Introduction to HTML5

CH # 1

- **1. Introduction to HTML5**
- 1.1. Basics of HTML5 – Introduction, features,
- form new elements, attributes and
- semantics in HTML5
- 1.2. <canvas>, <video>, <audio>.
- 1.3. Introduction to Scalable Vector Graphics
- (SVG)
- 1.4. Introduction to Version compatibility
- 1.5. Installation of Apache Tomcat
- (Xampp/Lampp/MySQL)
- Extra Reading: Geo location, Drag, Drop, Web
- Storage

- **Software application and Software product:**
- A Software that is designed according to client requirement is known as software application.
- Eg. IRCTC
- A software that is designed according to market requirement is known as software product.
- Eg. MS Office
- **Types of application and products:**
- 1. Desktop applications
- 2. Web applications
- 3. Mobile applications.

- **Web Terminology:**
- **1. Network:**
- A group of computer connected to each other for sharing information and resources.
- **2. Types of Network:**
- a) LAN b) MAN c) WAN
- **3. Internet**
- It resembles a wide area network that connects computers all over the world.
- **4. Web**
- Web is portion of internet with restricted access.

- **Web Server:**
- A web server resembles both hardware and software. It satisfies the client request by sending and receiving the data.
- Most popular web servers:
- Microsoft IIS
- Apache Tomcat
- **Web site:**
- A website is virtual directory on web server.
- It compresses of all the resources that client request.
- The resources for the website are kept in physical path(d:\vanita\images)
- The website resources are accessible using virtual path.
- (http://localhost/vanita/...)

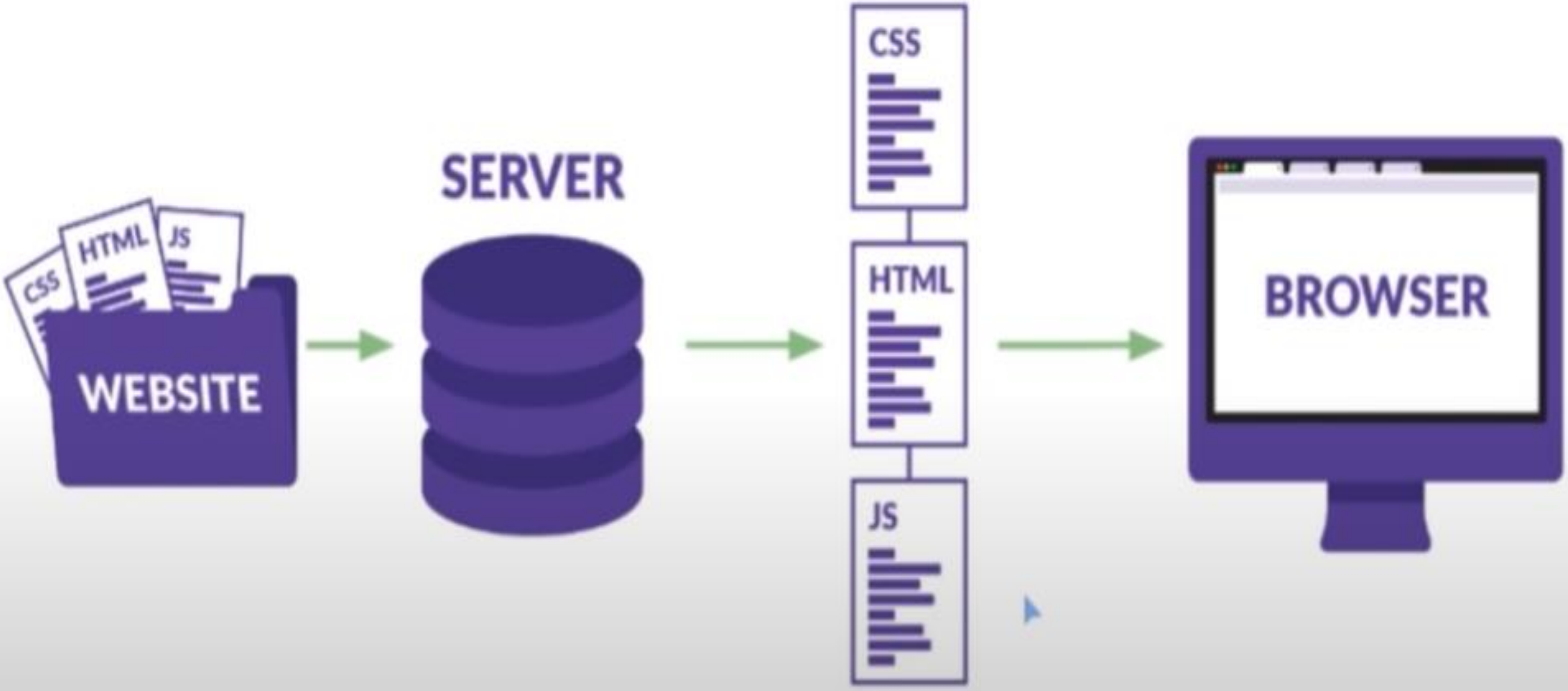
- **Web Page:**
- A web page is a hypertext document that provides an UI to interact with the resources in website.
- Types of web pages:
- a) Static page
- b)Dynamic page

HTML

- Hyper text markup language is a standard markup language for creating a web pages and web applications.
- Markup means with HTML you declare what is presented to viewer not how it is presented.
- Visual representation is defined by CSS (Cascading style sheet).
- An element usually consist of an opening tag (<element_name>) and closing tag (</element_name>).
- There are some HTML elements which don’t have a closing tag. Eg.<img>, <link> etc.

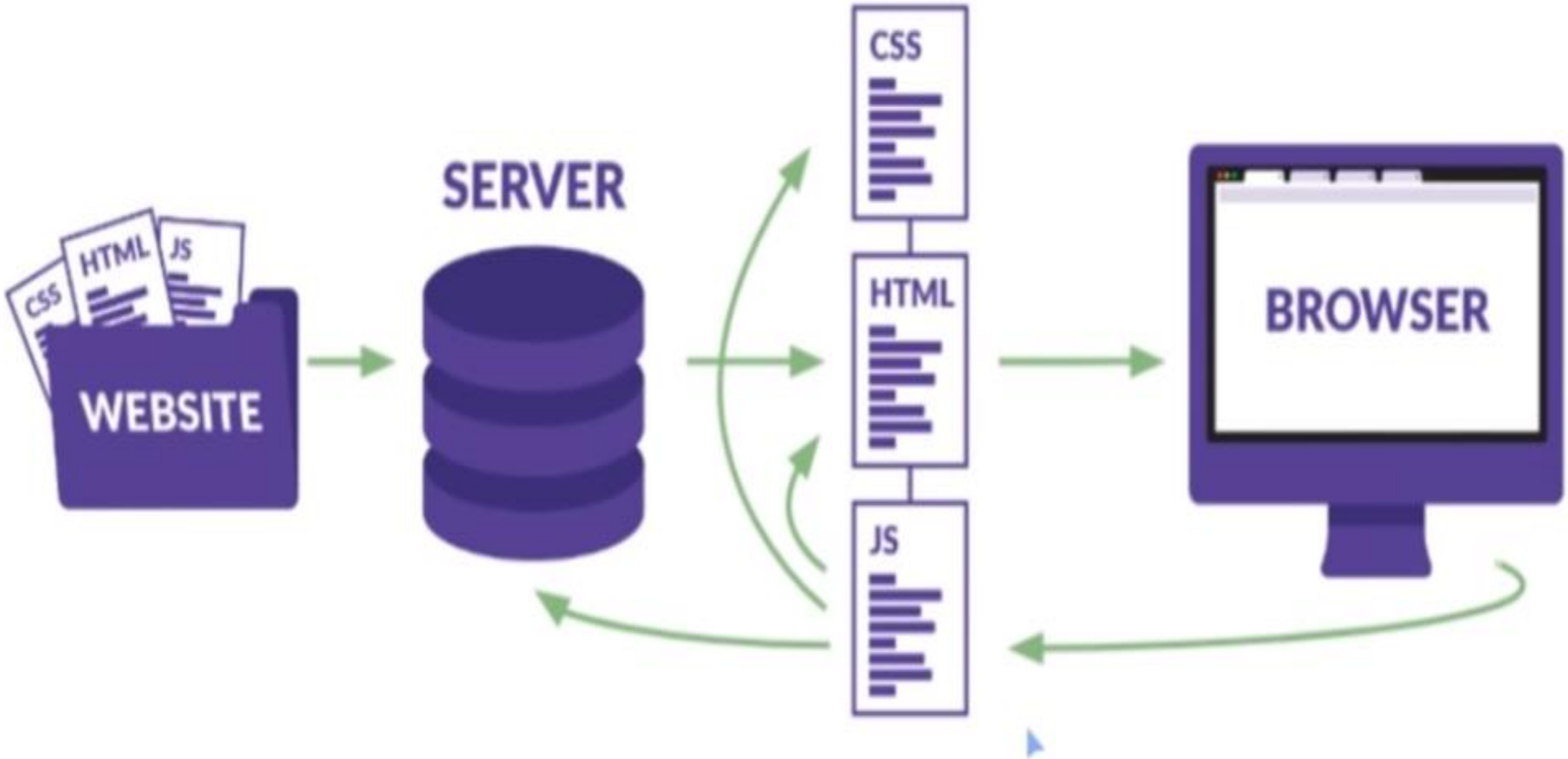
HOW THE WEB WORKS

IN ESSENCE



JAVASCRIPT

MAKES SITE DYNAMIC



- **<html> tag**
- The <html> tag is the topmost tag in HTML. It is the root of an HTML document. You should always use the lang attribute with the <html> tag. It declares the language of the webpage to the browsers and search engines.
- Example: <html lang="en">

- **<head> tag**
- The <head> tag contains information about the webpage that is not visible to the users. The head of HTML document may contain <title>,<meta>,<style>,<link>, <base>,<script> and <noscript> tags.
- **Example:**
- <head>
- <title>HTML - head tag</title>
- <meta name="author" content="john smith">
- <style>body {color: black}</style>
- <link rel="stylesheet" href="stylesheet.css">
- <base href="https://www.tutorialstonight.com">
- <script src="script.js"></script>
- <noscript>Your browser does not support javascript. </noscript>
- </head>

<body> tag

- The <body> tag contains all the visible parts of the webpage. Anything like text, image, audio, video, animations, etc all lies in the body tag.
- Example:
- <body>
- ----
- </body>

- **Heading tags**
- Heading tags in HTML are used to create headings on the webpage. There are 6 different types of heading in HTML h1 to h6.
- The headings h1 to h6 are ordered on the basis of their font size and importance. The <h1> defines most important heading and <h6> defines least important.
- Example:
- <h1>This is heading 1.</h1>
- <h2>This is heading 2.</h2>
- <h3>This is heading 3.</h3>
- <h4>This is heading 4.</h4>
- <h5>This is heading 5.</h5>
- <h6>This is heading 6.</h6>

- **Paragraph tag**
- A paragraph in webpage is defined by <p> tag. A paragraph is a block-level element used to represent text on the browser.
- Example:
- <p>This is paragraph 1.</p>
- <p>This is paragraph 2.</p>
- **comment tag**
- In HTML you can write comment using
- <!-- ... -->.
- Example:
- <!-- This is a single line comment -->

- **<br> tag (line break)**
- The <br> is used to create a line break in the text.
- Example: The br tag <br>
- **<hr> tag (horizontal line)**
- The <hr> is used to create a break between paragraphs using a horizontal line.
- Using CSS you can modify and style as per your own need.
- Example:
- <p>Scene 1:</p>
- <hr>
- <p>Scene 2:</p>

- **<b> tag:** format the text to bold.
- **<strong> tag:** used to make the content very important.
- **<i> tag:** makes text italic
- **<u> tag:** it underlines the text content
- **<sup> tag:** used to display text as a superscript.
- **<sub> tag:** used to display text as a subscript.
- **<pre> tag:** also called a preformatted tag. The whitespaces used in the element are displayed the same as written.
- **<small> tag:** used to define smaller text like copyright, comments, etc.

- **HTML <font> Tag:** (not supported in HTML5)
- The <font> tag was used in HTML 4 to specify the font face, font size, and color of text.

  Attributes: color=rgb(x,x,x) colorname

  size=number

  face=font_family

Instead use:

<p style="color:red">This is a paragraph.</p>

<p style="font-family:verdana">This is a paragraph.</p>

- **LIST TAGS :** Used to list out our items, subjects etc

1. <LI> tag – Defines list items

   Appearance: < LI >   </LI >

   Attributes: TYPE=disc|square|circle
   TYPE=1|a|A|i|I
   VALUE=n--→ Specify value of list item.

2. <OL> tag – Ordered List

   Appearance: < Ol >   </Ol >

   Attributes: TYPE=1|a|A|i|I
   START=n --→ Specify start value of ordered list

3. <UL> tag – Unordered List

   Appearance: < UL >   </ UL >

   Attributes: TYPE=disc|square|circle

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

- Syntax: <a href = "..........."> Link Text </a>
- Attributes:

download	*filename*	Specifies that the target will be downloaded when a user clicks on the hyperlink
href	*URL*	Specifies the URL of the page the link goes to
hreflang	*language_code*	Specifies the language of the linked document
target	_blank _parent _self _top	Specifies where to open the linked do

- Example: <a href=“abc.html">Click Here!</a>
- **Hyperlink To image:**
- <a href=“abc.html">
  <img border="0" alt=“Good Morning”
  src=“good_morning.gif" width="100" height="
  100">
  </a>

- **<img> Tag**
- The <img> tag is used to embed an image in an HTML page.
- The <img> tag has two required attributes:
- **src -** Specifies the path to the image
- **alt -** Specifies an alternate text for the image, if the image for some reason cannot be displayed
- **height:** Specifies the height of an image
- **width:** Specifies the width of an image
- **usemap:** Specifies an image as a client-side image map
- **Example:**
- <img src=“pic.jpg" alt=“Pic 1” width="500" height="600">

Tables

- Table is used to display data in tabular format.
- Table define with the <table> tag.
- Table header is defines <th> tag
- A table is divided into rows <tr> tag.
- Each row is divided into data cells <td> tag.
- A table cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables etc.

Table Elements

1. TABLE

Appearance: <TABLE>…..</TABLE>

Attributes:

ALIGN= left|right|center

BORDER=n → Border thickness

WIDTH=n → specify width of table

CELLPADDING=n → space between cell wall and cell content

CELLSPACING=n→ distance of the cells from one another

2. <TR> tag: Defines row in a table

Appearance:<TR> ..</TR>

Attributes:

ALIGN= left|right|center=> Horizontal

VALIGN=top|middle|bottom=> vertical

3. <TD> tag: Defines cell in HTML

Appearance:<TD>…….</TD>

Attributes:

ALIGN=left|right|center=> Horizontal

VALIGN=top|middle|bottom=> vertical

ROWSPAN=N →Sets no of rows a cell should span

COLSPAN=N → Sets no of columns a cell should span

WIDTH=n → height of cell

HEIGHT=n → Width of cell

NOWRAP → Content in a cell should not wrap.

4. <TH> tag: Define table heading

Appearance:<TH>…..</TH>

Attributes:

ALIGN=left|right|center=> Horizontal

VALIGN=top|middle|bottom=> vertical

ROWSPAN=N →Sets no of rows a cell should span

COLSPAN=N → Sets no of columns a cell should span

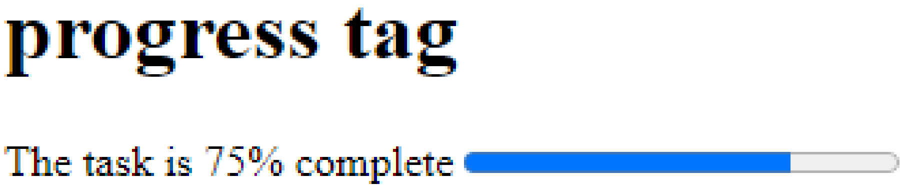
WIDTH=n → height of cell

HEIGHT=n → Width of cell

NOWRAP → Content in a cell should not wrap.

- **Example:**
-
```
<table>
  <tr>
    <th>Roll No</th>
    <th>Name</th>
  </tr>
  <tr>
    <td>AAA</td>
    <td>101</td>
  </tr>
</table>
```

- **<address> tag**
- is used to display the contact or address of a person or an organization.
- <address>
- Mailto: <a href= mailto:office@gmail.com> ABC </a>
- <br> Phone: 1234567890 <br>
- Twitter: 🐦@example_ABC
- </address>

- **<progress> tag**
- used to display an indicator to show the completion progress of a task. It is displayed as a progress bar.
- The tag accepts 2 attribute
- **max** - Describes the highest value of the progress bar
- **value** - Describes the current value of the progress bar.
- Example:
- <body>
- <h1>progress tag</h1>
- <p>The task is 75% complete <progress max="100" value="75"> </progress>
- </p>
- </body>
- 

  progress tag

  The task is 75% complete

- **<time> tag**
- It is used to represent time in HTML. It contains a datetime attribute that stores the exact date and time of an event, which is used by search engines to provide better results.
- Example:
- <p>I have a meeting at
- <time>02:30 pm</time>.</p> <p><time datetime="2021-06-23 17:00:00">My birthday</time> is in the summer.</p>

- **<figure> tag**
- The <figure> behaves like a container for images and also gives an option for the image caption defined by <figcaption>.
- **<figcaption> tag**
- The <figcaption> tag is used to create a caption for an image.
- **Example:**
- <figure>
- <img src=“sunset.jpg"   alt=“Sunset">
- <figcaption>Sunset Pic</figcaption>
- </figure>

Marquee Tag

- The <marquee> tag is a container tag of HTML is implemented for creating scrollable text or images within a web page from either left to right or vice versa, or top to bottom or vice versa.
- The different attributes of <marquee> tag are:

Attribute	Description
width	provides the width or breadth of a marquee. For example width="10" or width="20%"
height	provides the height or length of a marquee. For example height="20" or height="30%"
direction	provides the direction or way in which your marquee will allow you to scroll. The value of this attribute can be: left, right, up or down
scrolldelay	provides a feature whose value will be used for delaying among each jump.

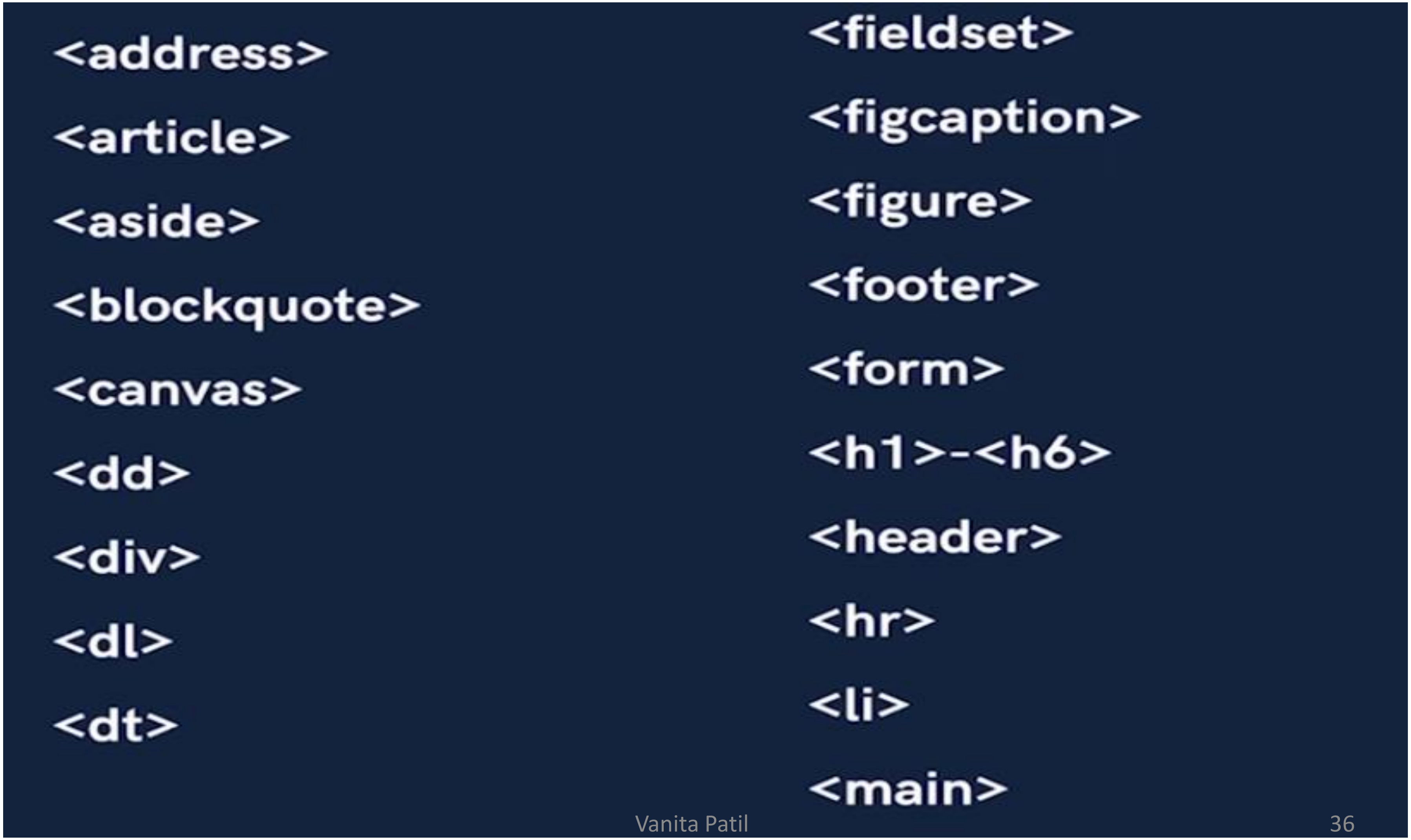
scrollamount	provides value for speeding the marquee feature
behavior	provides the scrolling type in a marquee. That scrolling can be like sliding, scrolling or alternate
loop	provides how many times the marquee will loop
bgcolor	provides a background color where the value will be either the name of the color or the hexadecimal color-code.
vspace	provides a vertical space and its value can be like: vspace="20" or vspace="30%"
hspace	provides a horizontal space and its value can be like: hspace="20" or hspace="30%"

- **Example:**
- <marquee direction="down" behavior="alternate" height="500px">
- <marquee scrollamount="20" direction="right"><img src="car.gif"> </marquee>
- </marquee>

Div Tag

- <div> tag defines a division or a section in an HTML document.
- <div> element is used as a container for other elements.
- Block element (takes full width)
- By default browser place a line break before and after the <div> element.
- This can be changed with CSS.

