- # **HTML Introduction**

HTML describe webpage structure with tags marking the documents appropriately to tell the browser what each tagged element is.

Here, defines that document is html5, is root element of an html page,

contains meta information about the html page, <title> specifies title for the page(which is shown in the browser's title bar or page's tab. is a container for all visible contents of a site.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

`<br>` is an empty element...elements without content

- Browsers use tag to determine how to display the content of a

- UTF-8 is preferred way to encode html file and save type as "All Files(.)"

.htm or .html extensions are the same thing. You decide whichever one you want. `return`

HTML BASICS

`<!DOCTYPE>` declaration represents the document type, and helps browsers display web pages correctly. Must appear once in the webpage, not case sensitive

- This is how to declare doctype for HTML5 `<!DOCTYPE html>`

*Attributes provide additional information about html elements `<a>` defines html links `'href'` attribute defines link destination
`<img>` defines html images. The attributes src(source file), alternative text(alt), width,and height
When you inspect a site, you can edit the html or css on-the-fly.

Example of using the imag tag

```
<!DOCTYPE html>
<html>
<body>

<h2>HTML Images</h2>
<p>HTML images are defined with the img tag:</p>

<img src=" s.jpg" alt=" s.com" width="104" height="142">

<a href="https://www. s.com">This is a link</a>

</body>
</html>
return
```

HTML ELEMENTS

- HTML element is everything from start to end of tag `<tagname>Content goes here...</tagname>`
- The `<html>` element is the root element and it defines the whole html document.

**Even though skipping the end tag can work, skipping the end tag is not reliable

Never rely on the fact that some tags could run without end tag. Unexpected results and errors may occur if you forget the end tag.

- Empty elements have no content. `<br>` defines line break and it is an empty element without a closing tag.

HTML is not case sensitive but xhtml is more strict. So always use lower case letter do denote tags

```
return
```

HTML ATTRIBUTES

**Attribute comes as name="value"

- HTML links: `<a href="https://www. s.com">Visit  s</a>`

The SRC attribute:

```
<h2>The src Attribute</h2>
<p>HTML images are defined with the img tag, and the filename of the image source is
specified in the src attribute:</p>
<img src="img_girl.jpg" width="500" height="600">
```
**specifying the url in the `src` attribute
Absolute url: links to image on another site: src="a.jpg"

Relative url: Links to image hosted within the website. No domain name included here.
src="img_girl.jpg" src="/images/img_girl.jpg" **in this case, the slash is relative to the
site url

**Width and height attribute(pixels) with `<img>`
```
<img src="img_girl.jpg" width="500" height="600">
```
**Alt attribute includes alternative text for the image incase a user can not read the

image:

**style attribute adds style elements like color, font, size, etc to an image

```
<p style="color:red;">This is a red paragraph.</p>
```
**Lang attribute: Should always be included inside an tag the declare website language
to assist search engines locate you

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```
*You can also include country attribute.

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```
**The title attribute: Gives extra information about an element and the attribute will be
displayed as a tooltip when mouse pointer runs over it.

```
<p title="I'm a tooltip">This is a paragraph.</p>
```
having attributes in quote and in cases where there is a space in the attribute, you have
to have the space.

*when attribute value contains one of the quotes, use the other quote

```
<p title='John "ShotGun" Nelson'>
```
Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```
***Attributes Summary

- All HTML elements can have attributes
- The href attribute of `<a>` specifies the URL of the page the link goes to
- The src attribute of `<img>` specifies the path to the image to be displayed
- The width and height attributes of `<img>` provide size information for images
- The alt attribute of `<img>`provides an alternate text for an image
- The style attribute is used to add styles to an element, such as color, font, size, and more
- The lang attribute of the `<html>` tag declares the language of the Web page
- The title attribute defines some extra information about an element

```
return
```

HTML HEADINGS

Headings go from `<h1>` to `<h6>`. Search engines use them to understand the structure of the webpage
Use h1 for main headings followed by h2 and so on

Each heading has its default size. You can change that by specifying inside a style element.

*Change your style like this:

```
<h1 style="font-size:60px;">Heading 1</h1>
return
```

HTML PARAGRAPHS

In html, you cannot be sure of how html will be displayed on each screen size. Adding extra lines or spaces to the document does not add new lines to the page. Browser removes extra spaces once page displays

```
<p>This is a paragraph</p>
```
- Horizontal rules: `<hr>` Used to separate content(define change) and defines a thematic break in an HTML page. It shows up as an horizontal line. `<hr>` tag has no closing tag since it is an empty tag.
- Line breaks: `<br>` element Use this when you want a line break without starting a new paragraph.

The Poem Problem: Poems have to be included in the `<pre>` preformatted text tag

The `<pre>` tag displays text in fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Tags summary

```
<p>      Defines a paragraph
<hr>     Defines a thematic change in the content
<br>     Inserts a single line break
<pre>    Defines pre-formatted text
return
```

HTML STYLES

The HTML Style attribute syntax:

```
<tagname style="property:value;">
```

- Where The property is a CSS property and value is a CSS value.

- Background color: Defines the background color for an HTML element

```
<body style="background-color:powderblue;">
```

- Set background color for two different elements:

```
<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:tomato;">This is a paragraph.</p>
```
**Text color

```
 <h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```
**Fonts

```
 <h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```
**Text Size

```
 <h1 style="font-size:300%;">This is a heading</h1>
```

```
<p style="font-size:160%;">This is a paragraph.</p>
```
**Text alignment

```
 <h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```
Styles summary

- Use the style attribute for styling HTML elements
- Use background-color for background color
- Use color for text colors
- Use font-family for text fonts
- Use font-size for text sizes
- Use text-align for text alignment

```
return
```

HTML FORMATTING

```
<b> - Bold text
<strong> - Important text
<i> - Italic text...often used to indicate a technical term, a phrase from another
language, a thought, a ship name, etc.
<em> - Emphasized text. A screen reader will emphasize the text using verbal stress
<mark> - Marked text. Element that should be highlighted. You can use style to change
the color.
<small> - Smaller text
<del> - Deleted text. Browsers usually strike a line through
<ins> - Inserted text... (into a document) Browsers usually underline this. Can be
used after del.
<sub> - Subscript text...appears half a character below the normal line
<sup> - Superscript text...can be used for footnotes citation
*<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
return
```

HTML QUOTATIONS

HTML QUOTATIONs AND CITATION ELEMENTS

```
<abbr>  Defines an abbreviation or acronym. Marking them give information to
browsers, translation systems and search engines. Use global attribute title with it
to show full meaning of the abbreviation.
```

```
<address>        Defines contact information for the author/owner of a document or an
article. This can be email, url, physical, mobile no, social media handles etc.
Browsers add line break before and after <address> element.
<bdo>   Defines the text direction.Used to override current text direction
<blockquote>     Defines a section that is quoted from another source. Browser
usually indents this. cite is used here to show the source, not href.
<cite>  Defines the title of a creative work. Creator's name is not a title.Browsers
will usually cite this element in italic
<q>        Defines a short inline quotation.Browser inserts quotation marks around the
quotation
```

- Quote

```
 <p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

- Short Quotations

```
 <p>WWF's goal is to: <q>Build a future where people live in harmony with
nature.</q></p>
```

- The use of abbreviation `<abbr>`

```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```
Address

```
<address>
Written by John Doe.<br>
Visit us at:<br>
Example.com<br>
Box 564, Disneyland<br>
USA
</address>
```

- Cite for work title

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

- Bi-Directional Override(BDO)

```
 <bdo dir="rtl">This text will be written from right to left</bdo>
return
```

HTML COMMENTS

There is only exclamation at the start not at the end

```
<!-- Write your comments here -->
```
Example usage

```
<!-- This is a comment -->

<p>This is a paragraph.</p>

<!-- Remember to add more information here -->
```

- Comments can help you to debug your html lines of code

```
return
```

HTML COLORS

You specify them with color names, RGB, HEX, HSL,RGBA(the alpha is for transparency),HSLA values.

HTML supports 140 standard color names

- Specify background color as shown below:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

- Set color of text

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

- Set Color of border (the square box) around text

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

RGB

- These are all the same

```
rgb(255, 99, 71)
#ff6347
hsl(9, 100%, 64%)
```

- RGBA, HSLA

```
rgba(255, 99, 71, 0.5)
hsla(9, 100%, 64%, 0.5)
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
return
```

RGBA

RGB and RGBA Coloring

Red, Green, Blue, and Alpha

`rgb(r,g,b)` each parameter has a value between 0 and 255 i.e 256 in total
Total number of colors: 256 x 256 x 256 = 16777216 possible colors!

```
Red color: rgb(255, 0, 0),
Green color: rgb(0, 255, 0)
Black: rgb(0, 0, 0)
White:rgb(255, 255, 255).
Grey: rgb( x,x,x) where x is not zero or 255. Black and white are at the two ends of
the color spectrum.
```
RGBA

`rgba(red, green, blue, alpha)` alpha 0 means fully transparent while 1.0 is not transparent at all.
```
return
```

HTML HEX COLORS

RR(RED), GG(GREEN)BB(BLUE)...HEXADECIMAL INTEGERS SPECIFY THE COMPONENTS OF THE COLORS, EACH WITH VALUES RANGING FROM 00 TO ff (same as decimal 0-255)

`#ff0000`- red since red has the highest value (ff) while the green and blue appear as 00 and 00

`#00ff00` appears as green since green is set to the highest value(ff) and the remaining two have zero values

To display black, set all color parameters to 00, like this: `#000000`. To display white, setll all color parameters to ff: `#ffffff`

Shades of gray: Equal values for all the parameters: Each one of these are different shades of gray.

```
#404040
#686868
#a0a0a0
return
```

HSL AND HSLA

HSL: Hue, Saturation, and, lightness

HSLA: Hue, Saturation, Lightness, and Alpha.

Hue: Degree on the colorwheel from 0 to 360 0 is red, 120 gree, 240 blue

Saturation: It is a percentage value. 0% is black, and 100 % is full color

Lightness:Alsp a percentage value: 0% is black and 100% white

Saturation can be described as the intensity of a color. It is about how much shade of gray you want to have showing up on the color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

0% is completely gray, you can no longer see the color.

** The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).

Shades of Gray: Can be obtained by setting hue and saturation to 0, and adjust the lighness from 0 % to 100 % to get darker/lighter shades:

```
hsl(0, 0%, 20%)   --darker shade of grey
hsl(0, 0%, 70%)== lighter shade of grey
```
HSLA Color Values:

Alpha rep transparency. 0.00 means fully transparent and 1.0 not transparent at all.

```
return
```

___

HTML CSS

HTML STYLES-CSS

CSS can control layout of multiple pages at same time.

CSS: CAscading style sheets: To control color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes etc

Tip: The word cascading means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

CSS can be added to HTML documents in 3 ways:

Inline - by using the style attribute inside HTML elements Internal - by using a `<style>` element in the `<head>` section External - by using a `<link>` element to link to an external CSS file
```
return
```

CSS INLINE

Inline CSS: To apply a unique style to a single HTML element, and it uses the style attribute of such element.

- Set h1 color to blue and p to red

```
<h1 style="color:blue;">A Blue Heading</h1>
```

```
<p style="color:red;">A red paragraph.</p>
return
```

CSS INTERNAL

Internal CSS: To define style for a single HTML Page and it defined in section of an html page within <style> element

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
return
```

CSS EXTERNAL

External CSS: Define style for many HTML pages. To use the external style, you add a link to it in the head section of the html. File must not contain any HTML Code and can be written in any text editor. Save with a .css extension.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

- Styles.css looks like this

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

return

CSS FONTS COLORS AND SIZES

CSS Colors, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.

The CSS color property defines the text color to be used.

The CSS font-family property defines the font to be used.

The CSS font-size property defines the text size to be used.

- Use of CSS color, font-family, and font-size properties.

