

[Type text]

# UNIT 1

## XHTML

### ORIGINS AND EVOLUTION OF HTML AND XHTML

HTML : Hyper Text Mark-up Language

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

### HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<a href="#">WHATWG HTML5 Living Standard</a>

[Type text]

2014	<a href="#">W3C Recommendation: HTML5</a>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<a href="#">W3C Recommendation: HTML5.1 2nd Edition</a>
2017	<a href="#">W3C Recommendation: HTML5.2</a>

## XHTML :eXtensible Hyper Text Mark-up Language

- XHTML stands for **EX**tensible **H**yper**T**ext **M**arkup **L**anguage
- XHTML is a stricter, more XML-based version of HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers
- XML is a markup language where all documents must be marked up correctly (be "well-formed").
- XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML). In addition, browsers ignore errors in HTML pages, and try to display the website even if it has some errors in the markup. So XHTML comes with a much stricter error handling.

## The Most Important Differences from HTML

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

HTML	XHTML
HTML is much easier to write	XHTML requires a level of discipline many of us naturally resist
huge number of HTML documents available on the Web, browsers will continue to support HTML as far as one can see into the future.	some older browsers have problems with some parts of XHTML.
HTML has few syntactic rules, and HTML processors (e.g., browsers) do not enforce the rules it does have. Therefore, HTML authors have a high degree of freedom to use their own syntactic preferences to create documents. Because of this freedom, HTML documents lack consistency, both in low-level syntax and in overall structure.	XHTML has strict syntactic rules that impose a consistent structure on all XHTML documents. Another significant reason for using XHTML is that when you create an XHTML document, its syntactic correctness can be checked, either by an XML browser or by a validation tool
Used for displaying the data	Used for describing the data

[Type text]

## BASIC SYNTAX

- The fundamental syntactic units of HTML are called **tags**.
- In general, tags are used to specify categories of content.
- The syntax of a tag is the tag's name surrounded by *angle brackets* (< and >).
- Tag names must be written in all lowercase letters.
- Most tags appear in pairs: an *opening tag* and a *closing tag*.
- The name of a closing tag is the name of its corresponding opening tag with a slash attached to the beginning. For example, if the tag's name is p, the corresponding closing tag is named /p.
- Whatever appears between a tag and its closing tag is the **content** of the tag. Not all tags can have content.
- The opening tag and its closing tag together specify a container for the content they enclose.
- The container and its content together are called an **element**.
- Example: <p> This is DR.AIT Web Programming Notes. </p>
- The paragraph tag, <p>, marks the beginning of the content; the </p> tag marks the end of the content of the paragraph element.
- Attributes, which are used to specify alternative meanings of a tag, can appear between an opening tag's name and its right angle bracket.
- They are specified in keyword form, which means that the attribute's name is followed by an equal's sign and the attribute's value.
- Attribute names, like tag names, are written in lowercase letters.
- Attribute values must be delimited by double quotes.
- Comments in programs increase the readability of those programs. Comments in XHTML have the same purpose. They can appear in XHTML in the following form:  
**<!-- anything except two adjacent dashes -->**
- Browsers ignore XHTML comments—they are for people only. Comments can be spread over as many lines as are needed. For example, you could have the following comment:  
**<!-- Welcome to Web Technologies -->**

## Standard XHTML Document Structure

- Every XHTML document must begin with an xml declaration element that simply identifies the document as being one based on XML. This element includes an attribute that specifies the version number 1.0.
- The xml declaration usually includes a second attribute, encoding, which specifies the encoding used for the document [utf-8].
- Following is the xml declaration element, which should be the first line of every XHTML document:  
**<?xml version = "1.0" encoding = "utf-8"?>**
- Note that this declaration must begin in the first character position of the document file.
- The xml declaration element is followed immediately by an SGML DOCTYPE command, which specifies the particular SGML document-type definition (DTD) with which the document complies, among other things.
- The following command states that the document in which it is included complies with the XHTML 1.0 Strict standard:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- An XHTML document must include the four tags <html>, <head>, <title>, and <body>.
- The <html> tag identifies the root element of the document. So, XHTML documents always have an <html> tag immediately following the DOCTYPE command, and they always end with the closing html tag, </html>.
- The html element includes an attribute, xmlns, that specifies the XHTML namespace, as shown in the following element:

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

- Although the xmlns attribute's value looks like a URL, it does not specify a document. It is just a name that

[Type text]

- happens to have the form of a URL.
- An XHTML document consists of two parts, named the *head* and the *body*.
  - The `<head>` element contains the head part of the document, which provides information about the document and does not provide the content of the document.
  - The body of a document provides the content of the document.
  - The content of the title element is displayed by the browser at the top of its display window, usually in the browser window's title bar.

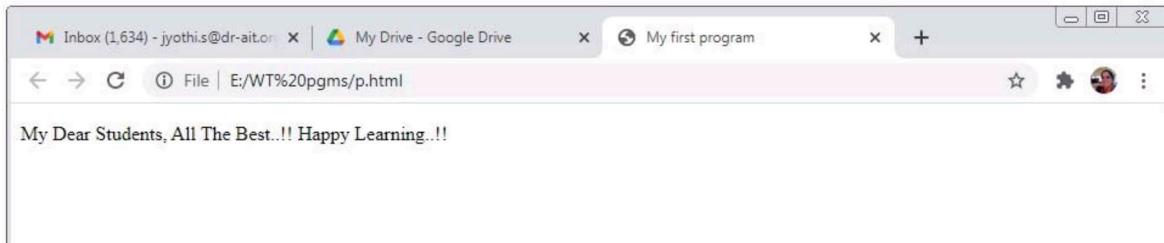
## **BASIC TEXT MARKUP**

We will have a look at a complete XHTML document:

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- complete.html
```

A document which must be followed throughout the notes

```
-->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
    <title> My first program </title>
</head>
<body>
    <p>
        My Dear Students, All The Best..!! Happy Learning..!!
    </p>
</body>
</html>
```



**PLEASE NOTE:** From here onwards programming in XHTML will begin. Please add the following compulsory document structure to all programs in the first 4 lines and skip the simple `<html>` tag of first line.

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
```

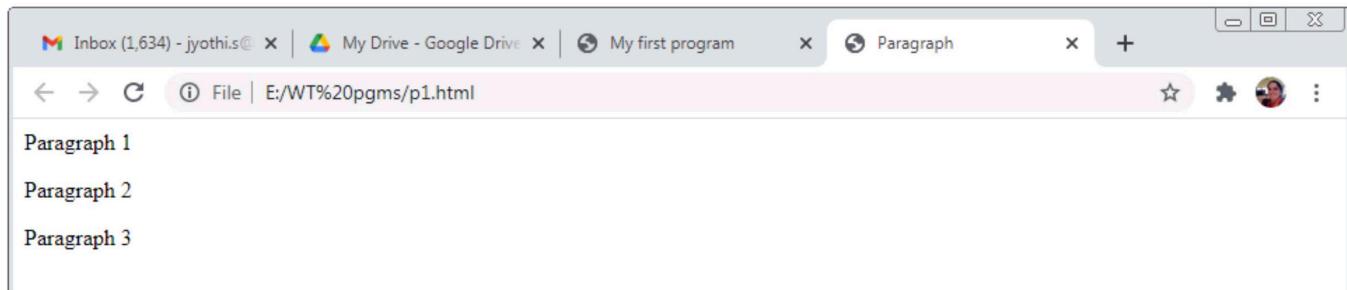
### **Paragraphs:**

It begins with `<p>` and ends with `</p>`. Multiple paragraphs may appear in a single document.

```
<html>
<head>
    <title> Paragraph </title>
</head>
<body>
    <p> Paragraph 1 </p>
    <p> Paragraph 2 </p>
    <p> Paragraph 3 </p>
```

[Type text]

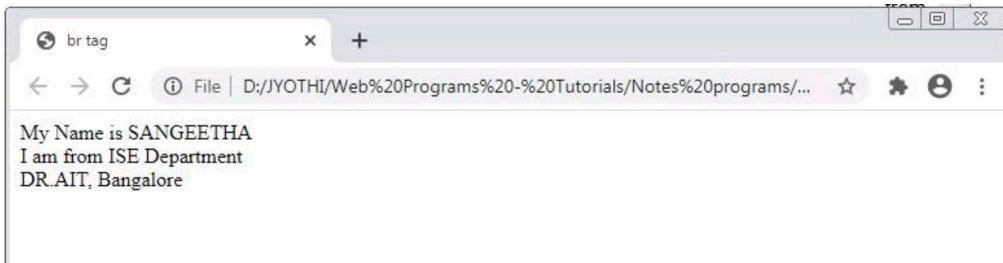
```
</body>
</html>
```



## Line Breaks:

The break tag is specified as `<br />`. The slash indicates that the tag is both an opening and closing tag.

```
<html>
<head>
    <title> br tag </title>
</head>
<body>
    <p>
        My Name is SANGEETHA <br/>
        I am from ISE Department<br/> DR.AIT,
        Bangalore
    </p>
</body>
</html>
```

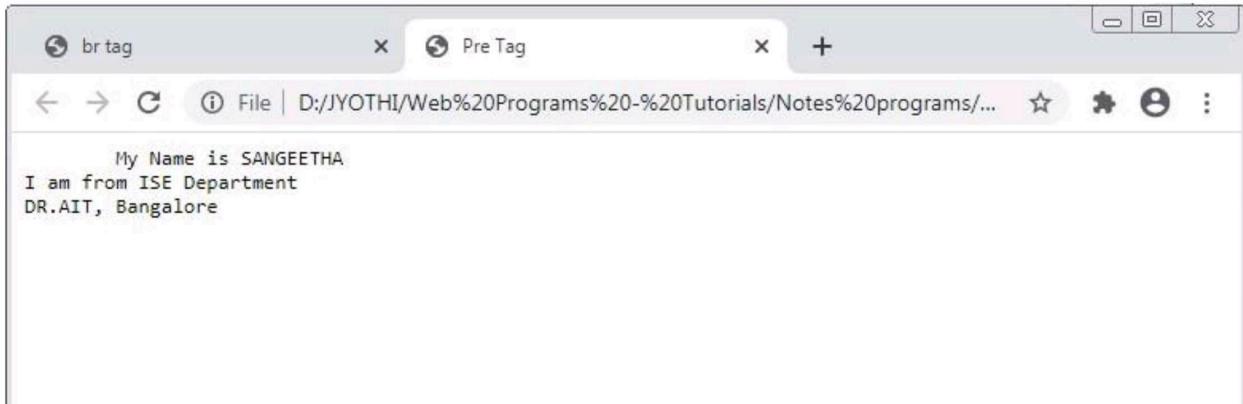


## Preserving White Space

Sometimes it is desirable to preserve the white space in text—that is, to prevent the browser from eliminating multiple spaces and ignoring embedded line breaks. This can be specified with the `<pre>` tag.

```
<html>
<head>
    <title> Pre Tag </title>
</head>
<body>
    <p><pre>           My Name is SANGEETHA
        I am from ISE Department
        DR.AIT, Bangalore
    </pre></p>
</body>
</html>
```