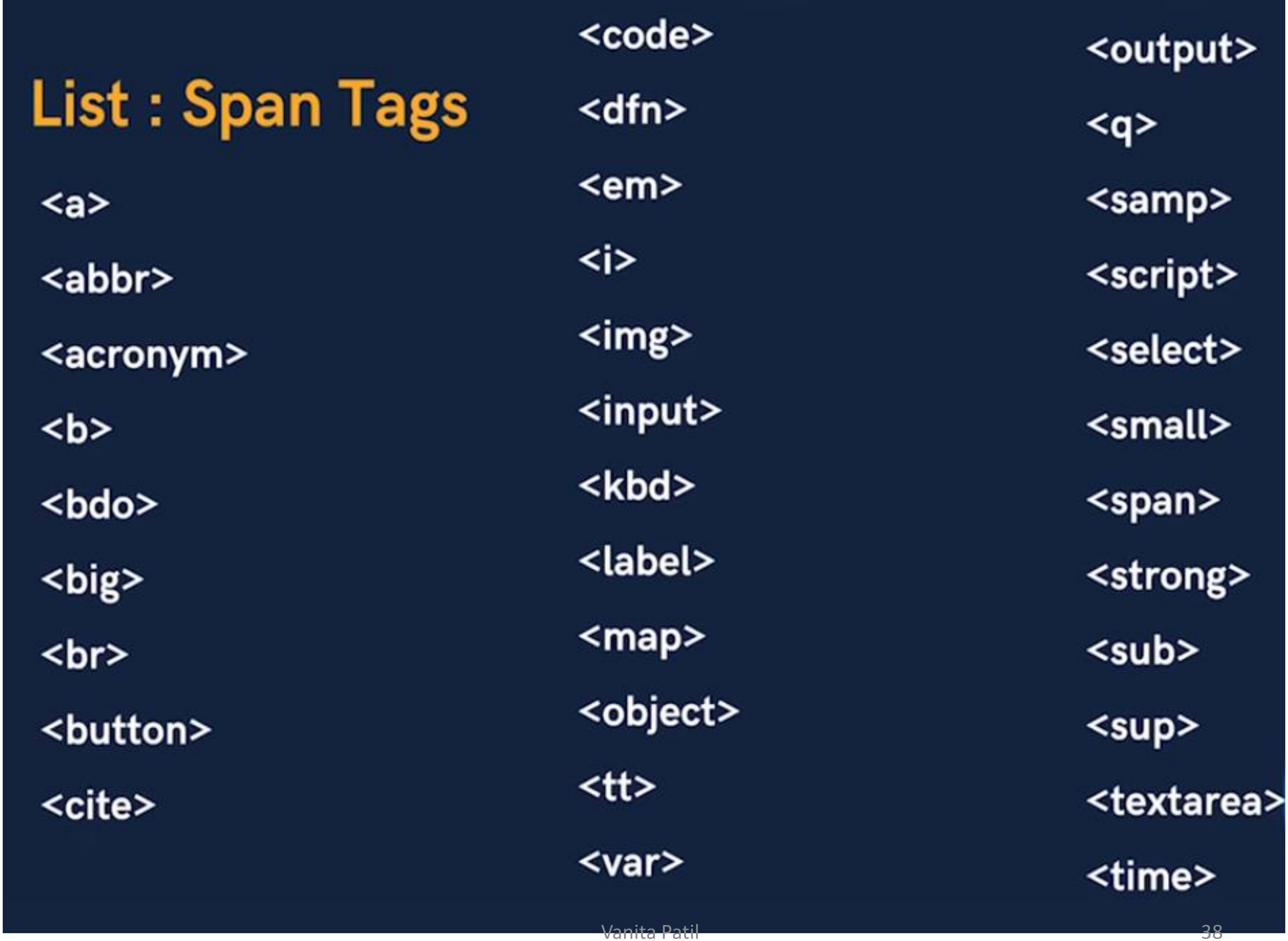
List of Block Tags



- <address>
- <article>
- <aside>
- <blockquote>
- <canvas>
- <dd>
- <div>
- <dl>
- <dt>
- <fieldset>
- <figcaption>
- <figure>
- <footer>
- <form>
- <h1>-<h6>
- <header>
- <hr>
- <li>
- <main>

Span Tag

- Span is also a container for other HTML elements.
- Inline element (takes width as per size)
- Eg.
- <span>
- <p>Hello</p>
- </span>



List : Span Tags

<a>

<abbr>

<acronym>

<b>

<bdo>

<big>

<button>

<cite>

<code>

<dfn>

<em>

<i>

<img>

<input>

<kbd>

<label>

<map>

<object>

<tt>

<var>

<output>

<q>

<samp>

<script>

<select>

<small>

<span>

<strong>

<sub>

<sup>

<textarea>

<time>

HTML Input Controls

- **1. Text input controls**
- Ex. <label for="username">Username:</label><br>
-     <input type="text" id="username“ name="username">
- **2. Password input controls**
- <label for="pwd">Password:</label><br>
- <input type="password" id="pwd" name="pwd">
- **3. Multiple-Line Text Input Controls**

- <input type="submit"> defines a button for **submitting** form data to a **form-handler**.
- <input type="reset"> defines a **reset button** that will reset all form values to their default values.
- <input type="button"> defines a **button.**
- Eg. <input type="button" onclick="alert('Hello World!')" value="Click Me!">

- **Example:**
- <form action=“home.html">
- <label for="pwd">Password:</label><br>
- <input type="password" id="pwd" name="pwd"><br><br>
- <input type="submit" value="Submit">
- </form>

- **<input type="radio">** defines a radio button.
- Exmple:
- <input type="radio" id="html" name="fav_language" value="HTML">
- <label for="html">HTML</label><br>
- **<input type="checkbox">** defines a checkbox.
- Example:
- <input type="checkbox" id="vehicle1" name=" vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike </label>

- The <input type="file"> defines a file-select field and a "Browse" button for file uploads.
- **Example:**
- <form action=“a.html">
- <label for="myfile">Select a file:</label>
- <input type="file" id="myfile" name="myfile"><br><br>
- <input type="submit" value="Submit">
- </form>

- <textarea> Control:
- Example:
- <textarea rows = "5" cols = "50" name = "description"> Enter description here... </textarea>
- Select Box Control:
- Example:
- <select name = "dropdown">
- <option value = "Maths" selected> Maths </option>
- <option value = "Physics">Physics</option> </select>

New Input Controls in HTML

- There are almost 13 new input types introduced in HTML5 form.
- **color:** This input type allows the user to select a color from a color picker.
- **date:** This input type allows the user to select a date from a drop-down calendar.
- **time:** This input type allows the user to enter a time.
- **datetime:** This input type allows the user to select date and time along with timezone.
- **datetime-local:** This input type allows the user to select both local date and time.
- **week:** This input type allows the user to select week and year from the drop-down calendar.
- **email:** This input type allows the user to enter an e-mail address.

- **month:** This input type allows the user to select a month and year from a drop-down calendar.
- **number:** This input type allows the user to enter a numerical value.
- **range:** This input type allows the user to enter a numerical value within a specified range.
- **search:** This input type allows the user to enter a search string within the input field.
- **tel:** This input type allows the user to enter a telephone number.
- **url:** This input type allows the user to enter the URL.

- **Example of color, date and time:**
- Date Syntax: <input type="date">
- Time Syntax: <input type="time">
- Color Syntax: <input type="color">
- **Example:**

```
<body>
    <form>
        <label for="color">Select Color : </label>
        <input type="color" /><br />
         <label for="date">Select Date : </label>
        <input type="date" /><br />
        <label for="time">Select Time :</label>
        <input type="time" />
    </form>
</body>
```

- **Example of datetime, datetime-local and week input type.**
- Datetime : <input type=”datetime”>
- Datetime-local : <input type="datetime-local">
- Week: <input type="week">
- **Note: datetime-local** and **week** are not supported by Firefox, Safari, and Internet Explorer browsers.
- **datetime** input type for entering a date and time, but it is now obsolete.
- **Example:**
- <body>
- <form>
- <label for="date">Enter Date-Time : </label>
- <input type="datetime" /> <br />
- <label for="date">Select Date-Time Local : </label>
- <input type="datetime-local" /> <br />
- <label for="time">Select Week : </label>
- <input type="week" />
- </form>
- </body>

- **Example of email, month, number and range input type.**
- **Email Syntax:** <input type="email">
- **Month Syntax:** <input type="month">
- **Note: month** is not supported by Firefox, Safari and Internet Explorer browsers.
- **Number Syntax:** <input type="number">
- **Range Syntax:** <input type="range">

- **Example**:
- <body>
- <form>
- <label for="mail">Enter Email : </label>
- <input type="email" required/><br />
- <label for="date">Select Month : </label>
- <input type="month" /><br />
- <label for="number">Select Number : </label>
- <input type="number" /> <br />
- <label for="number">Select Number : </label>

<input type="number" min="1" max="10" step="0.5“ id="mynumber"> <br />

- <label for="number">Select Number in Range : </label>
- <input type="range" min="1" max="10" step="1" />
- </form>
- </body>

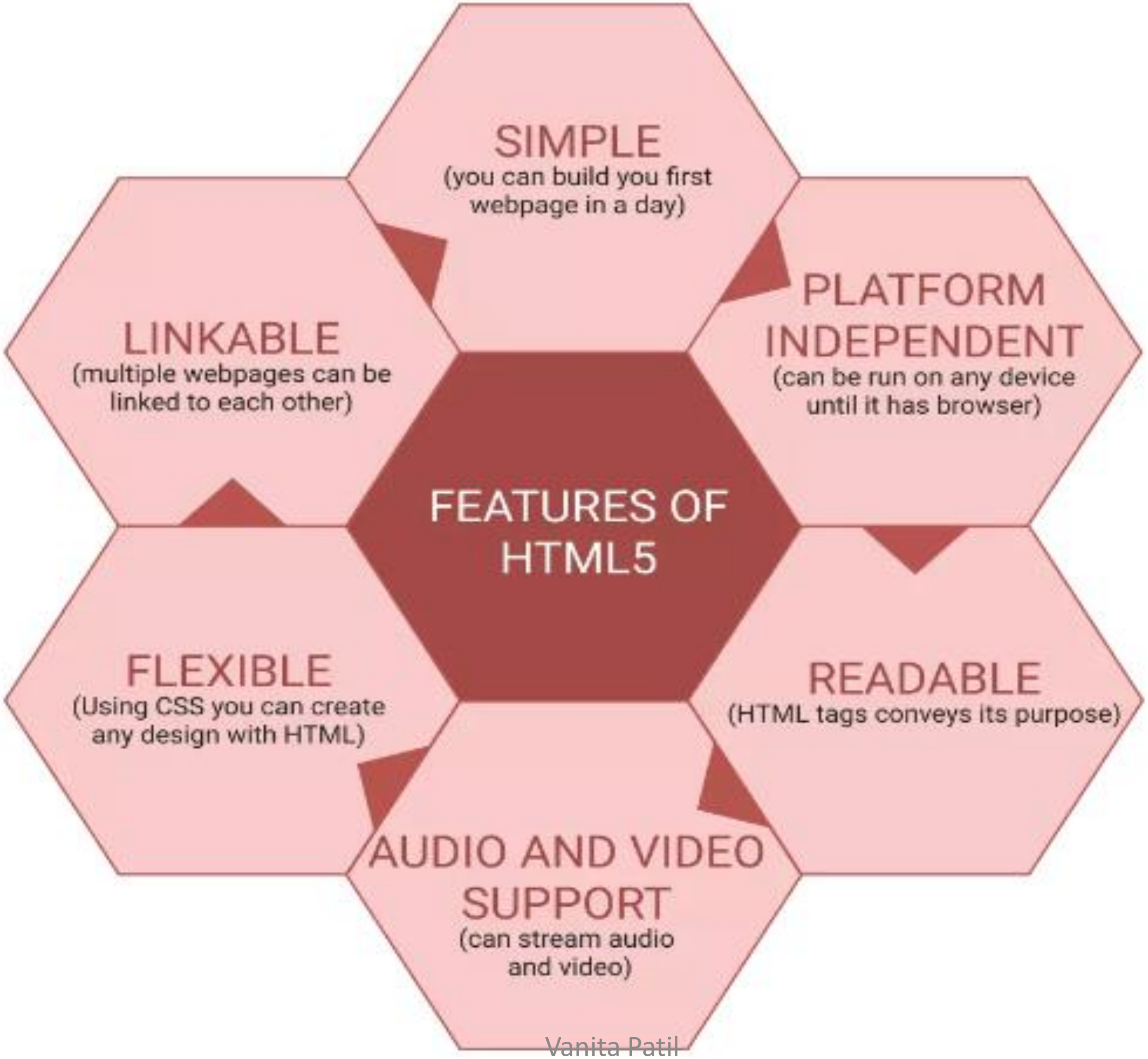
- **Example of search, tel and url input types.**
- **Search Syntax:** <input type="search">
- **Tel Syntax:** <input type="tel">
- **Note:** Currently **tel** is not supported by any browser.
- **Url Syntax:** <input type="url">
- **Example:**

```
<body>
    <form>
        <label for="search">Enter Search string : </label>
        <input type="search" placeholder="search ..." />  <br />
        <label for="number">Enter Telephone No. : </label>
        <input type="tel" placeholder="xxx-xxx-xxxx" /> <br />
         <label for="url">Enter Url : </label>
        <input type="url" placeholder="https://www..." />
    </form>
</body>
```

Basics of HTML5 – Introduction

- **Html5** is the combination of three web technologies:
  HTML → to build webpage structure,
  CSS → to enhance look and feel(presentation layer), and
  Javascript → to add functionality to HTML elements.
- **HTML + CSS + JAVASCRIPT = HTML5**
- The term **HTML5** means not only HTML, it is a combination of HTML, CSS and JavaScript with APIs.
- For example, drawing and animation using **canvas**, **offline storage**, **microdata**, **audio and video**, **drag and drop**, **geolocation**, **embedded fonts**, web APIs etc.
- **HTML5** includes new semantic tags and some old tags(*with redefinition*).

Features Of HTML



SIMPLE
(you can build you first webpage in a day)
PLATFORM INDEPENDENT
(can be run on any device until it has browser)
READABLE
(HTML tags conveys its purpose)
AUDIO AND VIDEO SUPPORT
(can stream audio and video)
FLEXIBLE
(Using CSS you can create any design with HTML)
LINKABLE
(multiple webpages can be linked to each other)
FEATURES OF HTML5

- HTML5 Changed Elements
- These elements were already there in HTML4/XHTML. But in **HTML5**, the definition or usage has been changed.
- **a** tag is now **Hyperlink**, not anchor.
- **hr** tag is *Thematic Break*, not *Horizontal Rule*
- **input tag** is *Input Control*.
- **dl** tag is *description list*, not *definition list*.
- **b** tag is *offset text conventionally styled in bold*
- **i** tag is *offset text conventionally styled in italic*
- **u** tag is *offset text conventionally styled with an underline*

- Difference Between HTML and HTML5

Sr No	HTML	HTML5
1	Hyper text mark-up language is acronym for HTML which is primary language for developing web pages	HTML5 is a new version of HTML which has new functionality with mark-up language as the core technology to interact with internet technologies for structuring and presenting the content.
2	It does not support audio and video without the use of a flash player.	It supports audio and video controls with the use of <audio> and <video> tags.
3	HTML has a support of tracking users location who are visiting the site but the process is cumbersome and difficult to find users location when logged from mobile devices	HTML5 has using JavaScript Geolocation API which can be used to identify the location of any user who is accessing the website.
4	HTML uses browser cache memory as temporary storage.	HTML5 offers multiple storage options, such as an SQL database, application cache, and web storage.

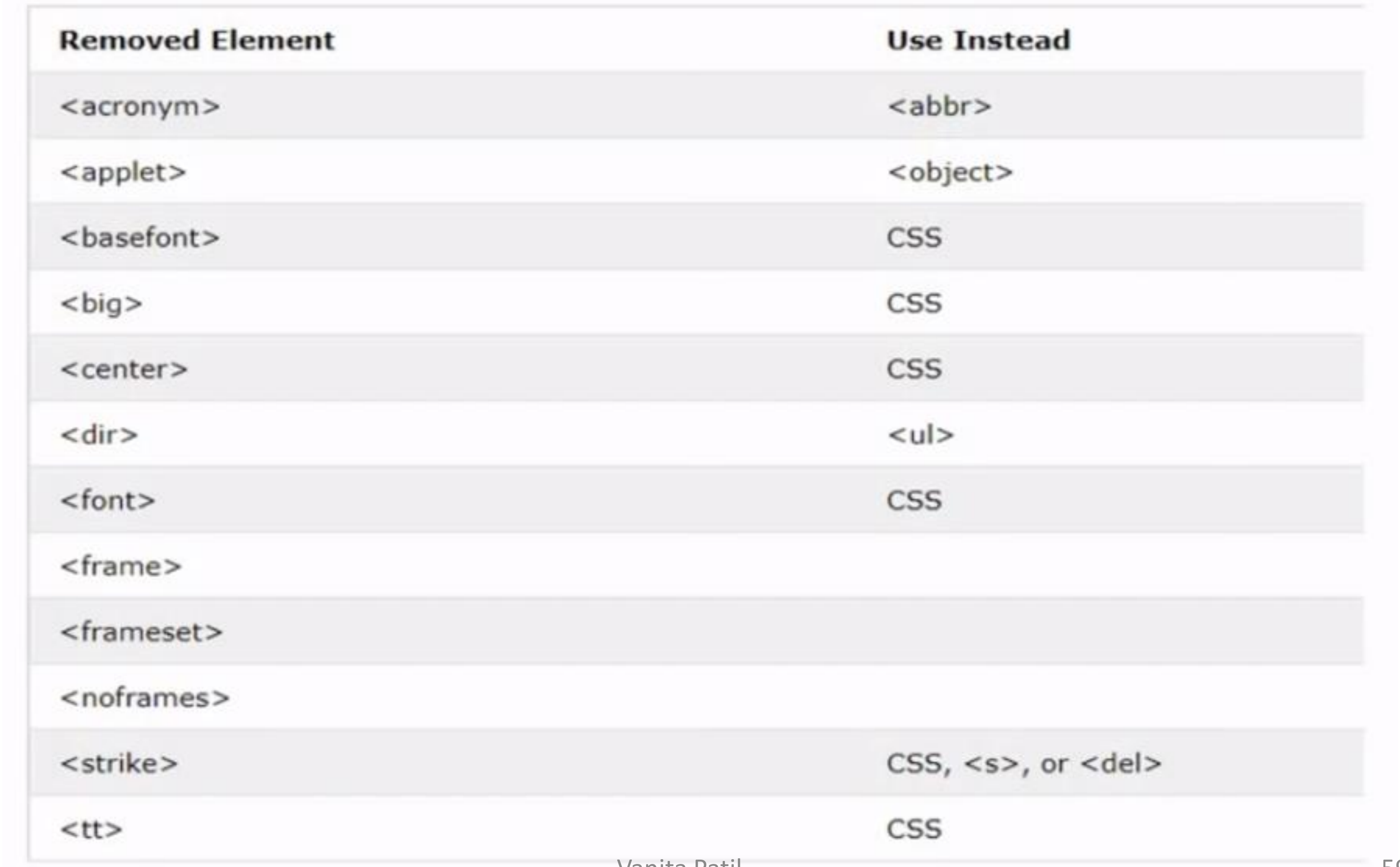
5	HTML is compatible with almost all browsers as it has existed for a long time, and browsers did enough modification to support all features in HTML	In HTML5, we have many new tags and elements and removed/modified a few tag elements, so only a few browsers are fully compatible with HTML5.
6	In HTML, Vector Graphics support is possible with the help of other tools such as Silverlight, Adobe Flash, VML, etc.	HTML5 supports Vector Graphics by default because it has built-in canvas and SVG.
7	HTML cannot handle incorrect syntax and any other errors.	HTML5 is capable of handling incorrect syntax and other errors.
8	The doctype declaration is long and complicated in HTML.	The doctype declaration is comparatively quite simple and easy to understand in HTML5.

- HTML Versions

Year	Achievement
1989	HTML Formed
1995	HTML 2
1995	CSS
1995	JAVASCRIPT
1997	HTML 4
1998	CSS 2
2000	XHTML 1
2001	XHTML 1.1
2009	HTML5
2015	HTML5.1
2017	HTML5.2

- **HTML5 Browsers Support:**
- **HTML5** is supported by latest browsers only.
- The first browser support of html5 was seen in 2011.
- Here is the list of supported browsers for html5.
- Chrome 4 and above
- Firefox 3.6 and above
- Opera 11 and latest versions
- Safari 5.1 for Mac, Iphones and Ipads
- Internet Explorer 9 and above
- Edge 12 and above Browsers

- **HTML5 Removed Tags:**
- With HTML5, some *presentational tags* are removed. These tags still get browser support, but W3C validator shows errors. Here is a list of some **removed elements in html5**.



Removed Element	Use Instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	<ul>
<font>	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, or <del>
<tt>	CSS

HTML5 Document

- **1. The HTML5 Doctype:**
- The first line of every HTML5 document is a special code called *doctype*.
- It clearly announces to anyone who's reading the document markup that HTML5 content follows:
- Syntax: <!DOCTYPE html>
- **2. Character Encoding:**
- The *character encoding* is the standard that tells a computer how to convert your text into a sequence of bytes when it's stored in a file (and how to convert it back again when the file is opened).