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
return
```

CSS Border

The CSS border property defines a border around an HTML element.

Tip: You can define a border for nearly all HTML elements.

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

```
return
```

CSS Padding

The CSS padding property defines a padding (space) between the text and the border.

Use of CSS border and padding properties:

```
p {
  border: 2px solid powderblue;
  padding: 30px;
}
return
```

CSS Margin

The CSS margin property defines a margin (space) outside the border.

```
p {
  border: 2px solid powderblue;
  margin: 50px;
}
return
```

CSS link to external

External style sheets can be referenced with a full URL or with a path relative to the current web page.

*Example, linking to a full style sheet

```
<link rel="stylesheet" href="https://www. s.com/html/styles.css">
```

- Links to a style sheet located in the html folder on the current web site:

```
<link rel="stylesheet" href="/html/styles.css">
```
HTML CSS SUMMARY: Use the HTML style attribute for inline styling Use the HTML <style> element to define internal CSS Use the HTML element to refer to an external CSS file Use the HTML element to store <style> and elements Use the CSS color property for text colors Use the CSS font-family property for text fonts Use the CSS font-

size property for text sizes Use the CSS border property for borders Use the CSS padding property for space inside the border Use the CSS margin property for space outside the border

`return`

HTML LINKS

They are hyperlinks that can be clicked to be redirected to another page. When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. A link can be an image or any other HTML element!

`<a> tag defines hyperlinks:`

`<a href="url">link text</a>`

`href attribute is the link destination, and the link text is what the use sees.`

` <a href="https://www. s.com/">Visit  s.com!</a>`
** By default, links will appear as follows in all browsers: An unvisited link is underlined and blue A visited link is underlined and purple An active link is underlined and red

You can also style a link with CSS `return`

Links

HTML Links - The target Attribute By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

Possible Values for the target attribute:

```
_self - Default. Opens the document in the same window/tab as it was clicked
_blank - Opens the document in a new window or tab
_parent - Opens the document in the parent frame
_top - Opens the document in the full body of the window
```

- Absolute vs Relative URLs Absolute URLs use the full web address in href attribute.

A link to a page within the current is specified with a relative URL(OMITTING HTTPS://WWW.)

- Examples of relative and absolute URL:

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

** HTML Links - Use an Image as a Link: To do this, just use in image   tae

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

** Link to an Email Address: use mailto: inside href attribute to create a link that oopens in the user's email program (to let them send a new email):

** `<a href="mailto:someone@example.com">Send email</a>`

** To use a button as an HTML Link; you need to add some Js CODES. JS allows you to specify what happends at cetain events, such as a click of a button:

** Here is the JS code `<button onclick="document.location='default.asp'">HTML Tutorial</button>`

** LInk titles: Shown as tooltip when mouse pointer is moved on element.

```
<a href="https://www. s.com/html/" title="Go to  s HTML section">Visit our HTML Tutorial</a>
```

** More on relative and absolute URLS:

- Use a full URL to link to a web page:

```
<a href="https://www. s.com/html/default.asp">HTML tutorial</a>
```

- Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

- Link to a page located in the same folder (within the html folder of the current website) as the current page: `<a href="default.asp">HTML tutorial</a>`

Use the `<a>` element to define a link Use the `href` attribute to define the link address Use the target attribute to define where to open the linked document Use the `<img>` element

(inside `<a>`) to use an image as a link Use the mailto: scheme inside the href attribute to create a link that opens the user's email program

```
<a href="https://www. s.com/html/default.asp">HTML tutorial</a>
```

CHAPTER SUMMARY:

Use the element to define a link Use the href attribute to define the link address Use the

target attribute to define where to open the linked document Use the   element (inside) to use an image as a link Use the mailto: scheme inside the href attribute to create a link that opens the user's email program

```
return
```

Colors

HTML Links - Different Colors

By default, a link will appear like this (in all browsers):

An unvisited link is underlined and blue A visited link is underlined and purple An active link is underlined and red

**The default css for links is :

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```

**style a link like a button here:

```
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
return
```

Link Bookmarks

HTML Links - Create Bookmarks: can enable users to jump to certain part of a webpage. This helps for a long webpage.

- To create bookmark, create bookmark then add link to it. Clicking the page automatically redirects to boookmarked location in the webpage

-Use id attribute to create a bookmark:

```
<h2 id="C4">Chapter 4</h2>
```
-Add link to the bookmark from within the same page.

```
 <a href="#C4">Jump to Chapter 4</a>
```
-Add link to a bookmark on another page:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```
**To remove the underline from link, remove text decoration:

```
<a href="html_images.asp" style="text-decoration:none">HTML Images</a>
```
Chapter Summary:

Chapter Summary

Use the id attribute (`id="value"`) to define bookmarks in a page Use the href attribute (`href="#value"`) to link to the bookmark
```
return
```

HTML IMAGES

IMAGES

```
<img src="pic_trulli.jpg" alt="Italian Trulli">
```
Image tag is used to insert images in a webpage. Images are linked to a webpage not embeded.

SRC attributes src - Specifies the path to the image alt - Specifies an alternate text for the image

Syntax `<img src="url" alt="alternatetext">`
Browsers get image from a web server and inserts into the page each time yu load. Image stays in same spot in relation to a webpage...else the alt text shows up to indicate that the link is brocken.

**Value of alt attribute shows in cases where: because of slow connection, an error in the src attribute, or if the user uses a screen reader which reads html code and reads out for the user...usually used by the visually impaired people.

**You can also specify width and height in the style attribute of the image

```
<img src="img_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;">
```
**With and height attribute as width and height attribute...which is always in pixels `<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">`
**It is better to use style attribute so the stylesheet does not end up changing image size.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  width: 100%;
}
</style>
</head>
<body>

<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>
</html>
```
**For images in the subfolder, you must include the folder name

```
<img src="/images/html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">
```
If your site point to an external image on another server, specify full URL to point to the image.

```
mg src="https://www. s.com/images/ s_green.jpg" alt=" s.com">
```
NB on external images: Might be removed by site owner anytime. Also getting permission may be necessary to avoid copyright violation

**Animated Images:: HTML allows GIFs

```
<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">
```

**Image as a tag:: Put   tag inside the tag

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```
**Let image float to the right or to the left of a text using CSS float property

```
<p><img src="smiley.gif" alt="Smiley face"
style="float:right;width:42px;height:42px;">
The image will float to the right of the text.</p>
```

```
<p><img src="smiley.gif" alt="Smiley face"
style="float:left;width:42px;height:42px;">
The image will float to the left of the text.</p>
```
**Common image formats supported on a website

bbreviation File Format File Extension APNG Animated Portable Network Graphics .apng GIF Graphics Interchange Format .gif ICO Microsoft Icon .ico, .cur JPEG Joint Photographic Expert Group image .jpg, .jpeg, .jfif, .pjpeg, .pjp PNG Portable Network Graphics .png SVG Scalable Vector Graphics .svg

**Loading large images takes time and can slow down your webpage...so use images carefully.

```
return
```

IMAGE MAP

Image Maps::Allows you to perform actions depending on where in image you click. You need an image, some html code that describes the clickable area.

The image: Inserted using image tag, but you must add usemap attribute in this case.

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">
```

-The usemap value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.

You can use any image as an image map.

-add a `<map>` element linked to the image using the name attribute.
`<map name="workmap">`

name attribute must have the same value as the   's usemap attribute

-Add clickable area to the image: You define the clickable area using an element.

- Define shape of the clickable area using any of these values: rect - defines a rectangular region circle - defines a circular region poly - defines a polygonal region default - defines the entire region

-Define coordinate to be able to place the clickable area onto the image.

Shape="rect"
The coordinates for `shape="rect"` come in pairs, one for the x-axis (distance from left) and one for the y-axis(distance from top).
So, the coordinates 34,44 is located 34 pixels from the left margin and 44 pixels from the top:

The coordinates 270,350 is located 270 pixels from the left margin and 350 pixels from the top:

Shape="circle"
To add a circle area, first locate the coordinates of the center of the circle:

337,300

then specify the radius: 44 pixels

**

`<area shape="circle" coords="337, 300, 44" href="coffee.htm">`
Shape=poly"
The `shape="poly"` contains several coordinate points, which creates a shape formed with straight lines (a polygon).
This can be used to create any shape. Such as a croissant. You have to find x and y coordinates for all edges of the croissant.

- Each point has x and y...specify all the coordinates around the perimeter of the polygonal shape.

```
<area shape="poly"
coords="140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,322,40,259,1
03,161,128,147" href="croissant.htm">
```
**You can also trigger a JS function by clicking the poissant shape. Add a click event to the element to execute a JavaScript function:

```
  <map name="workmap">
    <area shape="circle" coords="337,300,44" onclick="myFunction()">
</map>

<script>
function myFunction() {
    alert("You clicked the coffee cup!");
}
</script>
```

You can use href with the function to inform user to confirm what they are trying to do before they follow the hyperlink on the picture.

**Image Map summary: =

```
<img>    Defines an image
<map>    Defines an image map
<area>   Defines a clickable area inside an image map
<picture>        Defines a container for multiple image resources
return
```

Background Images

Background Images: Repeats itself if it is smaller

*Add background image on an HTML element using style attribute and css background-image.

```
  <div style="background-image: url('img_girl.jpg');">
```
**You can also specify background image in the style element of the head section

```
  <style>
div {
    background-image: url('img_girl.jpg');
}
</style>
```
**Adding background image for an entire page:specify background image on the body element

```
<style>
body {
    background-image: url('img_girl.jpg');
}
</style>
```

**Background image repeat:: Background image smaller than the element will repeat itself, horizontally and vertically,until it reaches the end of the element.

```
<style>
body {
   background-image: url('img_girl.jpg');
}
</style>
```

**You can set background repeat property to no repeat to prevent the background image from repeating itself.

```
 <style>
body {
   background-image: url('example_img_girl.jpg');
   background-repeat: no-repeat;
}
</style>
```

- Background Cover: You can set background-size property to cover to make background image to cover the entire element. set the background-attachment property to fixed to make sure the entire element is always covered...This keeps image original proportions.

```
<style>
body {
   background-image: url('img_girl.jpg');
   background-repeat: no-repeat;
   background-attachment: fixed;
   background-size: cover;
```
** Background stretch:: Allows you to make the background image to stretch to fit the entire element. Set background-size property to 100% 100%

```
 <style>
body {
   background-image: url('img_girl.jpg');
   background-repeat: no-repeat;
   background-attachment: fixed;
   background-size: 100% 100%;
}
</style>
return
```

PICTURE ELEMENT

`<picture>` element allows you to display different pictures for different devices or screen sizes.

```
 <picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  <img src="img_girl.jpg">
</picture>
```

- `<img>` element should be specified last in the picture element. This element is used by the browsers that do not support the picture elment or if none of the tags work.

When to use the picture element:: -Bandwidth:: Wnen on a small screen device, browser searches through the source elements to find the matching element first.

-Format Support:: You can also add all image formats so that different browser image supports can be accommodated. In this case, the device use the first image format that it supports and ignore the rest.

```
<picture>
  <source srcset="img_avatar.png">
  <source srcset="img_girl.jpg">
  <img src="img_beatles.gif" alt="Beatles" style="width:auto;">
</picture>
```
HTML IMAGE TAGS SUMMARIZED:::

```
<img>    Defines an image
<map>    Defines an image map
<area>   Defines a clickable area inside an image map
<picture> Defines a container for multiple image resources
return
```

HTML TABLES

```
<table> tag defined the HTML table.
<tr> defines table row
<th> defines table header. By default bold and centered.
<td> defines table data/cell.. By default regular and left-aligned. They are data
containers for text, images, lists, other tables, etc.
```

- A Simple HTML Table

```
 <table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
```

```
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
    </tr>
    <tr>
      <td>Eve</td>
      <td>Jackson</td>
      <td>94</td>
    </tr>
</table>
```
** Table looks like this:: Firstname Lastname Age Jill Smith 50 Eve Jackson 94 John Doe 80

- Add a border to HTML table:::

```
table, th, td {
    border: 1px solid black;
}
```
** Collapsed Table Border:: Add Border collapse property to enable all borders to collapse into one border.