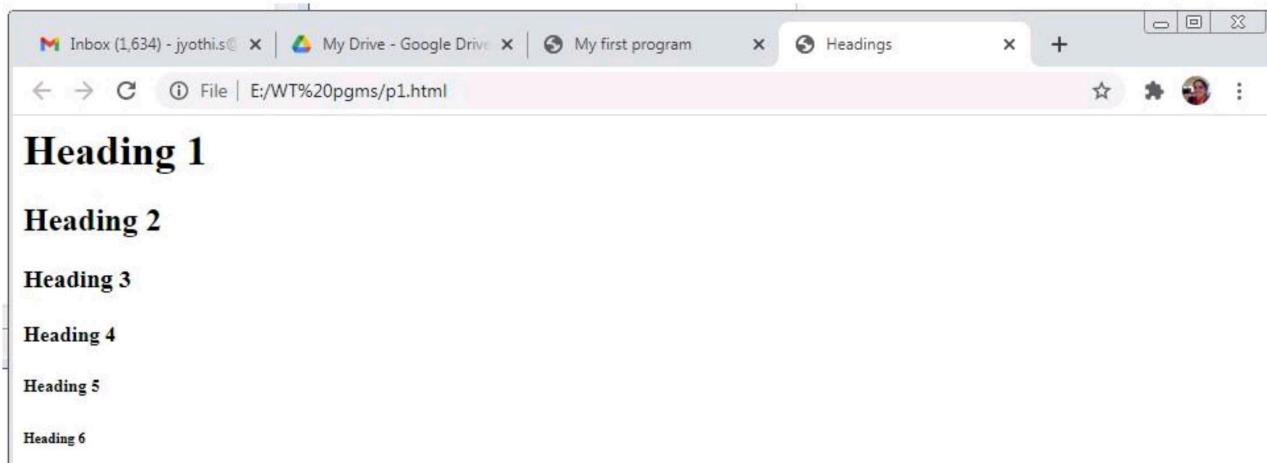
[Type text]



## Headings:

In XHMTL, there are six levels of headings, specified by the tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`, where `<h1>` specifies the highest-level heading. Headings are usually displayed in a boldface font whose default size depends on the number in the heading tag. On most browsers, `<h1>`, `<h2>`, and `<h3>` use font sizes that are larger than that of the default size of text, `<h4>` uses the default size, and `<h5>` and `<h6>` use smaller sizes. The heading tags always break the current line, so their content always appears on a new line. Browsers usually insert some vertical space before and after all headings.

```
<html>
<head>
    <title> Headings </title>
</head>
<body>
    <h1> Heading 1 </h1>
    <h2> Heading 2 </h2>
    <h3> Heading 3 </h3>
    <h4> Heading 4 </h4>
    <h5> Heading 5 </h5>
    <h6> Heading 6 </h6>
</body>
</html>
```

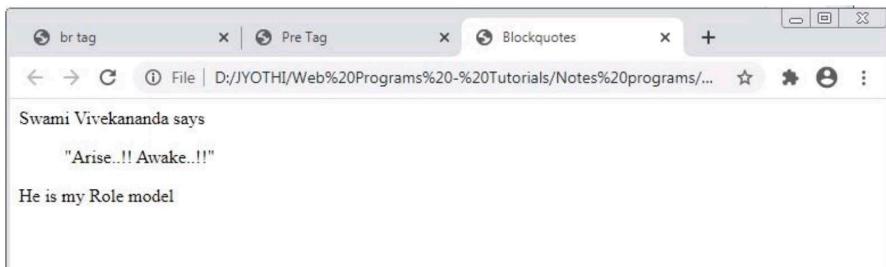


[Type text]

## Block Quotations:

The `<blockquote>` tag is used to make the contents look different from the surrounding text.

```
<html>
<head> <title> Blockquotes </title>
</head>
<body>
<p> Swami Vivekananda says </p>
<blockquote>
<p> "Arise..!! Awake..!!" </p>
</blockquote>
<p> He is my Role model </p>
</body>
</html>
```



## Font Styles and Sizes:

- `<b>`, `<i>` and `<u>` specifies bold, italics and underline respectively.
- The emphasis tag, `<em>`, specifies that its textual content is special and should be displayed in some way that indicates this distinctiveness. Most browsers use italics for such content.
- The strong tag, `<strong>` is like the emphasis tag, but more so. Browsers often set the content of strong elements in bold.
- The code tag, `<code>`, is used to specify a monospace font, usually for program code.

```
<html>
<head> <title> font styles and sizes </title>
</head>
<body>
<p><pre>
Illustration of Font Styles
<b> This is Bold </b>
<i> This is Italics </i>
<u> This is Underline </u>
<em> This is Emphasis </em>
<strong> This is strong </strong>
<code> Total = Internals + Externals //this is code</code>
</pre></p>
<p><pre>
Illustration of Font Sizes (subscripts and superscripts)
x<sub>2</sub><sup>3</sup> + y<sub>1</sub><sup>2</sup>
</pre></p>
</body>
</html>
```

[Type text]



The screenshot shows a browser window with several tabs open: "br tag", "Pre Tag", "Blockquotes", and "font styles and sizes". The main content area displays text illustrating different font styles and sizes. It includes examples of bold, italics, underline, emphasis, strong, and code styles. It also shows mathematical notation like  $x_2^3 + y_1^2$ .

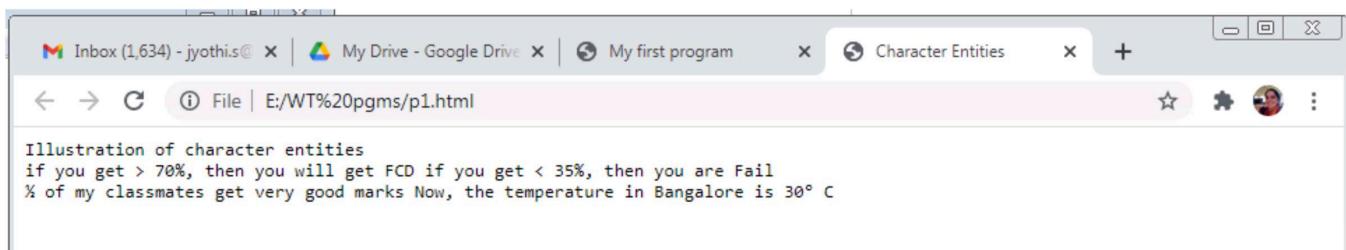
```
Illustration of Font Styles
This is Bold
This is Italics
This is Underline
This is Emphasis
This is strong
Total = Internals + Externals //this is code

Illustration of Font Sizes (subscripts and superscripts)  $x_2^3 + y_1^2$ 
```

## Character Entities:

XHTML provides a collection of special characters that are sometimes needed in a document but cannot be typed as themselves. In some cases, these characters are used in XHTML in some special way—for example, >, <, and &. In other cases, the characters do not appear on keyboards, such as the small raised circle that represents “degrees” in a reference to temperature. These special characters are defined as entities, which are codes for the characters. An entity in a document is replaced by its associated character by the browser.

```
<html>
<head> <title> Character Entities </title>
</head>
<body>
<p><pre>
  Illustration of character entities
  if you get &gt; 70%, then you will get FCD if you get &lt;
  35%, then you are Fail
  &frac12 of my classmates get very good marks Now, the
  temperature in Bangalore is 30&deg C
</pre></p>
</body>
</html>
```



The screenshot shows a browser window with three tabs: "Inbox (1,634) - jyothi.s@...", "My Drive - Google Drive", and "My first program". The current tab is titled "Character Entities". The main content area displays text illustrating character entities. It includes examples of >, <, &, &gt;, &lt;, &frac12;, &deg;, and &nbsp;.

```
Illustration of character entities
if you get > 70%, then you will get FCD if you get < 35%, then you are Fail
½ of my classmates get very good marks Now, the temperature in Bangalore is 30° C
```

## Horizontal Rules:

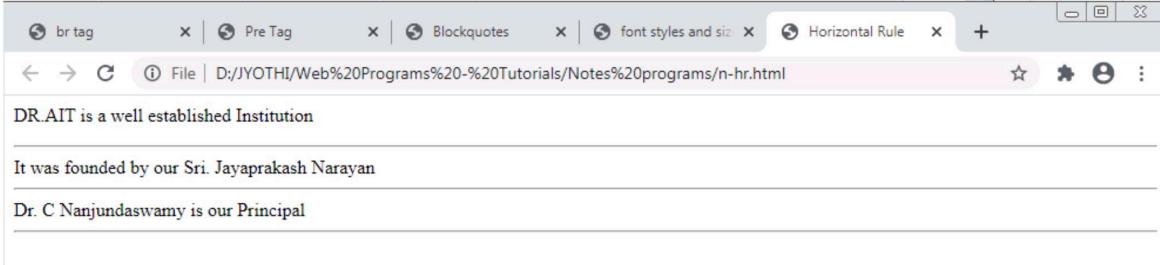
The parts of a document can be separated from each other, making the document easier to read, by placing horizontal lines between them. Such lines are called horizontal rules. The block tag that creates them is `<hr />`. The `<hr />` tag causes a line break (ending the current line) and places a line across the screen.