- Today, virtually all English websites use an encoding called UTF-8, which is compact, fast, and supports all the non-English characters.
- you need to add the meta element shown below at the very beginning of your <head> section.
- <head>
- **<meta charset="utf-8">**
- <title>A Tiny HTML Document</title>
- </head>

- **3. The Language**
- This information is occasionally useful to other people—for example, search engines can use it to filter search results so they include only pages that match the language of the searcher.
- To specify the language of some content, you use the *lang* attribute on any element, along with the appropriate language code. That’s ***en*** for plain English.
- Syntax: <html lang="en">

- **4. Adding a Style Sheet**
- Virtually every web page in a properly designed, professional website uses style sheets. You specify the style sheets you want to use by adding <link> elements to the <head> section of an HTML5 document, like this:
- <head>
- <meta charset="utf-8">
- <title>A Tiny HTML Document</title>
- **<link href="styles.css" rel="stylesheet">**
  </head>
- there’s no need to add the *type=“text/css”* attribute that web pages used to require.

- **5. Adding JavaScript**
- You add JavaScript to an HTML5 page in much the same way that you add it to a traditional HTML page.
- <head>
- <meta charset="utf-8">
- <title>A Tiny HTML Document</title>
- **<script src="scripts.js"></script>**
- </head>
- There’s no need to include the *language=“JavaScript”* attribute.

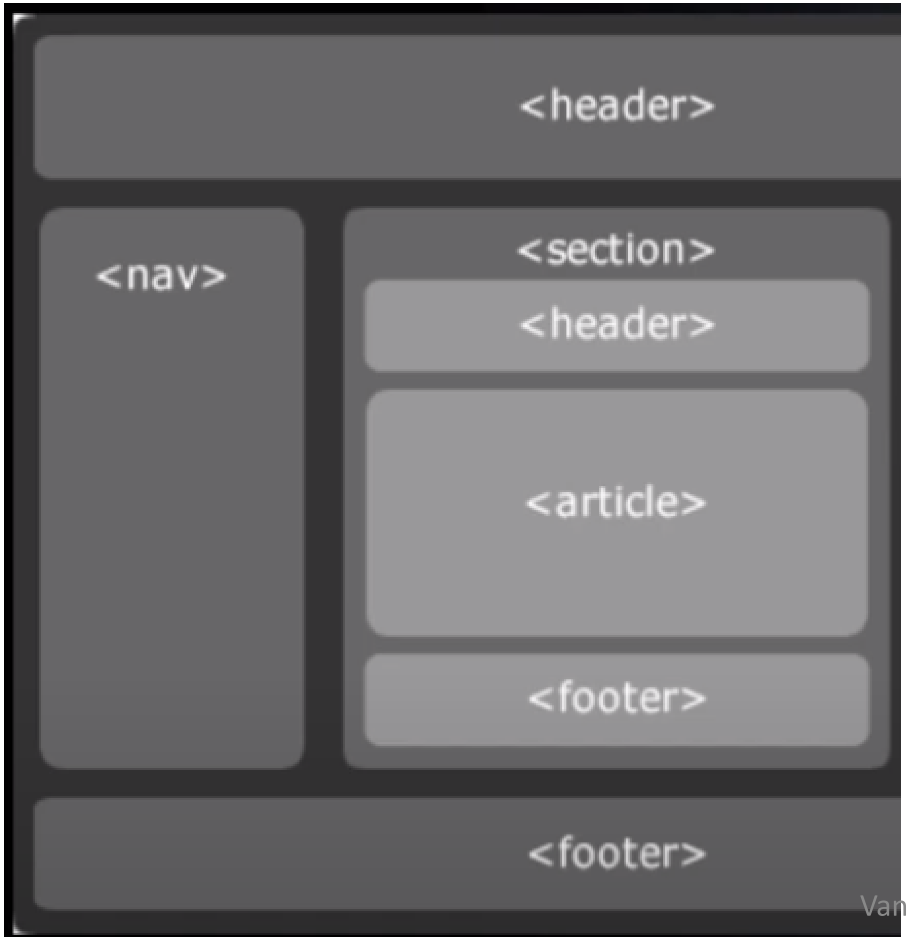
HTML5 Semantics

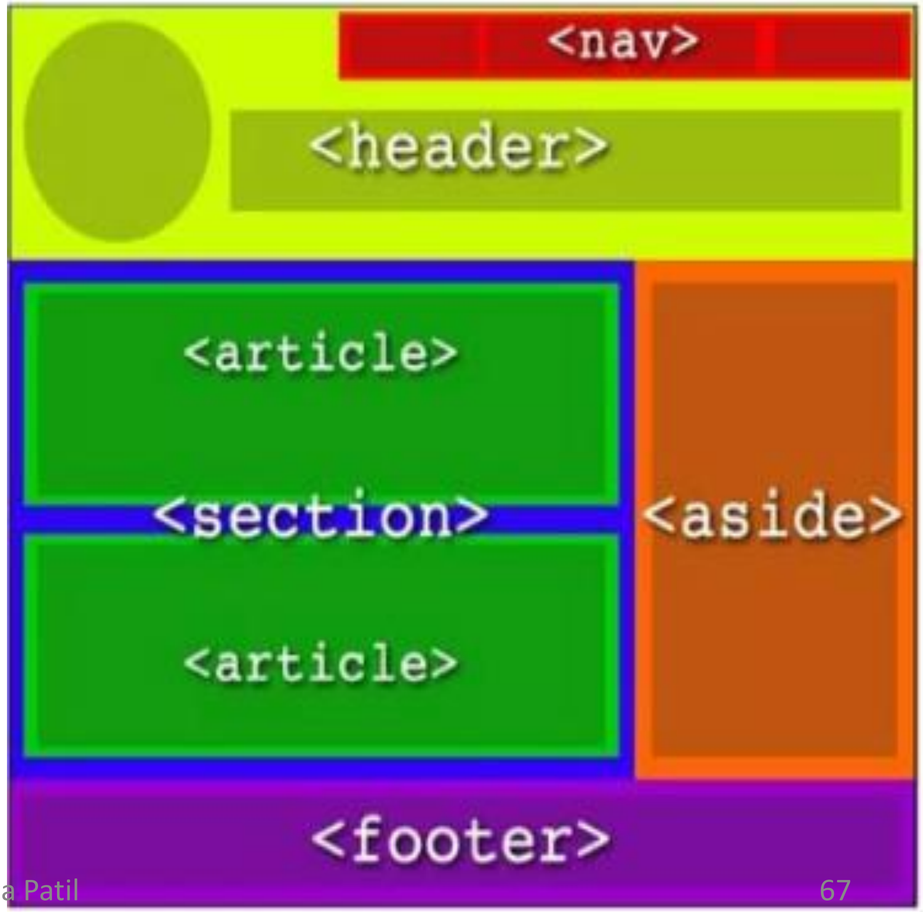
- HTML tags are classified in two types.
- Semantic
- Non-Semantic
- **Semantic Elements:** Semantic elements have meaningful names which tells about type of content. For example header, footer, table, … etc. HTML5 introduces many semantic elements as mentioned below which make the code easier to write and understand for the developer as well as instructs the browser on how to treat them.

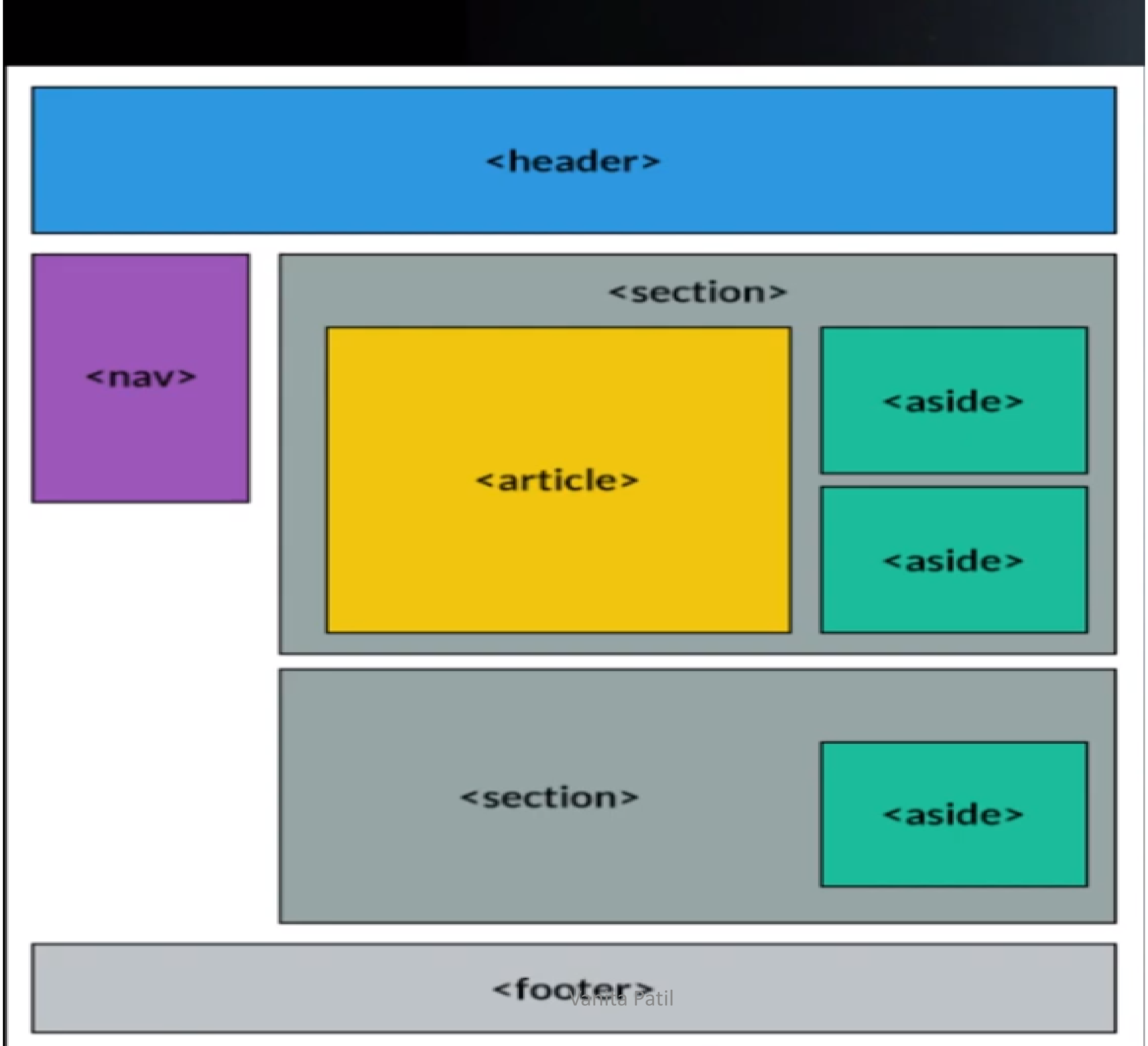
- **HTML5 Semantic Elements:**
- 1. Article
- 2. section
- 3. Aside
- 4. header
- 5. Footer
- 6. nav
- 7. figcaption
- 8. figure
- 9. details
- 10. main
- 11. mark

HTML Layout

- It has several elements that allows to control the structure of our web page.

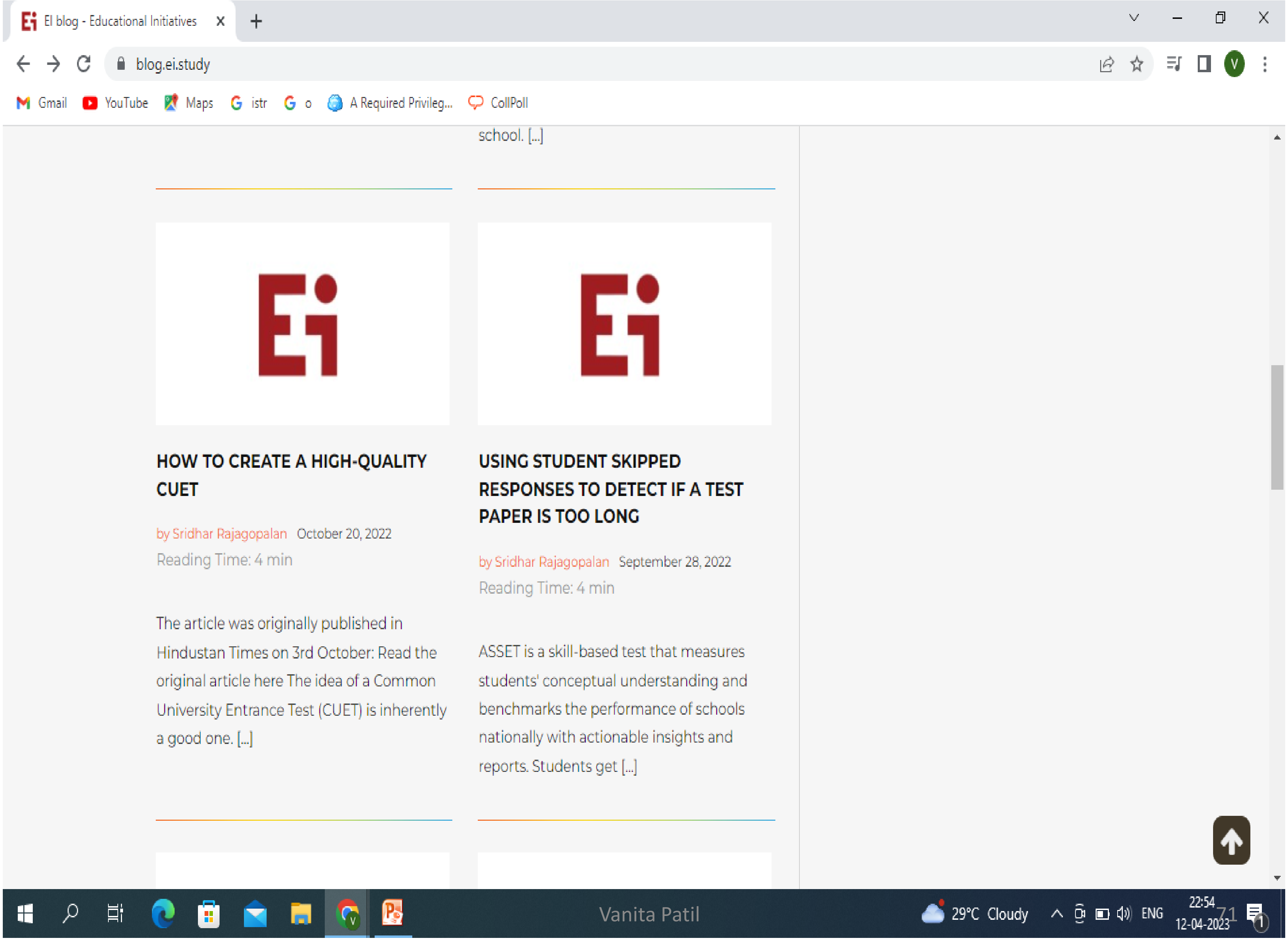






- **1. Article**
- The **<article>** tag is one of the new sectioning element in HTML5. The HTML **<article>** tag is used to represent an article.
- The content within the <article> tag is independent of the other content of the site (even though it can be related).
- An **Article** could be : A magazine/newspaper article, A blog entry, A forum post, A user-submitted a comment
- **Note:** This tag does not render as anything special in a browser, you have to use CSS for that.
- Syntax: <article>-------------</article>
- **Example:**
- <article>
- <p> Some contents here </p>
- </article>

- This tag contains independent content that doesn’t require any other context.
- So the <article> tag can be placed inside the main content.
- But each of the articles will contain independent content within it.
- Like YouTube use to contain different kinds of videos on a single page, each video is independent or you can see the course page , each course is independent, each course can have its own page.



school. [...]

HOW TO CREATE A HIGH-QUALITY CUET

by Sridhar Rajagopalan October 20, 2022

Reading Time: 4 min

The article was originally published in Hindustan Times on 3rd October: Read the original article here The idea of a Common University Entrance Test (CUET) is inherently a good one. [...]

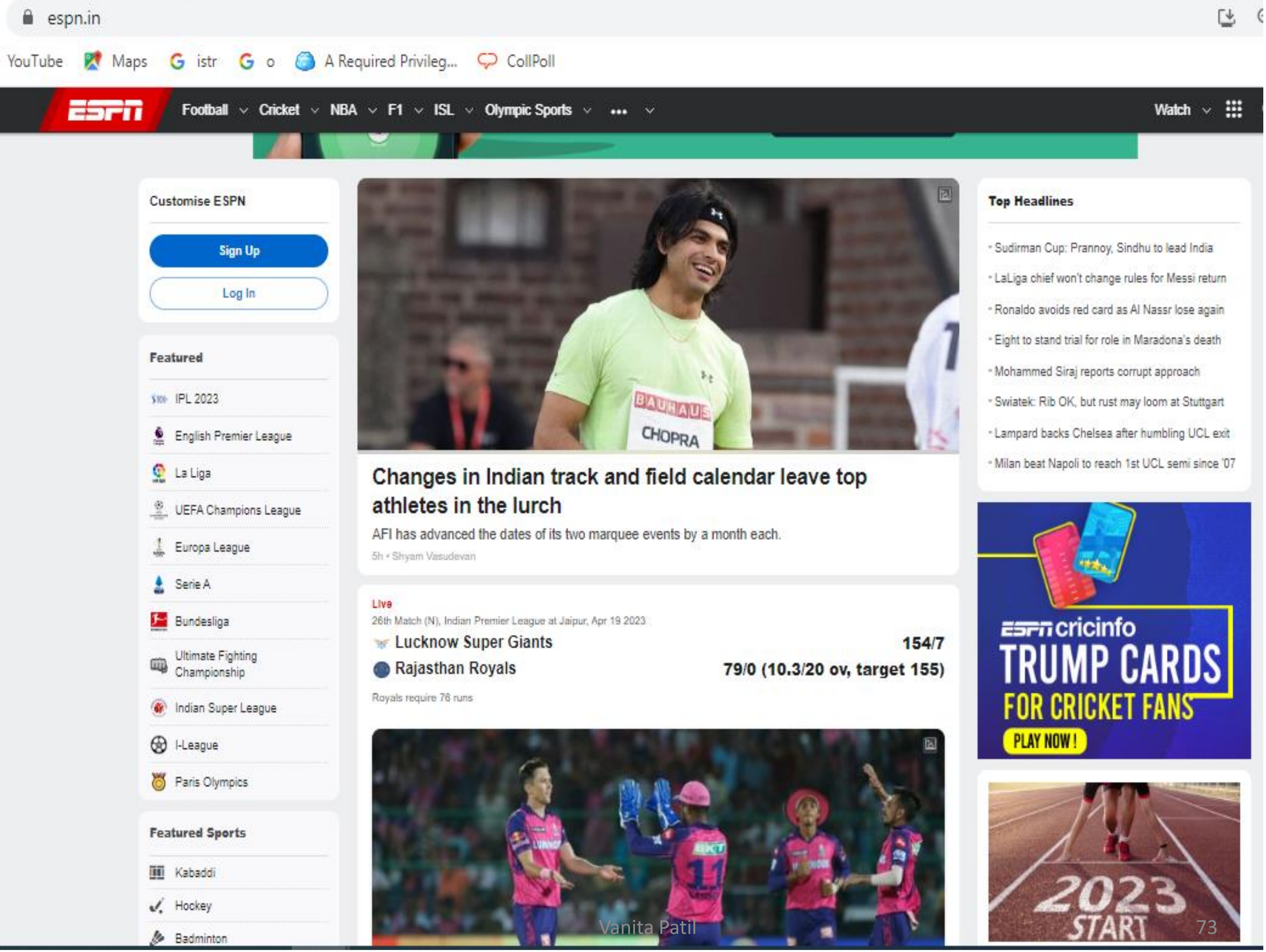
USING STUDENT SKIPPED RESPONSES TO DETECT IF A TEST PAPER IS TOO LONG

by Sridhar Rajagopalan September 28, 2022

Reading Time: 4 min

ASSET is a skill-based test that measures students' conceptual understanding and benchmarks the performance of schools nationally with actionable insights and reports. Students get [...]

- 2. Section Tag:
- This tag is used to split a page into sections like Introduction, Contact Information, Details, etc and each of these sections can be in a different <section> tag.
- The <section> tag divides the content into sections and subsections.
- The section tag is used when requirements of two headers or footers or any other section of documents needed.
- **Note:** Placing <article> tag inside of <section> tag is good practice, like section basically defines the types and the articles will contain the specific contents in that type of section.



espn.in

YouTube Maps istr o A Required Privileg... CollPoll

Customise ESPN

Featured

- IPL 2023
- English Premier League
- La Liga
- UEFA Champions League
- Europa League
- Serie A
- Bundesliga
- Ultimate Fighting Championship
- Indian Super League
- I-League
- Paris Olympics

Featured Sports

- Kabaddi
- Hockey
- Badminton

Changes in Indian track and field calendar leave top athletes in the lurch

AFI has advanced the dates of its two marquee events by a month each.

