

INTRODUCTION

HTML Stands for Hyper Text Markup language.
HTML is a markup language for describing web documents (web pages).

HTML 5 is the next major revision of the HTML Standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1.

HTML 5 is a standard for structuring and presenting content on the www.

HTML 5 is a cooperation between the World wide web Consortium (W3C) and the Web Hypertext Application Technology Working group (WHATWG).

The new Standard incorporates feature like video playback and drag-and-drop that have been previously dependent on third - Party browser Plug-ins such as Adobe Flash, Microsoft Silverlight, and google chrome.

Browser Support

MOTIVATION

HTML 5 Support all types of Modern browser.

1) Google chrome.

2) Mozilla Firefox.

3) Apple Safari.

4) Microsoft Internet Explorer.

5) Opera mini etc.

New Features

- **Multimedia Support :** You can embed video and Audio without third-party Proprietary plug-ins or codes. Video becomes as easy as embedding an image.
- **Web Workers :** Certain web application use heavy script to perform functions. Web workers use separate background threads for processing and it does not affect the performance of a web page.
- **Canvas :** This feature allow a web developer to render graphics on the fly. As with video, there is no need for a plugin.
- **Application Caches :** web pages will start storing more and more information locally on the visitor's computer. It works like cookies, but where cookies are small, the new feature allow for much larger files. Google Gears is an excellent example of this in action.

- Geolocation : Best known for use on mobile devices.
it is based on latitude, longitude. (Map related information)
geolocation is coming with HTML5 standard.
- New Tags and Attributes

HTML 5 supports new tags and attributes like :- placeholder, required, Embed, Svg, canvas, and new input type attributes.

These attributes help in making better user interface.

placeholder attribute :- It is used to give hint to user.

required attribute :- It is used to force user to fill the field.

Embed attribute :- It is used to embed video or audio file.

Svg attribute :- It is used to draw graphics.

HTML5 Document

• A small HTML document

```
<!DOCTYPE html>
<html>
  <head>
    <title> Web Page Title </title>
  </head>
  <body>
    <h1> This is a heading </h1>
    <p> This is a paragraph </p>
  </body>
</html>
```

• Document Explained

- The DOCTYPE declaration defines the document type to be HTML.
- The text between `<HTML>` and `</HTML>` describe an HTML document.
- The text between `<head>` and `</head>` provides information about the document.
- The text between `<title>` and `</title>` provides a title for the document.
- The text between `<body>` and `</body>` describes the visible page content.
- The text between `<h1>` and `</h1>` describes a heading.

- Basic structure of HTML tags.

`<h1>` `</h1>`

Opening tag closing tag

Attribute ←

`<h1 id="main">` this is a h1 tag `</h1>`

Attribute Name ← Attribute Value →

Attribute ← Attribute Value →

entity additional information

entity additional information

- A Brief History

► IBM wanted to set a documentation system in which we can mark the title, headings, paragraphs and font type selections (in the 1980s).

They came out with a set of mark-up system called it Generalized Mark-up Language (GML).

► In 1986, International Standardizing organization (ISO) took up this concept and standardized it as Standard Generalized Mark-up Language (SGML).

> In 1989 by Tim Berners Lee and his team in the European Laboratory for Particle Physics (CERN) designed the present form of the documentation language and called it HTML.

Many versions of HTML:

VERSION

YEAR

HTML 1.0 1991

HTML 2.0 1995

HTML 3.2 1997

HTML 4.01 1999

XHTML 1.0 2000

HTML 5 2014

Tags and their use

List of HTML 5 tag.

Below you see element in

o, v, x, >

- > Elements in x are in Previous version of HTML, but are not supported in HTML 5.
- > Elements in v are new to HTML 5 and have broad browser support.
- > Elements in o are new to HTML 5, but don't yet have broad browser support.
- > Elements in > are avail in Previous version of HTML and also avail in HTML 5.