Note again the slash in the `<hr />` tag, indicating that this tag has no content and no closing tag.

```
<html>
<head>
  <title> Horizontal Rule </title>
</head>
<body>
<p>
  DR.AIT is a well established Institution<hr/>
  It was founded by our Sri. Jayaprakash Narayan <hr/>
```

[Type text]

```
Dr. C Nanjundaswamy is our Principal <hr>
</p>
</body>
</html>
```



### The meta Element:

The meta element is used to provide additional information about a document. The meta tag has no content; rather, all of the information provided is specified with attributes. The two attributes that are used to provide information are name and content. The user makes up a name as the value of the name attribute and specifies information through the content attribute. One commonly chosen name is keywords; the value of the content attribute associated with the keywords are those which the author of a document believes characterizes his or her document. An example is

```
<meta name = "Title" content = "Web Technologies" />
```

```
<meta name = "Author" content = "Jyothi S" />
```

Web search engines use the information provided with the meta element to categorize Web documents in their indices.

### IMAGES

- Image can be displayed on the web page using <img> tag.
- When the <img> tag is used, it should also be mentioned which image needs to be displayed. This is done using **src** attribute.
- Attribute means extra information given to the browser
- Whenever <img> tag is used, alt attribute is also used.
- Alt stands for alert.
- Some very old browsers would not be having the capacity to display the images.
- In this case, whatever is the message given to alt attribute, that would be displayed.
- Another use of alt is when image display option has been disabled by user. The option is normally disabled when the size of the image is huge and takes time for downloading.

```
<html>
  <head>
    <title>display image</title>
  </head>
  <body>
    
  </body>
</html>
```

#### NOTE:

- JPEG - Joint Photographic Experts Group
- GIF - Graphic Interchange Format
- PNG - Portable Network Graphics

[Type text]

## XHTML Document Validation:

The W3C provides a convenient Web-based way to validate XHTML documents against its standards. Step 1: The URL of the service is <http://validator.w3.org/file-upload.html>. Copy & paste this link.

Step 2: You will be driven to “Validate by File Upload” option automatically.

Step 3: Browse for a XHTML program file in your computer. (example: F:/complete.html)

Step 4: Click on “More Options” and select your criteria like show source

Step 5: After all the settings, click on “Check” button

Now you will be navigated to another page which shows success or failure.

In our example, the file *complete.html* is a valid XHTML file. So the output shows success..!!

## Output:

This screenshot shows the W3C Markup Validation Service interface in Mozilla Firefox. A series of numbered callouts (1 through 5) guide the user through the validation process:

- Step 1: The URL [validator.w3.org/#validate\\_by\\_upload+with\\_options](http://validator.w3.org/#validate_by_upload+with_options) is pasted into the address bar.
- Step 2: The "Validate by File Upload" tab is selected.
- Step 3: A file named "F:\complete.html" is selected from the "File" input field.
- Step 4: The "More Options" dropdown menu is open, with the "Show Source" checkbox selected.
- Step 5: The "Check" button is highlighted.

Below the form, a note states: "Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#) on the W3C QA Website."

At the bottom, a note reads: "This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SML, MathML, etc. If you wish to validate specific content such as RSS/Atom feeds or CSS stylesheets, [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).

This screenshot shows the validation results page for the file "complete.html". A red oval highlights the green header bar which displays the message: "This document was successfully checked as XHTML 1.0 Strict!". To the right of this message is a blue speech bubble containing the text "Hey Success..!!".

The page includes various metadata fields and a footer with community support information.

Metadata Fields (from top to bottom):

- File:  Browse...  
Use the file selection box above if you wish to re-validate the uploaded file complete.html/
- Modified: (undefined)
- Server: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1
- Size: (undefined)
- Content-Type: text/html
- Encoding: utf-8
- Doctype: XHTML 1.0 Strict
- Root Element: html
- Root Namespace: <http://www.w3.org/1999/xhtml>

Footer:

The W3C validators rely on community support for hosting and development.  
[Donate](#) and help us build better tools for a better web.

3523 [Help](#)

[Type text]

## HYPertext Links

### Links:

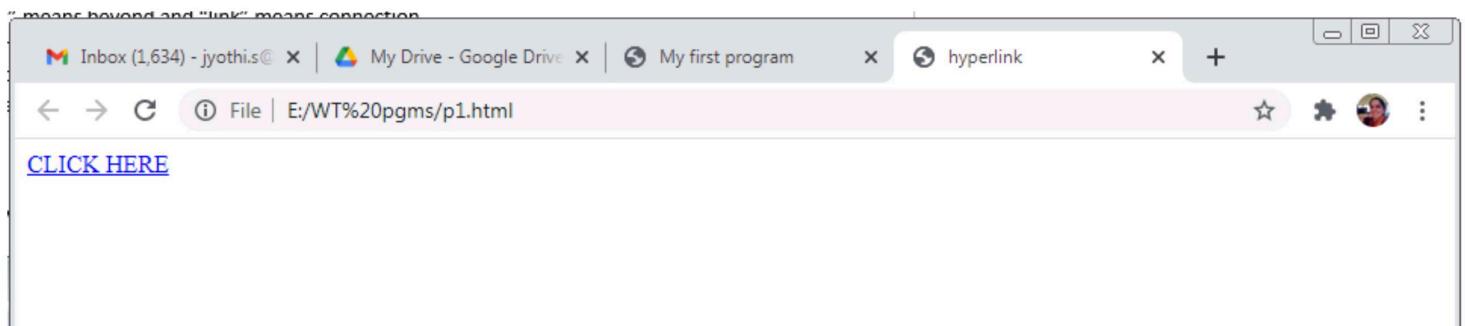
- Hyperlinks are the mechanism which allows the navigation from one page to another.
- The term “hyper” means beyond and “link” means connection
- Whichever text helps in navigation is called hypertext
- Hyperlinks can be created using [\(anchor tag\)](#)
- The attribute that should be used for [is \*\*href\*\*](#)

### Program: *hyper.html*

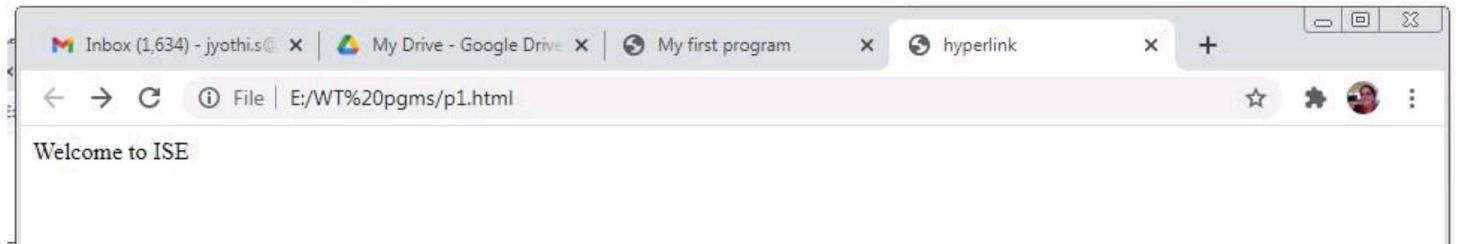
```
<html>
<head>
<title> hyperlink </title>
</head>
<a href = "link.html"> CLICK HERE </a>
</html>
```

### Program: *link.html*

```
<html>
<head>
<title> hyperlink </title>
</head>
<p> Welcome to ISE </p>
</html>
```



After clicking on the above text, we can navigate to another page “link.html” as shown below



[Type text]

### Targets within Documents:

If the target of a link is not at the beginning of a document, it must be some element within the document, in which case there must be some means of specifying it. The target element can include an id attribute, which can then be used to identify it in an href attribute. (observe the scroll bar in the outputs given)

```
<html>
<head>
<title> target link</title>
</head>
<body>
<h1> Puneeth Rajkumar </h1>
<a href = "#bottom"> Click Here For His Autobiography </a>
<p><pre>
Appu
Abhi
Veera Kannadiga
Maurya
Akaash Namma
Basava Ajay
Arasu
Milana
Bindaas
Vamshi
Raaj Raam
Prithvi
Jackie
Hudugaru
Paramathma
Anna Bond
</pre></p>
<h2> AutoBiography </h2>
<p id = "bottom"> <pre>
Puneeth Rajkumar was born on 17th of March, 1975.
His father Dr. Rajkumar is the Legend of Kannada Film Industry.
His mother is Smt. Parvathamma Rajkumar who is a renowned producer in the industry. His
brothers ShivaRajkumar and RaghavendraRajkumar are very popular heroes.
He is married to Smt. Ashwini Revnath
He has two daughters namely Dhrithi and Vanditha..
At present, Puneeth is the greatest star of Kannada Film Industry.
</pre></p>
</body>
</html>
```

[Type text]

A screenshot of a Mozilla Firefox browser window titled "target link - Mozilla Firefox". The address bar shows "file:///F:/7th semester/3. web programming/WP assignments/targetLink.html". The page content is a list of names:

```
Appu
Abhi
Veera Kannadiga
Maurya
Akaash
Namma Basava
Ajay
Arasu
Milana
Bindaas
Vamshi
Raaj
Raam
Prithvi
Jackie
Hudugaru
Paramathma
Anna Bond
```

Actually, here we are not creating two separate files, but we are specifying a target within the same document itself. If you click on the above link, you will be redirected to the bottom of the page which contains Autobiography of Puneeth Rajkumar. This is useful for lengthy documents like e-newspaper, e-magazine etc.,

A screenshot of a Mozilla Firefox browser window titled "target link - Mozilla Firefox". The address bar shows "file:///F:/7th semester/3. web programming/WP assignments/targetLink.html". The page content is identical to the first screenshot, displaying the same list of names.

## AutoBiography

Puneeth Rajkumar was born on 17th of March, 1975.  
His father Dr. Rajkumar is the Legend of Kannada Film Industry.  
His mother is Smt. Parvathamma Rajkumar who is a renowned producer in the industry.  
His brothers ShivaRajkumar and RaghavendraRajkumar are very popular heroes.  
He is married to Smt. Ashwini Revnath  
He has two daughters namely Dhrithi and Vanditha..  
At present, Puneeth is the greatest star of Kannada Film Industry.