5h • Shyam Vasudevan

Live

26th Match (N), Indian Premier League at Jaipur, Apr 19 2023

Lucknow Super Giants	154/7
Rajasthan Royals	79/0 (10.3/20 ov, target 155)

Royals require 76 runs

Top Headlines

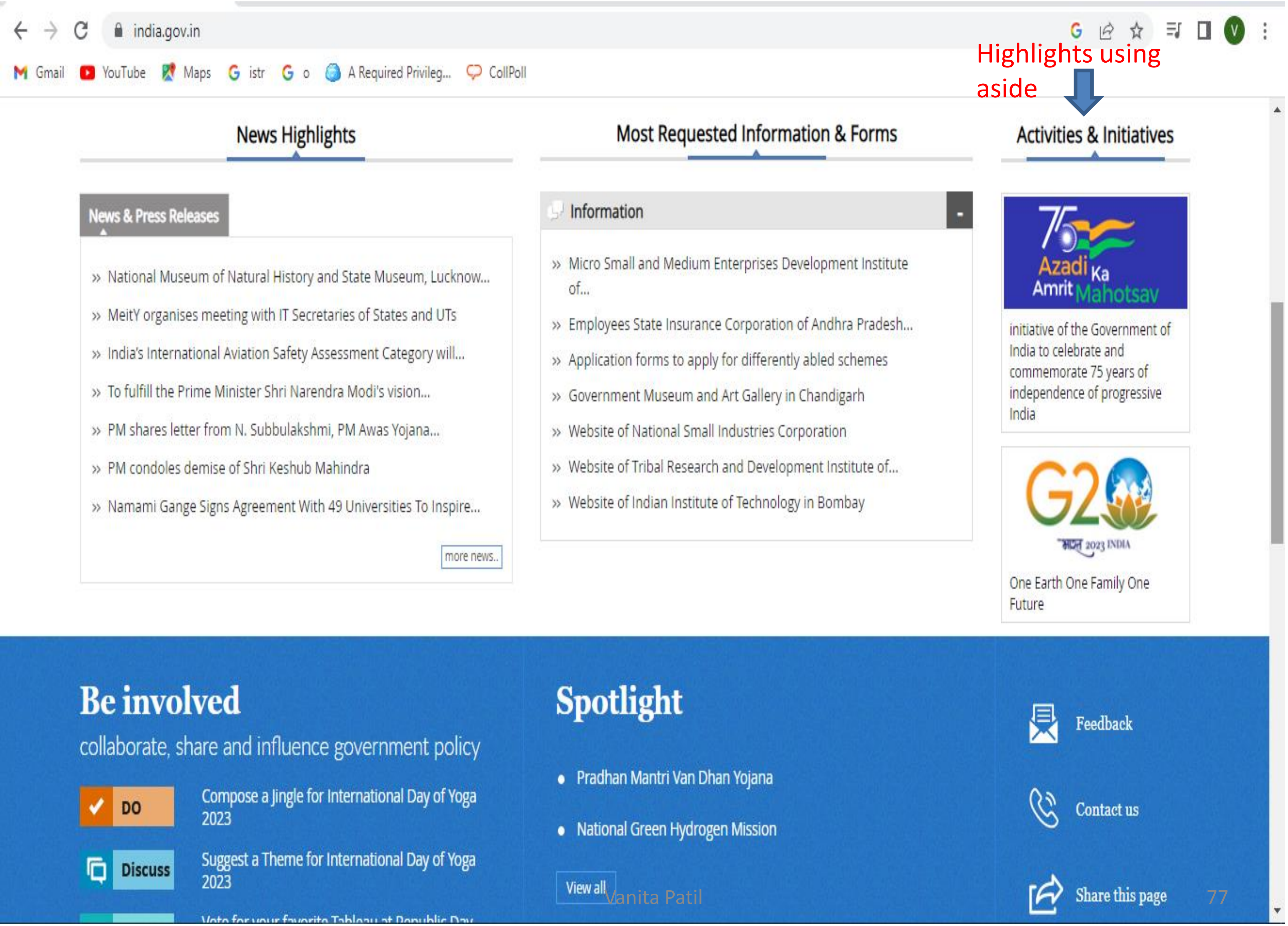
- Sudirman Cup: Prannoy, Sindhu to lead India
- LaLiga chief won't change rules for Messi return
- Ronaldo avoids red card as Al Nassr lose again
- Eight to stand trial for role in Maradona's death
- Mohammed Siraj reports corrupt approach
- Swiatek: Rib OK, but rust may loom at Stuttgart
- Lampard backs Chelsea after humbling UCL exit
- Milan beat Napoli to reach 1st UCL semi since '07

- Body contains n number of sections
- **Section** contains n number of articles.
- **article** contains n number of paragraphs.
- **Syntax:** <section> -------- </section>
- **Example:**
- <section>
- <article> This is the article element </article>
- <article> This is the another article element </article>
- </section>

- **3. <aside>**
- **Aside tag is used to represent the content with in the article or web page.**
- Able to display the existing text aside that is left or right.
- The HTML <aside> tag is used to represent a portion of a document that is indirectly related to the main content.
- It is most commonly used as a sidebar in the document.
- **Example.**
- <aside>
- Hello Welcome!!
- <p>HTML stands for **HyperText Markup Language**. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes. </p>
- </aside>

- **Note:** The <aside> tag is new in HTML5. This tag does not render as anything special in a browser you have to use CSS for that.

Highlights using aside



india.gov.in

News Highlights

News & Press Releases

- National Museum of Natural History and State Museum, Lucknow...
- MeitY organises meeting with IT Secretaries of States and UTs
- India's International Aviation Safety Assessment Category will...
- To fulfill the Prime Minister Shri Narendra Modi's vision...
- PM shares letter from N. Subbulakshmi, PM Awas Yojana...
- PM condoles demise of Shri Keshub Mahindra
- Namami Gange Signs Agreement With 49 Universities To Inspire...

more news..

Most Requested Information & Forms

Information

- Micro Small and Medium Enterprises Development Institute of...
- Employees State Insurance Corporation of Andhra Pradesh...
- Application forms to apply for differently abled schemes
- Government Museum and Art Gallery in Chandigarh
- Website of National Small Industries Corporation
- Website of Tribal Research and Development Institute of...
- Website of Indian Institute of Technology in Bombay

Activities & Initiatives

75 Azadi Ka Amrit Mahotsav

initiative of the Government of India to celebrate and commemorate 75 years of independence of progressive India

G20 भारत 2023 INDIA

One Earth One Family One Future

Be involved

collaborate, share and influence government policy

✓ DO — Compose a Jingle for International Day of Yoga 2023

Discuss — Suggest a Theme for International Day of Yoga 2023

Vote for your favorite Tableau at Republic Day

Spotlight

- Pradhan Mantri Van Dhan Yojana
- National Green Hydrogen Mission

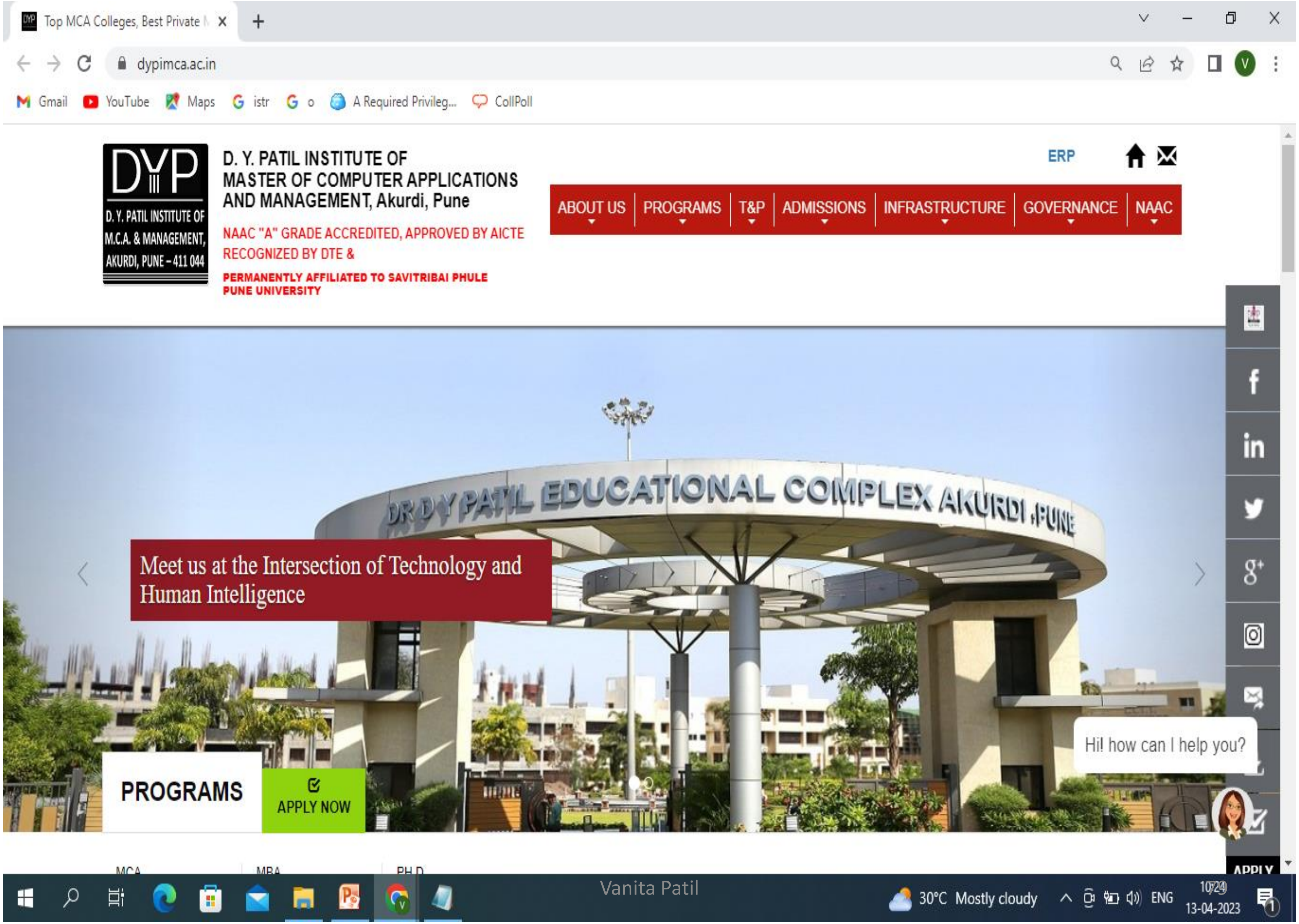
View all

Feedback

Contact us

Share this page

- **4.<header>**
- Defines a **header** for a document, section or an article. **Header** can be used more than once on a single webpage. But try to use single **header** in a section, or an article.
- Syntax: <header> -------- </header>
- **Example:**
- <section>
- <header>
- <h2>Header for Section<h2>
- </header>
- <p>Section Content Goes Here...</p> </section>



Top MCA Colleges, Best Private M

dypimca.ac.in

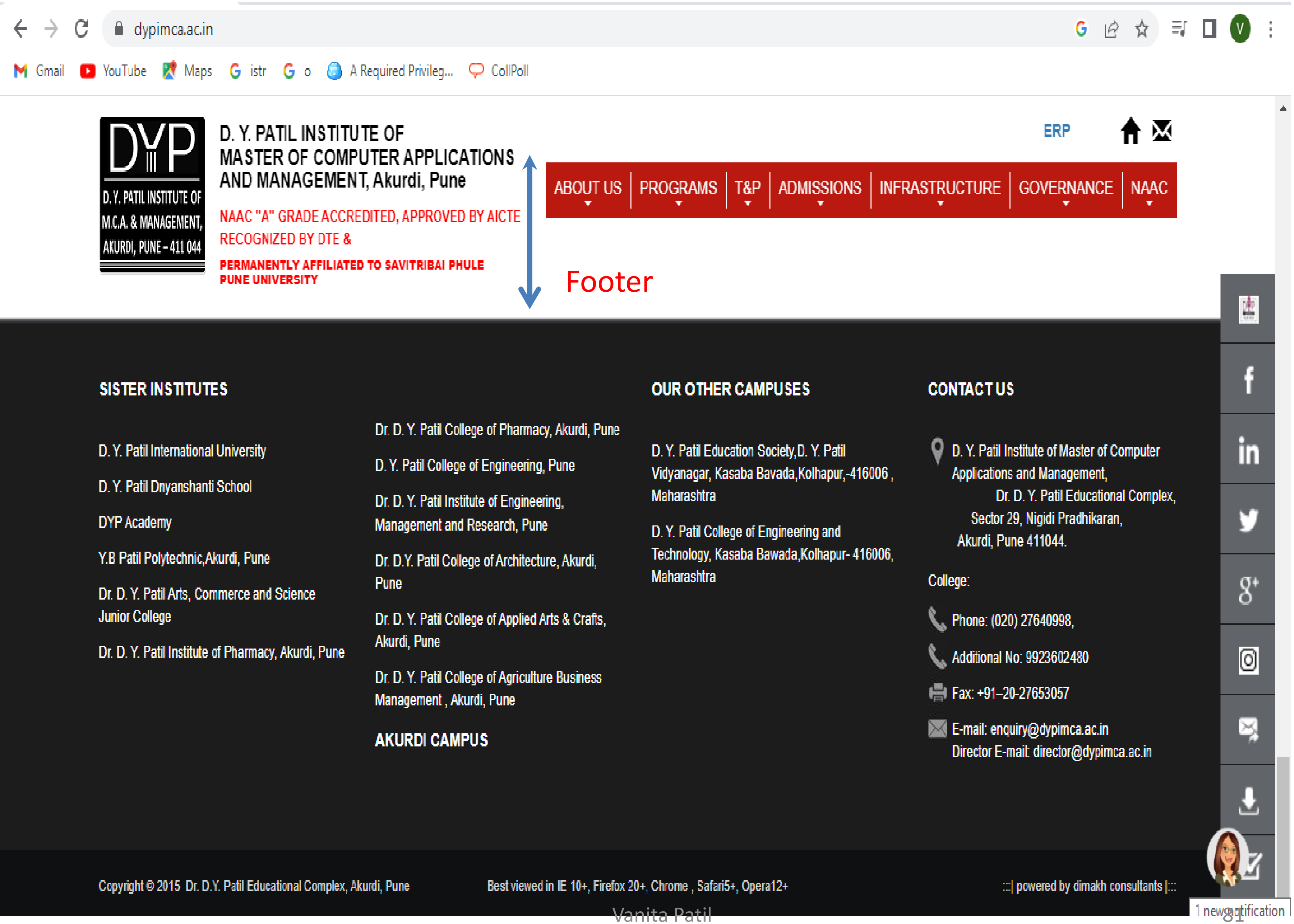
D. Y. PATIL INSTITUTE OF
MASTER OF COMPUTER APPLICATIONS
AND MANAGEMENT, Akurdi, Pune

NAAC "A" GRADE ACCREDITED, APPROVED BY AICTE
RECOGNIZED BY DTE &

PERMANENTLY AFFILIATED TO SAVITRIBAI PHULE
PUNE UNIVERSITY

Meet us at the Intersection of Technology and Human Intelligence

- **5. <Footer>**
- **Footer tag** defines footer of an document, section or an article.
- Footer usually contains the information such as author of a document, copyright information, links to terms of use, privacy policy etc.
- Syntax: <footer> ------ </footer>
- Example:
- <footer> <p>Content for footer<p> </footer>



D. Y. PATIL INSTITUTE OF
M.C.A. & MANAGEMENT,
AKURDI, PUNE – 411 044

D. Y. PATIL INSTITUTE OF
MASTER OF COMPUTER APPLICATIONS
AND MANAGEMENT, Akurdi, Pune

NAAC "A" GRADE ACCREDITED, APPROVED BY AICTE
RECOGNIZED BY DTE &

PERMANENTLY AFFILIATED TO SAVITRIBAI PHULE
PUNE UNIVERSITY

Footer

SISTER INSTITUTES

D. Y. Patil International University

D. Y. Patil Dnyanshanti School

DYP Academy

Y.B Patil Polytechnic,Akurdi, Pune

Dr. D. Y. Patil Arts, Commerce and Science Junior College

Dr. D. Y. Patil Institute of Pharmacy, Akurdi, Pune

Dr. D. Y. Patil College of Pharmacy, Akurdi, Pune

D. Y. Patil College of Engineering, Pune

Dr. D. Y. Patil Institute of Engineering, Management and Research, Pune

Dr. D.Y. Patil College of Architecture, Akurdi, Pune

Dr. D. Y. Patil College of Applied Arts & Crafts, Akurdi, Pune

Dr. D. Y. Patil College of Agriculture Business Management , Akurdi, Pune

AKURDI CAMPUS

OUR OTHER CAMPUSES

D. Y. Patil Education Society,D. Y. Patil Vidyanagar, Kasaba Bavada,Kolhapur,-416006 , Maharashtra

D. Y. Patil College of Engineering and Technology, Kasaba Bawada,Kolhapur- 416006, Maharashtra

CONTACT US

D. Y. Patil Institute of Master of Computer Applications and Management,
Dr. D. Y. Patil Educational Complex,
Sector 29, Nigidi Pradhikaran,
Akurdi, Pune 411044.

College:

Phone: (020) 27640998,

Additional No: 9923602480

Fax: +91-20-27653057

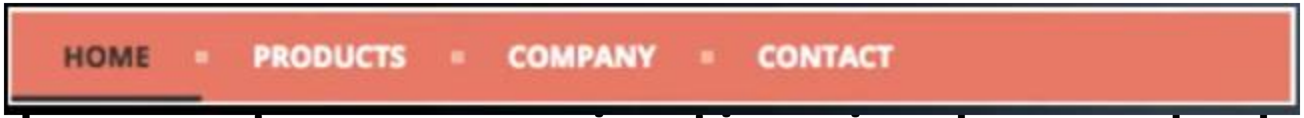
E-mail: enquiry@dypimca.ac.in
Director E-mail: director@dypimca.ac.in

Copyright © 2015 Dr. D.Y. Patil Educational Complex, Akurdi, Pune

Best viewed in IE 10+, Firefox 20+, Chrome , Safari5+, Opera12+

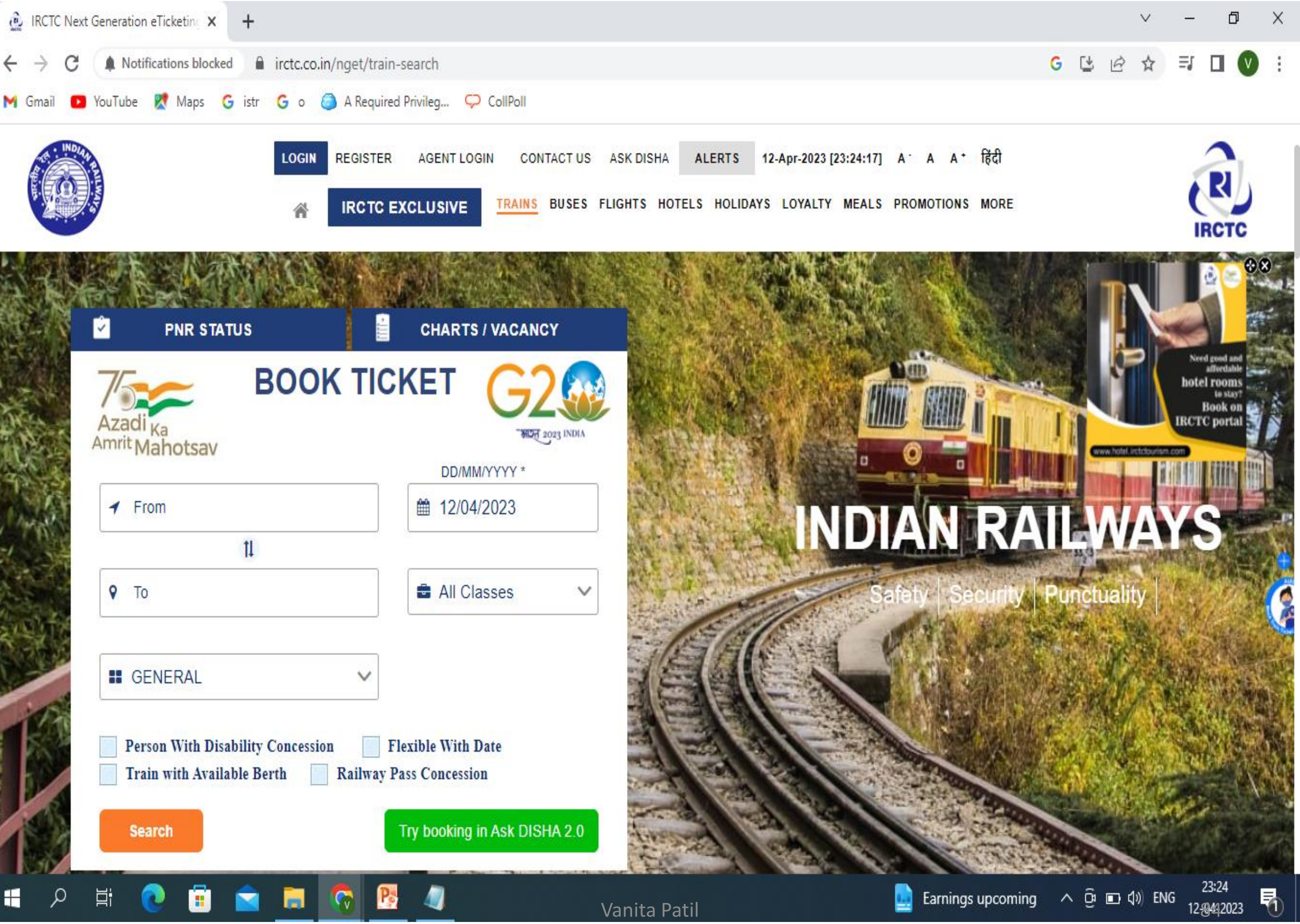
:::| powered by dimakh consultants |:::

- **6. <NAV> Tag:**
- <nav> tag defines set of navigation links.