> <!-- comment --> > <bdi>

> <!DOCTYPE> > <bdo>

> <a> > <big>

> <abbr> <> > <blockquote>

x <acronym> > <body>

> <address> >

x <applet> > <button>

> <area> <> > <canvas>

✓ <article> > <caption>

✓ <aside> <> > <center>

✓ <audio> <> > <cite>

> <> > <code>

> <base> <> > <col>

x <basefont> > <colgroup>

✓ <command>	<✓ <figure>>
o <datalist>	(✗) >
> <dd>	<✓ <footer>>
> <dd>	> <form>
o <details>	x <frame> x
> <dfn>	x <frameset>
> <div>	> <h1> to <h6>
> <div>	> <head>
> <dc>	✓ <header>
> <dt>	<✓ <hgroup>
> 	> <hr>
✓ <embed>	> <html>
> <fieldset>	> <i>
✓ <figcaption>	> <iframe>

> > x <noframes>

> <input> > <noscript>

> <ins> > <object>

> <kbd> >

✓ <keygen> > <optgroup>

> <label> > <option>

"> <legend> ✓ <output>

> > <p>

> <link> > <param>

> <map> > <area>

> <mark> ✓ <progress>

> <menu> > <q>

> <meta> ✓ <np>

o <meters> ✓ <rt>

✓ <nav> ✓ <rb>

> <s> </s> > <tbody>

> <samp> </samp> > <td> </td>

> <script> </script> > <textarea>

✓ <section> </section> > <tfoot>

> <select> </select> > <th> </th>

> <small> </small> > <thead> </thead>

✓ <source> </source> ✓ <time> </time>

> > <title> </title>

x <strike> </strike> > <tr> </tr>

> ✓ <track> </track>

> <style> </style> x <tt> </tt>

> > <u> </u>

✓ <summary> >

> > <var> </var>

> <table> </table> ✓ <video> </video>

o <wbr>

~~HTML~~ Tags and their use.

- `<!-- Comment -->`

Definition and usage: The `<!-- Comment -->`

is used to insert comments in the HTML source code. Comments are not displayed in the browsers.

Comments are basically used for explaining program code.

Example :

`<!-- this is comment -->`

- `<!DOCTYPE>`

Definition and usage: The `<!DOCTYPE>`

declaration must be the very first thing in your HTML document, before the `<HTML>` tag.

The `<!DOCTYPE>` declaration is not an HTML tag it is an instruction to the web browser about what version of HTML the page is written in.

Eg.

```
<!DOCTYPE html>
<html>
  <head>
    <title> ----- </title>
  </head>
  <body>
    </body>
</html>
```

* [Link](#)

The [tag defines a hyperlink, which is used to link from one page to another.](#)

The most important attribute of the [element is the href attribute which indicates the link's destination.](#)

By default, links will appear as follows in all browsers:

- An unvisited link is underline and blue.
- A visited link is underline and purple.
- An active link is underline and red.

Attributes

- > href : it just specifies the URL of the page the link goes to.
- > hreflang : it specifies the language of the linked document.
- > media : it specifies what media/device the linked document is optimized for.
- > rel :
 - alternate, author, bookmark, help, licence
 - next, nofollow, noindex, prefetch, prev
 - search, tag.It specifies the relation between the current document and the linked document.
- > target :
 - blank, -parent, -self, -topIt specifies when to open the linked document.
- > type : specifies the Mime type of the linked doc.

Eg: ` click me `

- <form>

The form tag is used to create an HTML form for user input.

The form tag element can contain one or more of the following form elements:

- <input>

- <textarea>

- <button>

- <select>

- <option>

- <optgroup>

- <fieldset>

- <label>

eg. <form action = "demo.php" method = "get">

First name : <input type = "text" name = "fname">

Last name : <input type = "text" name = "lname">

<input type = "submit" value = "Submit">

</form>

- Attributes

> accept-charset : Specifies the character encoding that are to be used for the form submission.

> action : Specifies where to send the form-data when a form is submitted.

> autocomplete : Specifies whether a form on/off 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 : get, post

→ specifies the HTTP method to use when sending form-data.

> name → specifies the name of the form.

> id → specifies the id of the form.

> nonvalidate : specifies that the form should not be validated when submitted.

> target :

-blank, -Parent, -Self, -top

specifies where to display the response that is received after submitting the form.

- <input> : introduction & basic idea

HTML element. The <input> tag specifies an input field where the user can enter data.

e.g. <input type="text" name="fname">

- Attributes of <input>

> accept

> accept : audio, video, image

> alt : image

specifies the type of files that the server accepts (only for file input type="file").