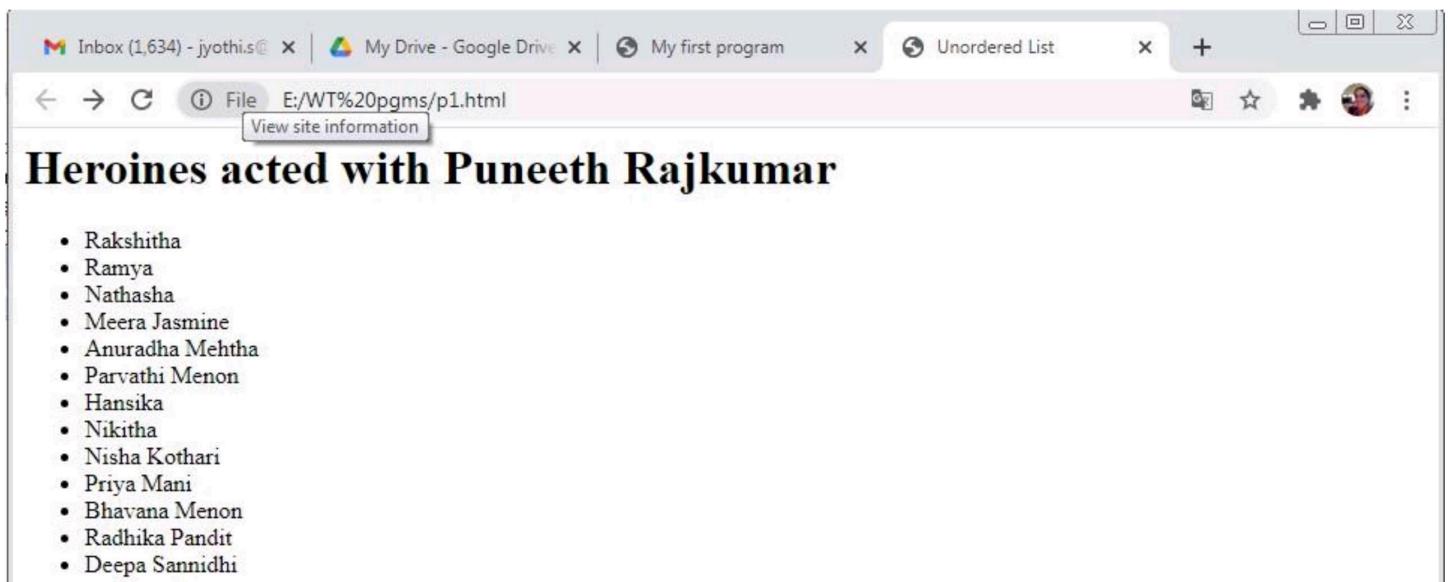
[Type text]

## LISTS

### Unordered Lists:

The `<ul>` tag, which is a block tag, creates an unordered list. Each item in a list is specified with an `<li>` tag (`li` is an acronym for *list item*). Any tags can appear in a list item, including nested lists. When displayed, each list item is implicitly preceded by a bullet.

```
<html>
<head>
<title> Unordered List </title>
</head>
<body>
<h1> Heroines acted with Puneeth Rajkumar </h1>
<ul>
    <li>Rakshitha</li>
    <li>Ramya</li>
    <li>Nathasha</li>
    <li>Meera Jasmine</li>
    <li>Anuradha Mehta</li>
    <li>Parvathi Menon</li>
    <li>Hansika</li>
    <li>Nikitha</li>
    <li>Nisha Kothari</li>
    <li>Priya Mani</li>
    <li>Bhavana Menon</li>
    <li>Radhika Pandit</li>
    <li>Deepa Sannidhi</li>
</ul>
</body>
</html>
```



A screenshot of a web browser window titled "Unordered List". The address bar shows the file path "E:/WT%20pgms/p1.html". The main content area displays the heading "**Heroines acted with Puneeth Rajkumar**" followed by an unordered list of names:

- Rakshitha
- Ramya
- Nathasha
- Meera Jasmine
- Anuradha Mehta
- Parvathi Menon
- Hansika
- Nikitha
- Nisha Kothari
- Priya Mani
- Bhavana Menon
- Radhika Pandit
- Deepa Sannidhi

[Type text]

## Ordered Lists:

Ordered lists are lists in which the order of items is important. This orderedness of a list is shown in the display of the list by the implicit attachment of a sequential value to the beginning of each item. The default sequential values are Arabic numerals, beginning with 1.

An ordered list is created within the block tag `<ol>`. The items are specified and displayed just as are those in unordered lists, except that the items in an ordered list are preceded by sequential values instead of bullets.

```
<html>
<head><title> ordered List </title></head>
<body>
<h1>Chicken Masala</h1>
<ol>
    <li>For 1 kg of chicken, add 20g Teju Chicken Masala</li>
    <li>Fry 2 big onions with 3tbsp ghee/oil till golden brown</li>
    <li>Add 2 tomato, 1tsp ginger garlic paste, 2-3 green chillies and fry</li>
    <li>When tomato becomes soft, add chicken and 100ml water</li>
    <li>Add 25g coriander leaves and cook till the chicken is soft and gravy turns thick</li>
    <li>Ready to serve</li>
</ol>
</body>
</html>
```

The screenshot shows a web browser window with the title bar 'ordered List'. The address bar shows the file path 'E:/WT%20pgms/p1.html'. The main content area displays a heading 'Vegetable Pulav' followed by a numbered list of six steps:

1. For 1/4 kg of Rice, add 20g MTR Pulav Masala
2. Fry 2 big onions with 3tbsp ghee/oil till golden brown
3. Add 1tsp ginger garlic paste, 2-3 green chillies, 2 tomato and fry
4. When tomato becomes soft, add cut Vegetables(peas,carrot, beans) along with Rice and 100ml water
5. Add 25g coriander leaves and cook till the Rice is soft
6. Ready to serve

## Nested Lists:

```
<html>
<head><title> nested lists </title></head>
<ol>
    <li> Information Science </li>
    <ol>
        <li>OOMD</li>
        <li>Java & J2ee</li>
        <ul>
            <li>classes and methods</li>
            <li>exceptions</li>
            <li>applets</li>
            <li>servelets</li>
        </ul>
    </ol>
</ol>
```

[Type text]

```
<li>Computer Networks</li>
<ul>
  <li>Part 1</li>
  <li>Part 2</li>
</ul>
<li>DBMS</li>
<li>Operations Research</li>
</ol>
<li> Computer Science</li>
<ol>
  <li>Compiler Design</li>
  <li>FLAT</li>
  <ul>
    <li>NFA</li>
    <li>DFA</li>
    <li>CFG</li>
  </ul>
  <li>Computer Graphics</li>
  <li>Artificial Intelligence</li>
</ol>
</ol>
</html>
```

The screenshot shows a web browser window with the following tabs: 'Inbox (1,634) - jyothi.s@...', 'My Drive - Google Drive', 'My first program', and 'nested lists'. The current page is 'E:/WT%20pgms/p1.html'. The content of the page is a nested list:

- 1. Information Science
  - 1. OOMD
  - 2. Java & J2ee
    - classes and methods
    - exceptions
    - applets
    - servelets
  - 3. Computer Networks
    - Part 1
    - Part 2
  - 4. DBMS
  - 5. Operations Research

2. Computer Science

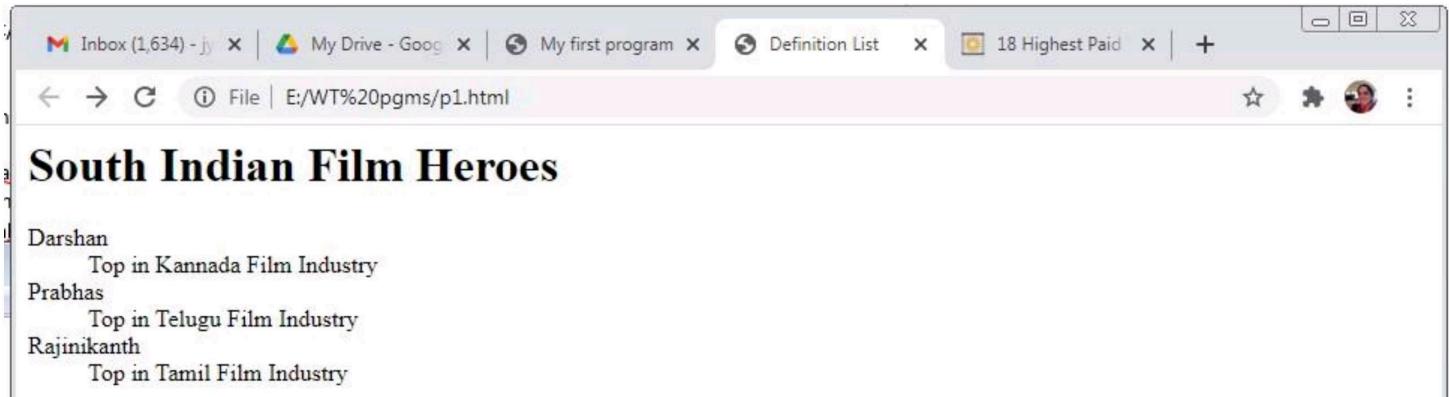
  - 1. Compiler Design
  - 2. FLAT
    - NFA
    - DFA
    - CFG
  - 3. Computer Graphics
  - 4. Artificial Intelligence

[Type text]

### Definition Lists:

As the name implies, definition lists are used to specify lists of terms and their definitions, as in glossaries. A definition list is given as the content of a `<dl>` tag, which is a block tag. Each term to be defined in the definition list is given as the content of a `<dt>` tag. The definitions themselves are specified as the content of `<dd>` tags. The defined terms of a definition list are usually displayed in the left margin; the definitions are usually shown indented on the line or lines following the term.

```
<html>
<head>
<title> Definition List </title>
</head>
<body>
<h1> South Indian Film Heroes </h1>
<dl>
    <dt> Puneeth Rajkumar </dt>
    <dd> Top in Kannada Film Industry </dd>
    <dt> Mahesh Babu </dt>
    <dd> Top in Telugu Film Industry </dd>
    <dt> Suriya </dt>
    <dd> Top in Tamil Film Industry </dd>
</dl>
</body>
</html>
```



[Type text]

## TABLES

A table is a matrix of cells. The cells in the top row often contain column labels, those in the leftmost column often contain row labels, and most of the rest of the cells contain the data of the table. The content of a cell can be almost any document element, including text, a heading, a horizontal rule, an image, and a nested table.