- In most case navigation is shown below the header.
- The <nav> tag defines a set of navigation links.
- Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major blocks of navigation links.
- **Syntax: <nav> Links </nav>**

- Example:
- <nav>
- <a href=“index.html”>Home</a>
- <a href=“product.html”>Products</a>
- <a href=“company.html”>Company</a>
- <a href=“contact.html”>Contac</a>
- </nav>



BOOK TICKET

From

DD/MM/YYYY *

12/04/2023

To

All Classes

GENERAL

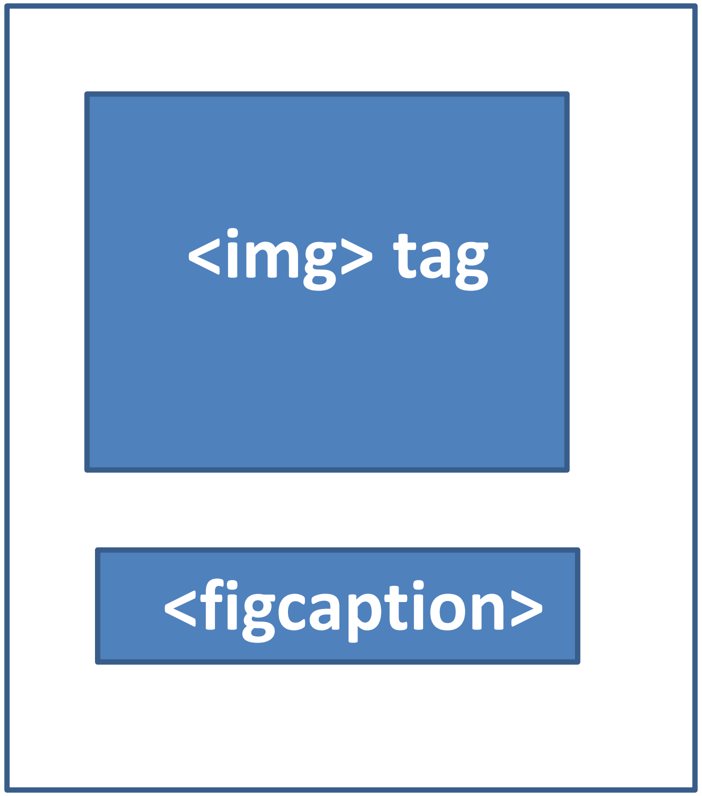
□ Person With Disability Concession □ Flexible With Date

□ Train with Available Berth □ Railway Pass Concession

Vanita Patil

- **7. Figure**
- **8. Figcaption**
- **Figure tag** Specifies **self-contained content**, like images, illustrations, diagrams, code listings, etc. Figure can have **figcaption** child to explain what **figure** is showing.
- Figcaption
- **Figcaption** is the caption of figure element.
- <figure>
- <img src="img/path" alt="...">
- <figcaption> caption for image</figcaption>
</figure>

<figure> tag



<img> tag

<figcaption>

- **9. Details**
- The <details> tag is often used to create an interactive widget that the user can open and close.
- By default, the widget is closed.
- When open, it expands, and displays the content within.
- The **summary tag** is used with the **details** tag for specifying visible heading.
- **Attribite:**
- **open**: Specifies that the details should be visible (open) to the user
- **Syntax:**
- <details>
- <summary> Text content </summary>
- <div> Content . . . >
- </details>

- **Example:**
- <html>
- <body>
- <h1>The summary element</h1>
- <details>
- <summary>HTML</summary>
- <p>HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.
  </p>
- </details>
- </body>
- </html>

- **10. main**
- The **<main>** element represents the dominant content of the <body> of a document.
- The content inside the <main> element should be unique to the document.
- It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.
- **Note:** There must not be more than one <main> element in a document.
  The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

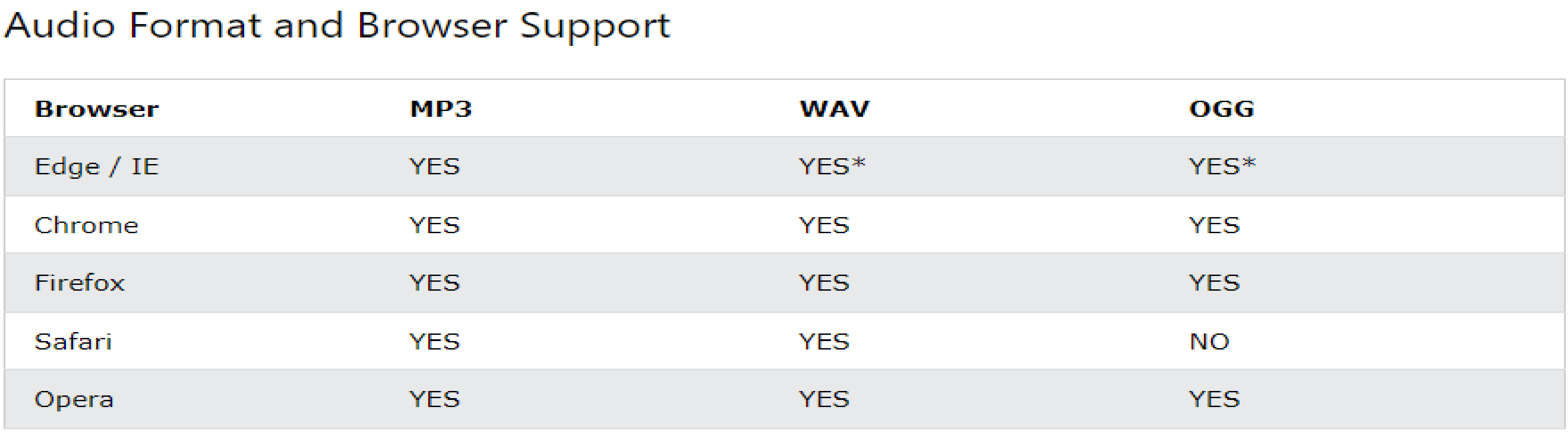
- **Example:**

```
•   <body>
•      <header> <h1>Welcome to our website</h1> </header>
•      <main>
•         <h1>Core Web Technologies</h1>
•         <p>HTML and CSS are two of the core technologies for building web pages.</p>
•         <article>
•            <h2>What is HTML?</h2>
•           <p>HTML stands for HyperText Markup Language. HTML is the standard markup
•                language  for describing the structure of web pages.</p>
•         </article>
•         <article>
•            <h2>What is CSS?</h2>
•            <p>CSS stands for Cascading Style Sheet. CSS allows you to specify various style
•                 properties  for HTML elements such as colors, backgrounds, fonts etc.</p>
•         </article>
•      </main>
•      <footer> <p>copyright &copy; tutorialrepublic.com</p></footer>
•   </body>
```

- **11. Mark**
- The <mark> element defines a marked section of text.
- You can use this tag if you want to highlight a section of your text for reference purposes.
- **Syntax:** <mark> ... </mark>
- **Example:**
- <html lang="en">
- <head><title>Example of HTML mark Tag</title></head>
- <body>
-         <p>This is some **<mark>**highlighted**</mark>** text.</p>
-         <p>Here are **<mark>**some more highlighted**</mark>** text.</p>
- </body>
- </html>

Audio Tag

- The <audio> tag is used to embed sound content in a document, such as music or other audio streams.
- The <audio> tag contains one or more <source> tags with different audio sources.
- There are three supported audio formats in HTML: MP3, WAV, and OGG.



Audio Format and Browser Support

Browser	MP3	WAV	OGG
Edge / IE	YES	YES*	YES*
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

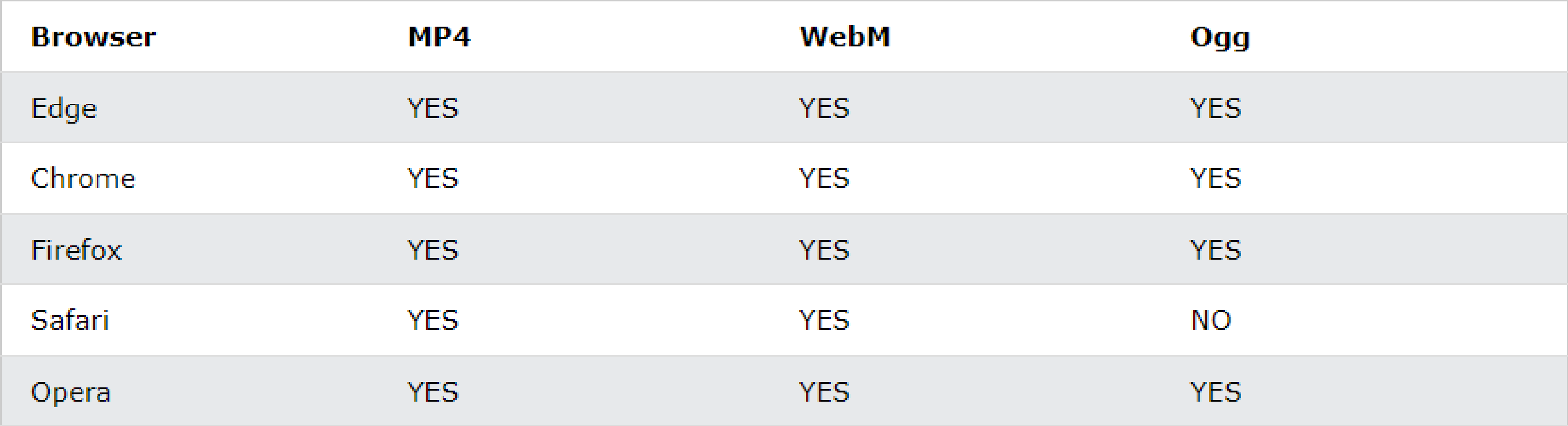
- Attributes:

Attribute	Value	Description
autoplay	autoplay	Specifies that the audio will start playing as soon as it is ready
controls	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
loop	loop	Specifies that the audio will start over again, every time it is finished
muted	muted	Specifies that the audio output should be muted
src	*URL*	Specifies the URL of the audio file

- **Example:**
- <audio controls autoplay>
- <source src = "/html5/audio.ogg" type = "audio/ogg" />
- <source src = "/html5/audio.wav" type = "audio/wav" />
- </audio>

Video Tag

- The <video> tag is used to embed video content in a document, such as a movie clip or other video streams.
- The <video> tag contains one or more <source> tags with different video sources. The browser will choose the first source it supports.
- The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.
- There are three supported video formats in HTML: MP4, WebM, and OGG.



Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Attribute:

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	*pixels*	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
src	*URL*	Specifies the URL of the video file
width	*pixels*	Sets the width of the video player

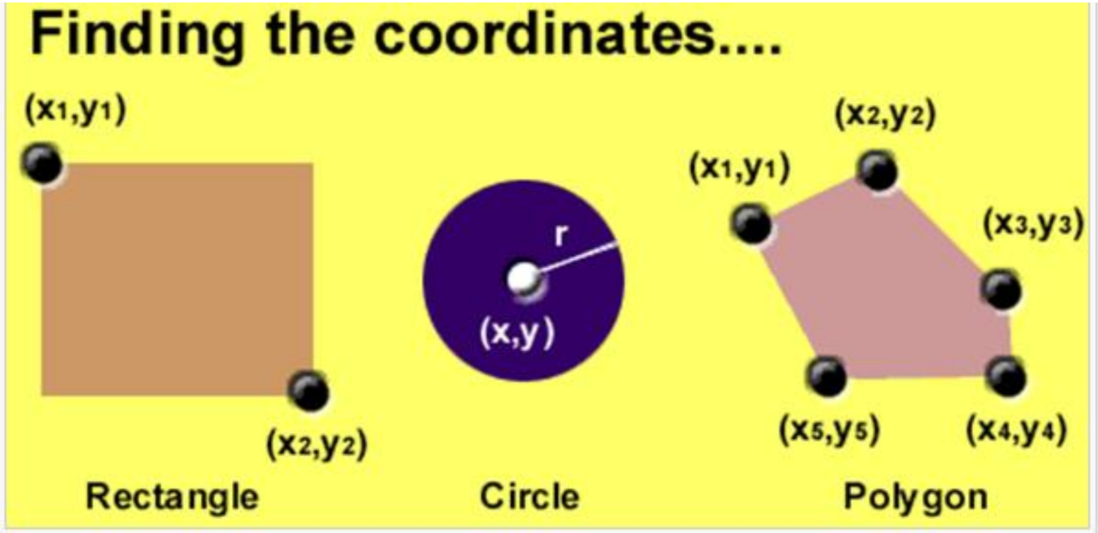
- Example:
- <video width = "300" height = "200" controls autoplay>
- <source src = "/html5/foo.ogg" type ="video/ogg" />
- <source src = "/html5/foo.mp4" type = "video/mp4" />
- Your browser does not support the <video> element. </video>

<map> tag

- The <map> tag specifies a client-side image map.
- Using HTML map tags, you can create a **clickable map**.
- By clicking the HTML map area, the user will open **links** provided.
- The clickable places are defined using <area> elements.
- **How to Create a Clickable Map**
- The map tags define an HTML **image map**. It is an image with clickable areas associated with **links**

- The required **name** attribute of the <map> element is associated with the <img>'s **usemap** attribute and creates a relationship between the image and the map.
- The <map> element contains a number of <area> elements, that defines the clickable areas in the image map.
- Syntax: **<img usemap="#mapname">**
- **Attribute: mapname:** It is used to hold the map name containing hash (#) character.

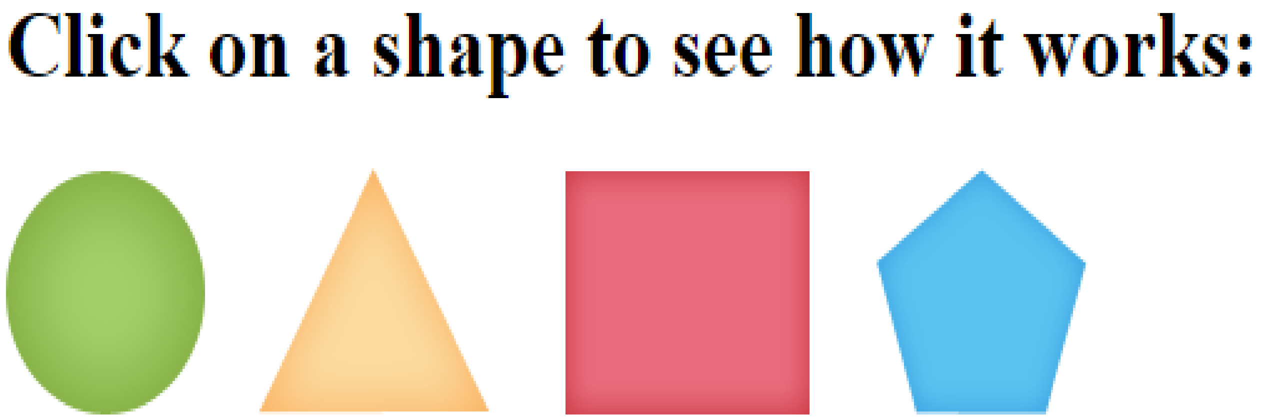
- If you wish to create a clickable image map, HTML <map> tags should also include a <name> attribute. It is required to properly link the **map** with the **image**:



- **<area> tag**
- The <area> is used to define an area inside an image and create a clickable link. It uses attributes to define shape, coordinates, URL, etc.
- **shape** - It defines shapes like circles, rectangles, polygons, etc
- **coords** - It specifies coordinates of the shape in x1,y1,x2,y2... pattern
- **href** - It provides the URL of the hyperlink for the target area
- **target**- Specifies the context in which to open the linked resource.

- **Example:**
- <img src="shapes.png" usemap="#shapes" alt="Geometrical Shapes">
- <map name="shapes">
- <area shape="circle" coords="40,40,40" href="circle.html" alt="Circle">
- <area shape="poly" coords="148,2,104,80,193,80" href="triangle.html" alt="Triangle">
- <area shape="rect" coords="226,2,323,80" href="rectangle.html" alt="Rectangle">
- <area shape="poly" coords="392,2,352,32,366,80,418,80,432,32" href="pentagon.html" alt="Pentagon">
- </map>

- Output:



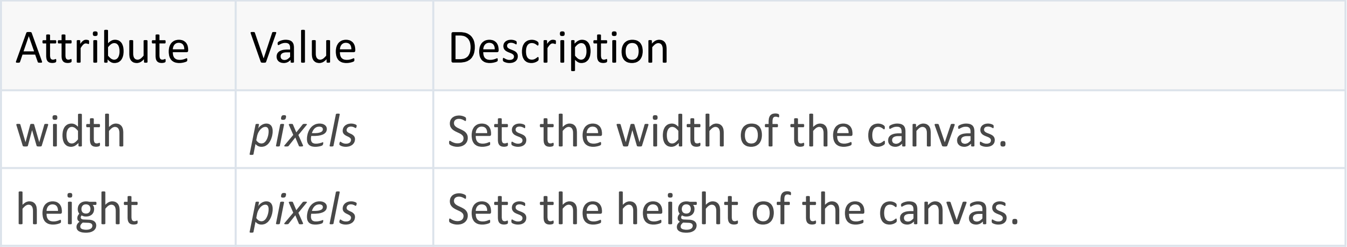
Click on a shape to see how it works:

Canvas

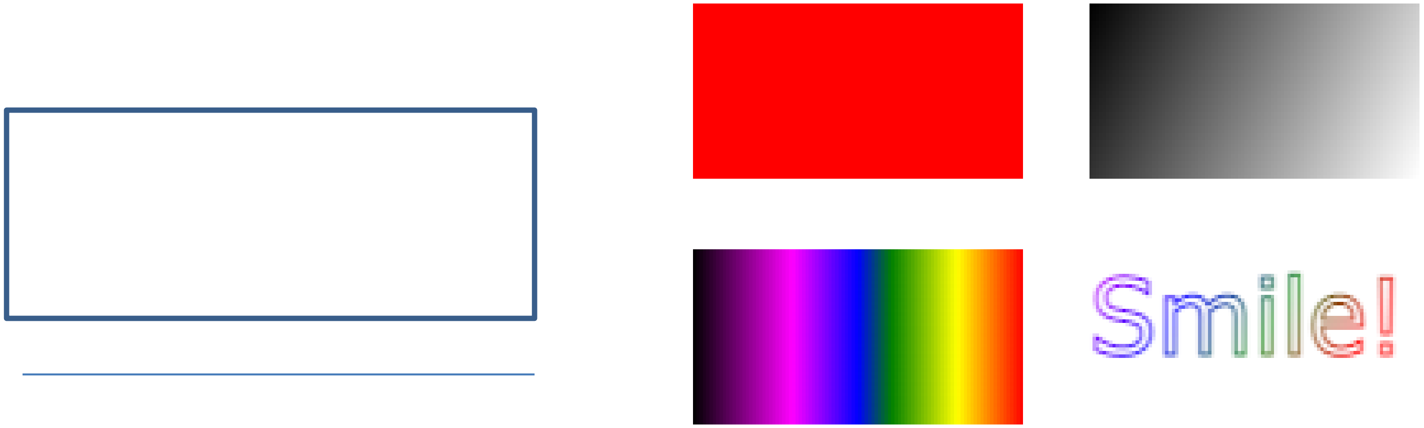
- HTML5 element <canvas> gives you an easy and powerful way to draw graphics using JavaScript.
- HTML5 Canvas API includes a 2D context that allows a programmer to draw various shapes, render text, and display images directly onto a defined area of the browser window.
- Attributes:

Attribute	Value	Description
height	*pixels*	Specifies the height of the canvas. Default value is 150
width	*pixels*	Specifies the width of the canvas Default value is 300

- A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.
- Syntax:
  <canvas id="myCanvas" width="200" height="100"></canvas>
- Attributes:



Attribute	Value	Description
width	*pixels*	Sets the width of the canvas.
height	*pixels*	Sets the height of the canvas.



- Canvas Default Coordinate:The default size of the canvas is 300 * 150 px without any border or content.
- Using Canvas you can draw –
- Rectangle
- Line
- Circle
- Quadratic curve or Arc
- Polygon
- Gradient

- The JavaScript should be used to draw on the canvas. We are going to do with step by step.
- **1. Find the Canvas Element**
- To find the canvas element use the HTML DOM method: getElementById():
- var canvas=document.*getElementById*("canvas");
- **2. Create a drawing object**
- Use the getContext() HTML object, with properties and methods:
- var ctx = canvas.*getContext*("2d");
- **3. Draw on the Canvas Element**
- Use the fillStyle property can be a CSS color, a pattern, or a gradient.
- ctx.*fillStyle* = "#1c87c9";

- **1. Draw rectangle in canvas:**
- The rect() method creates a rectangle.
- **Parameters:**
- X-The x-coordinate of the upper-left corner of the rectangle
- Y-The y-coordinate of the upper-left corner of the rectangle
- width -The width of the rectangle, in pixels
- Height-The height of the rectangle, in pixels
- **Example:**
- <body>
- <canvas id="canvas" width="350px" height="250px" style="border: 2px solid blue;"></canvas>
- <script>
- var getId = document.getElementById('canvas');
- var ctx = getId.getContext('2d');
- ctx.fillStyle = 'grey';
- ctx.fillRect(25,25,300,200);
- </script>
- </body>

- **Example 2:**

```
<html>
<body>
<canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;">
<script>
   var c = document.getElementById("myCanvas");
   var ctx = c.getContext("2d");
  ctx.beginPath();
  ctx.lineWidth = "6";
  ctx.strokeStyle = "red";
  ctx.rect(5, 5, 290, 140);
  ctx.stroke();
</script>
</body>
</html>
```

- **Drawing a Line**
- The methods below are used to draw a straight line on a canvas:
- moveTo(x,y), which specifies the starting point of the line
- lineTo(x,y), which specifies the ending point of the line