> alt : specifies an alternate text for images (only for type="image")

> autocomplete : specifies whether an <input> element should have auto-complete enabled.

> autofocus : specifies that an <input> element should automatically get focus when the page loads.

> checked : specifies that an element should be preselected when the page loads (for type = "checkbox" or type = "radio").

> disabled : specifies that an element should be disabled.

> height : specifies the height of an element (only for type = "image")

> list : refers to a element that contains predefined option for element.

> max
number

date

Specifies the maximum value for an element.

> maxlength
number

Specifies the maximum number of characters allowed in an element.

> min : specifies a minimum value for

number, date etc.

length

Specifies a minimum value for length of an <input> element.

> multiple : specifies that a user can enter more than one value in an <input> element.

> name : specifies the name of an <input> element.

> Pattern : specifies a regular expression that an <input> element's value is checked against.

> Placeholder : specifies a short hint that describe the expected value of an <input> element.

> Readonly : specifies that an input field is read-only.

> Required : specifies that an input field must be filled out before submitting the form.

> Size : Specifies the width, in characters, of an <input> element.

> Src : Specifies the URL of the image to use as a submit button (only for type = "image").

> Step : Specifies the legal number intervals for an input field.

> Type of input :

- button
- checkbox
- color
- date
- datetime
- datetime-local
- email
- file
- hidden
- image
- month
- number
- password
- radio
- range
- reset
- search
- submit
- tel
- text
- time
- url
- week

II Specifies the type of input element to display.

> Value : specifies the value of an element.

> width : specifies the width of an element (only for type = "image").

e.g.

maxlength = "15" id = "ABC"
name = "text" autofocus = "autofocus" >

<datalist>

<option value = "india">

<option value = "Pak">

<option value = "china">

</datalist>

(And A placeholder "Placeholder" - placeholder attribute)

(input elements are single-line input fields)

Design a form

First name	<input type="text"/>
Last name	<input type="text"/>
uname	<input type="text"/>
choose Password	<input type="password"/>
Confirm password	<input type="password"/>
Country	<input type="text"/> choose <input checked="" type="checkbox"/>
Sex	<input type="radio"/> male <input type="radio"/> female
Sav game	<input type="checkbox"/> cricket <input type="checkbox"/> chess <input type="checkbox"/> hock
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

Code

```
<form name="form1" Action="A.php">
```

```
<input type="text" name="fname" /><br>
<input type="text" name="lname" /><br>
<input type="text" name="uname" /><br>
<input type="password" name="psw" /><br>
<input type="password" name="psw1" /><br>
```

Country

<select> country

<option> option 1 </option>

<option> option 2 </option>

<option> option 3 </option>

</select>

Sex: <input type="radio" name="sex"

<input value="male"> male

<input type="radio" name="sex"

<input value="female"> female

fav game: <input type="checkbox"

<input checked="" name="game" value="cricket">

cricket

<input type="checkbox" name="game"

value="chess"> chess

<input type="checkbox" name="game"

value="hockey"> hockey

<input type="submit" value="Submit">

<input type="reset" value="Reset">

</form>

Layouting a Webpage

- Layouting a webpage is done by using three methods.

1) Using Table tag

2) Using FRAMES

3) Using Div tag.

⇒ Table are used for two main purposes.

1) To store the data.

2) To ^{Layout} ~~store~~ the webpage.

⇒ TABLE TAG : <table> </table>

⇒ TABLE Row : <tr> </tr>

⇒ TABLE Cell : <td> </td>

⇒ TABLE Header : <th> </th>

⇒ TABLE Body : <tbody> </tbody>

⇒ TABLE head : <thead> </thead>

⇒ TABLE footer : <tfoot> </tfoot>

- Attributes of table tag

1) width : it controls the width of the table

2) height : it controls the height of the table.

3) border : it creates borders of the table

a) Cell Padding : gap between cell borders and its content.

5) Cellspacing : gap between two cells

6) Bgcolor : Background color of the table.

- TABLE Cell attributes

- Colspan : it controls the no. of columns a table cell span.

- Rowspan : it controls no of rows a table cellspan.

- Valign : it controls the vertical alignment of object.

- Halign : it controls the horizontal alignment of object.