```
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
```
** Add Cell Padding::: Specifies the space between the cell and its borders. If not specified, table cells are displayed without padding. Use css padding property to set padding:

```
th, td {
    padding: 15px;
}
```
** Left-align Headings::Table headings are by default bold and centred.

To left-align table headings, use CSS text-align property.

```
th {
    text-align: left;
}
```
** Add border spacing:: Specifies space between cells. Use CSS border-spacing property. Border spacing has no effect for table with collapsed borders

```
table {
    border-spacing: 5px;
}
```
** Cells that Spans Many Columns:: use colspan attribute

```
<table style="width:100%">
  <tr>
```

```
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```
** Looks like this:Name Telephone Bill Gates 55577854 55577855

** Make cell that spans many rows::use rowspan. rowspan=2 means that you will specify two row elements to contain the content spanning two rows

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```
** Add a caption to the table:: use tag...inserted immediately after the table tag

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```
** Define a special style for one table. Add an id to the table and then define the special style next

```
<table id="t01">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
```

```
    <tr>
      <td>Eve</td>
      <td>Jackson</td>
      <td>94</td>
    </tr>
</table>
```

```
#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

you can also add more styles:

```
#t01 tr:nth-child(even) {
  background-color: #eee;
}
#t01 tr:nth-child(odd) {
  background-color: #fff;
}
#t01 th {
  color: white;
  background-color: black;
}
```

- CHAPTER SUMMARY _HTML CHAPTER SUMMARY Use the HTML `<table>` element to define a table Use the HTML `<tr>`element to define a table row Use the HTML `<td>` element to define a table data Use the HTML `<th>` element to define a table heading Use the HTML `<caption>` element to define a table caption Use the CSS border property to define a border Use the CSS border-collapse property to collapse cell borders Use the CSS padding property to add padding to cells Use the CSS text-align property to align cell text Use the CSS border-spacing property to set the spacing between cells Use the colspan attribute to make a cell span many columns Use the rowspan attribute to make a cell span many rows Use the id attribute to uniquely define one table

```
return
```

HTML LISTS

** unordered list has bullets ** ordered lists have numberings or some kind of sequence labels

- Unordered html list:: each starts with `<ul>` tag and each list item starts with `<li>` tag by default, list item is marked with small black circles as bullets.

```
<ul>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ul>
return
```

ORDERED

- Ordered HTML list: Marked by

  and each list item is marked by

  i.   tag. Numbers are usd to mark list items by default.

```
<ol>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ol>
```

HTML Description lists:: List of terms with description of each term in the list. description list is marked by `<dl>` tag and each description term (name) is marked by `<dt>` xyz and the description of each description list term is tagged by `<dd>`:

```
<dl>
 <dt>Coffee</dt>
 <dd>- black hot drink</dd>
 <dt>Milk</dt>
 <dd>- white cold drink</dd>
</dl>
```

- HTML List tags summary Tag Description `<ul>` Defines an unordered list `<ol>` Defines an ordered list `<li>` Defines a list item `<dl>` Defines a description list `<dt>` Defines a term in a description list `<dd>` Describes the term in a description list

```
return
```

UNORDERED

**HTML Unordered lists cont'd-List marker

To choose list marker, use css list-style-type property to define style of the list item market.

Value Description disc Sets the list item marker to a bullet (default) circle Sets the list item marker to a circle square Sets the list item marker to a square none The list items will not be marked

- Disc(filled circle)

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Open(unfilled circle)

```
 <ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```
** Square (filled with black)

```
 <ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```
** None (No marker)

```
 <ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```
** Nested HTML List: List inside list

```
 <ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```
** A list item (`<li>`) can contain a new list, and other HTML elements, like images and links, etc.

- Horizontal list with CSS

You can style lists horizontally, to create a navigation menu.

** You can style list horizontally to create a navigation menu. The float style left sets the navigation menu to the left

```html
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```
** Chapter Summary Chapter Summary

Use the HTML `<ul>` element to define an unordered list Use the CSS list-style-type property to define the list item marker Use the HTML `<li>` element to define a list item

Lists can be nested List items can contain other HTML elements Use the CSS property float:left to display a list horizontally

** HTML List Tags Summary

```
HTML List Tags
Tag      Description
<ul>     Defines an unordered list
<ol>     Defines an ordered list
<li>     Defines a list item
<dl>     Defines a description list
<dt>     Defines a term in a description list
<dd>     Describes the term in a description list
return
```

HTML Ordered List:: <ol> tag defines an ordered list which can be numerical or alphabetical. List are marked by numbers by default

** Ordered HTML List - The Type Attribute With the type attribute of the

tag, you can define the type of the list item marker

```
Type       Description
type="1"   The list items will be numbered with numbers (default)
type="A"   The list items will be numbered with uppercase letters
type="a"   The list items will be numbered with lowercase letters
type="I"   The list items will be numbered with uppercase roman numbers
type="i"   The list items will be numbered with lowercase roman numbers
```

   o   Numbers

```
 <ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

   o   Uppercase Letters:

```
 <ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

   o   Lowercase Letters:

```
 <ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

o Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

o Lowercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

o Control List Counting: Enables you to be able to start counting controlled list at a certain number other than 50

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
return
```

OTHER

o Nested HTML Lists(list inside list)

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ol>
      <li>Black tea</li>
      <li>Green tea</li>
    </ol>
  </li>
  <li>Milk</li>
</ol>
```

o Chapter summary:

Use the HTML `<ol>` element to define an ordered list Use the HTML type attribute to define the numbering type Use the HTML `<li>` element to define a list item Lists can be nested List items can contain other HTML elements
HTML List Tags

```
<ul>     Defines an unordered list
<ol>     Defines an ordered list
<li>     Defines a list item
<dl>     Defines a description list
<dt>     Defines a term in a description list
<dd>     Describes the term in a description list
```

- o Description lists: List of terms with a description of each term. `<dl>` tag defines the list, `<dt>` defines the term (name), and `<dd>` tag describes each term:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

- o Description list summary: Use the HTML `<dl>` element to define a description list Use the HTML `<dt>` element to define the description term Use the HTML `<dd>` element to describe the term in a description list

HTML List Tags

```
**Tag     Description
<ul>     Defines an unordered list
<ol>     Defines an ordered list
<li>     Defines a list item
<dl>     Defines a description list
<dt>     Defines a term in a description list
<dd>     Describes the term in a description list
return
```

HTML BLOCK AND INLINE

HTML BLOCK AND INLINE ELEMENTS

Every HTML element has a default display value, depending on what type of element it is. Display values can be block or inline

- o Block-level Elements Block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can)

`<div>` element is a block-level element
Other block-level element include: Here are the block-level elements in HTML:

```
<address>
<article>
<aside>
<blockquote>
<canvas>
<dd>
<div>
<dl>
<dt>
<fieldset>
<figcaption>
<figure>
<footer>
<form>
<h1>-<h6>
<header>
<hr>
<li>
<main>
<nav>
<noscript>
<ol>
<p>
<pre>
<section>
<table>
<tfoot>
<ul>
<video>
```
** Inline Elements: It does not start on a new line but only takes as much width as necessary. example: This is a elements inside a paragraph.

```
<span>Hello World</span>
```

- o   The list of inline elements:

Here are the inline elements in HTML:

```
<a>
<abbr>
<acronym>
<b>
<bdo>
<big>
<br>
<button>
<cite>
<code>
<dfn>
```

```
<em>
<i>
<img>
<input>
<kbd>
<label>
<map>
<object>
<output>
<q>
<samp>
<script>
<select>
<small>
<span>
<strong>
<sub>
<sup>
<textarea>
<time>
<tt>
<var>
```

- An inline element cannot contain a block-level element

The `<div>` Element
`<div>` element is used as a container for other HTML elements.
`<div>` element has no required attribute. the common attributes of the div element are: style, class, id
When used together with CSS, the `<div>` element can used to style blocks of content:

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in
the United Kingdom, with a metropolitan area of over 13 million
inhabitants.</p>
</div>
```

The `<span>` Element: Used to mark up a part of a text, or part of a document. `<span>` element has no required attribute...style,class and id are common ones. When used with css, the span element can be used to style parts of a text.

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes
and my father has <span style="color:darkolivegreen;font-weight:bold">dark
green</span> eyes.</p>
```

** Chapter Summary There are two display values: block and inline A block-level element always starts on a new line and takes up the full width available An inline element does not start on a new line and it only takes up as much width as necessary The `<div>` element is a block-level and is often used as a container for

other HTML elements The `<span>` element is an inline container used to mark up a part of a text, or a part of a document
** HTML Tags summarized

```
Tag       Description
<div>     Defines a section in a document (block-level)
<span>    Defines a section in a document (inline)
return
```

HTML CLASSES

** html class attribute is used to specify a class for an html element.

- o Multiple html elements can also share the same class.
- o Class attribute is used to point to a class to manipulate together in a style sheet. Javascript can also manipulate elements in the same class together.

** In the example below, all three div elements with class city are style equally together equally according to the .city style definition in the header.

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
```

```
    <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```
** Here we can also style note class in `<span>` elements with .note style definition
```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```
NB

Tip: The class attribute can be used on any HTML element.

Note: The class name is case sensitive!

   o   The Syntax for Class.

To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>
```

```
<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```

- o Multiple classes.

HTML elements can belong to multiple classes. Define multiple classes by separating the class names with a space. The element will be styled according to all classes specified.

Below, `<h2>` elements belongs to both city class and also main class(`<div class="city main">`)
Below, h2 element belongs to both city class and also to the main class and will get CSS styles from both of the classes

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

- o The full html

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}

.main {
  text-align: center;
}
</style>
</head>
<body>

<h2>Multiple Classes</h2>
<p>Here, all three h2 elements belongs to the "city" class. In addition,
London also belongs to the "main" class, which center-aligns the text.</p>

<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>

</body>
</html>
```

Different HTML elements like `<p>` and `<h2>` points to city class ans share same style

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

** Use ofthe class Attribute in Javascript Class name can also be used by JS to perform certain tasks for specific elements. JS usually access elemetns with a specific class name with the `getElementByClassName()` method:

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
return
```

HTML ID

HTML ID ATTRIBUTE: Used to specify a unique id for an HTML element.

Only one element has a particular id.

The id attribute is used to point to a specific style declaration in a style sheet. Also used by JS to access and manipulate elements with the specific id.

id syntax: write hash `(#)` followed by an id name. Then define the CSS properties with curly braces {}. in the sheet below, `<h1>` element is styled according to the HashmyHeader style definition in the header section

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
```

```
</html>
```
** The id name is case sensitive! And it must contain at least one character, and must not contain whitespaces (spaces, tabs, etc.).

Diff between Class and ID: Class can be used by multiple HTML elements while id name must only be used by one HTML element within the page.

** Below we illlustrate the difference between Class and ID

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Difference Between Class and ID</h2>
<p>A class name can be used by multiple HTML elements, while an id name must
only be used by one HTML element within the page:</p>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```
** HTML Bookmarks with ID and Links.

Html bookmarks allows readers to jump to a specific part of a webpage, useful for very long pages.

To use a bookmark, create it and then add a link to it. Once link is clicked, page scrolls to the location with the bookmark.

** To create a bookmark,

```
<h2 id="C4">Chapter 4</h2>
```

  o   To add a link to it:

```
<a href="#C4">Jump to Chapter 4</a>
```

  o   To add a link to the bookmark (Jump to Chapter 4) from another page.

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```
** See the demonstration below:

```
<!DOCTYPE html>
<html>
<body>

<p><a href="#C4">Jump to Chapter 4</a></p>
<p><a href="#C10">Jump to Chapter 10</a></p>

<h2>Chapter 1</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>
<p>This chapter explains ba bla bla</p>

<h2 id="C4">Chapter 4</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>
```

```
<p>This chapter explains ba bla bla</p>

<h2 id="C10">Chapter 10</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 18</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 19</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 20</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 21</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 22</h2>
<p>This chapter explains ba bla bla</p>

<h2>Chapter 23</h2>
<p>This chapter explains ba bla bla</p>

</body>
</html>
```

o   Using the id Attribute in JavaScript

The id attribute can also be used by JavaScript to perform some tasks for that specific element.

JavaScript can access an element with a specific id with
the `getElementById()` method:

```
<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

- o Chapter Summary:: Chapter Summary The id attribute is used to specify a unique id for an HTML element The value of the id attribute must be unique within the HTML document The id attribute is used by CSS and JavaScript to style/select a specific element The value of the id attribute is case sensitive The id attribute is also used to create HTML bookmarks JavaScript can access an element with a specific id with the getElementById() method

```
return
```

HTML Iframes

It is used to display a webpage within another webpage

- o HTML Iframe syntax

HTML Iframe Syntax:

`<iframe>` tag specifies an inline frame. This inline frame is used to embed another document within the current html document
** Syntax example below

```
<iframe src="url" title="description">
```
NB: It is a good practice to always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.