### Basic Table Tags:

- A table is specified as the content of the block tag **<table>**.
- There are two kinds of lines in tables: the line around the outside of the whole table is called the *border*; the lines that separate the cells from each other are called *rules*.
- It can be obtained using **border** attribute. The possible values are “border” or any number.
- The table heading can be created using **<caption>** tag.
- The table row can be created using **<tr>** tag.
- The column can be created either by using **<th>** tag (stands for table header which is suitable for headings) or **<td>** tag (stands for table data which is suitable for other data).

```
<html>
<head>
<title> Table with text and image </title>
</head>
<body>
<table border = "border">
<caption>Flowers</caption>
<tr>
<th> Type</th>
<th> Image </th>
</tr>
<tr>
<td> Sun Flower</td>
<td> <img src = "puneeth.jpg" alt = "cant display"/></td>
</tr>
<tr>
<td> Rose</td>
<td> <img src = "deepa.jpg" alt = "cant display"/></td>
</tr>
</table>
</body>
</html>
```

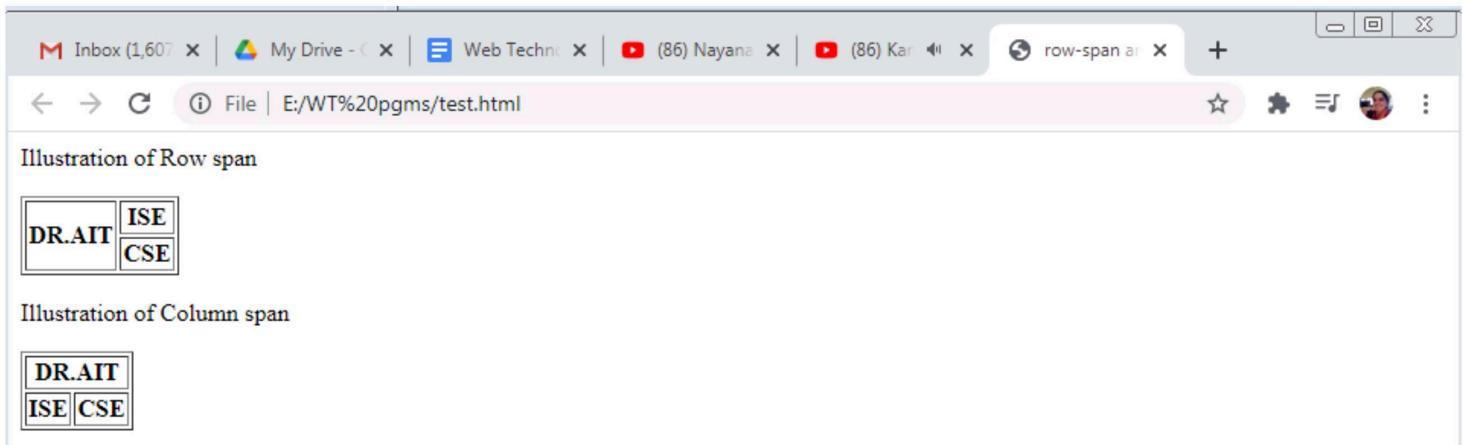
Type	Image
Sun Flower	
Rose	

[Type text]

### The rowspan and colspan Attributes:

Multiple-level labels can be specified with the rowspan and colspan attributes.

```
<html>
  <head>
    <title>row-span and column-span</title>
  </head>
  <body>
    <p> Illustration of Row span</p>
    <table border="border">
      <tr>
        <th rowspan="2"> DR.AIT </th>
        <th>ISE</th>
      </tr>
      <tr>
        <th>CSE</th>
      </tr>
    </table>
    <p> Illustration of Column span</p>
    <table border="border">
      <tr>
        <th colspan="2"> DR.AIT </th>
      </tr>
      <tr>
        <th>ISE</th>
        <th>CSE</th>
      </tr>
    </table>
  </body>
</html>
```



[Type text]

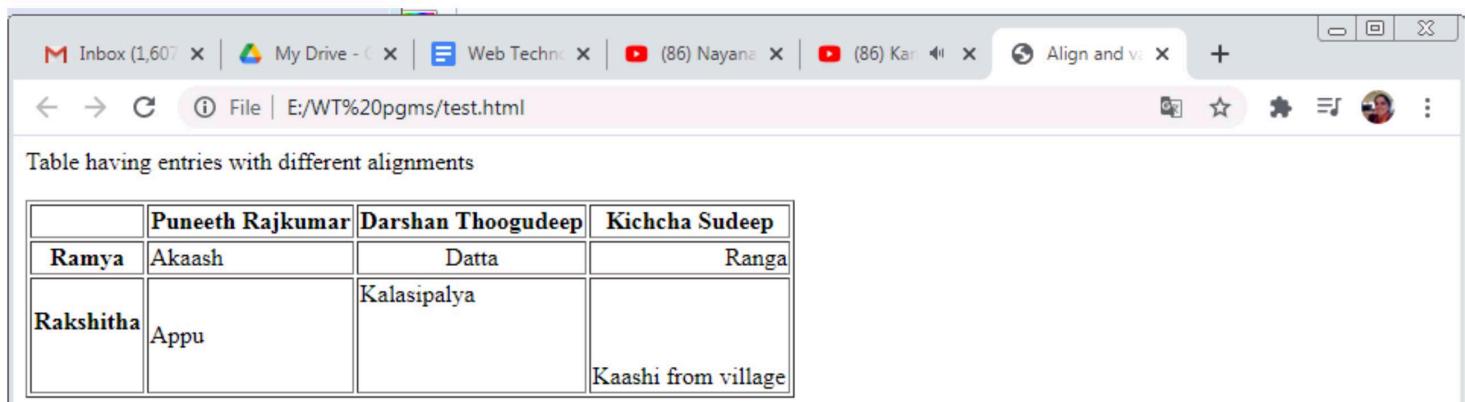
### The align and valign Attributes:

The placement of the content within a table cell can be specified with the align and valign attributes in the <tr>, <th>, and <td> tags.

The align attribute has the possible values left, right, and center, with the obvious meanings for horizontal placement of the content within a cell. The default alignment for th cells is center; for td cells, it is left.

The valign attribute of the <th> and <td> tags has the possible values top and bottom. The default vertical alignment for both headings and data is center.

```
<html>
  <head>
    <title> Align and valign </title>
  </head>
  <body>
    <p>Table having entries with different alignments</p>
    <table border="border">
      <tr align = "center">
        <th> </th>
        <th> Puneeth Rajkumar </th>
        <th> Darshan Thoogudeep</th>
        <th> Kichcha Sudeep </th>
      </tr>
      <tr>
        <th> Ramya </th>
        <td align = "left"> Akaash </td>
        <td align = "center"> Datta </td>
        <td align = "right"> Ranga </td>
      </tr>
      <tr>
        <th> <br/>Rakshitha <br/><br/><br/></th>
        <td> Appu </td>
        <td valign = "top"> Kalasipalya </td>
        <td valign = "bottom"> Kaashi from village </td>
      </tr>
    </table>
  </body>
</html>
```



The screenshot shows a web browser window with the title bar "Align and valign". The address bar shows the file path "E:/WT%20pgms/test.html". The main content area displays the heading "Table having entries with different alignments" followed by a table with four columns and three rows. The first row has column headers: "Puneeth Rajkumar", "Darshan Thoogudeep", and "Kichcha Sudeep". The second row contains the name "Ramya" in the first column and the names "Akaash", "Datta", and "Ranga" in the subsequent columns. The third row contains the name "Rakshitha" in the first column and the names "Appu", "Kalasipalya", and "Kaashi from village" in the subsequent columns. The table has a border and the text is aligned as specified in the HTML code.

	Puneeth Rajkumar	Darshan Thoogudeep	Kichcha Sudeep
Ramya	Akaash	Datta	Ranga
Rakshitha	Appu	Kalasipalya	Kaashi from village

[Type text]

### The cellpadding and cellspacing Attributes:

Cellspacing is the distance between cells.

Cellpadding is the distance between the edges of the cell to its content.

```
<html>
<head>
<title> cell spacing and cell padding </title>
</head>
<body>
<h3>Table with space = 10, pad = 50</h3>
<table border = "7" cellspacing = "10" cellpadding = "50">
<tr>
<td> Divya </td>
<td>Chethan </td>
</tr>
</table>
<h3>Table with space = 50, pad = 10</h3>
<table border = "7" cellspacing = "50" cellpadding = "10">
<tr>
<td> Divya </td>
<td>Chethan </td>
</tr>
</table>
</body>
</html>
```

### Table Sections:

Tables naturally occur in two and sometimes three parts: header, body, and footer. (Not all tables have a natural footer.) These three parts can be respectively denoted in XHTML with the `thead`, `tbody`, and `tfoot` elements. The header includes the column labels, regardless of the number of levels in those labels. The body includes the data of the table, including the row labels. The footer, when it appears, sometimes has the column labels repeated after the body. In some tables, the footer contains totals for the columns of data above. A table can have multiple body sections, in which case the browser may delimit them with horizontal lines that are thicker than the rule lines within a body section.

## FORMS

The most common way for a user to communicate information from a Web browser to the server is through a form. XHTML provides tags to generate the commonly used objects on a screen form. These objects are called *controls* or *widgets*. There are controls for single-line and multiple-line text collection, checkboxes, radio buttons, and menus, among others. All control tags are inline tags.

### The `<form>` Tag:

All of the controls of a form appear in the content of a `<form>` tag. A block tag, `<form>`, can have several different attributes, only one of which, `action`, is required. The `action` attribute specifies the URL of the application on the Web server that is to be called when the user clicks the *Submit* button. Our examples of form elements will not have corresponding application programs, so the value of their `action` attributes will be the empty string ("").