Code for storing data

~~Code~~ <thead>

<tn>

(th) Month (th)

Current <th> savings </th>

L(tn) \rightarrow lim (b))

</thead>

$\langle t_{\text{foot}} \rangle$ vs. walking rate

Lens

$\langle \text{fd} \rangle \text{ sum } \langle \text{ifd} \rangle$

Ltd) + 180 L/Ltd)

1/15

L1(Esoft)

<tbody>

Ltrs

~~(td) January 21 (td)~~

$\langle \text{Ed} \rangle + 100 \quad \langle \text{IEd} \rangle$

Letters

Lens

<fd> Feb </fd>

<td> \$80 </td>

1-657

</tbody>

</table>

[Code for Creating layout]

<HTML>

<body>

<table width="780px" height="720px"
border="0px" align="center">

<tr width="780px" height="120px"

background-color="green">

<td> </td>

</tr>

<tr width="780px" height="400px"

background-color="blue">

<td> </td>

</tr>

<tr width="780px" height="250px"

background-color="red">

<td> </td>

</tr>

<tr width="780px" height="250px"

bgcolor="pink">

<td>

</td>

<td width="260px" height="250px"

bgcolor="yellow"></td>

<td width="260px" height="250px"

bgcolor="green"></td>

<td width="260px" height="250px"

bgcolor="red"></td>

</table>

</td>

</tr>

<tr width="780px" height="60px"

bgcolor="black">

<td><td>

</td>

</table>

</body>

</html>

Frames and Frameset

→ It is used to create web layout.

frameset are used to put several webpage into a single webpage.

eg. <HTML>

<frameset rows="25%",*,25%">

<frame src="">
</frame>

<frame src="">
</frame>

<frame src="">
</frame>

</frameset>

</HTML>

<HTML>

<frameset rows = "25%, *, 25%">

<frame src = "">

</frame>

<frameset cols = "25%, *, 25%">

<frame src = "">

</frame>

<frame src = "">

</frame>

<frame src = "">

</frame>

</frameset>

<frame src = "">

</frame>

</frameset>

</HTML>

- Attribute of frameset

- 1) framespacing

- 2) frameborder

- 3) border

- 4) bordercolor

- 5) rows, cols

eg: <frameset rows="25%, 50%, *"

framespacing=5 frameborder=3

border=3 bordercolor="#FF0000">

- Attribute of frame

- 1) name-size

- 2) scrolling

- 3) name

- 4) id

- 5) Target

eg. <html> <frame src="left.html" name="left">

<HTML>

<body>

Home

About

Service

</body>

</HTML>

<HTML>

<frameset cols="20%, * ">

<frame src="left.html" name="left"

id="left">

</frame>

<frame src=" " name="main"

id="main">

</frame>

</frameset>

</HTML>

- **Iframe:** This tag specifies an infinite inline frame which is used to embed another document within the current html document.

eg: `<iframe src=" " width="500px" height="500px"></iframe>`

- Attributes

> height, width.

> name

> src, srcdoc

> sandbox

• allow - forms

• allow - same - origin

• allow - scripts

• allow - top - navigation.

- **<Div>** : This tag is used to define a division or a section in an HTML document and is used to group block-elements to format them with CSS.

-Ex: In <HTML> tag first is <div> tag

and in <body> tag there is <div> tag

span class="left"

<div id="header">

</div>

<div> 09.

<div id="content"> <div id="left">

</div>

<div id="right">

</div>

</div>

<div> 12

<div id="footer">

</div> 13 go continue

<div> 14

</HTML>

- attributes : id, name, style

Multimedia Tags (viQ)

- audio
- video
- embed

- <audio> : This tag is used to play music in our web page.

e.g. <body>

<audio controls autoplay loop>

<source src="1.mp3"

type="audio/mpeg">

</audio>

</body>

- Attributes of Audio tag

- Controls : autoplay, loop

- <Video> tag This tag is used to display video by playing video in our browser with web pages.

e.g. <body>

<video width="640" height="360" controls>

<source src="2.1MP4" type="video/mp4">

</video>

</body>

Attribute of video tag

- width

- height

- Controls

- Embed > This tag is used for add external application interactive content like flash (.swf) file in our document.

Q3.

```
Embed src = "file.swf" >
```

- Attributes of embed tag
 - height
 - width
 - src

- List tags are standard collection tags.