** Use height and width to specify the size ofthe iframe

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe
Example"></iframe>
```
** You can also add style attribute and use css height and width properties:

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;" title="Iframe
Example"></iframe>
```

** Iframe-Remove the border

- o To remove border add style attribute and use the CSS border property.

```
<iframe src="demo_iframe.htm" style="border:none;" title="Iframe
Example"></iframe>
```

- o You can also use css style to change the size and color of the iframe's border:

```
<iframe src="demo_iframe.htm" style="border:2px solid red;" title="Iframe
Example"></iframe>
```
Iframe - Target for a Link

An iframe can be used as the target frame for a link. The target attribute of the link must refer to the name attribute ofthe iframe

```
<iframe src="demo_iframe.htm" name="iframe_a" title="Iframe Example"></iframe>

<p><a href="https://www. s.com" target="iframe_a"> s.com</a></p>
```
** When the target attribute of a link matches the name of an iframe, the link will open in the iframe

In essence, the target of a url can be a new (_blank) page and can also be the name (attribute) of an iframe

```
return
```

HTML JavaScript

Javascript inclusion in HTML makes it more interactive and dynamic.

** The below gives a button that shows date and time.

```
<!DOCTYPE html>
<html>
<body>

<h1>My First JavaScript</h1>

<button type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>
```

```
</body>
</html>
```

- HTML Script tag.

It defines client-side JavaScript…the the <script> element contains script statement or points to an external script through an src attribute. JS is used for image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the `document.getElementById()` method.
This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

```
<h2>Use JavaScript to Change Text</h2>
<p>This example writes "Hello JavaScript!" into an HTML element with
id="demo":</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```
** Use JavaScript to change the style of an HTML document

```
<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function myFunction() {
  document.getElementById("demo").style.fontSize = "25px";
  document.getElementById("demo").style.color = "red";
  document.getElementById("demo").style.backgroundColor = "yellow";
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>
```
** Use Javascript to change the src attribute of an
image. `document.getElementById("image").src = "picture.gif";`

- see full code below

```
<h1>My First JavaScript</h1>
<p>Here, a JavaScript changes the value of the src (source) attribute of an
image.</p>

<script>
function light(sw) {
```

```
  var pic;
  if (sw == 0) {
    pic = "pic_bulboff.gif"
  } else {
    pic = "pic_bulbon.gif"
  }
  document.getElementById('myImage').src = pic;
}
</script>

<img id="myImage" src="pic_bulboff.gif" width="100" height="180">

<p>
<button type="button" onclick="light(1)">Light On</button>
<button type="button" onclick="light(0)">Light Off</button>
</p>
```

- o HTML `<noscript>` tag

Defines alternate content to be displayed to users that have disabled scripts in their browsers or have browsers with no support for scripts

- o This is the script

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
<noscript>Sorry, your browser does not support JavaScript!</noscript>

**The full script is:
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

<noscript>Sorry, your browser does not support JavaScript!</noscript>

<p>A browser without support for JavaScript will show the text written inside
the noscript element.</p>
return
```

HTML FILE PATH

A file path describes the location of a file in a website's folder structure.

Examples of filepath

```
Path       Description
<img src="picture.jpg">    The "picture.jpg" file is located in the same
folder as the current page
<img src="images/picture.jpg">     The "picture.jpg" file is located in the
images folder in the current folder
<img src="/images/picture.jpg">     The "picture.jpg" file is located in the
images folder at the root of the current web
<img src="../picture.jpg">          The "picture.jpg" file is located in the
folder one level up from the current folder
Path       Description
<img src="picture.jpg">    The "picture.jpg" file is located in the same
folder as the current page
<img src="images/picture.jpg">     The "picture.jpg" file is located in the
images folder in the current folder
<img src="/images/picture.jpg">     The "picture.jpg" file is located in the
images folder at the root of the current web
<img src="../picture.jpg">          The "picture.jpg" file is located in the
folder one level up from the current folder
```

- A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like:

Web pages Images Style sheets JavaScripts

Absolute file paths: the full url to a file

```
 <img src="https://www. s.com/images/picture.jpg" alt="Mountain">
```
Relative file paths: Points to a file relative to the current page.

- File path pointing to a file in the images folder located at the root of the current web.

```
 <img src="/images/picture.jpg" alt="Mountain">
```

- File path pointing to a file in the images folder located in the current folder:

```
<img src="images/picture.jpg" alt="Mountain">
```

- File paths pointing to a file in the images folder located in the folder one level up from the current folder:

```
<img src="../images/picture.jpg" alt="Mountain">
```
Advice:

Best Practice

It is best practice to use relative file paths (if possible).

When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.

`return`

HTML HEAD

- o   HTML HEAD- The Head Element

`<head>` element contains the `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, `<base>` elements.
`<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.
HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

`<title>` element:
Defines document title, it must contain only text and it is shown in the browser title bar or in page's tag. Title element is required in html documents. The content ofthis page title helps for Search Engine Optimization of a webpage which is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element: defines a title in the browser toolbar provides a title for the page when it is added to favorites displays a title for the page in search engine-results
So, try to make the title as accurate and meaningful as possible

The HTML `<style>` element
This element is used to define style information for a single HTML page.

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```
The HTML `<link>` Element.

Defines relationship between current document and an external resource. Link tag is most used to link to external style sheets:

```
<link rel="stylesheet" href="mystyle.css">
```
** The link element in action

```
<head>
  <title>Page Title</title>
  <link rel="stylesheet" href="mystyle.css">
</head>
```
The HTML `<meta>` Element

Typically used to specify the character set, page description, kyewords, author of document, and viewport settings.

The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

** Examples

Examples

Define the character set used:

```
<meta charset="UTF-8">
```
Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```
Define a description of your web page:

```
<meta name="description" content="Free Web tutorials">
Define the author of a page:
<meta name="author" content="John Doe">
```

```
Refresh document every 30 seconds:
```

```
<meta http-equiv="refresh" content="30">
```

```
Setting the viewport to make your website look good on all devices:
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
Example of <meta> tags:
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
```

```
<meta name="author" content="John Doe">
```

- o Setting The Viewport Viewport if user's visible area of a webpage. Varies with the device and is smaller on a mobile phone than PC screen.

You should include the following element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```
This gives the browser instructions on how to control the page's dimensions and scaling.

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

- o HTML `<script>` Element.

The `<script>` element is used to define client-side JavaScript.

- o The script below writes "Hello JavaScript" into an HTML element with id="demo"

```
 <script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```
The HTML `<base>` Element
`<base>` element specifies the base URL and/or target for all relative URLs in a page.
The `<base>` tag must have either an href or a target attribute present, or both.
There can only be one single element in a document!

** Specify a default URL and a default target for all links on a page:

```
<head>
<base href="https://www. s.com/" target="_blank">
</head>

<body>
<img src="images/stickman.gif" width="24" height="39" alt="Stickman">
<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```
** Chapter Summary

```
Chapter Summary

   The <head> element is a container for metadata (data about data)
   The <head> element is placed between the <html> tag and the <body> tag
   The <title> element is required and it defines the title of the document
   The <style> element is used to define style information for a single
document
   The <link> tag is most often used to link to external style sheets
   The <meta> element is typically used to specify the character set, page
description, keywords, author of the document, and viewport settings
   The <script> element is used to define client-side JavaScripts
   The <base> element specifies the base URL and/or target for all relative
URLs in a page
```
** HTML Head Elements

```
Tag       Description
<head>    Defines information about the document
<title>   Defines the title of a document
<base>    Defines a default address or a default target for all links on a
page
<link>    Defines the relationship between a document and an external resource
<meta>    Defines metadata about an HTML document
<script>  Defines a client-side script
<style>   Defines style information for a document
return
```

HTML LAYOUT

HTML LAYOUT--Layout Elements and Techniques

HTML has semantic elements

```
<header> - Defines a header for a document or a section
<nav> - Defines a set of navigation links
<section> - Defines a section in a document
<article> - Defines an independent, self-contained content
<aside> - Defines content aside from the content (like a sidebar)
<footer> - Defines a footer for a document or a section
<etails> - Defines additional details that the user can open and close on
demand
<summary> - Defines a heading for the <details> element
```
Techniques for creating HTML layouts CSS framework CSS float property CSS
flexbox CSS grid

  o   CSS Frameworks: Enable you to fastly create your layout: and Boostrap
      are examples

CSS Float layout: Use CSS float property to do web layout. You just need to remember float and clear properties to use this. Its main disadvantage is the it is tied to the document flow which may harm flexibility.

CSS Flexbox layout::Ensures that elements behave predictably when multiple sizes of user screen must display site well.

CSS Gridlayout: Offers grid-based layout system which facilitate designing webpages without floats and positioning and using rows and columns to achieve this.

```
return
```

HTML RESPONSIVE

HTML RESPONSIVE--Responsive Web Design;

Responsive web looks good irrespecive of the device and it automatically adjust for different sizes and viewports.

Here, HTML and CSS automatically resize, hide, shrink, or enlarge a site to make it look good on all devices (desktops, tablets, and phones)

To create a responsive site, add the following tag to all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- o Responsive Images: Scale nicely to fit any browser size. To make image responsive set CSS width property to 100 % and image will become responsive according to the needs. See code below for a responsive image.

```
<img src="img_girl.jpg" style="width:100%;">
```
To avoid image scaling up beyond original size, use max-width property instead of the width:100%. For exaMple:

```
<img src="img_girl.jpg" style="max-width:100%;height:auto;">
```
** To show different images depending on browser width, use

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
```

```
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  <img src="img_smallflower.jpg" alt="Flowers">
</picture>
```
** Responsive Text Size.

The text size can be set with "vw" unit, which means the "viewport width" to enable text size to follow the size ofthe browser window

- o the code below enables responsive text size

```
<h1 style="font-size:10vw">Hello World</h1>
```

- o Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

- o Media queries: Enables you to define completely different styles for different browser sizes.

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller
*/
  }
}
</style>
```
** Example of a full responsive web

```
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www. s.com/w3css/4/  ">
<body>

<div class="w3-container w3-green">
  <h1> s Demo</h1>
  <p>Resize this responsive page!</p>
</div>
```

```
<div class="w3-row-padding">
  <div class="w3-third">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
    with a metropolitan area of over 13 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Paris</h2>
    <p>Paris is the capital of France.</p>
    <p>The Paris area is one of the largest population centers in Europe,
    with more than 12 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan.</p>
    <p>It is the center of the Greater Tokyo Area,
    and the most populous metropolitan area in the world.</p>
  </div>
</div>

</body>
</html>
```

** Bootstrap: another framework that uses HTML, CSS, and JQuery to make web pages:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
  <div class="jumbotron">
    <h1>My First Bootstrap Page</h1>
  </div>
  <div class="row">
    <div class="col-sm-4">
      ...
    </div>
    <div class="col-sm-4">
```

```
        ...
    </div>
    <div class="col-sm-4">
    ...
        </div>
    </div>
</div>

</body>
</html>
return
```

HTML COMPUTERCODE

HTML Computer Code Elements

These are the different html elements for defining use input and computer code.

- o Code element

```
<code>
x = 5;
y = 6;
z = x + y;
</code>
```

- o HTML `<kbd>` For Keyboard Input

`<kbd>` element defines keyboard input... and the content inside this tag is displayed inside the browser's default monospace font.
```
<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>
```
The code above shows:Save the document by pressing Ctrl + S

HTML `<samp>` For Program Output Samp element can be used to define sample output from a computer program. The content is displayed inside browser's default monospace font.

- o To define some text as sample output from a computer program:

```
<p>Message from my computer:</p>
<p><samp>File not found.<br>Press F1 to continue</samp></p>
```
HTML `<code>` For Computer Code.
This element defines a piece of computer code. Content is displayed inside the browser's default monospace font.