### The `<input>` Tag:

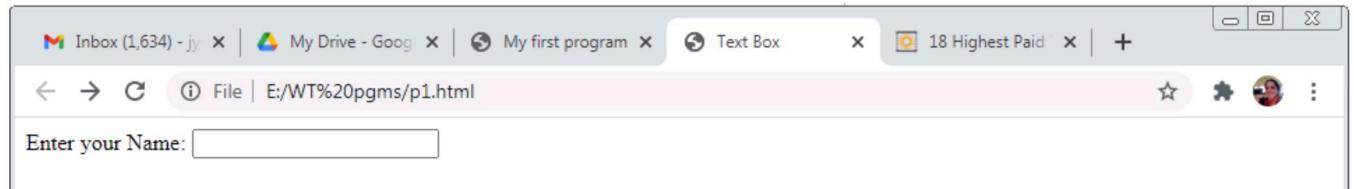
Many of the commonly used controls are specified with the inline tag `<input>`, including those for text, passwords, checkboxes, radio buttons, and the action buttons *Reset*, *Submit*, and *plain*.

[Type text]

❖ Text Box

- ✓ It is a type of input which takes the text.
- ✓ Any type of input can be created using <input>
- ✓ The *type* attribute indicates what type of input is needed for the text box, the value should be given as text.
- ✓ For any type of input, a name has to be provided which is done using *name* attribute.
- ✓ The size of the text can be controlled using *size* attribute.
- ✓ Every browser has a limit on the number of characters it can collect. If this limit is exceeded, the extra characters are chopped off. To prevent this chopping, *maxlength* attribute can be used. When *maxlength* is used, users can enter only those many characters that is given as a value to the attribute.

```
<html>
  <head>
    <title>Text Box</title>
  </head>
  <body>
    <form action = " ">
      <p>
        <label>Enter your Name:</label>
        <input type = "text" name = "myname" size = "20" maxlength = "20" />
      </p>
    </form>
  </body>
</html>
```



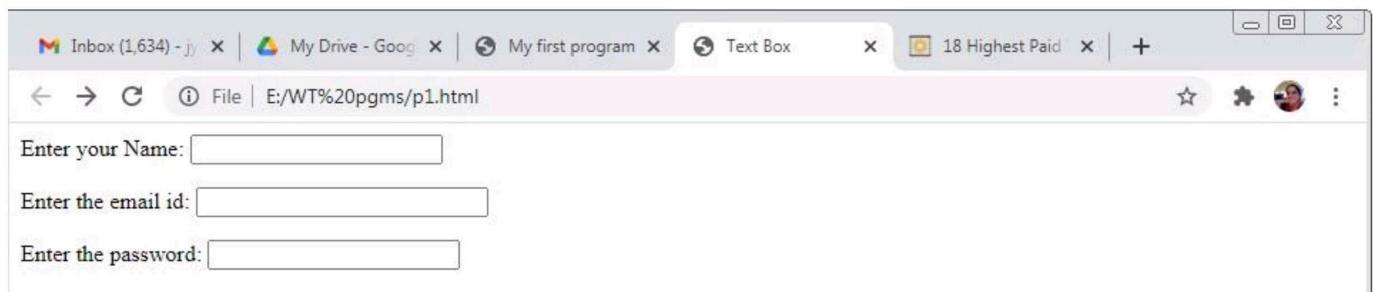
❖ Password Box

- ✓ If the contents of a text box should not be displayed when they are entered by the user, a password control can be used.
- ✓ In this case, regardless of what characters are typed into the password control, only bullets or asterisks are displayed by the browser.

```
<html>
  <head>
    <title>Password Box</title>
  </head>
  <body>
    <form action = " ">
      <p>
        <label>Enter the email id:</label>
        <input type = "text" name = "myname" size = "24" maxlength = "25" />
      </p>
      <p>
        <label>Enter the password:</label>
        <input type = "password" name = "mypass" size = "20" maxlength = "20" />
      </p>
    </form>
  </body>
</html>
```

[Type text]

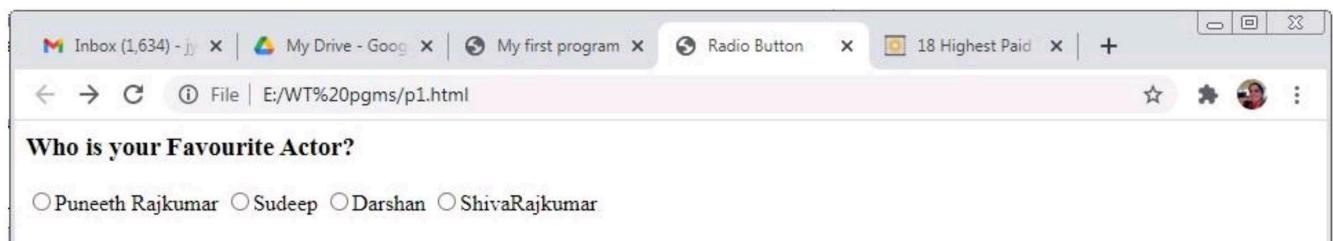
```
</p>
</form>
</body>
</html>
```



### ❖ Radio Button

- ✓ Radio buttons are special type of buttons which allows the user to select only individual option
- ✓ Radio buttons are created using the input tag with the *type* attribute having the value **radio**.
- ✓ When radio buttons are created, values must be provided with the help of *value* attribute.
- ✓ All the radio buttons which are created would have same name. This is because the radio buttons are group elements.
- ✓ If one of the radio buttons has to be selected as soon as the web page is loaded, checked attribute should be used. The value also would be checked.

```
<html>
  <head>
    <title>Radio Button</title>
  </head>
  <body>
    <h3>Who is your Favourite Actor?</h3>
    <form action = " ">
      <p>
        <label><input type="radio" name="act" value="one"/>Puneeth Rajkumar</label>
        <label><input type="radio" name="act" value="two"/>Sudeep</label>
        <label><input type="radio" name="act" value="three"/>Darshan</label>
        <label><input type="radio" name="act" value="four"/>ShivaRajkumar</label>
      </p>
    </form>
  </body>
</html>
```



[Type text]

### ❖ Check Box

- ✓ Check box is a type of input using which multiple options can be selected.
- ✓ Check box can also be created using the `<input>` tag with the `type` having the value “checkbox”.
- ✓ During the creation of check box, the value should be provided using the `value` attribute.
- ✓ All the checkbox which are created would have the same name because they are group elements.
- ✓ If one of the check box have to be selected as soon as the page is loaded, checked attribute should be used with the value checked.

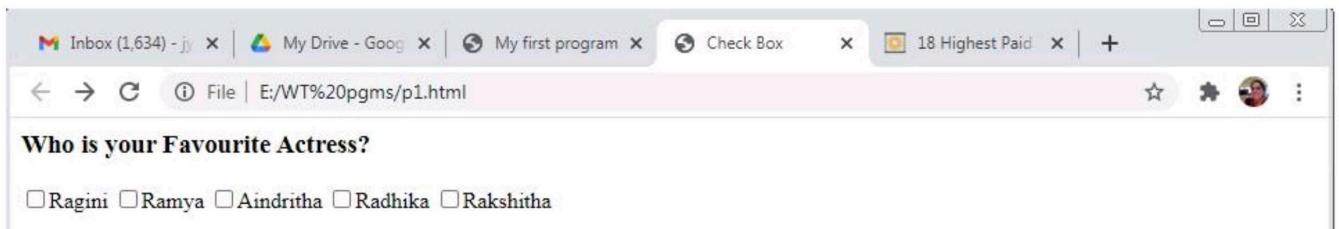
```
<html>
<head>
<title>Check Box</title>
</head>
<body>
<h3>Who is your Favourite Actress?</h3>
<form action = " ">
<p>
<label><input type="checkbox" name="act" value="one"/>Ragini</label>

<label><input type="checkbox" name="act" value="two"/>Ramya</label>

<label><input type="checkbox" name="act" value="three"/>Aindritha</label>

<label><input type="checkbox" name="act" value="four"/>Radhika</label>

<label><input type="checkbox" name="act" value="four"/>Rakshitha</label>
</p>
</form>
</body>
</html>
```



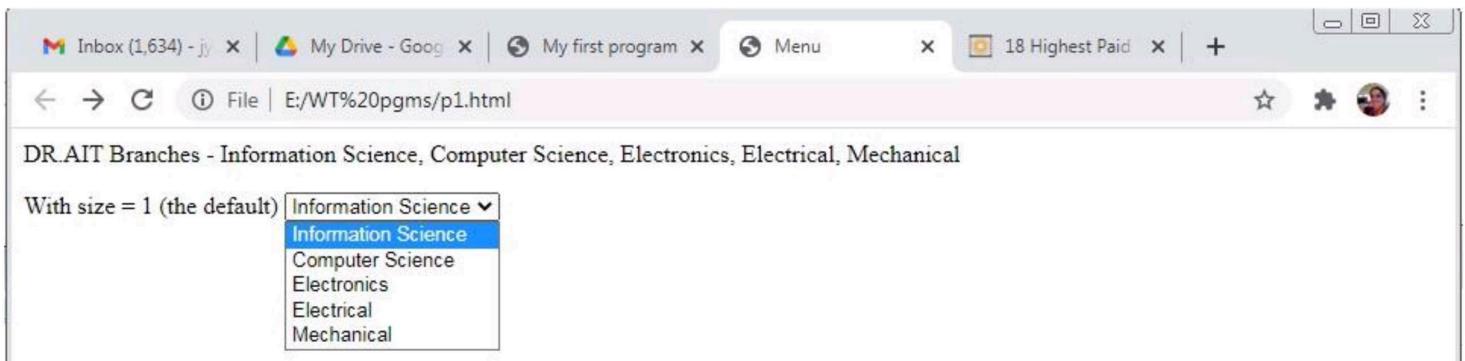
### The `<select>` Tag:

- Menu items is another type of input that can be created on the page.
- To create the menu item, `<select>` tag is used.
- To insert the item in the menu, `<option>` tag is used.