1) ordered list (OL).

2) unorderd list (UL).

3) Definition list (DL).

- ordered list : An ordered list can be numerical or alphabetical. Use the tag to define list items and CSS to style lists.

list item (LI) is a child of list (UL).

e.g. <ol start=50>

 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

 a b c d e f g h i j k l m n o p q r s t u v w x y z

 (ordered list)

⇒ attributes of list items

• type : (button, checkbox, radio, etc.)

• type : (button, checkbox, radio, etc.)

• start

- **Unordered List** : it is used to create unordered (bulleted) list.

eg,

```
<ul> (1a) first paragraph <br>
    <li> a </li>
```

```
    <li> b </li>
```

→ handles only +2i+ which

↳ Attributes : `list-type` : none, square, circle

- **Definition List** : It is used in conjunction with `<dt>` (defines the item in the list) and `<dd>` (describes the item in the list).

eg. `<dt>`

```
<dt> Coffee </dt>
```

```
<dd> Black hot drink </dd>
```

```
<dt> Milk </dt>
```

```
<dd> white cold drink </dd>
```

`<dl>`

- **<marquee>** : It is a scrolling ~~block of text display~~ either horizontal across or vertically down in your ~~own~~ webpage.

Attributes

- width
- Height
- Direction
- behaviour
- scroll amount
- backcolor
- loop
- hspace
- vspace

e.g. **<marquee scrollamount = "5" width = "40">**

This is marquee **</marquee>**

<marquee width = "60" direction = "up"
behaviour = "alternate">

<marquee direction = "right" behaviour = "alternate">

This is a Marquee **</marquee></marquee>**

- Some HTML ~~is~~ tags.

- SVG

- Canvas

- header

- footer

- nav

- article

- **<SVG>**: SVG Stands for Scalable vector graphics, and is used to draw shapes with HTML-style mark up.

~~to~~ It offers several methods for drawing Paths, boxes, circles, text, and graphics images.

Note: SVG is not pixel-based, so it can be magnified infinitely with no loss of quality.

- Drawing a Circle using <svg>

To draw shapes with SVG, you first need to create an SVG element tag with two attributes: width and height.

```
<SVG width = "1000" height = "1000">  
</svg>
```

To create a circle, add a <circle> tag:

```
<SVG width = "2000" height = "2000">  
  <circle cx = "80" cy = "80" r = "50"  
    fill = "green" />  
</svg>
```

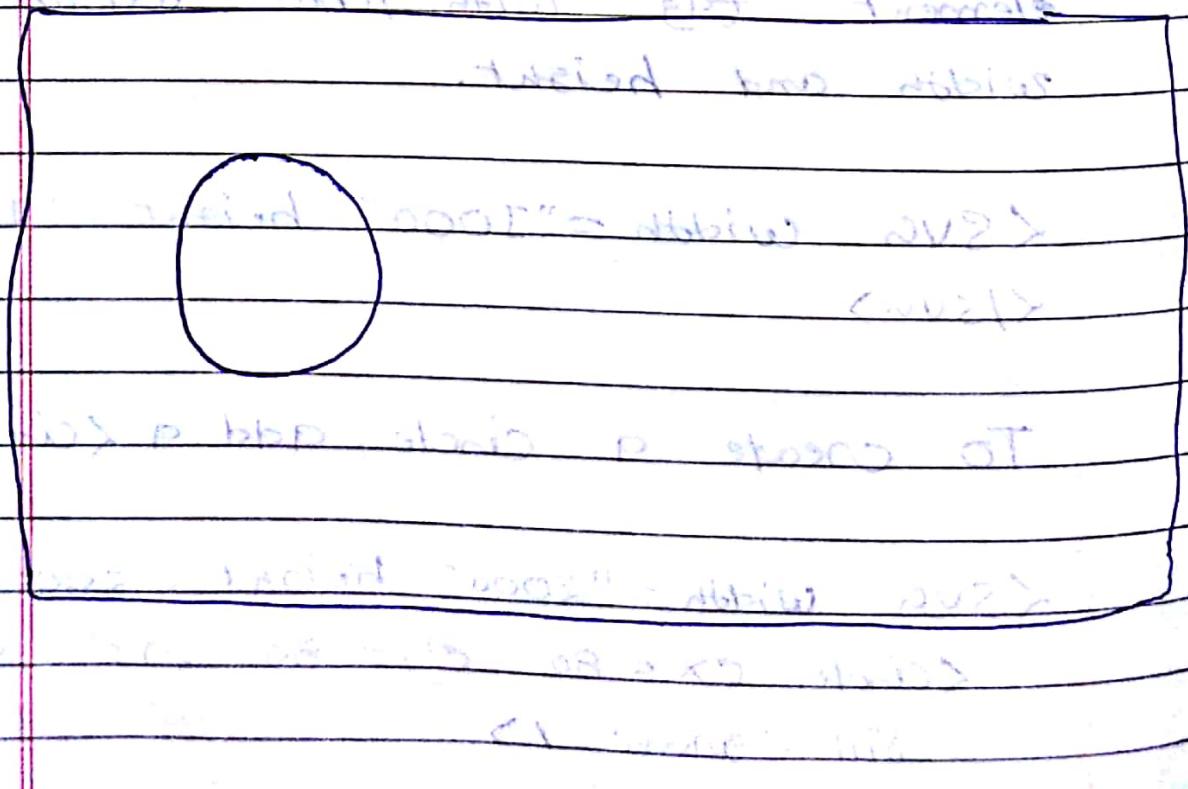
- cx Pushes the center of the circle further to the right of the screen.