```
<h2>The code Element</h2>
<p>Programming code example:</p>

<code>
x = 5;
y = 6;
z = x + y;
</code>
```

     o   To preserve the extra white spaces in a `<code>` element, use a `<pre>` element to enclose the `<code>` element.

```
<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>
```

HTML `<var>` For Variables It is used to define variable in programming or in a mathematical expression. The content inside is typically displayed in italics.

```
<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>, where
<var>b</var> is the base, and <var>h</var> is the vertical height.</p>
```

The result is:

The area of a triangle is: `1/2 x b x h, where b is the base, and h is the vertical height.`

     o   Chapter Summary

```
  The <kbd> element defines keyboard input
  The <samp> element defines sample output from a computer program
  The <code> element defines a piece of computer code
  The <var> element defines a variable in programming or in a mathematical
expression
  The <pre> element defines preformatted text
return
```

HTML SEMANTICS

HTML semantic Elements

Semantic elements are the elements that have a meaning and this meaning describes the content to both browser and the developer.

Non-semantic element: `<div>` and `<span>` tells nothing about the content
Semantic elements: `<form>`, `<table>`, and `<article>` defines content Sites usually
define elements:`<div id="nav">` `<div class="header">` `<div id="footer">` to
indicate navigation, header, and footer.

- o   emantic Elements in HTML

```
<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<main>
<mark>
<nav>
<section>
<summary>
<time>
```

HTML `<section>` Element: A Thematic grouping of content typically with a
heading. You can split a webpage into sections for introduction, content, and
contact information.
```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an international organization
working on issues regarding the conservation, research and restoration of the
environment, formerly named the World Wildlife Fund. WWF was founded in
1961.</p>
</section>

<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF
originated from a panda named Chi Chi that was transferred from the Beijing
Zoo to the London Zoo in the same year of the establishment of WWF.</p>
</section>
```
HTML `<article>` Element
The `<article>` element specifies independent self-contained content. Article
should make sense and should be distributable independently from the rest of
the site. Article element can be used in: Forum post, Blog post, Newspaper
Article.

- o   Example of the usage of article elements

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!</p>
```

```
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox
has been the second most popular web browser since January, 2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
</article>
```

- o Use of CSS to style Article elements:

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<article class="all-browsers">
  <h1>Most Popular Browsers</h1>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January, 2018.</p>
  </article>
```

```
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in
2015. Microsoft Edge replaced Internet Explorer.</p>
  </article>
</article>

</body>
</html>
```

Nesting `<article>` in `<section>` or Vice Versa.

The `<article>` element specifies independent self-contained content while eleemnt defines a section in a document.

You will find HTML pages with `<section>` elements containing `<article>` element and `<article>` elements containing `<section>`

HTML `<header>` Element

`<header>`element rep a containere for introductory content or a set of navigational links and typically contains one or more `<h1>` -`<h6>`, logo or icon, authorship information etc.

- o Note: You can have several `<header>` elements in one HTML document. However, `<header>` cannot be placed within a `<footer>`, `<address>` or another `<header>` element.

Example of header:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

- o HTML `<footer>` Element

This element typically contains authorship information copyright information contact information sitemap back to top links related documents

- o One document can have several `<footer>` elements in one document

- o Examples of footer section in a document

```
<footer>
  <p>Author: Hege Refsnes</p>
```

```
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```

HTML `<nav>` Element:Defines a set of navigation links.

`<nav>` element define a set of navigation links which are the major blocks of navigation links. So you should not use

for all links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

** A set 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>
```

HTML `<aside>` Element Defines some contents aside from the content it is placed in like sidebar. The

contentshould be indirectly related to the surrounding content.

```
<p>My family and I visited The Epcot center this summer. The weather was nice,
and Epcot was amazing! I had a great summer together with my family!</p>

<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting
attractions, international pavilions, award-winning fireworks and seasonal
special events.</p>
</aside>
```

   o   Using CSS to style the aside element.

```
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>

<p>My family and I visited The Epcot center this summer. The weather was nice,
and Epcot was amazing! I had a great summer together with my family!</p>
```

```
<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring
exciting attractions, international pavilions, award-winning fireworks and
seasonal special events.</p>
</aside>

<p>My family and I visited The Epcot center this summer. The weather was nice,
and Epcot was amazing! I had a great summer together with my family!</p>
<p>My family and I visited The Epcot center this summer. The weather was nice,
and Epcot was amazing! I had a great summer together with my family!</p>

</body>
</html>
```

HTML `<figure>` and `<figcaption>` Elements

The `<figure>` tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The `<figcaption>` tag defines a caption for a `<figure>` element.

The `<figcaption>` element can be placed as the first or as the last child of a `<figure>` element.

The `<img>` element defines the actual image/illustration.

- o   Sample code for figure and figure caption

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>
</figure>
```

- o   Essence of Semantic Elements According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

Semantic Elements in HTML

Below is a list of some of the semantic elements in HTML.

```
Tag        Description
<article> Defines independent, self-contained content
<aside>    Defines content aside from the page content
<details> Defines additional details that the user can view or hide
<figcaption>       Defines a caption for a <figure> element
<figure>  Specifies self-contained content, like illustrations, diagrams,
photos, code listings, etc.
<footer>  Defines a footer for a document or section
<header>  Specifies a header for a document or section
<main>    Specifies the main content of a document
<mark>    Defines marked/highlighted text
<nav>     Defines navigation links
```

```
<section> Defines a section in a document
<summary> Defines a visible heading for a <details> element
<time>    Defines a date/time
return
```

HTML STYLE GUIDE

HTML Style Guide and Coding Conventions

- o   always declare doctype

```
<!DOCTYPE html>

* Use lower case elements
 <body>
<p>This is a paragraph.</p>
</body>

* Close all HTML elements

 <section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```

- o   Use lowercase attribute Name

```
<a href="https://www. s.com/html/">Visit our HTML tutorial</a>
```

- o   Always Quote Attribute Values

```
<table class="striped">
```

- o   Always specify alt attribute for images

```
<img src="html5.gif" alt="HTML5" style="width:128px;height:128px">
```

- o   Try not to include spaces around equal sign.

```
<link rel="stylesheet" href="styles.css">
```

- o   Avoid too long code lines. So scrolling right and left can be avoided.

- Avoid blank lines, spaces, or indentations without a reason. Make code readable by adding blank lines to separate large of logical code blocks. Add two spaces of indentation for readability.

- This is good

```
<body>

<h1>Famous Cities</h1>

<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.
It is the seat of the Japanese government and the Imperial Palace,
and the home of the Japanese Imperial Family.</p>

</body>
```
** And example of a good table:

```
<table>
  <tr>
    <th>Name</th>
    <th>Description</th>
  </tr>
  <tr>
    <td>A</td>
    <td>Description of A</td>
  </tr>
  <tr>
    <td>B</td>
    <td>Description of B</td>
  </tr>
</table>
```
*An example of a good list:

```
<ul>
  <li>London</li>
  <li>Paris</li>
  <li>Tokyo</li>
</ul>
```

- Never Skip the `<title>` Element

The content of a page title is important for SEO and is used to decide order when listing page results by search engines.

`<title>` elements: define a title in the browser toolbar, provide title for the page when it is added to favorites, display title for the page in search-engine result.

- Make sure title is accurate is as meaningful as possible

```
<title>HTML Style Guide and Coding Conventions</title>
```

   o   Always add `<html>` and `<body>` tags

Always add the html and body tags although the html will validate without them

   o   Omitting `<head>`

The `<head>` tag can also be omitted. Browsers add all elements before `<body>` to a default element. However, it is recommended that you use it.

   o   Close Empty HTML Elements

You have the option to close empty elements or leave it open.. in HTML.

```
<meta charset="utf-8">
```
However, for XML/XHTML, you have to close it

```
 <meta charset="utf-8" />
```

   o   Add the lang attribute

Always add the lang attribute inside your `<html>` tag, to declare the language of the Web page. This helps search engines and browsers.
for example:

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <title>Page Title</title>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

   o   Meta Data

Always specify both language and character encoding `<meta charset="charset">` should be defined as early as possible in an HTML document:
```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <meta charset="UTF-8">
```

```
  <title>Page Title</title>
</head>
```

- o Setting the Viewport

Viewport is the user's visible area of a webpage and varies with screen size of the user's device.

- o include the following element in all y our web pages to instruct the browser on how to control the page's dimension.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```
The `width=device-width` sets the width of the page to follow the screen-width or the device which will very depending on the device.
The `initial-scale=1.0` sets initial zoom level when the page is first loaded by browser.

- o Short comments should be written on one line:

```
<!-- This is a comment -->```
```

and for comments that occupy more than one line:
```
<!--
  This is a long comment example. This is a long comment example.
  This is a long comment example. This is a long comment example.
-->
```

- o Using Style Sheets

Use simple syntax for linking to style sheets(not necessary to include the type attribute)

```
<link rel="stylesheet" href="styles.css">
```
Short CSS rules can be written compressed like this:

```
p.intro {font-family:Verdana;font-size:16em;}
```
Long CSS rules can be written over multiple lines:

```
body {
  background-color: lightgrey;
  font-family: "Arial Black", Helvetica, sans-serif;
  font-size: 16em;
  color: black;
}
```

- o Some advice on how to properly include the CSS style sheet Place the opening bracket on the same line as the selector Use one space before the opening bracket Use two spaces of indentation Use semicolon after each property-value pair, including the last Only use quotes around values if the value contains spaces Place the closing bracket on a new line, without leading spaces

- o Loading JS in HTML

```
<script src="myscript.js">
```

- o Accessing HTML Elements with Js

The use of untidy html code will result into Js errors.

The below produce different results

```
getElementById("Demo").innerHTML = "Hello";

getElementById("demo").innerHTML = "Hello";
```

- o Use Lower Case File Names

Consistently use lower case file names because some servers(like Apache, Unix) are case sensitive about file names while others (microsoft, IIS) are not. Annd when you make a switch while having a mix, your web will be brocken. Use lower case file names to avoid this problem. So that london.jpg and London.jpg issues do not arise.

- o File Extensions

HTML files should have a .html extension (.htm is allowed).

CSS files should have a .css extension.

JavaScript files should have a .js extension.

- o Differences between .htm and .html

There is no difference between them. Consistency is good, however.

- o Default filenames:

When url does not specify filenames (such as `"https:www.abc.com/"`), the server adds a default filename
like `"index.html"` and `"index.htm"`, `"default.html"`,`"default.htm"`.
You can always configure servers with more than one default filename, and you can usually set up as many default filename as you want. However, if your server is only configured with `"index.html"` as the default filename, your file must be "index.html", not default .html.
`return`

HTML ENTITIES

- o   HTML Entities Replace reserved characters in HTML with character entities.

HTML entities Greater than and less than are reserved html signs and browser might mix them up.

Character entity looks like this:

`&entity_name;`

`OR`
`&#entity_number;`

- o   to display less than < sign, write `< or <`

NB: Use entity number preferably since it has more browser support.

Advantage of using an entity name: An entity name is easy to remember. Disadvantage of using an entity name: Browsers may not support all entity names, but the support for entity numbers is good.

- o   Non-breaking Space   is a space that will not break into a new line. Two words separated by a non-breaking space will stick together, not break into a new line. Use this when breaking words may not be disruptive.

- o   examples

`§ 10`
`10 km/h`
`10 PM`
- o   Tip: The non-breaking hyphen (`‑`) is used to define a hyphen character (-) that does not break into a new line.

- o Some useful HTML Character Entities

```
Result     Description          Entity Name      Entity Number
  non-breaking space              
<          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;
```

- o Combining Diacritical Marks (Useful when writing in yoruba for example and you need the ami signs on the alphabets)

A diacritical mark is a "glyph" added to a letter.

Some diacritical marks, like grave ( ) and acute ( ) are called accents.

Diacritical marks can appear both above and below a letter, inside a letter, and between two letters.

Diacritical marks can be used in combination with alphanumeric characters to produce a character that is not present in the character set (encoding) used in the page.