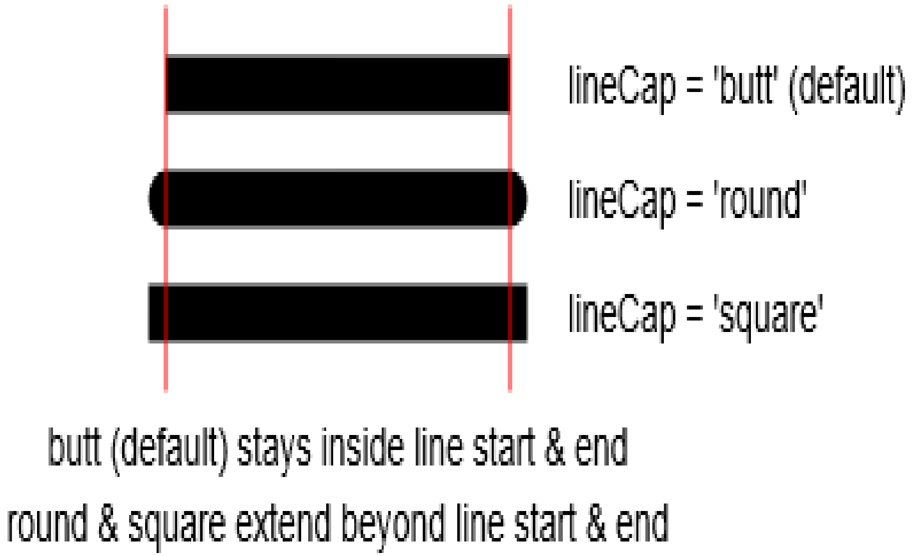
- **Example:**

```
<html lang="en">
<head>
<title>Example of HTML canvas Tag</title>

</head>
<body>
   <canvas id="myCanvas" width="300" height="200" style="border: 1px solid #000;"></canvas>
        <script type="text/javascript">

                                   var canvas = document.getElementById("myCanvas");
                                   var context = canvas.getContext("2d");
                                   context.moveTo(50, 150);
                                   context.lineTo(250, 50);
                                   context.stroke();

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

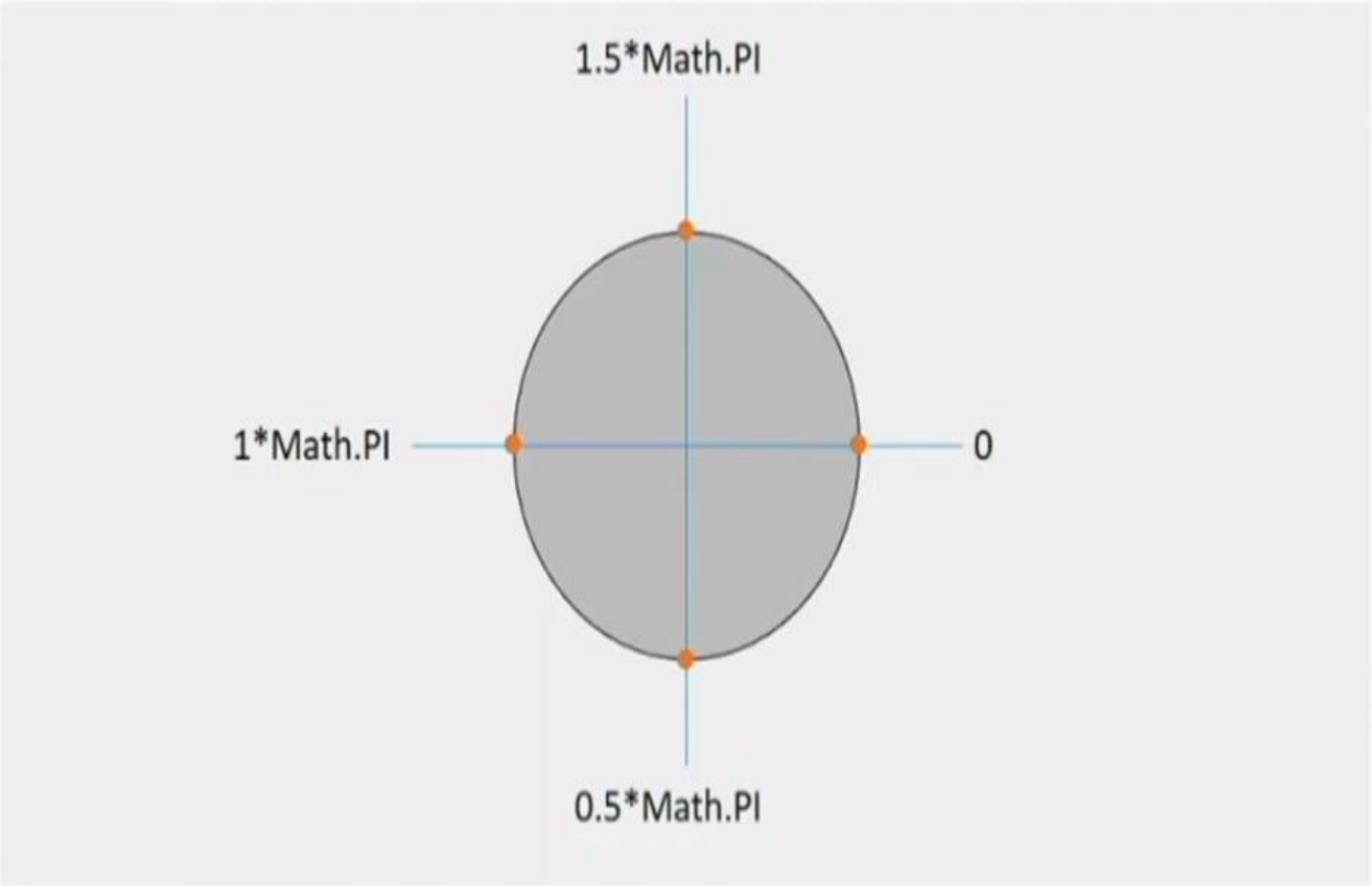
- **HTML5 Canvas Line Cap**
- To add a cap to an HTML5 Canvas line, we can use the lineCap property.
- Lines can have one of three cap styles: butt, round, or square.
- Unless otherwise specified, HTML5 Canvas lines are defaulted with the butt cap style.
- The lineCap property must be set before calling stroke().

- Sets the cap style of line starting points and ending points.
- **butt**, the default lineCap style, shows squared caps that do not extend beyond the line's starting and ending points.
- **round**, shows rounded caps that extend beyond the line's starting and ending points.
- **square**, shows squared caps that extend beyond the line's starting and ending points.

- Example:
- <script>
-   context.lineCap = 'round
- </script>


lineCap = 'butt' (default)
lineCap = 'round'
lineCap = 'square'
butt (default) stays inside line start & end
round & square extend beyond line start & end

- **Drawing a Circle**
- The methods below are used to draw a circle on a canvas:
- beginPath(), which begins a path
- arc(x,y,r,startangle,endangle), which creates an arc/curve. If you want to create a circle with arc(): set start angle to 0 and end angle to 2*Math.PI. The x and y specify the x- and y-coordinates of the circle’s center. The r parameter specifies the radius of the circle.


1.5*Math.PI
1*Math.PI
0
0.5*Math.PI

- Example:
- <!DOCTYPE html>
- <html>
- <head> <title>Title of the document</title> </head>
- <body>
- <canvas id="exampleCanvas" width="250" height="200" style="border:1px solid #dddddd;">
- </canvas>
- <script>
- var canvas = document.getElementById("exampleCanvas");
- var ctx = canvas.getContext("2d");
- ctx.beginPath();
- ctx.arc(125, 95, 70, 0, 2 * Math.PI);
- ctx.fillStyle = "#FB8B89";
- ctx.fill();
- ctx.lineWidth = 5;
- ctx.strokeStyle = "black";
- ctx.stroke();
- </script>
- </body>
- </html>

- **Canvas Text**
- HTML5 canvas allows creating text using different font and text properties presented below:
- **Properties and Methods**
-

Property	Description
font	It returns the current font settings and can be set to change the font.
textAlign	It returns the current text alignment settings and can be set to change the alignment. The property has the following values: start, end, left, right, and center.
textBaseline	It returns the current baseline alignment settings and can be set to change the baseline alignment. The property has the following values: top, hanging, middle, alphabetic, ideographic, and bottom.
fillText(text, x, y)	It draws a filled text at the position indicated by the given coordinates x and y.
strokeText(text, x, y)	It strokes the text at the position indicated by the given coordinates x and y.

- **Example:**
- <!DOCTYPE html>
- <html>
- <body>
- <canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;"></canvas>
- <script>
- var c = document.getElementById("myCanvas");
- var ctx = c.getContext("2d");
- ctx.font = "30px Arial";
- ctx.strokeText("Hello World",10,50);
- </script>
- </body>
- </html>

Gradients

- Gradients consist of continuously smooth color transitions along a vector from one color to another, possibly followed by additional transitions along the same vector to other colors.
- HTML5 canvas (2D) supports two kinds of gradient : linear and radial.

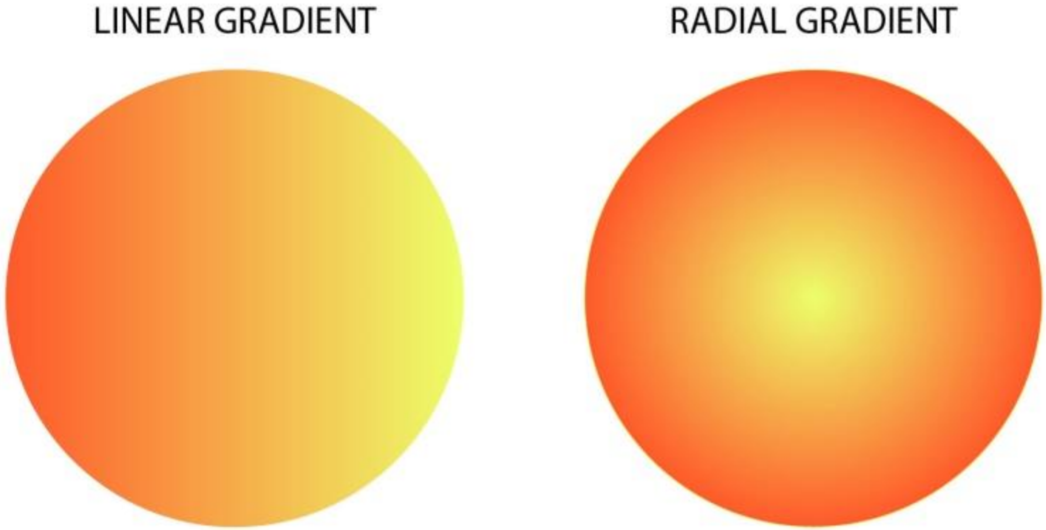
- **1. Linear gradient:**
- A linear gradient defines color change along a line between two points. You can create a linear gradient by calling createLinearGradient() method.
- Syntax:
- var gradient = ctx.createLinearGradient(x0, y0, x1, y1);

Parameters	Type	Description
x0	number	The x-coordinate (in pixels), of the start point of the gradient.
y0	number	The y-coordinate (in pixels), of the start point of the gradient.
x1	number	The x-coordinate (in pixels), of the end point of the gradient.
y1	number	The y-coordinate (in pixels), of the end point of the gradient.

- The next step in defining a gradient is to add at least two color stops. To do this use addColorStop() method. Here is the syntax and parameters of the method :
- Syntax: gradient.addColorStop(offset, color);

Parameters	Type	Description
offset	number	A floating point value between 0.0 and 1.0 that represents the position where 0 is the beginning of the gradient and 1 is the end..
color	number	A CSS color string to display at the position that the offset parameter specifies.

- **Example:**
- <body>
- <canvas id="demoCanvas" width="500" height="600"></canvas>
- <script>
- var canvas = document.getElementById("demoCanvas");
- var ctx = canvas.getContext("2d");
- var gradient = ctx.createLinearGradient(10, 90, 200, 90);
- gradient.addColorStop(0, 'black');
- gradient.addColorStop(1, 'white');
- ctx.fillStyle = gradient;
- ctx.fillRect(10, 10, 200, 250);
- </script>
- </body>


LINEAR GRADIENT
RADIAL GRADIENT

- **2. Radial gradient:**
- A radial gradient defines a color change along a cone (a three-dimensional geometric shape) between two circles.
- You can create a radial gradient by calling createRadialGradient() method and passing in two circles and also providing source and destination circle radius.
- **Syntax:**
- var gradient = ctx.createRadialGradient(x0, y0, r0, x1, y1, r1)

Parameters	Type	Description
x0	number	The x-coordinate (in pixels), of the starting circle of the gradient.
y0	number	The y-coordinate (in pixels), of the starting circle of the gradient.
rad0	number	The radius of the starting circle.
x1	number	The x-coordinate (in pixels), of the ending circle of the gradient.
y1	number	The y-coordinate (in pixels), of the ending circle of the gradient.
rad1	number	The radius of the ending circle.

The next step is to add at least two color stops. To do this use addColorStop() method.

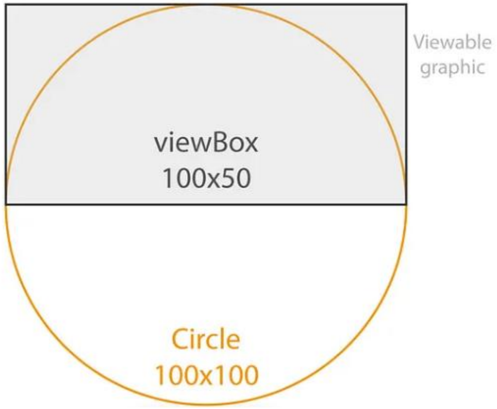
- **Example:**
- <body>
- <canvas id="demoCanvas" width="500" height="600"></canvas>
- <script>
- var canvas = document.getElementById("demoCanvas");
- var ctx = canvas.getContext("2d");
- var gradient = ctx.createRadialGradient(85, 88, 5, 88, 90, 69);
- gradient.addColorStop(0, 'white');
- gradient.addColorStop(1, 'black');
- ctx.fillStyle = gradient;
- ctx.arc(90, 90, 60, 0, 2 * Math.PI);
- ctx.fill();
- </script>
- </body>

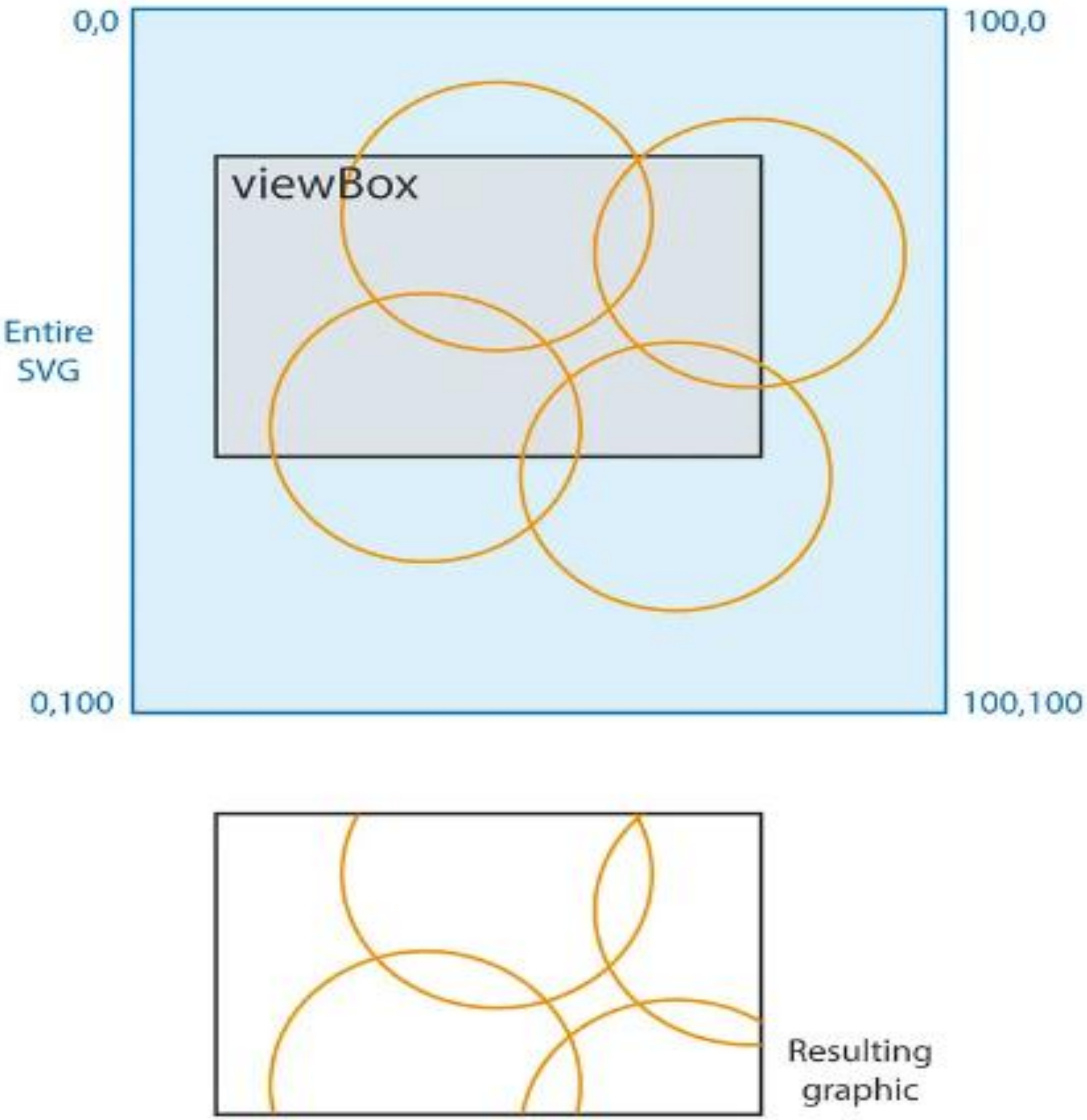
<SVG> tag

- SVG helps in creating 2 Dimensional vector graphics for the website.
- The full form of SVG is **S**calable **V**ector **G**raphics and it depicts the image or its related object in **XML** format which is like a special text format.
- Vector Graphics are made using **line**, **points,** and **arc** by using some mathematical formula.
- Images like PNG, GIF and JPEG uses **Raster Graphics**. Raster Graphic is also called as **Bitmap Image,** and they use pixels(set of dots), unlike Vector Graphics.

- **SVG Charateristics:**
- 1. SVG can be edited in any text editor.
- 2. SVGs are scalable.
- 3. SVG can be printed with high quality at any resolution.
- 4.SVGs are zoomable.
- SVG files are pure XML.
- SVG can be searched, indexed, scripted and compressed.

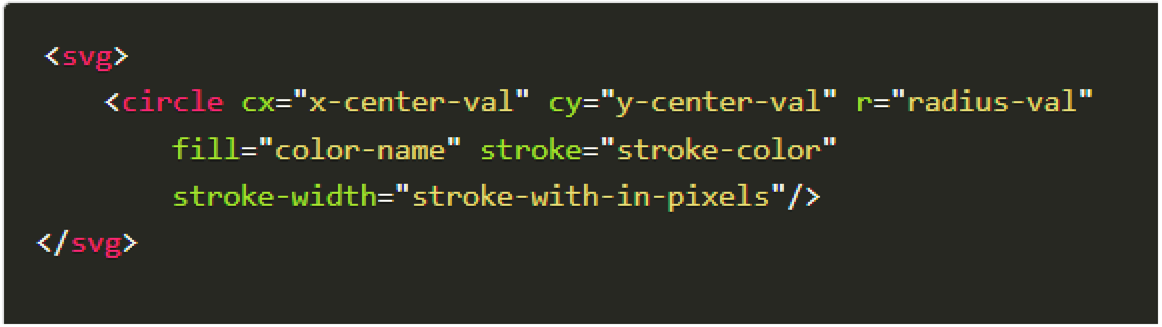
- Syntax: <svg> </svg>
- Example: <svg width="100px" height="100px" viewBox="0 0 100 100"></svg>
- width/height: The total width and height of the SVG element.
- viewBox: represents the viewable area of the SVG canvas .



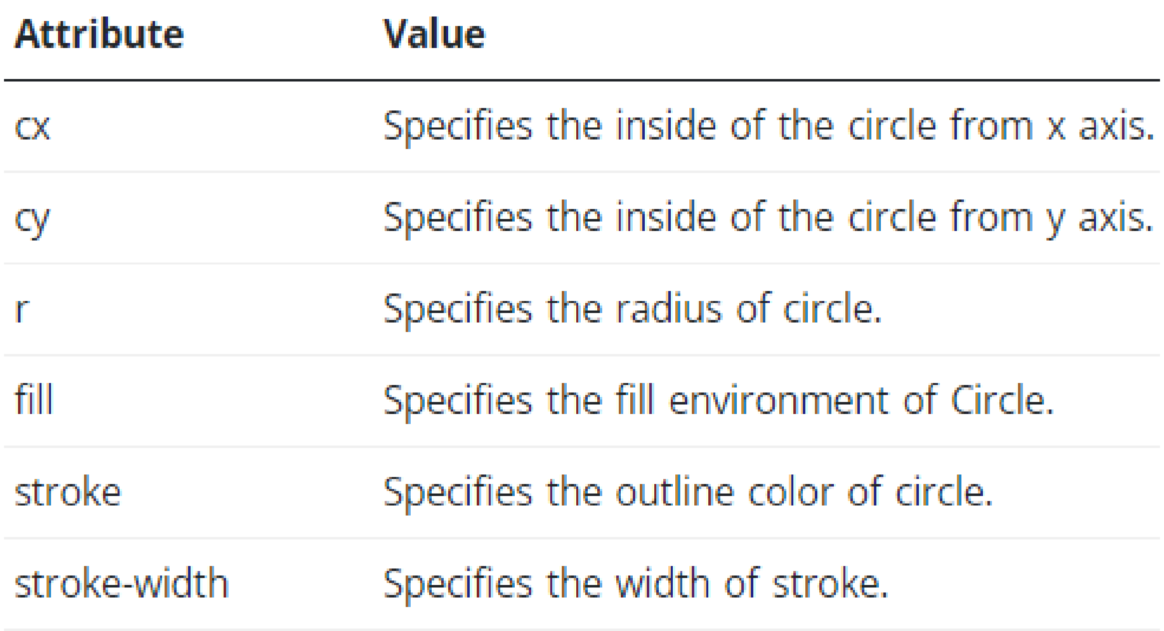

0,0
100,0
viewBox
Entire SVG
0,100
100,100
Resulting graphic

- Predefined shapes:
- Circle: <circle>
- Rectangle: <rect>
- Ellipse: <ellipse>
- Polyline: <polyline>
- Line: <line>
- Polygon: <polygon>
- Path: <path>

- **1. <circle> : Syntax:**

- 

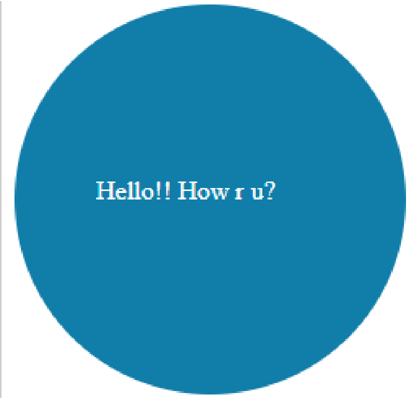
```
 <svg>
    <circle cx="x-center-val" cy="y-center-val" r="radius-val"
        fill="color-name" stroke="stroke-color"
        stroke-width="stroke-with-in-pixels"/>
</svg>
```



Attribute	Value
cx	Specifies the inside of the circle from x axis.
cy	Specifies the inside of the circle from y axis.
r	Specifies the radius of circle.
fill	Specifies the fill environment of Circle.
stroke	Specifies the outline color of circle.
stroke-width	Specifies the width of stroke.

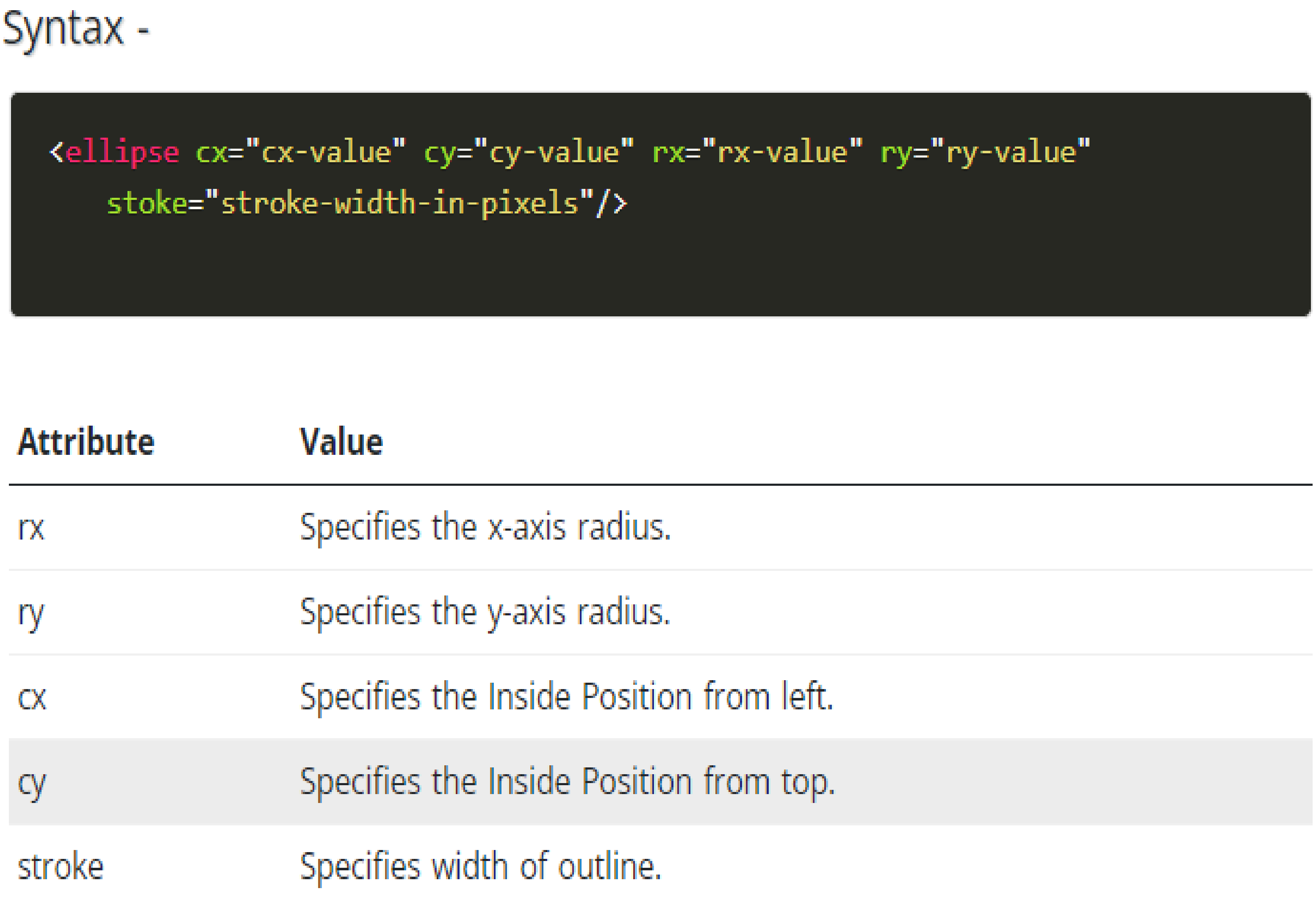
- **Example:**
- <html>
- <body>
- <svg width="100" height="100">
- <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />
- </svg>
- </body>
- </html>

- **Circle with Text:**
- <!DOCTYPE html>
- <html>
-   <body>
-     <h2>Example of SVG Circle with Text</h2>
-     <svg height="400" width="400">
-      <circle cx="120" cy="120" r="30%" fill="#117da9" /> <text x="50" y="120" fill="white">Hello! How r u?</text>
-     </svg>
-   </body>
- </html>



Hello!! How r u?

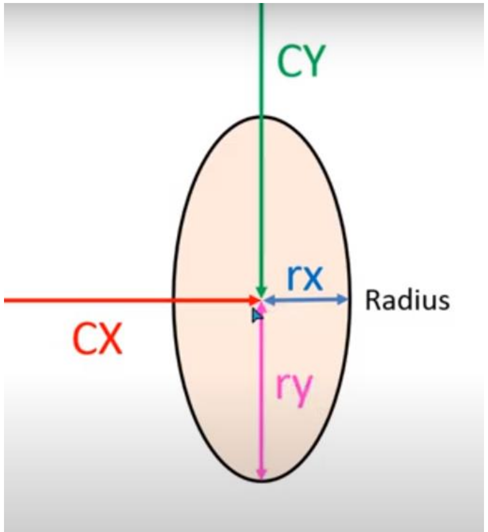
- **2.Ellipse:**



Syntax -