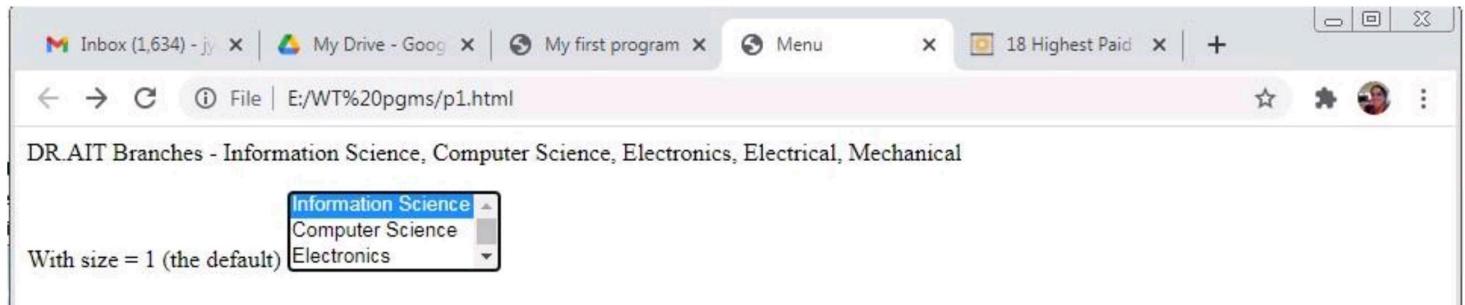
```
<html>
<head> <title> Menu </title>
</head>
<body>
<p>
DR.AIT Branches - Information Science, Computer Science, Electronics, Electrical, Mechanical
</p>
<form action = "">
```

[Type text]

```
<p>
With size = 1 (the default)
<select name = "branches">
<option> Information Science </option>
<option> Computer Science </option>
<option> Electronics </option>
<option> Electrical </option>
<option> Mechanical </option>
</select>
</p>
</form>
</body>
</html>
```



For the above program, If you give `<select name = "branches" size = "3">`, then you will get a scroll bar instead of drop down menu. It is as shown in the output given below:



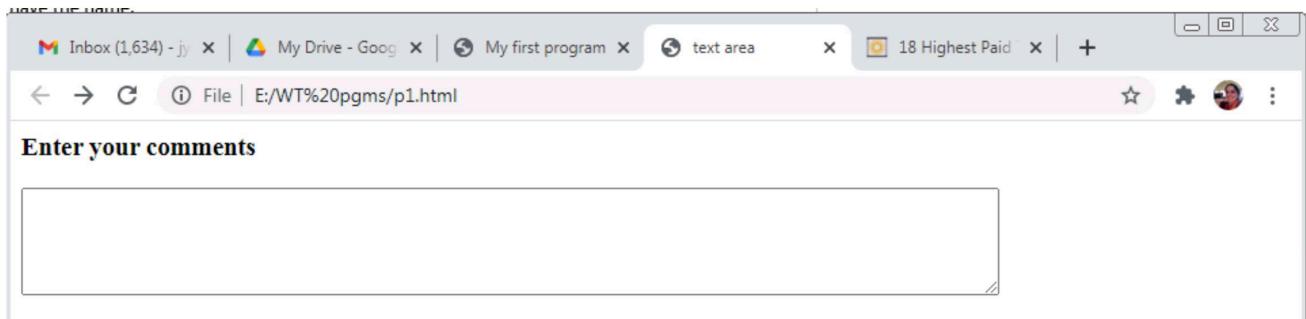
### The `<textarea>` Tag:

- Text area is a type of input using which multiple statements can be entered.
- Text area is created using `<textarea>` tag.
- Text area should have the name.
- During the creation of text area, it should be mentioned how many sentences can be entered. This is done using `rows` attribute.
- Similarly, it should also be mentioned how many characters can be entered in a line. This is done using `cols` attribute.
- If the value given to `rows` is exceeded i.e. if users enter sentences more than specified, the `scrollbar` automatically appears.

```
<html>
<head>
<title> text area </title>
```

[Type text]

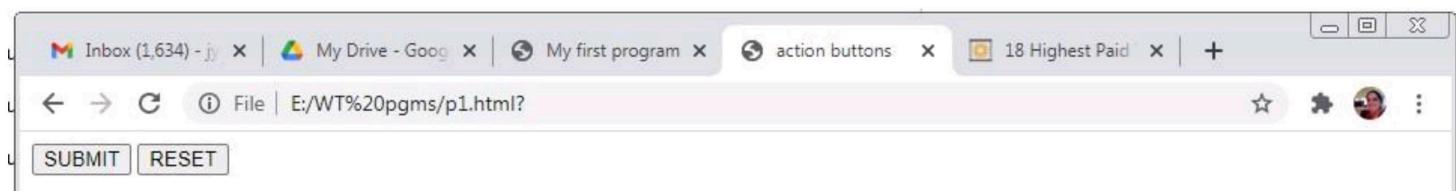
```
</head>
<body>
<form action="">
  <h3> Enter your comments</h3>
  <p>
    <textarea name="feedback" rows="5" cols="100">
    </textarea>
  </p>
</form>
</body>
</html>
```



### The Action Buttons:

The *Reset* button clears all of the controls in the form to their initial states. The *Submit* button has two actions: First, the form data is encoded and sent to the server; second, the server is requested to execute the server- resident program specified in the action attribute of the `<form>` tag. The purpose of such a server-resident program is to process the form data and return some response to the user. Every form requires a *Submit* button. The *Submit* and *Reset* buttons are created with the `<input>` tag.

```
<html>
<head>
<title> action buttons </title>
</head>
<body>
<form action="">
<p>
  <input type="SUBMIT" value="SUBMIT"/>
  <input type="RESET" value="RESET"/>
</p>
</form>
</body>
</html>
```



NOTE: A *plain* button has the type button. *Plain* buttons are used to choose an action.

[Type text]

### Example of a Complete Form:

```
<html>
<head>
<title> CompleteForm</title>
</head>
<body>
<h1>Registration Form</h1>
<form action="">
<p>
<label>Enter your email id:</label>
<input type = "text" name = "myname" size = "24" maxlength = "25" />
</p>
<p>
<label>Enter the password:</label>
<input type = "password" name = "mypass" size = "20" maxlength = "20" />
</p>
<p>Sex</p>
<p>
<label><input type="radio" name="act" value="one"/>Male</label>
<label><input type="radio" name="act" value="two"/>Female</label>
</p>
<p>Which of the following Accounts do you have?</p>
<p>
<label><input type="checkbox" name="act" value="one"/>Gmail</label>
<label><input type="checkbox" name="act" value="two"/>Facebook</label>
<label><input type="checkbox" name="act" value="three"/>Twitter</label>
<label><input type="checkbox" name="act" value="four"/>Google+</label>
</p>
<p> Any Suggestions?</p>
<p>
<textarea name="feedback" rows="5" cols="100">
</textarea>
</p>
<p>Click on Submit if you want to register</p>
<p>
<input type="SUBMIT" value="SUBMIT"/>
<input type="RESET" value="RESET"/>
</p>
</form>
</body>
</html>
```

[Type text]

The screenshot shows a web browser window with the following details:

- Tab Bar:** Shows multiple tabs including "Inbox (1,607)", "My Drive - ", "Web Techno", "(86) Nayana", "(86) Kai", and "CompleteFc".
- Address Bar:** Displays the URL "E:/WT%20pgms/test.html".
- Content Area:**
  - Section Header:** "Registration Form"
  - Text Input:** "Enter your email id:
  - Text Input:** "Enter the password:
  - Text Label:** "Sex"
  - Radio Buttons:** " Male" and " Female"
  - Text Label:** "Which of the following Accounts do you have?"
  - Checkboxes:** " Gmail", " Facebook", " Twitter", and " Google+"
  - Text Label:** "Any Suggestions?
  - Text Label:** "Click on Submit if you want to register"
  - Buttons:** "SUBMIT" and "RESET"

[Type text]

## **FRAMES**

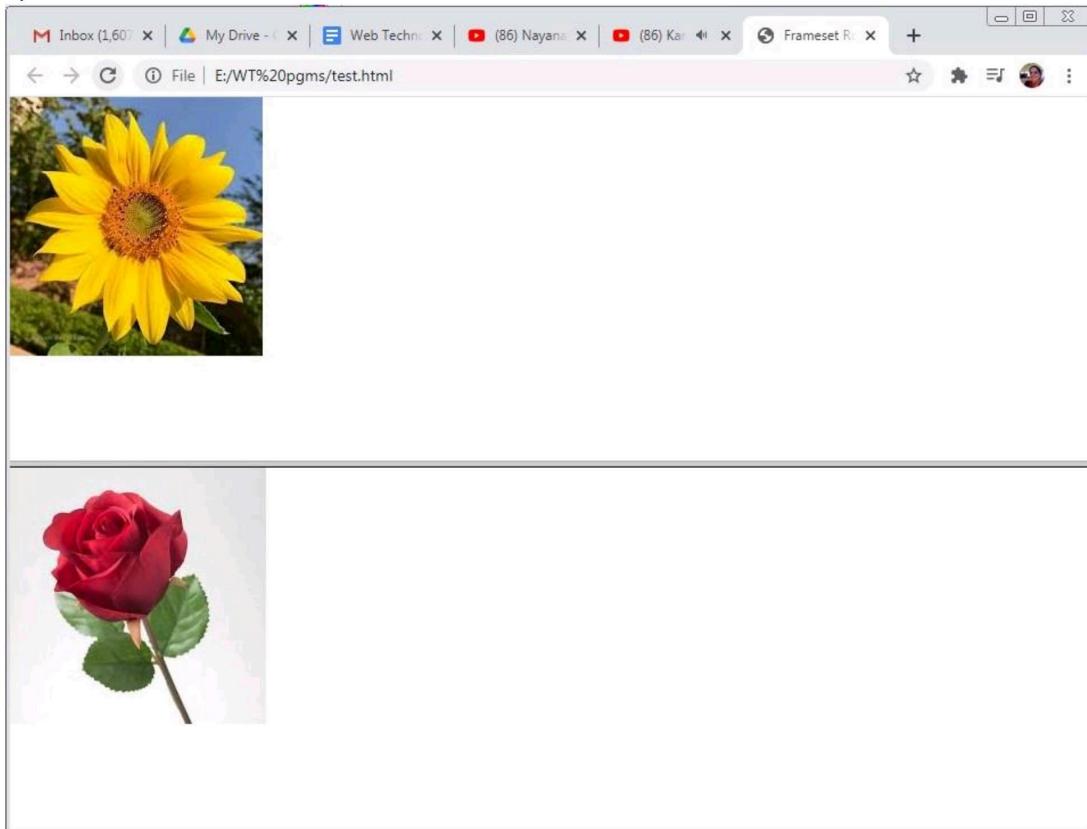
The browser window can be used to display more than one document at a time. The window can be divided into rectangular areas, each of which is a *frame*. Each frame is capable of displaying its own document.