```
* Mark     Character        Construct        Result
  `        a         a&#768; à
  ´        a         a&#769; á
ˆ a        a&#770; â
  ~        a         a&#771; ã
  `        O         O&#768; Ò
  ´        O         O&#769; Ó
ˆ O        O&#770; Ô
  ~        O         O&#771; Õ
return
```

HTML SYMBOLS

- o HTML Symbols

You can add symbols absent on your keyboard to html using entities.

Many mathematical symbols are present in HTML. You also have greek letters and many more entities.

NB: THIS SECTION HAS ABOUT 3 TABLES. Create HTML table in github later.

`return`

HTML EMOJIS

- o   Using Emojis in HTML

Emojis are characters from UTF-8 character set:

They look like images, or icons, but they are not. and they are letters(characters) from UTF-8 (Unicode) character set.

UTF-8 covers almost all existing characters and symbols

`return`

HTML CHARSET

HTML charset attribute: To display an HTML page correctly, a web browser must know the character set used in the page.

```
<meta charset="UTF-8">
```
if you do not specify the charset, "UTF-8" is the default in HTML.

UTF-8 Characters: You can't type them on the keyboard but they can always be displayed using numbers (entity numbers)

- o   Both `<p>` in the paragraph below are the same

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
```

```
<body>

<p>I will display A B C</p>
<p>I will display &#65; &#66; &#67;</p>

</body>
</html>
```

- A, B, C are 65, 66, and 67 respectively

To let browser recognize that you are displaying a utf-8 character, you must start with &# and end it with a semi-colon (;)

Emojis are also characters from utf-8.

- You can size emojis like other characters

```
<p style="font-size:48px">
&#128512; &#128516; &#128525; &#128151;
</p>
```

- HTML Encoding (Character sets)

A browser must know the character set to use to display it correctly.

** Excerpts from the HTML ENCODING CHAPTER From ASCII to UTF-8

ASCII was the first character encoding standard. ASCII defined 128 different characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! $ + - ( ) @ < > .

ISO-8859-1 was the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

ANSI (Windows-1252) was the original Windows character set. ANSI is identical to ISO-8859-1, except that ANSI has 32 extra characters.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

** HTML charset attribute you specify it in the tag of the HTML page.

```
<meta charset="UTF-8">
return
```

HTML URL ENCODE

HTML Uniform Resource Locators.

WEbbrowsers use URL to request pages from web servers. URL is used to address documents (or other data) on the web.

```
scheme://prefix.domain:port/path/filename
```
Explanation:

scheme - defines the type of Internet service (most common is http or https) prefix - defines a domain prefix (default for http is www) domain - defines the Internet domain name (like s.com) port - defines the port number at the host (default for http is 80) path - defines a path at the server (If omitted: the root directory of the site) filename - defines the name of a document or resource

- o Common URL Schemes

Common URL Schemes

```
The table below lists some common schemes:
Scheme     Short for         Used for
http       HyperText Transfer Protocol      Common web pages. Not encrypted
https      Secure HyperText Transfer Protocol        Secure web pages.
Encrypted
ftp        File Transfer Protocol   Downloading or uploading files
file                A file on your computer
```

- o URLs can only be sent over the internet using the ASCII character set. The URL has to be converted if it contains outside characters. non-ASCII characters are replaced by % followed by hexadecimal units. and Spaces are normally replaced by + (plus) sign.

- o Your browser will encode character-set used in your page, default character-set in HTML5 is UTF-8.

```
return
```

HTML VS XHTML

XHTML is more strict and more XML-based version of HTML.

What is XHTML? XHTML stands for EXtensible HyperText Markup Language
XHTML is a stricter, more XML-based version of HTML XHTML is HTML defined as
an XML application XHTML is supported by all major browsers

- o   Differences between HTML and XHTML in XHTML

```
<!DOCTYPE> is mandatory
The xmlns attribute in <html> is mandatory
<html>, <head>, <title>, and <body> are mandatory
Elements must always be properly nested
Elements must always be closed
Elements must always be in lowercase
Attribute names must always be in lowercase
Attribute values must always be quoted
Attribute minimization is forbidden
```

- o   XHTML - OCTYPE ....> is Mandatory

The `<html>`, `<head>`, `<title>`, and `<body>` elements must also be present. The
xmlns attribute in must specify the xml namespace for the document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Title of document</title>
</head>
<body>

  some content here  fslfksm

</body>
</html>

* XHTML Elements Must be Properly Nested
 <b><i>Some text</i></b>
don't switch the tags.
```

- o   XHTML Elements Must Always be Closed.

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

- o   XHTML Empty Elements Must Always be Closed.

```
A break: <br />
A horizontal rule: <hr />
An image: <img src="happy.gif" alt="Happy face" />
```
XHTML Elements Must be in Lowercase.

```
<body>
<p>This is a paragraph</p>
</body>
```

- o XHTML Attribure Names Must be in Lowercase.

```
<a href="https://www. s.com/html/">Visit our HTML tutorial</a>
```

- o XHTML Attribute Values Must be Quoted

```
<a href="https://www. s.com/html/">Visit our HTML tutorial</a>
```
Correct:

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
<input type="text" name="lastname" disabled="disabled" />
Wrong:
<input type="checkbox" name="vehicle" value="car" checked />
<input type="text" name="lastname" disabled />
```
W3C has XHTML Validator

```
https://validator.w3.org/nu/?doc=https%3A%2F%2Fwww.
s.com%2Fhtml%2Fhtml_validate.html
return
```

** Part 2: HTML Forms

HTML Forms

- o Filling a form that contains user name and password and that redirects the form to a specific page after servers process the input.

```
<h2>HTML Forms</h2>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page
called "/action_page.php".</p>
```

- The `<form>` Element

The HTML `<form>` element is used to create an HTML form for user input:
```
<form>

form elements

</form>
```
The form element is like a container for all kinds of input elements like: text fields, checkboxes, radio buttons, submit buttons, etc.

- The `<input>` element: is the most used form element and can be displayed on the 'type' attribute.

```
Type        Description
<input type="text">      Displays a single-line text input field
<input type="radio">     Displays a radio button (for selecting one of many
choices)
<input type="checkbox">  Displays a checkbox (for selecting zero or more of
many choices)
<input type="submit">    Displays a submit button (for submitting the form)
<input type="button">    Displays a clickable button
```

- Text Fields: `<input type="text">`

This defines a single-line input field for text input. The form itself is not visible and the number of characters is by default 20 characters.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```
The `<label>` tag defines a label for many form elements.
The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.
The for attribute of the `<label>` tag should be equal to the id attribute of the element to bind them together.

- Radio Buttons

```
<input type="radio"> defines the radio button which lets use select ONLY one
among a few choices.
```

o   see sample radio buttons below:

```
<form>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label>
</form>
```

o   Checkboxes: Enables users select zero or more among a limited no of choices

```
<input type="checkbox"> defines a checkbox
```
NB: the value attribute can indicate the default option

o   See example of form below:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

o   The Submit Button

```
<input type="submit"> defines the button for submitting form data to a form-
handler.
```
Example of a form with a submit button below:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

o   he Name Attribute for `<input>`

o   Each input field must have a name to be submitted. If the value of the name attribute is omitted, the input field value will not be sent at all.

```
return
```

HTML Form Attributes

- o   The HTML Form Attribute: A discussion of different attributes

The Action Attribute: defines action to be performed when form is submitted. Form data is usually sent to a file on server when submit button is clicked. If you omit action attribute, the action is set to the current page.

- o   To send form data to a file called action_page.php containing a server-side script that handles the form data:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o   The Target Attribute This specified where to display response received after submitting the form.

See below for the possible values of the target attribute

Value Description

```
_blank    The response is displayed in a new window or tab
_self     The response is displayed in the current window (default)
_parent   The response is displayed in the parent frame
_top      The response is displayed in the full body of the window
framename The response is displayed in a named iframe
```

- o   Example of the use of target attribute:

```
<p>When submitting this form, the result will be opened in a new browser
tab:</p>

<form action="/action_page.php" target="_blank">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o   The Method Attribute

This specifies the method attribute specifies HTTP method used when submitting form data.

- o The get method: `<form action="/action_page.php" method="get">`
- o The post method: `<form action="/action_page.php" method="post">`

- o Tip: Always use POST if the form data contains sensitive or personal information!

Notes on GET: Appends the form data to the URL, in name/value pairs NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!) The length of a URL is limited (2048 characters) Useful for form submissions where a user wants to bookmark the result GET is good for non-secure data, like query strings in Google

Notes on POST: Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL) POST has no size limitations, and can be used to send large amounts of data. Form submissions with POST cannot be bookmarked

Tip: Always use POST if the form data contains sensitive or personal information!

- o The Autocomplete Attribute Specifies whether a form should have autocomplete on or off. Autocomplete when left on enables browser automatically complete values based on previously entered values in the same browser.

`<form action="/action_page.php" autocomplete="on">`
The Novalidate Attribute This is a boolean attribute which specifies form-data (input) should not be validated when submitted.

`<form action="/action_page.php" novalidate>`

- o List of All Attributes

```
Attribute Description
accept-charset     Specifies the character encodings used for form submission
action    Specifies where to send the form-data when a form is submitted
autocomplete       Specifies whether a form should have autocomplete on or off
enctype   Specifies how the form-data should be encoded when submitting it to
the server (only for method="post")
method    Specifies the HTTP method to use when sending form-data
name      Specifies the name of the form
novalidate         Specifies that the form should not be validated when
submitted
```

```
rel        Specifies the relationship between a linked resource and the current
document
target     Specifies where to display the response that is received after
submitting the form
return
```

HTML Form Elements

HTML `<form>` Elements
The HTML element can contain one or more of the following form elements:

```
<input>
<label>
<select>
<textarea>
<button>
<fieldset>
<legend>
<datalist>
<output>
<option>
<optgroup>
```

   o   The `<input>` Element Input element is one of the most used form element.
       Can be displayed in varying ways depending on the type attribute

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname">
```

   o   The `<label>` Element

`<label>` element defines a label for several form elements. Helps screen-readers
read out the label when the user focus on the input. It also helps users click on
text next to small regions (like checkboxes and buttons) to click those small
areas.
The 'for' attribute of the`<label>` tag should equal to the 'id' attribute of the
element to bind them together.

   o   The `<select>` Element

This element defvines a drop-down list:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
```

```
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

- o The option elements defines an option that can be selected. The first item is selected by default. Add the 'selected' attribute to an option to make it pre-selected.

```
<option value="fiat" selected>Fiat</option>
```

- o Visible Values: Use size attribute to specify the number of visible values (a small scroll bar is inserted if you have more than the visible value set)

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

- o Allow Multiple Selections: 'multiple' attribute allow user to select more than one value. To use this, you have to Hold down the Ctrl (windows) / Command (Mac) button to select multiple options

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

- o The `<textarea>` Element The `<textarea>` element defines a multi-line input field (a text area): The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```
CSS can also be used to define the size of the text area.

```
<textarea name="message" style="width:200px; height:600px;">
```

```
The cat was playing in the garden.
</textarea>
```

- o The `<button>` Element Defines a clickable button.