- cy Pushes the center of the circle further down from the top of the screen.

- r defines the radius

- fill determines the color of our circle in red or blue.

- stroke adds an outline to the circle in black.



• define a rectangle.

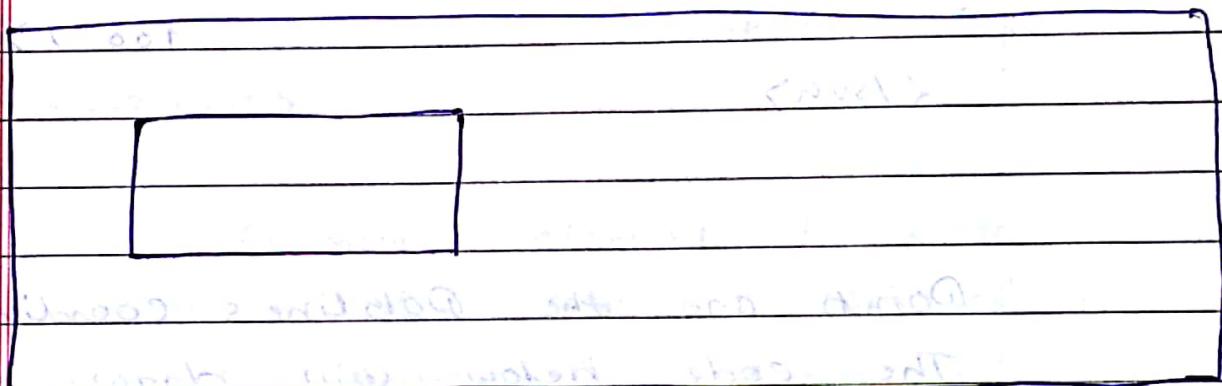
```
<svg width="2000" height="2000">
```

```
  <rect width="300" height="100">
```

```
    x="20" y="20" fill="green" />
```

```
</svg>
```

Output



• define a line segment.

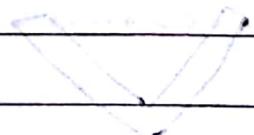
```
<svg width="400" height="410">
```

```
  <line x1="10" y1="10" x2="200" y2="100"
```

```
    style="stroke:#000000;stroke-linecap:round;
```

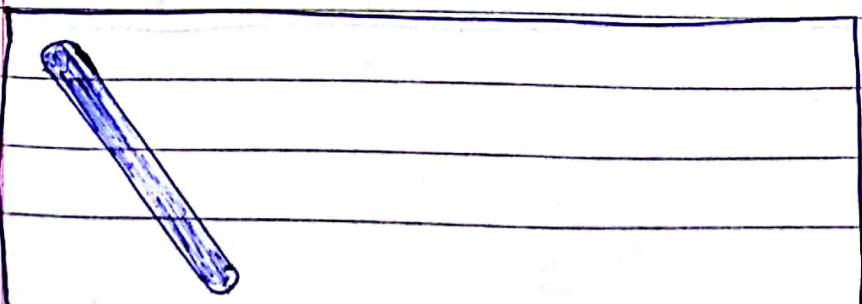
```
    stroke-width:20" />
```

```
</svg>
```



(x₁, y₁) defines the start coordinates

(x₂, y₂) defines the end coordinates.