```
<ellipse cx="cx-value" cy="cy-value" rx="rx-value" ry="ry-value"
    stoke="stroke-width-in-pixels"/>
```

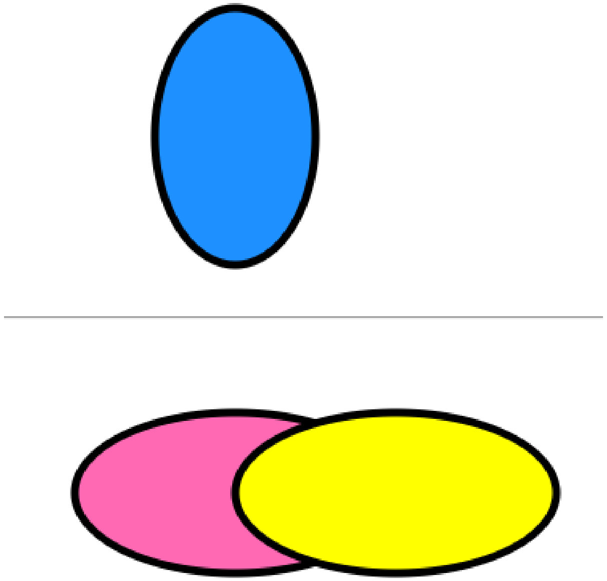
Attribute	Value
rx	Specifies the x-axis radius.
ry	Specifies the y-axis radius.
cx	Specifies the Inside Position from left.
cy	Specifies the Inside Position from top.
stroke	Specifies width of outline.



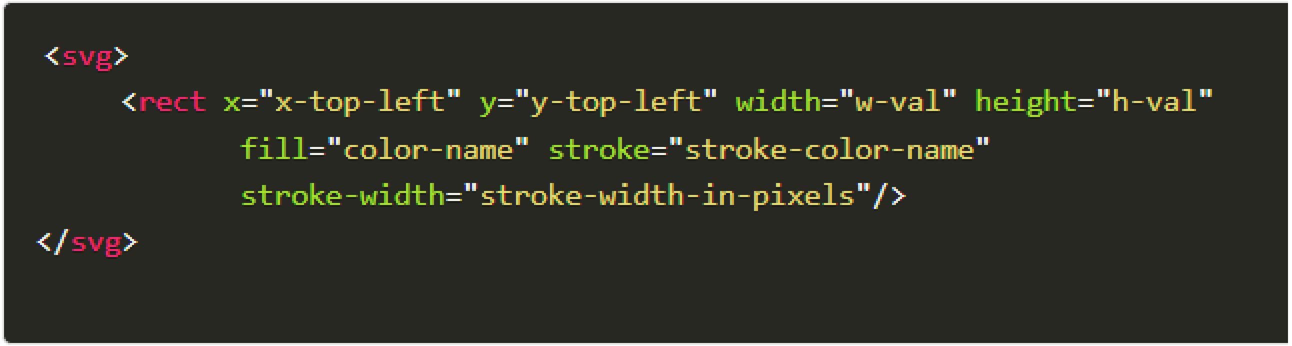
- Example:
- <svg width="400" height="150">
-   <ellipse cx="110" cy="70" rx="90" ry="50" fill="orange" />
- </svg>

- **Example:**

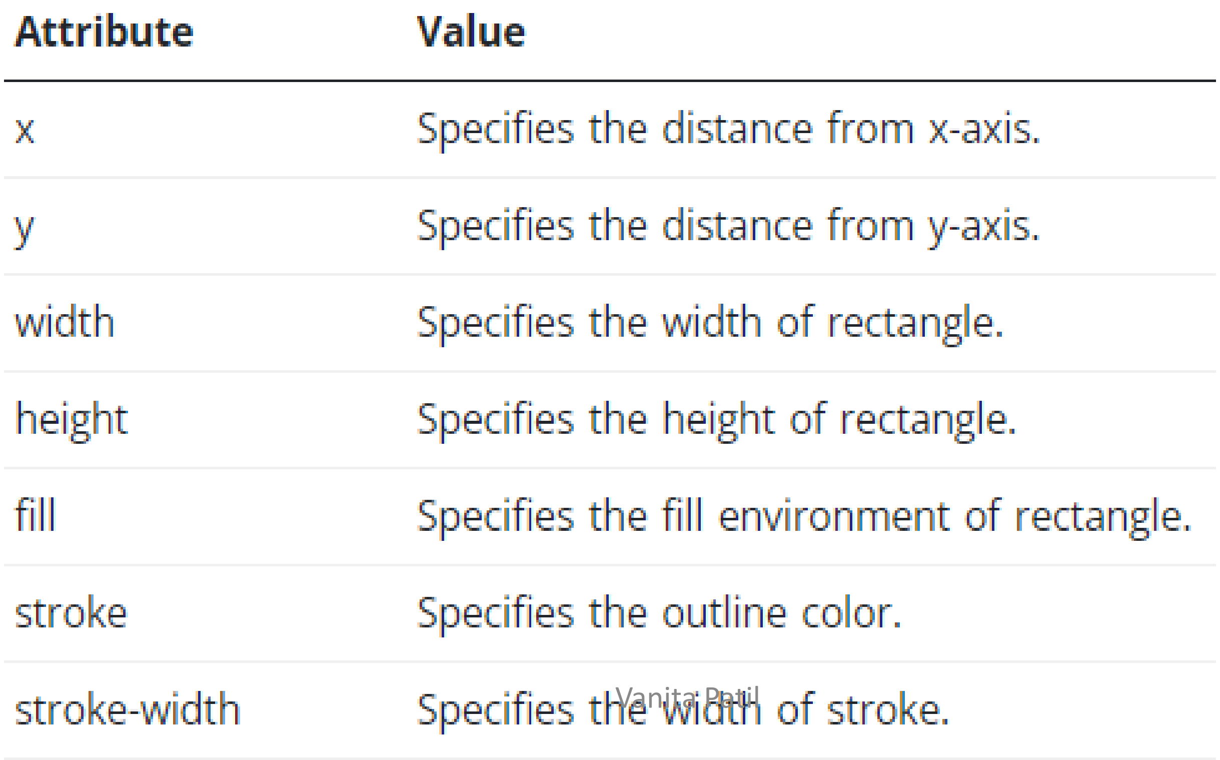
```
<html>
  <body>
    <h2>Example of SVG ELLIPSE</h2>
    <svg height="200" width="300">
      <ellipse cx="150" cy="100" rx="50" ry="80"
        fill="dodgerblue" stroke="black" stroke-width="5"/>
    </svg>
    <hr>
    <svg height="300" width="400">
      <ellipse cx="150" cy="100" rx="100" ry="50"
       fill="pink" stroke="black" stroke-width="5"/>

      <ellipse cx="250" cy="100" rx="100" ry="50"
       fill="yellow" stroke="black" stroke-width="5"/>
    </svg>
  </body>
</html>
```



- **3. <rect>: Syntax:**

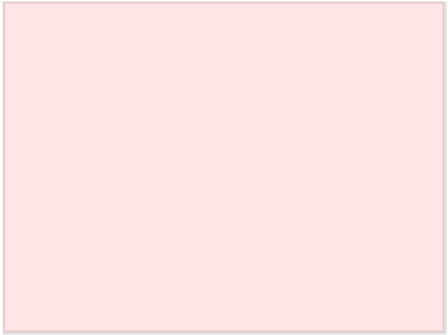