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```
Note: Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

- o The `<fieldset>` and `<legend>` Elements

The `<fieldset>` element is used to group related data in a form.
The `<legend>` element defines a caption for the element.
```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

- o The `<datalist>` Element Specifies a list of pre-defined options for an `<input>` element (works like a kind of autocomplete which list options you can select from once you type a text) This should be usable with the normal form input element--experiment.

Users will see a drop-down list of the pre-defined options as they input data.

The list attribute of the `<input>` element, must refer to the id attribute of the element
The datalist tag is not supported in Safari prior version 12.1

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

o The `<output>` Element Rep the result of a calculation (such as one performed by a script)

```
<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range"  id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```
** HTML Form Elements

```
Tag        Description
<form>     Defines an HTML form for user input
<input>    Defines an input control
<textarea>         Defines a multiline input control (text area)
<label>    Defines a label for an <input> element
<fieldset>         Groups related elements in a form
<legend>  Defines a caption for a <fieldset> element
<select>  Defines a drop-down list
<optgroup>         Defines a group of related options in a drop-down list
<option>  Defines an option in a drop-down list
<button>  Defines a clickable button
<datalist>         Specifies a list of pre-defined options for input controls
<output>  Defines the result of a calculation
return
```

HTML Input Types

o HTML Input Types The default is 'text'

```
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
```

```
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">
<input type="url">
<input type="week">
```
Tip: The default value of the type attribute is "text".

- o Input Type Text(default width=20 characters) `<input type="text">` defines a single-line text input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

- o Input Type Password

`<input type="password">` defines a password field.The characters in a password field are masked (shown as asterisks or circles)

```
 <form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

- o input Type Submit `<input type="submit">` defines a button for submitting form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

If you omit the submit button's value attribute, the button will get a default text (typically 'Submit Query')

```
 <form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

o Input Type Reset

`<input type="reset">` defines a reset button that will reset all form values to their default values.: YOu can also add value attribute to the reset button to customize what shows up on the button.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

o Input Type Radio -Defines a radio button which lets a user select one of a limited number of choices:

```
<form>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label>
</form>
```

o Input Type Checkbox

`<input type="checkbox">` defines a checkbox.
Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

o Input Type Button `<input type="button">` defines a button:

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

o Input Type Color The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

type="color" is not supported in Internet Explorer 11 or Safari 9.1 (or earlier)

The value attribute defines a preselected value for the color.

```
<form>
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000">
  <input type="submit" value="Submit">
</form>
```

- o   Input Type Date

The `<input type="date">` is used for input fields that should contain a date. Depending on browser support, a date picker can show up in the input field.

type="date" is not supported in Safari or Internet Explorer 11 (or earlier)

```
<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
  <input type="submit" value="Submit">
</form>
```
You can also add 'min' and 'max' attributes to add restrictions to dates

```
<form action="/action_page.php">
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <input type="submit" value="Submit">
</form>
```

- o   Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.
Depending on browser support, a date picker can show up in the input field.

```
<form action="/action_page.php">
  <label for="birthdaytime">Birthday (date and time):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">
  <input type="submit" value="Submit">
</form>
```

- o   Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.
Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input

```
<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
  <input type="submit" value="Submit">
</form>
```

- o Input Type File The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

```
<form action="/action_page.php">
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o Input Type Month The `<input type="month">` allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

```
<form action="/action_page.php">
  <label for="bdaymonth">Birthday (month and year):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">
  <input type="submit" value="Submit">
</form>
```

- o Input type Number The `<input type="number">` defines a numeric input field.

You can also set restrictions on what numbers are accepted.

Typically adds up and down button to decrease and increase the values between the ranges

The following example displays a numeric input field, where you can enter a value from 1 to 5:

```
<form action="/action_page.php">
  <label for="quantity">Quantity (between 1 and 5):</label>
```

```
  <input type="number" id="quantity" name="quantity" min="1" max="5">
  <input type="submit" value="Submit">
</form>
```
Input Restrictions

Here is a list of some common input restrictions: Attribute Description checked Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio") disabled Specifies that an input field should be disabled max Specifies the maximum value for an input field maxlength Specifies the maximum number of character for an input field min Specifies the minimum value for an input field pattern Specifies a regular expression to check the input value against readonly Specifies that an input field is read only (cannot be changed) required Specifies that an input field is required (must be filled out) size Specifies the width (in characters) of an input field step Specifies the legal number intervals for an input field value Specifies the default value for an input field

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

```
<form action="/action_page.php">
  <label for="quantity">Quantity:</label>
  <input type="number" id="quantity" name="quantity" min="0" max="100"
step="10" value="30">
  <input type="submit" value="Submit">
</form>
```

- o Input Type Range The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

```
<form action="/action_page.php" method="get">
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
  <input type="submit" value="Submit">
</form>
```

- o Input Type Search

```
<form action="/action_page.php">
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
  <input type="submit" value="Submit">
</form>
```

- o Input Type Search The `<input type="tel">` is used for input fields that should contain a telephone number.

```
<form action="/action_page.php">
  <label for="phone">Enter a phone number:</label><br><br>
  <input type="tel" id="phone" name="phone" placeholder="123-45-678"
pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required><br><br>
  <small>Format: 123-45-678</small><br><br>
  <input type="submit" value="Submit">
</form>
```

- o Input Type Time

The `<input type="time">` allows the user to select a time (no time zone). Depending on browser support, a time picker can show up in the input field.If the browser supports it, a time picker pops up when entering the input field.

```
<form action="/action_page.php">
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
  <input type="submit" value="Submit">
</form>
```

- o Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address. Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

```
<form action="/action_page.php">
  <label for="homepage">Add your homepage:</label>
  <input type="url" id="homepage" name="homepage">
  <input type="submit" value="Submit">
</form>
```

- o Input Type Week The `<input type="week">` allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.If the browser supports it, a date picker pops up when entering the input field

```
<form action="/action_page.php">
  <label for="week">Select a week:</label>
  <input type="week" id="week" name="week">
  <input type="submit" value="Submit">
```

```
</form>
return
```

HTML Input Attributes

HTML Input Type Attribute Tag Description `<input type="">` Specifies the input type to display

---

HTML Input Attributes: Attributes of input element

- o The value attribute Specifies the initial(default) value for the imput field

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The readonly Attribute Specifies an input field to be read-only. This input field cannot be modified but user can tab into it, highlight, and copy text from it. The value is sent when submitting the form.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The disabled attribute Specified that an input field is disabled, and such field is unusable and unclickable. Value will not be sent when submitting the form.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" disabled><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The size Attribute

Specifies the visible width, in characters, of an input field. Default value is size 20. This attribute works with Text, search, tel, url, email, and password.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o   The maxlength Attribute

Specifies the maximum number of characters allowed in an input field.

Note: When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o   Min and Max Attribute Specifies the min and max values for an input field. Works with number, range, date, datetime-local,month, time, and week.

Tip: Use the max and min attributes together to create a range of legal values.

```
<form action="/action_page.php">
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5"><br><br>

  <input type="submit" value="Submit">
</form>
```

- o   The multiple Attribute Specifies that the user is allowed to enter more than one value in an input field. multiple attribute works well with email and file types

```
<form action="/action_page.php">
  <label for="files">Select files:</label>
  <input type="file" id="files" name="files" multiple><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The pattern attribute

Specifies regular expressions that the input field's value is checked against when form is submitted. Pattern attribute works with text, date, search, url, tel, email, and password.

Tip: Use the global title attribute to describe the pattern to help the user.

Tip: Learn more about regular expressions in our JavaScript tutorial.

```
<form action="/action_page.php">
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The Placeholder Attribute The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

```
<form action="/action_page.php">
  <label for="phone">Enter a phone number:</label>
  <input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The required Attribute

Specifies that in input must be filled out before submitting the form. It works with the following input types: text, search, url, tel, email, passwork, date pickers, number, checkbox, radio, and file.

```
<form action="/action_page.php">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>
```

```
  <input type="submit" value="Submit">
</form>
```

- o he Step Attribute

Specifies legal number intervals for an input field (e.g. step="3" means that legal numbers could be -3, 0, 3, 6, etc.)

Tip: This attribute can be used together with the max and min attributes to create a range of legal values.

The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

```
<form action="/action_page.php">
  <label for="points">Points:</label>
  <input type="number" id="points" name="points" step="3">
  <input type="submit" value="Submit">
</form>
```
Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must also be checked by the receiver (the server)!

- o The Autofocus Attribute The input autofocus attribute specifies the an input field automatically get focus when page loads.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o The height and width Attributes

The input height and width attributes specify the height and width of an `<input type="image">` element.
Tip: Always specify both the height and width attributes for images. If height and width are set, the space required for the image is reserved when the page is loaded. Without these attributes, the browser does not know the size of the image, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the images load).

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
```

```
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="image" src="img_submit.gif" alt="Submit" width="48"
height="48">
</form>
```

NB: The input `type="image"` sends the X and Y coordinates of the click that activated the image button

- o The List Attribute The input list attribute refers to a element that contains pre-defined options for an element.

```
<form action="/action_page.php">
  <input list="browsers" name="browser">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit" value="Submit">
</form>
```

- o The Autocomplete Attribute

The input autocomplete attribute specifies whether a form or an input field should have autocomplete on or off.

Autocomplete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.

The autocomplete attribute works with and the following types: text, search, url, tel, email, password, datepickers, range, and color.

In the case below, the autocomplete is on for the form but off for the email field. Fill in the form and reload to see the effect.

```
<form action="/action_page.php" autocomplete="on">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" autocomplete="off"><br><br>
  <input type="submit" value="Submit">
</form>
```

- o Use maxlength for no of characters: `maxlength="40"` And use max for numeric value, use max: `max=40`

`return`

HTML Input Form Attributes

HTML Input form* Attributes

The form attribute: The input form attribute Specifies the form element element belongs to. It's value must be equal to the id attribute of the element it belongs to.

```
<p>The form attribute specifies the form an input element belongs to.</p>

<form action="/action_page.php" id="form1">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
</form>
<p>The "Last name" field below is outside the form element, but still part of
the form.</p>

<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname" form="form1">
```

- o The formaction Attribute The input to formaction attribute specifies the URL of the file that will process the input when the form is submitted.

Note: This attribute overrides the action attribute of the `<form>` element.
The formaction attribute works with the following input types: submit and image.

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formaction="/action_page2.php" value="Submit as Admin">
</form>
```

- o The forenctype Attribute The input formenctype attribute specifies how the form-data should be encoded when submitted (only for forms with method="post").

Note: This attribute overrides the enctype attribute of the `<form>` element. The formenctype attribute works with the following input types: submit and image.

```
<form action="/action_page_binary.asp" method="post">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/form-data" value="Submit as Multipart/form-data">
</form>
```

- The formmethod Attribute

The input formmethod attribute defines the HTTP method for sending form-data to the action URL.

Note: This attribute overrides the method attribute of the `<form>` element. The formmethod attribute works with the following input types: submit and image.

The form-data can be sent as URL variables (method="get") or as an HTTP post transaction (method="post").

Notes on the "get" method:

```
This method appends the form-data to the URL in name/value pairs
This method is useful for form submissions where a user want to bookmark the result
There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)
```

Notes on the "post" method:

```
This method sends the form-data as an HTTP post transaction
Form submissions with the "post" method cannot be bookmarked
The "post" method is more robust and secure than "get", and "post" does not have size limitations
<form action="/action_page.php" method="get" target="_blank">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit using GET">
  <input type="submit" formmethod="post" value="Submit using POST">
</form>
```

- o The formtarget Attribute

The input formtarget attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

Note: This attribute overrides the target attribute of the `<form>` element.
The formtarget attribute works with the following input types: submit and image.

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formtarget="_blank" value="Submit to a new window/tab">
</form>
```

- o The formnovalidate Attribute The input formnovalidate attribute specifies that an element should not be validated when submitted.

Note: This attribute overrides the novalidate attribute of the `<form>` element.
The formnovalidate attribute works with the following input types: submit.