- define shapes built from multiple line definitions:

```
<svg width="2000" height="500">
```

```
  <Polyline style="stroke-linejoin-  
  miter; stroke:black; stroke-width:12  
  fill:none;" Points="100 100, 150 150, 200  
  100" />
```

```
</svg>
```

Points are the Polyline's coordinates.

The code below will draw a black check sign.

Output



check

- <Canvas> : The HTML Canvas is an ~~HTML element~~ used to draw graphics that include everything from simple lines to complex graphics objects.

The <Canvas> element is defined by:

```
<Canvas id="Canvas1" width="200"  
height="100">  
</Canvas>
```

- ⇒ The <Canvas> element is only a container for graphics. You must use a script to actually draw the graphics (usually JavaScript).
- ⇒ The <Canvas> element must have an id attribute so it can be referred to by JavaScript.

```
<HTML> elements start here X  
<Head> </Head> script  
<body>  
<Canvas id="Canvas1"  
width="400" height="300">  
</Canvas>
```

```
<script>
var can = document.getElementById("canvas1");
var ctx = can.getContext("2d");

</script>
</body>
</HTML>
```

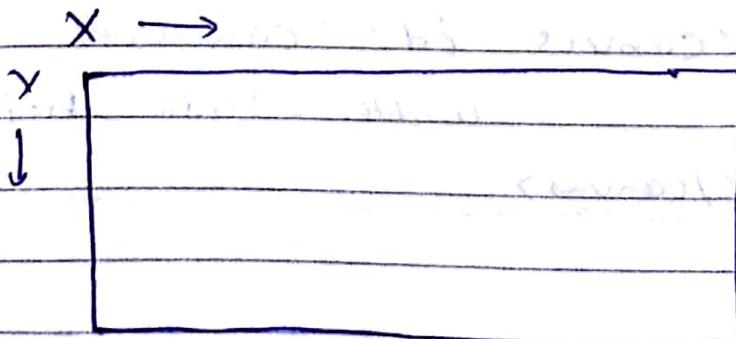
- Canvas Coordinates

The `HTMLCanvas` is a two-dimensional grid.

The upper-left corner of the canvas has the coordinates $(0,0)$.

X coordinate increases to the right.

Y coordinate increases toward the bottom of the canvas.



• Drawing Shapes

The `fillRect(x, y, w, h)` method draws a "filled" rectangle, in which `w` indicates width and `h` indicate height. The default fill color is black.

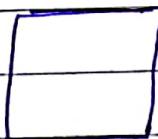
A black 100x100 pixel rectangle is drawn on the canvas at the position (20, 20),

e.g. `var c = document.getElementById("Canvas1");`

`var ctx = c.getContext("2d");`

`ctx.fillRect(20, 20, 100, 100);`

Output



Canvas

→ Good for

- Elements are drawn programmatically
- Drawing is done with pixels
- Animation are not built in
- High Performance for pixel-based drawing operations.
- Resolution dependent
- No support for event handlers
- You can save the resulting image as .png or .jpg
- Well suited for graphics-intensive games

VS

SVG.

- Elements are the part of the page's DOM (Document object model).
- Drawing is done with vectors.
- Effects, such as animations are built in.
- Based on standard XML syntax, which provides better accessibility.
- Resolution independent.
- Support for event handlers.
- Not suited for game applications.
- Best suited for application with large rendering areas (for e.g. google maps).

Layout

header

nav

article

aside

section

section

section

footer

Code

```
<HTML(>lang = "en">
  <head>
    <meta charset = "UTF-8">
  </head>

  <body>
    <header>
      <nav> _____ </nav>
    </header>

    <article>
      <aside> _____ </aside>
      <section> . . . </section>
      <section> . . . </section>
      <section> . . . </section>
    </article>

    <aside> . . . </aside>

    <footer> _____ </footer>
  </body>

</HTML>
```