```
<svg>
    <rect x="x-top-left" y="y-top-left" width="w-val" height="h-val"
          fill="color-name" stroke="stroke-color-name"
          stroke-width="stroke-width-in-pixels"/>
</svg>
```

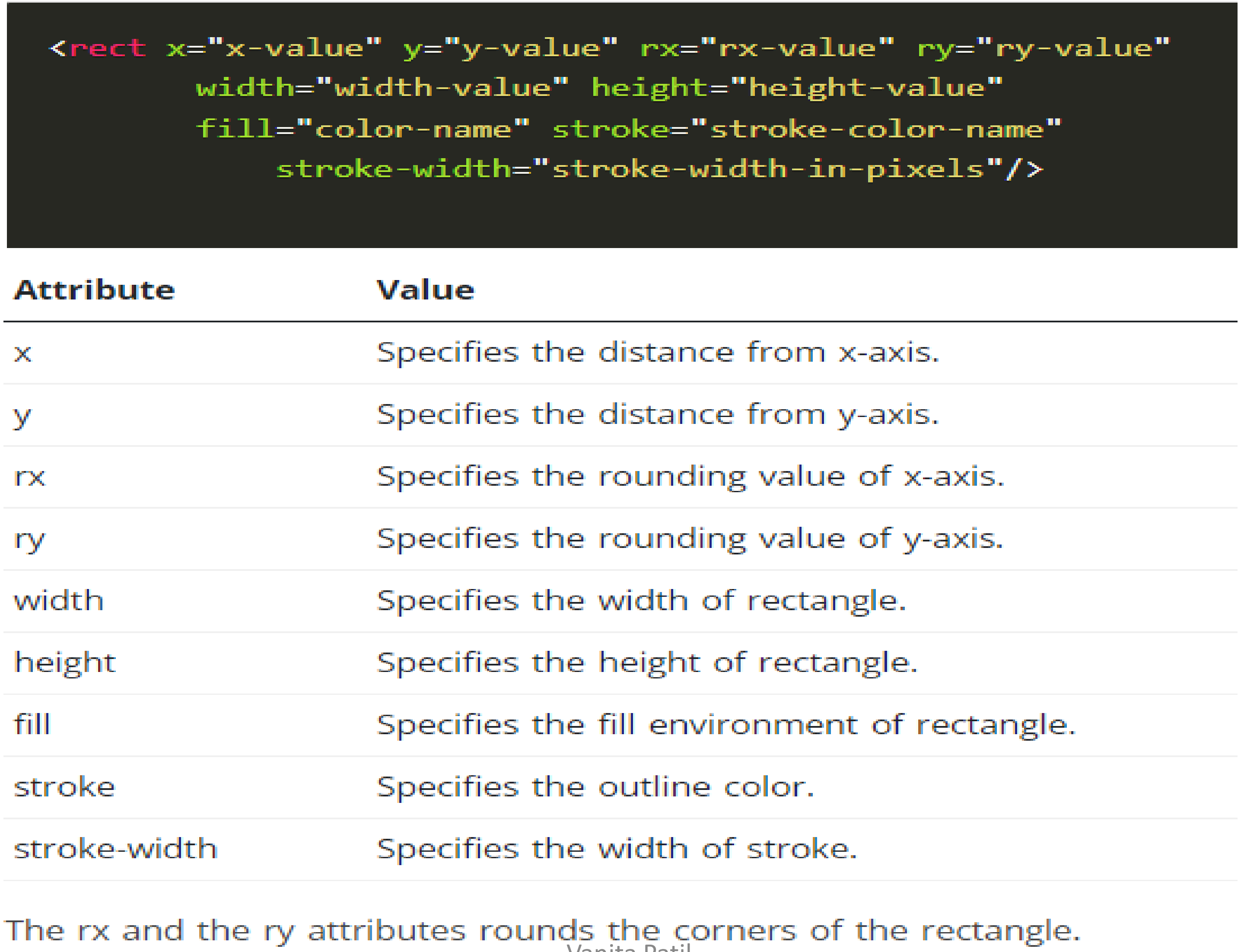


Attribute	Value
x	Specifies the distance from x-axis.
y	Specifies the distance from y-axis.
width	Specifies the width of rectangle.
height	Specifies the height of rectangle.
fill	Specifies the fill environment of rectangle.
stroke	Specifies the outline color.
stroke-width	Specifies the width of stroke.

- **Example:**
- <html>
- <body>
- <svg width="500" height="400">
- <rect width="400" height="300" stroke="black" stroke-width="4" fill="red" fill-opacity="0.1" stroke-opacity="0.1"/>
- </svg>
- </body>
- </html>



- **4. Rounded-Rect:** Syntax



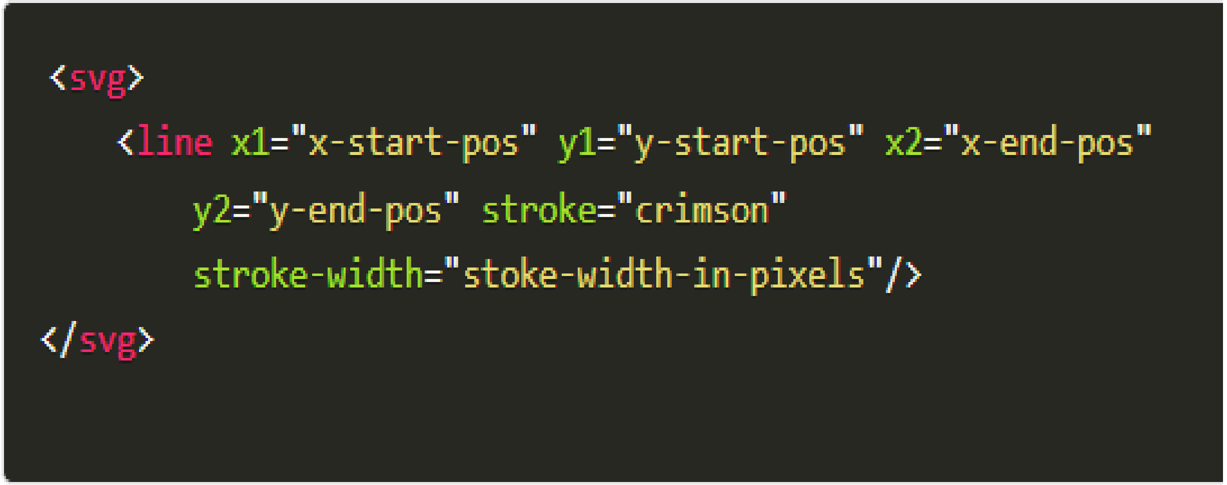
```
<rect x="x-value" y="y-value" rx="rx-value" ry="ry-value"
        width="width-value" height="height-value"
        fill="color-name" stroke="stroke-color-name"
            stroke-width="stroke-width-in-pixels"/>
```

Attribute	Value
x	Specifies the distance from x-axis.
y	Specifies the distance from y-axis.
rx	Specifies the rounding value of x-axis.
ry	Specifies the rounding value of y-axis.
width	Specifies the width of rectangle.
height	Specifies the height of rectangle.
fill	Specifies the fill environment of rectangle.
stroke	Specifies the outline color.
stroke-width	Specifies the width of stroke.

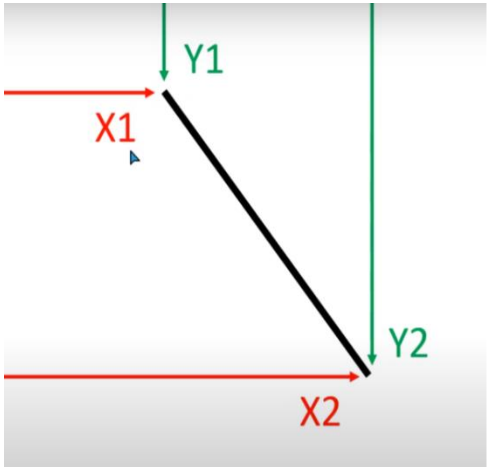
The rx and the ry attributes rounds the corners of the rectangle.

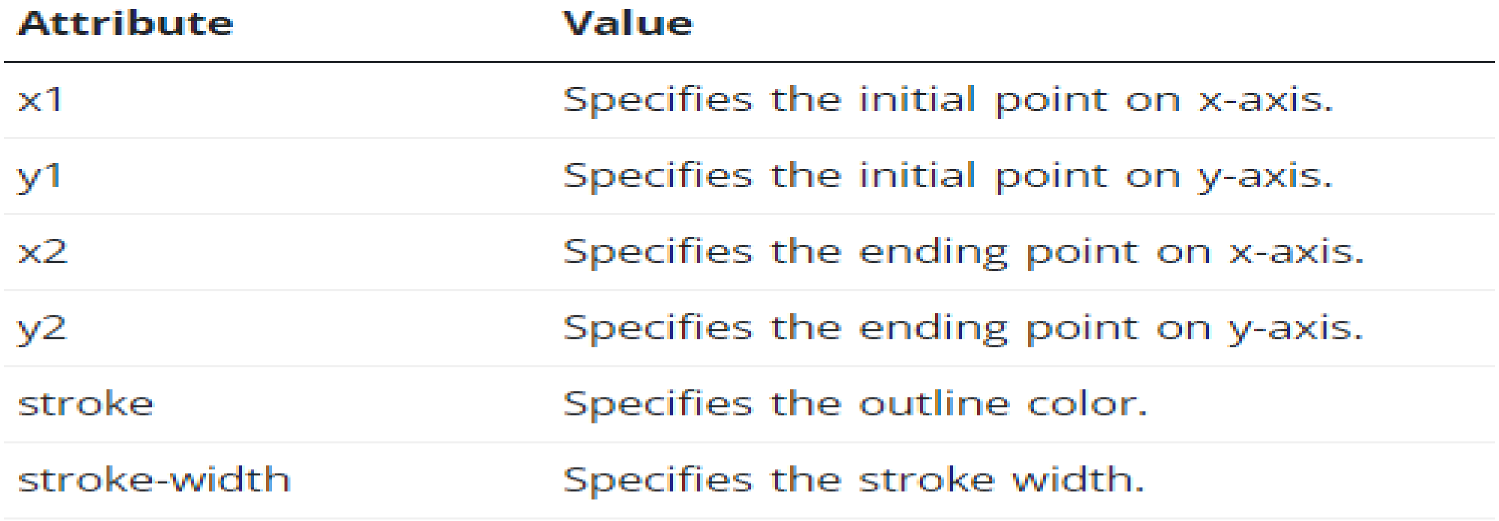
- Example:
- <html>
- <body>
- <h2>Example of SVG Rounded Rectangle</h2>
- <svg height="300" width="300">
- <rect x="20" y="20" width="200" height="100" rx="20" fill="purple" />
- </svg>
- </body>
- </html>

- **5.<line> : Syntax:**



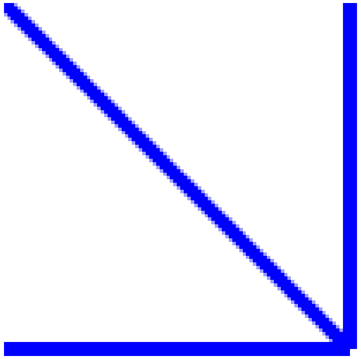
```
<svg>
   <line x1="x-start-pos" y1="y-start-pos" x2="x-end-pos"
       y2="y-end-pos" stroke="crimson"
       stroke-width="stoke-width-in-pixels"/>
</svg>
```



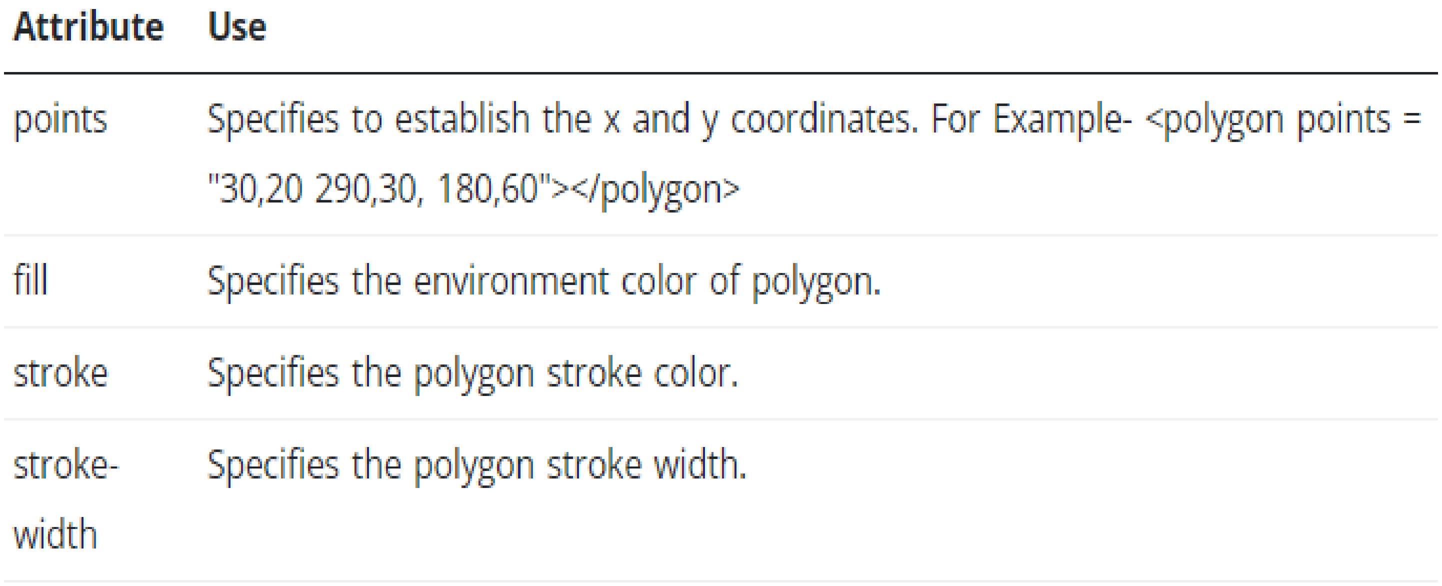


Attribute	Value
x1	Specifies the initial point on x-axis.
y1	Specifies the initial point on y-axis.
x2	Specifies the ending point on x-axis.
y2	Specifies the ending point on y-axis.
stroke	Specifies the outline color.
stroke-width	Specifies the stroke width.

- **Example:**
- <html>
- <body>
- <svg width="500" height="400">
- <line x1="0" y1="0" x2="100" y2="100" stroke="blue" stroke-width="4"/>
- <line x1="100" y1="0" x2="100" y2="100" stroke="blue" stroke-width="4"/> (horizontal line)
- <line x1="0" y1="100" x2="100" y2="100" stroke="blue" stroke-width="4"/> (vertical line>
- </svg>
- </body>
- </html>

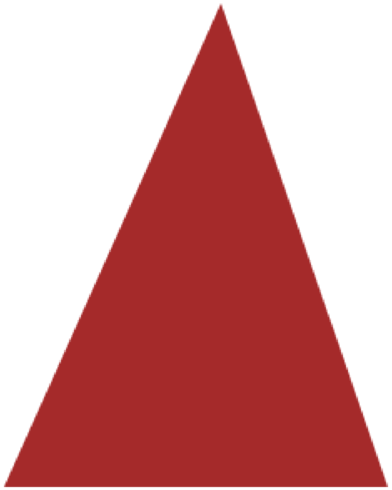


- **6. <polygon>:**
- Polygon should have at least 3 sides.
- Polygon are made up of straight lines and the shapes is “closed” (All the lines are connect up).
- Attribute:



Attribute	Use
points	Specifies to establish the x and y coordinates. For Example- <polygon points = "30,20 290,30, 180,60"></polygon>
fill	Specifies the environment color of polygon.
stroke	Specifies the polygon stroke color.
stroke-width	Specifies the polygon stroke width.

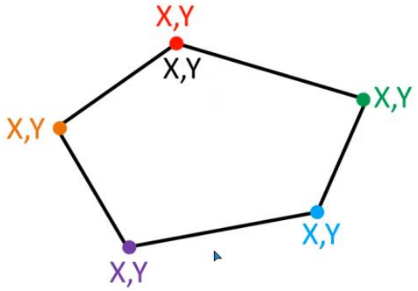
- **Example:(Triangle)**
- <html>
- <body>
- <h2>Example of SVG Triangle</h2>
- <svg height="400" width="400">
- <polygon points="250,60 120,350 350,350" fill="brown" />
- </svg>
- </body>
- </html>



- **Example: (Star)**

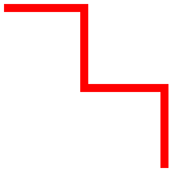
```
• <html>
•   <body>
•     <h2>Example of SVG Star</h2>
•     <svg height="300" width="300">
•      <polygon points="100,15 40,200 190,80, 10,80
•        160,200"  fill="navy" />
•     </svg>
•   </body>
• </html>
```

- **7.<polyline>:** The <polyline> element is used to create any shape that consists of only straight lines (that is connected at several points):
- Attributes:
- Points= x, y coordinates
- The points attribute defines the list of points (pairs of x and y coordinates) required to draw the polyline

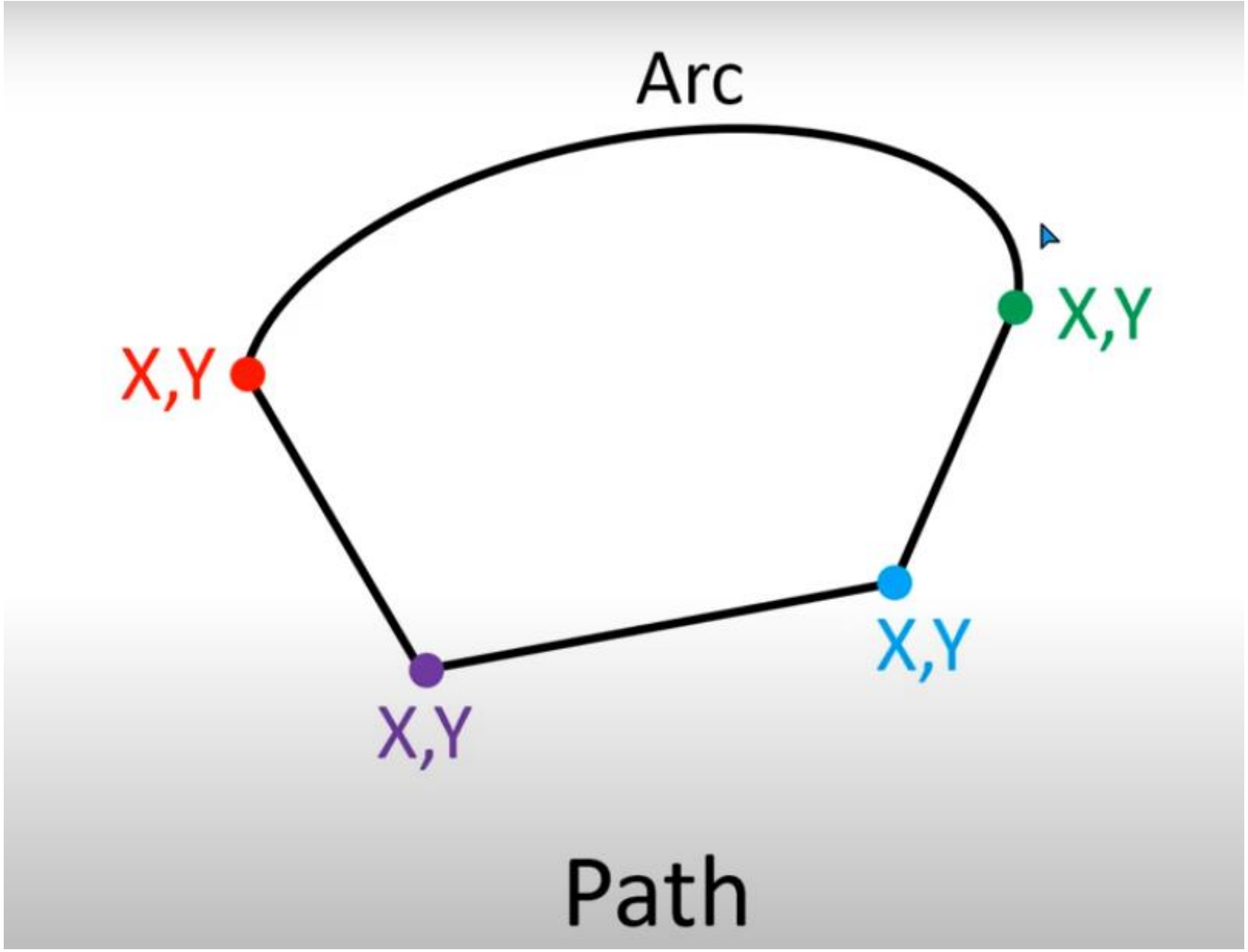


Polyline

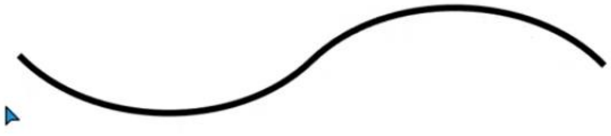
- **Example:**
- <html>
- <body>
- <svg width="300" height="300">
- <polyline points="0,40 40,40 40,80 80,80 80,120" fill="white" stroke="red" stroke-width="4">
- </svg>
- </body>
- </html>



- **8. <path>**
- The <path> element is used to define a path.
- **Attributes:**
- Points(d): x y coordinates
- **Commands available for path:**
- **M** — moves the cursor to a point in the SVG
- **v** — draws a vertical line (positive numbers go down, negative up)
- **h** — draws a horizontal line (positive numbers go right, negative number go left)
- **L** — draws a straight line from the current position to a specified coordinate
- **z** — draws a line from the current point to the starting point, effectively closing the shape
- **C** — cubic curve (**Curve:** this takes 3 coordinates in the form C x1,y1 x2,y2 x,y)

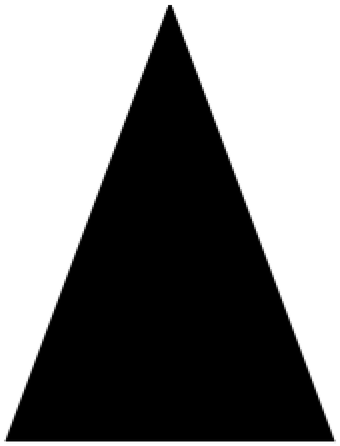

Arc
X,Y
X,Y
X,Y
X,Y
Path

- SVG Commands:
- 1. Line Command (L,H,V,Z)
- 2.Cubic Bezier Command (C,S)



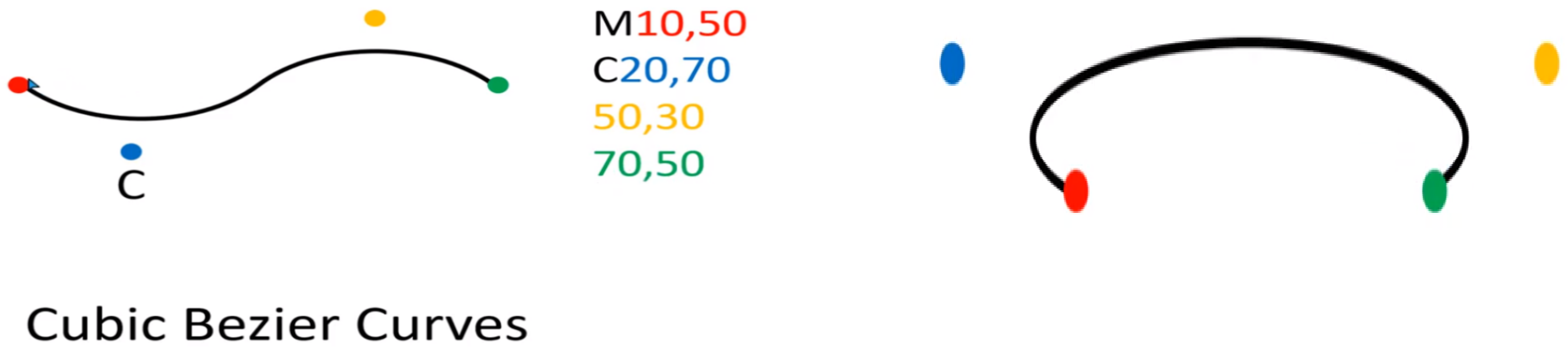
- 3.Quadratic Bezier Command (Q,T)
- 4.Arcs (A)

- **1. Line Command:**
- <html>
- <body>
- <svg height="210" width="400">
-   <path d="M150 0 L75 200 L225 200 Z" />
- </svg>
- </body>
- </html>



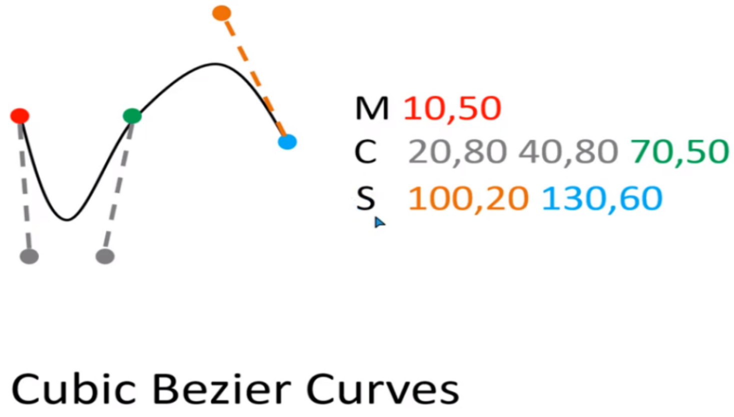
- Example:
- <html>
- <body>
- <svg width="500" height="400">
- <path d="M20,40 H80 V70 H20 Z" fill="none" stroke="blue">
- </svg>
- </body>
- </html>
- **O/P =?**

- 2. Cubic Bezier Curves



```
M10,50
C20,70
50,30
70,50
```

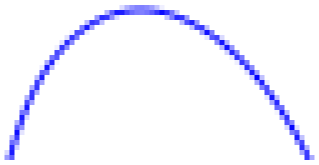
Cubic Bezier Curves



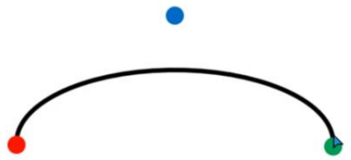
```
M 10,50
C  20,80 40,80 70,50
S  100,20 130,60
```

Cubic Bezier Curves

- Example:
- <html>
- <body>
- <svg width="500" height="400">
- <path d="M10,50 C 20,10 50,10,70,50" fill="none" stroke="blue">
- </svg>
- </body>
- </html>

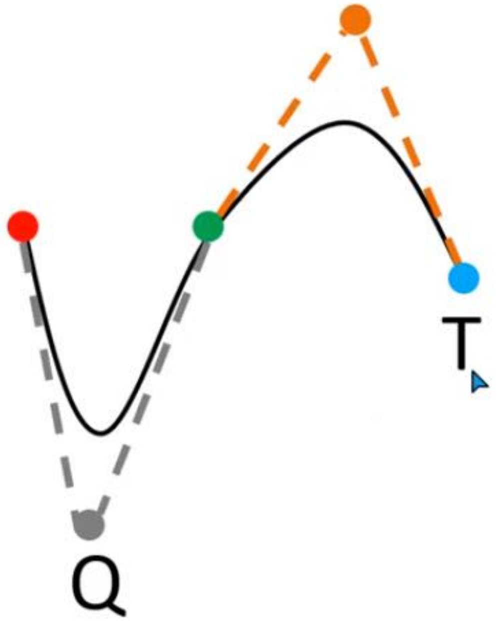


- 3. Quadratic Bezier Cureves

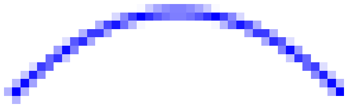


Quadratic Bezier Curves

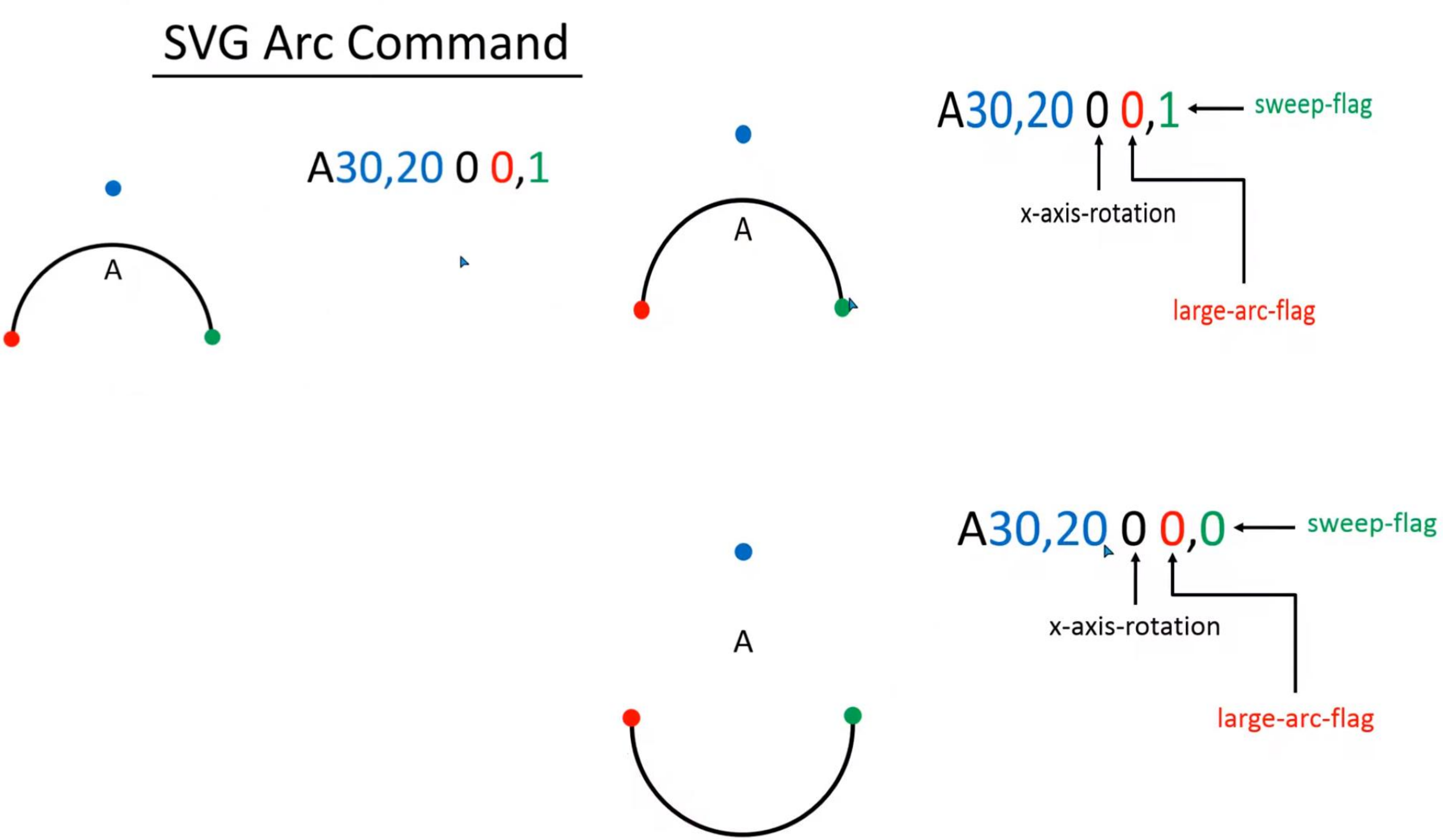
This point will assume



- **Example:**
- <html>
- <body>
- <svg width="500" height="400">
- <path d="M10,50 Q 30,30 50,50" fill="none" stroke="blue">
- </svg>
- </body>
- </html>



- Arc:



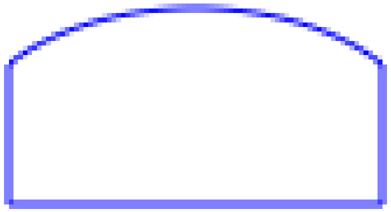
SVG Arc Command

```
A30,20 0 0,1
```

```
A30,20 0 0,1
```

```
A30,20 0 0,0
```

- **Example:**
- <html>
- <body>
- <svg width="500" height="400">
- <path d="M10,50 A 50,30 0 0,1 90,50 V80 H10 z " fill="none" stroke="blue">
- </svg>
- </body>
- </html>



- END