```
<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email" required><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formnovalidate="formnovalidate" value="Submit without
validation">
</form>
```

- o The novalidate attribute The novalidate attribute is a attribute.

When present, novalidate specifies that all of the form-data should not be validated when submitted.

```
<form action="/action_page.php" novalidate>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email" required><br><br>
  <input type="submit" value="Submit">
</form>
```
Chapter summary HTML Form and Input Elements Tag Description

Defines an HTML form for user input Defines an input control
```
return
```

PART 4: HTML GRAPHICS

HTML CANVAS

HTML cANVAS GRAPHICS

It is used to draw graphics on a webpage. You use element to create the canvas.

The HTML element is used to draw graphics, on the fly, via JavaScript.

The element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Example of the canvas markup:

```
 <canvas id="myCanvas" width="200" height="100"></canvas>
```
Note: Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Example:

```
 <canvas id="myCanvas" width="200" height="100" style="border:1px solid
#000000;">
</canvas>
```
NB: Canvas can draw line, circle, draw text, Stroke text, Draw linar gradient, draw circular gradient, draw image into an empty canvas line. Since Javascript is used to create the canvas, detailed study of canvas will be made under javascript training.

```
return
```

HTML SVG

SVG can draw circle, rectangle, rounded rectangle, svg star, svg logo.

Differences Between SVG and Canvas

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with a JavaScript).

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

Comparison of Canvas and SVG

The table below shows some important differences between Canvas and SVG:

Canvas | SVG

Resolution dependent |Resolution independent No support for event handlers | Support for event handlers Poor text rendering capabilities | Best suited for applications with large rendering areas (Google Maps) You can save the resulting image as .png or .jpg | Slow rendering if complex (anything that uses the DOM a lot will be slow) Well suited for graphic-intensive games | Not suited for game applications

return

HTML MEDIA

PART 5: HTML MEDIA

HTML Multimedia:

Multimedia on the web is sound, music, videos, movies, and animations.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

- o   Videoformats

Format File Description MPEG .mpg .mpeg MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML.

AVI .avi AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.

WMV .wmv WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.

QuickTime .mov QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.

RealVideo .rm .ram RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers.

Flash .swf .flv Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.

Ogg .ogg Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.

WebM .webm WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML.

MPEG-4 or MP4 .mp4 MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube.

Note: Only MP4, WebM, and Ogg video are supported by the HTML standard.

- Common Audio formats:Only MP3, WAV, and Ogg audio are supported by the HTML standard. MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music. If your website is about recorded music, MP3 is the choice.

Format File Description

MIDI .mid .midi MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.

RealAudio .rm .ram RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.

WMA .wma WMA (Windows Media Audio). Developed by Microsoft. Plays well on Windows computers, but not in web browsers.

AAC .aac AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.

WAV .wav WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML.

Ogg .ogg Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.

MP3 .mp3 MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.

MP4 .mp4 MP4 is a video format, but can also be used for audio. Supported by all browsers. `return`

HTML VIDEO

The `<video>` element is used to shows a video on a page.

```
<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>
```

"controls" attribute adds video controls like play, pause, and volume. It is good to include width and height attributes. Without setting these, the page may move irregularly or unsteadily as the video loads.

The `<source>` element allows specifying alternative video files that the browser may choose from. The browser by default chooses the first recognized format. The text between video tags will only be displayed if browser does not support the

- o HTML `<video>` Autoplay

- o Autoplay attribute starts video automatically.

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

The autoplay attribute does not work in mobile devices like iPad and iPhone.

- o HTML Video Formats

There are three supported video formats: MP4, WebM, and Ogg. The browser support for the different formats is: Browser MP4 WebM Ogg Edge YES YES YES Chrome YES YES YES Firefox YES YES YES Safari YES YES NO Opera YES YES YES

- o HTML Video - Media Types File Format Media Type MP4 video/mp4 WebM video/webm Ogg video/ogg

- o HTML Video - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the

This allows you to load, play, and pause videos, as well as setting duration and volume.

There are also DOM events that can notify you when a video begins to play, is paused, etc.

- See HTML video audio DOM reference for details of how to customize video in HTML

Summary HTML Video Tags Tag Description

```
<video>   Defines a video or movie
<source>  Defines multiple media resources for media elements, such as <video>
and <audio>
<track>   Defines text tracks in media players
return
```

HTML AUDIO

HTML `<audio>` element
The `<audio>` element is used to play an audio file.
```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

- HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume.

The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.
There are three supported audio format: MP3, WAV, OGG.

Opera, Firefox and Chrome support all audio formats. Edge/Internet explorer supports only MP3 while Safari supports only MP3 and WAV.

HTML Audio - Media Types File Format Media Type MP3 audio/mpeg OGG audio/ogg WAV audio/wav

- HTML Audio - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the `<audio>` element. This allows you to load, play, and pause audios, as well as set duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

For a full DOM reference, go to w3c HTML Audio/Video DOM Reference.

HTML Audio Tags Tag Description `<audio>` Defines sound content `<source>` Defines multiple media resources for media elements, such as

---

HTML Plug ins

They extend the standard functionality of the browser. Plug-ins can serve the following purposes: To run Java applets To run Microsoft ActiveX controls To display Flash movies To display maps To scan for viruses To verify a bank id

Warning !

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

The `<object>` Element
`<object>` is supported by all browsers and it defines embeded object within an HTML document. It was designed to embed plug-ins (like Java applets, PDF readers, and Flash Players) in web pages, but can also be used to include HTML in HTML:
`<object width="100%" height="500px" data="snippet.html"></object>`

- o   You can also embed an image as an object

`<object data="audi.jpeg"></object>`

- o   The `<embed>` Element It is supported in all major browsers and defines an embedded object within an HTML document. It is part of HTML5 spec even though it does not show up in the previous versions that it supported.

`<embed src="audi.jpeg">`
Note that the `<embed>` element does not have a closing tag. It can not contain alternative text.

o   The `<embed>` element can also be used to include HTML in HTML:

```
<embed width="100%" height="500px" src="snippet.html">
```
One thing you should note is that to select which to use between and , make sure you consider which is most widely supported by various browsers and your data type embeded. `return`

HTML YOUTUBE

HTML YouTube Videos

This is the easiest way to play videos in HTML because it removes the necessity of converting to different formats. YouTube can be allowed to play videos on your website to avoid the conversion problem.

o   YouTube Video Id

YouTube will display an id (like tgbNymZ7vqY), when you save (or play) a video.

You can use this id, and refer to your video in the HTML code.

o   Playing YouTube in HTML

To play your video on a web page, do the following: Upload the video to YouTube Take a note of the video id Define an <iframe> element in your web page Let the src attribute point to the video URL Use the width and height attributes to specify the dimension of the player Add any other parameters to the URL (see below)

```
<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY">
</iframe>
```

o   YouTube Autoplay + Mute

You can let your video start playing automatically when a user visits the page, by adding autoplay=1 to the YouTube URL.

Note: Automatically starting a video can annoy your visitor and end up causing more harm than good!

Chrome added stricter autoplay policies in 2018. Chromium browsers do not allow autoplay in all cases. However, muted autoplay is always allowed.

Add mute=1 after autoplay=1 to let your video start playing automatically (but muted).

```
<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1">
</iframe>
 <iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">
</iframe>
<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">
</iframe>
```

   o   YouTube Controls

Add controls=0 to display controls in the video player.

Value 0: Player controls does not display.

Value 1 (default): Player controls display.

```
<iframe width="420" height="345"
src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">
</iframe>
return
```

Part 6 HTML API

HTML GEOLOCATION

HTML GEOLOCATION API

   o   HTML Geolocation API is used to locate a user position.

   o   Locate the user's Position

The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

Note: Geolocation is most accurate for devices with GPS, like smartphones.

Note: As of Chrome 50, the Geolocation API will only work on secure contexts such as HTTPS. If your site is hosted on an non-secure origin (such as HTTP) the requests to get the users location will no longer function.

The `getCurrentPosition()` method is used to return the user's position. The example below returns the latitude and longitude of the user's position(a basic geolocation without error handling):

```
<p id="demo"></p>

<script>
var x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```
code above explained.

Example explained: Check if Geolocation is supported If supported, run the `getCurrentPosition() method`. If not, display a message to the user If the `getCurrentPosition()` method is successful, it returns a coordinates object to the function specified in the parameter (showPosition)
The `showPosition()` function outputs the Latitude and Longitude
The example above is a very basic Geolocation script, with no error handling.

  o Handling Errors and Rejections

The second parameter of the getCurrentPosition() method is used to handle errors. It specifies a function to run if it fails to get the user's location:

```
<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>
var x = document.getElementById("demo");
```

```
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition, showError);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}

function showError(error) {
  switch(error.code) {
    case error.PERMISSION_DENIED:
      x.innerHTML = "User denied the request for Geolocation."
      break;
    case error.POSITION_UNAVAILABLE:
      x.innerHTML = "Location information is unavailable."
      break;
    case error.TIMEOUT:
      x.innerHTML = "The request to get user location timed out."
      break;
    case error.UNKNOWN_ERROR:
      x.innerHTML = "An unknown error occurred."
      break;
  }
}
</script>
```

- o Displaying the Result in a Map

To display the result in a map, you need access to a map service, like Google Maps.

In the example below, the returned latitude and longitude is used to show the location in a Google Map (using a static image):

```
function showPosition(position) {
  var latlon = position.coords.latitude + "," + position.coords.longitude;

  var img_url = "https://maps.googleapis.com/maps/api/staticmap?center=
  "+latlon+"&zoom=14&size=400x300&sensor=false&key=YOUR_KEY";

  document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";
}
```

- o Location-specific Information

This page has demonstrated how to show a user's position on a map.

Geolocation is also very useful for location-specific information, like:

Up-to-date local information Showing Points-of-interest near the user Turn-by-turn navigation (GPS)

The `getCurrentPosition()` Method - Return Data
The `getCurrentPosition()` method returns an object on success. The latitude, longitude and accuracy properties are always returned. The other properties are returned if available: Property Returns coords.latitude The latitude as a decimal number (always returned) coords.longitude The longitude as a decimal number (always returned) coords.accuracy The accuracy of position (always returned) coords.altitude The altitude in meters above the mean sea level (returned if available) coords.altitudeAccuracy The altitude accuracy of position (returned if available) coords.heading The heading as degrees clockwise from North (returned if available) coords.speed The speed in meters per second (returned if available) timestamp The date/time of the response (returned if available) Geolocation Object - Other interesting Methods

The Geolocation object also has other interesting methods:

`watchPosition()` - Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car). `clearWatch()` - Stops the `watchPosition()` method.
The example below shows the watchPosition() method. You need an accurate GPS device to test this (like smartphone):

```
<p>Click the button to get your coordinates.</p>

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>
var x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.watchPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
    x.innerHTML="Latitude: " + position.coords.latitude +
```

```
    "<br>Longitude: " + position.coords.longitude;
}
</script>

return
```

HTML DRAG AND DROP

HTML Drag and Drop API: Can be used to make an element draggable.

HTML WEB STORAGE

HTML Web Storage API: Enables users to store data locally within user's browser. The storage limit is greater than that of cookies.the storage is per origin (per domain and protocol). Pages from one origin can store and access the same data.

HTML WEB WORKERS

HTML Web Workers API: It is a JS running in background without affecting the performance of the page. This prevents the page from becoming unresponsive until the script in the background finishes loading.

Since web workers are in external files, they do not have access to the following JavaScript objects: The window object The document object The parent object

HTML SSE

HTML Server Sent Events (SSE) API

o   Server-Sent Events - One Way Messaging

It allows a webpage to get updates from a server. An SSE is when a webpage gets updates from a server automatically.

Examples of server sent events: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

o   Receive Server-Sent Event Notifications

The EventSource object is used to receive server-sent event notifications.

You need a server that is capable of sending data updates like PHP or ASP

`return`