change the appearance of <div> \*/  
@media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {  
  div.example {  
    font-size: 50px;  
    padding: 50px;  
    border: 8px solid black;  
    background: yellow;  
  }  
}

Media queries can be used to check many things, such as:

* Width and height of the viewport
* Width and height of the device
* Orientation (is the tablet/phone in landscape or portrait mode
* Resolution

<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">

<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">

## Pseudo Class

a:link{color: green}

The selector is applied to links when they are in an unvisited state

a:visited{color: red}

This selector is applied to links when they are visited

a:hover{color: yellow}

This selector is applied when we hover over a link

p:nth-child(even){background-color:red;}

p:nth-child(2n){background-color:red;}

p:nth-child(odd){background-color:green;}

p:nth-child(2n + 1){background-color:green;}

## Pseudo Elements

p:first-letter{font-size: 3em}

First letter of the paragraph is becoming large

p:first-line{font-style: italic}

First line is becoming italic

## Cascading Styling Sheets

0 0 0

**First Digit:** Does this selector have any ids. One point for each

**Second Digit:** Does this selector have any classes or pseudo-classes. One point for each

**Third Digit:** Does this selector have any element names. One point for each

Override a style using

h1{font-size: 200% **!important**;}

**Specificity Point Examples**

|  |  |  |
| --- | --- | --- |
| h1.greentea | 011 |  |
| p img | 002 |  |
| a:link | 011 |  |
| ol li p | 003 |  |
| .green | 011 |  |
| #elixirs h1 | 102 | Winner |
| em | 001 |  |
| span.cd | 011 |  |
| #sidebar | 101 |  |

Reorder the rules, from highest score to lowest score in CSS

## Float

The float property specifies how an element should float.

Note: Absolutely positioned elements ignores the float property!

img {  
    float: right;  
}

## Position

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

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

**static** Default value. Elements render in order, as they appear in the document flow

**absolute** The element is positioned relative to its first positioned (not static) ancestor element

**fixed** The element is positioned relative to the browser window

**relative** The element is positioned relative to its normal position,

**sticky** The element is positioned based on the user's scroll position

#parent1 {  
    position: static;  
    border: 1px solid blue;  
    width: 300px;  
    height: 100px;  
}  
  
#child1 {  
    position: absolute;  
    border: 1px solid red;  
    top: 70px;  
    right: 15px;  
}  
  
#parent2 {  
    position: relative;  
    border: 1px solid blue;  
    width: 300px;  
    height: 100px;  
}

#child2 {  
    position: absolute;  
    border: 1px solid red;  
    top: 70px;  
    right: 15px;  
}

## Display

The display property specifies the display behavior (the type of rendering box) of an element.

p.ex1 {display: none;}  
p.ex2 {display: inline;}  
p.ex3 {display: block;}  
p.ex4 {display: inline-block;}

**inline** Displays an element as an inline element

Any height and width properties will have no effect

**block** Displays an element as a block element

It starts on a new line, and takes up the whole width

**flex** Displays an element as a block-level flex container

**grid** Displays an element as a block-level grid container

**inline**-**block** Displays an element as an inline-level block container.

The element itself is formatted as an inline element. We can apply height and width values

**inline**-**flex** Displays an element as an inline-level flex container

**inline**-**grid** Displays an element as an inline-level grid container

**inline**-**table** The element is displayed as an inline-level table

**list**-**item** Let the element behave like a <li> element

**run**-**in** Displays an element as either block or inline, depending on context

**table** Let the element behave like a <table> element

**table**-**caption** Let the element behave like a <caption> element

**table**-**column**-**group** Let the element behave like a <colgroup> element

**table**-**header**-**group** Let the element behave like a <thead> element

**table**-**footer**-**group** Let the element behave like a <tfoot> element

**table**-**row**-**group** Let the element behave like a <tbody> element

**table**-**cell** Let the element behave like a <td> element

**table**-**column** Let the element behave like a <col> element

**table**-**row** Let the element behave like a <tr> element

## Vertical Align

The vertical-align property sets the vertical alignment of an element.

img.a {vertical-align: baseline;}  
img.b {vertical-align: text-top;}  
img.c {vertical-align: text-bottom;}  
img.d {vertical-align: sub;}  
img.e {vertical-align: super;}

## Vendor Specific CSS Properties

-moz-transform

Moz stands for mozilla

div{

transform: rotate(45deg), **(General)**

-webkit-transform: rotate(45deg), **Safari & Chrome**

-moz-transform: rotate(45deg), (**Mozilla)**

-ms-transform: rotate(45deg), **(Internet Explorer)**

-o-transform: rotate(45deg) **(Opera)**

}

## CSS TERMS

|  |  |  |  |
| --- | --- | --- | --- |
| background | text-align | @media | -webkit-transform |
| border | letter-spacing | border-style | -ms-transform |
| color | font-weight | border-color | -o-transform |
| margin | font-style | table |  |
| padding | list-style | table-cell |  |
| font-family | line-height | table-row |  |
| position | font-size | vertical-align |  |
| top | @font-face | -moz-transform |  |