### **Framesets:**

- The number of frames and their layout in the browser window are specified with the <frameset> tag.
- A frameset element takes the place of the body element in a document. A document has either a body or a frameset but cannot have both.
- The <frameset> tag must have either a *rows* or a *cols* attribute. (or both)
- To create horizontal frames, *rows* attribute is used.
- To create vertical frames, *cols* attribute is used.
- The values for these attributes can be numbers, percentages and asterisks.
- Two or more values are separated by commas & given in quoted string.

### **To Demonstrate Horizontal Frames using rows Attribute**

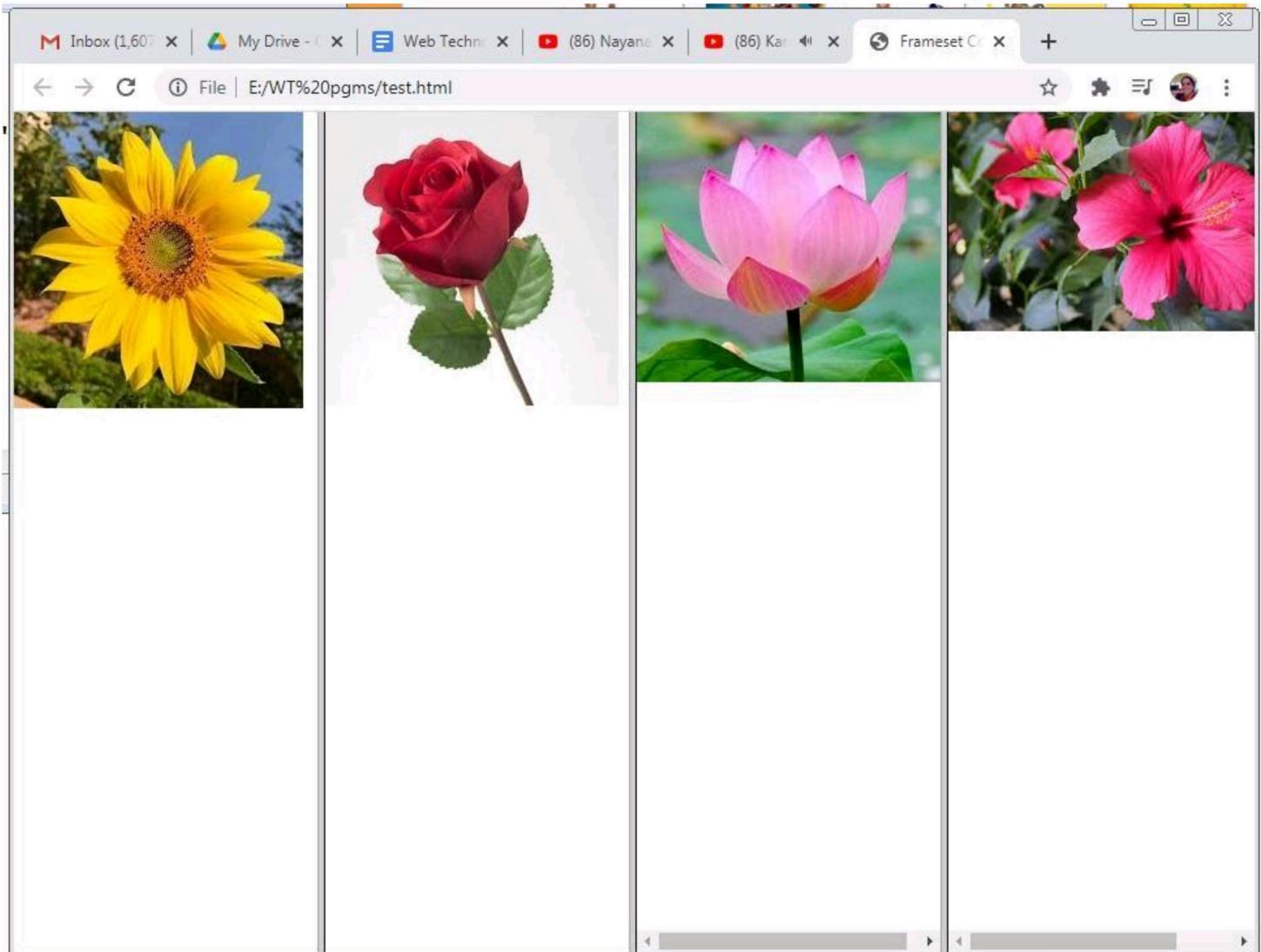
```
<html>
<head>
<title>Frameset Rows</title>
</head>
<frameset rows = "*,*">
<frame src = "flower1.jpg"/>
<frame src = "flower2.jpg"/>
</frameset>
</html>
```



[Type text]

### To Demonstrate Vertical Frames using *cols* Attribute

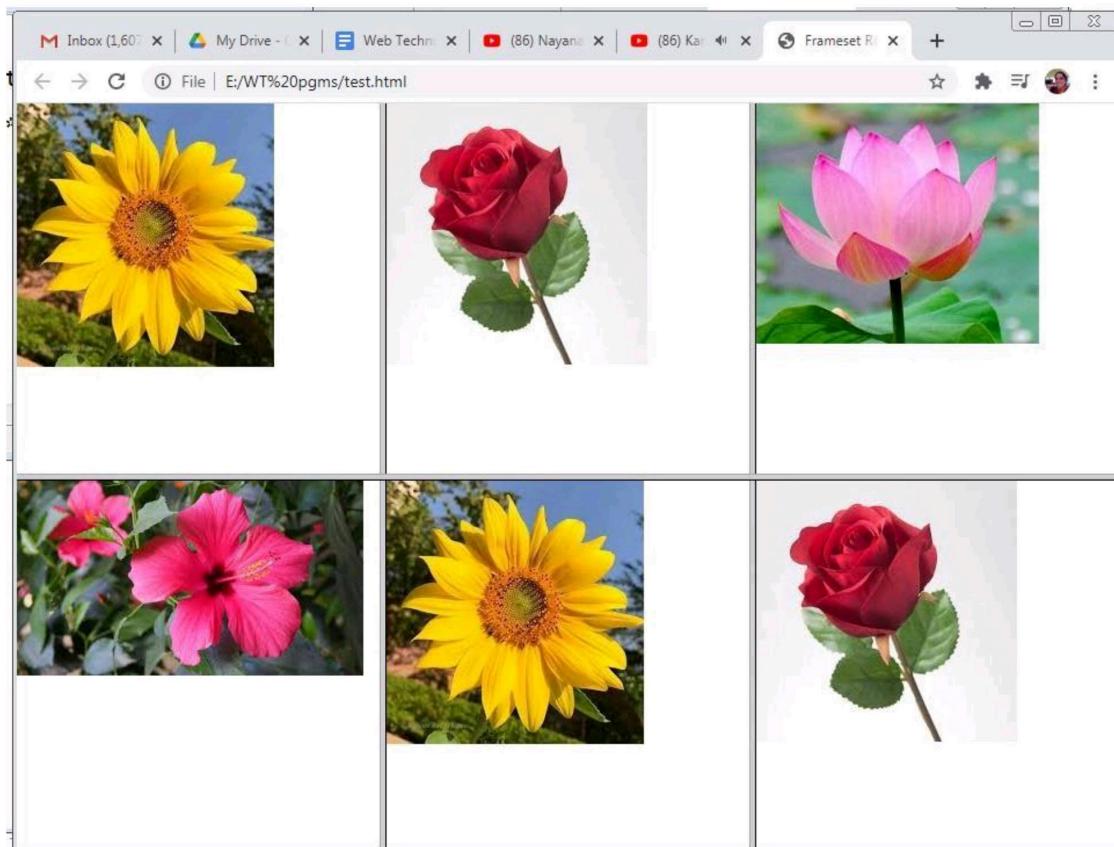
```
<html>
<head>
<title>Frameset Cols</title>
</head>
<frameset cols = "25%,25%,25%,25%">
<frame src = "flower1.jpg"/>
<frame src = "flower2.jpg"/>
<frame src = "flower3.jpg"/>
<frame src = "flower4.jpg"/>
</frameset>
</html>
```



[Type text]

## To Demonstrate Horizontal Frames using rows and cols Attribute

```
<html>
<head>
<title>Frameset Rows and cols</title>
</head>
<frameset rows = "50,50" cols = "*,*,*">
<frame src = "flower1.jpg"/>
<frame src = "flower2.jpg"/>
<frame src = "flower3.jpg"/>
<frame src = "flower4.jpg"/>
<frame src = "flower1.jpg"/>
<frame src = "flower2.jpg"/>
</frameset>
</html>
```



[Type text]

## SYNTACTIC DIFFERENCES BETWEEN HTML AND XHTML

PARAMETERS	HTML	XHTML
Case Sensitivity	Tags and attributes names are case insensitive	Tags and attributes names must be in lowercase
Closing tags	Closing tags may be omitted	All elements must have closing tag
Quoted attribute values	Special characters are quoted. Numeric values are rarely quoted.	All attribute values must be quoted including numbers
Explicit attribute values	Some attribute values are implicit. For example: <table border>. A default value for border is assumed	All attribute values must be explicitly stated
<i>id</i> and <i>name</i> attributes	Both <i>id</i> and <i>name</i> attributes are encouraged	Use of <i>id</i> is encouraged and use of <i>name</i> is discouraged
Element nesting	Rules against improper nesting of elements (for example: a form element cannot contain another form element) are not enforced.	All nesting rules are strictly enforced