

COMPUTER STUDIES

Standard 10



PLEDGE

India is my country.
All Indians are my brothers and sisters.
I love my country and I am proud of its rich and varied heritage.
I shall always strive to be worthy of it.
I shall respect my parents, teachers and all my elders and treat everyone with courtesy.
I pledge my devotion to my country and its people.
My happiness lies in their well-being and prosperity.

રાજ્ય સરકારની વિનામૂલ્યે યોજના હેઠળનું પુસ્તક



Gujarat State Board of School Textbooks
'Vidyayan', Sector 10-A, Gandhinagar-382010

© Gujarat State Board of School Textbooks, Gandhinagar
Copyright of this book is reserved by Gujarat State Board of School Textbooks.
No reproduction of this book in whole or in part, in any form is permitted without written permission of the Director, Gujarat State Board of School Textbooks.

Subject Adviser

Prof. R. P. Soni

Authors

Dr Harshal Arolkar (Convener)
Dr Sanjay M. Shah
Dr Priti S. Sajja
Dr Kuntal P. Patel

Reviewers

Shri Girish S. Brahmbhatt
Shri Saket A. Dave
Shri Bimal K. Raval
Shri Mayuri I. Shah
Shri Ramaniklal L. Gilatar
Shri Rajanikant A. Pandya
Shri Pankaj R. Shukla
Shri Rajshree N. Padia
Shri Balkrishna Patel
Shri Sejal Trivedi

Co-ordinator

Shri Ashish H. Borisagar
(Subject Co-ordinator : Mathematics)

Preparation and Planning

Shri Haresh S. Limbachiya
(Dy. Director : Academic)

Lay-out and Planning

Shri Haresh S. Limbachiya
(Dy. Director : Production)

PREFACE

The Gujarat State Secondary and Higher Secondary Education Board has prepared new syllabi based on the open source operating system and compatible open source software tools for various topics of Computer Studies. These syllabi are sanctioned by the Government of Gujarat.

It is a matter of pleasure for the Gujarat State Board of School Textbooks to place this textbook of **Computer Studies** prepared according to the new syllabus before the students of **Standard 10**.

Before publishing the textbook, its manuscript has been fully reviewed by experts and teachers teaching at this level. Carrying out suggestions given by teachers and experts, we have made necessary changes in the manuscript and then have published the textbook.

The board has taken special care to ensure that this textbook is interesting, useful and free from errors. However, we welcome suggestions to enhance the quality of the textbook.

Dr. Bharat Pandit

Director

Date : 3-3-2015

Dr. Nitin Pethani

Executive President

Gandhinagar

First Edition : 2014, Reprint 2014

Published by : Dr. Bharat Pandit, Director, on behalf of Gujarat State Board of School Textbooks,
'Vidyayan', Sector 10-A, Gandhinagar

Printed by :

FUNDAMENTAL DUTIES

It shall be the duty of every citizen of India :

- (a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;
- (b) to cherish and follow the noble ideals which inspired our national struggle for freedom;
- (c) to uphold and protect the sovereignty, unity and integrity of India;
- (d) to defend the country and render national service when called upon to do so;
- (e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;
- (f) to value and preserve the rich heritage of our composite culture;
- (g) to protect and improve the natural environment including forests, lakes, rivers and wild life, and to have compassion for living creatures;
- (h) to develop the scientific temper, humanism and the spirit of inquiry and reform;
- (i) to safeguard public property and to abjure violence;
- (j) to strive towards excellence in all spheres of individual and collective activity so that the nation constantly rises to higher levels of endeavour and achievement;
- (k) to provide opportunities for education by the parent or the guardian, to his child or a ward between the age of 6-14 years as the case may be.

CONTENTS

1. Introduction to HTML	1
2. Head and Body Sections	18
3. Handling Images in HTML	37
4. List and Table Handling in HTML	55
5. Introduction to Calc	81
6. Data Editing and Formating in Calc	97
7. Functions in Calc	127
8. Charts in Calc	153
9. Problem and Problem Solving	175
10. Introduction to C Language	192
11. Data Types, Operators and Expression in C Language	216
12. Using I/O Operations	239
13. Decision Structures	255
14. Loop Control Structures	270
15. Arrays	284
16. Function	296
● Appendix - I	309
● Appendix - II	310
● Appendix - III	311
● Appendix - IV	312



About This Textbook...

Dear Teachers,

With a mission to spread computer literacy on a fast track, the Gujarat Government has provided latest computer equipment to more than 6000 aided schools under the ICT@School program. As a new policy initiative, all the schools are given the Ubuntu (a variant of Linux) Operating System and other Open Source software packages so that schools can freely use and exchange the software without bothering about the licensing issues. Since earlier textbooks were largely based on proprietary software, there was a need to rewrite the textbooks based on new syllabus. This was also necessary in view of the fact that the 8th standard has been transferred to primary section. Therefore, new content has been provided for 9th to 12th standard in a phased manner based on the open source Operating System and compatible open source software tools for various topics of computer studies.

This textbook for 10th standard is the second in series for the subject of 'Computer Studies'. Students have already learnt Open Source Operating System-Ubuntu, Open Office Word Processor Writer, Presentation tool Impress and some concepts of Internet surfing, searching, email, file downloading etc. as well as security in 9th standard. In this textbook of 10th standard, they will be initially introduced to basic markup language HTML for presentation of web pages and contents. The Spreadsheet component of Open Office Calc is also discussed in detail. Later they will be learning the problem solving method and the C programming language. For easy comprehension of the language, numerous examples have been given along with explanation. We believe, learning programming will help students to develop logical thinking capabilities.

We hope this coverage will be useful to the students in 10th standard and you will enjoy teaching and conducting practicals using open source Ubuntu operating system and tools.

Dear Students,

Since you now are quite familiar with various definitions around computers, open source operating system Ubuntu as well as other operating systems available in public domain, Open Office components of word processing and presentation tools, Internet and its uses. It is now time to move further and learn advanced topics such as HTML, Spreadsheet Calc and C programming language.

In this textbook the essentials of markup language HTML are covered in chapters 1 to 4. In these chapters, full explanation is given for evolution of HTML, Structure of HTML document, necessary tags for creating documents with examples. Head and Body elements along with their attributes are again explained with relevant examples. Finally how to handle images, lists and tables within HTML code is presented and explained through various illustrations. Chapters 5 to 8 cover the Open Office Spreadsheet component Calc, Coverage of Calc includes Data Editing and Formatting, Calc Functions and generating Charts. Since elementary knowledge of problem solving and programming is very essential, Introduction to C Programming covering Data types, Operators, Expressions, I/O operations, Decision Structures, Arrays and Loop Control Structures and C Functions are covered in chapters 9 to 16. Numerous examples covering the described features have been presented at appropriate place with explanation.

It is expected that if you carefully study the text and practice the laboratory exercises, you will develop reasonable confidence in working with HTML, Calc and C Programming which are although elementary but quite essential topics in understanding computer applications.





Introduction to HTML

Working of the Internet and HTML

Computers are widely used in variety of applications. When computers are connected with each other they can share resources. Such group of connected computers is known as a computer network. The Internet is collection of such multiple computer networks, hence known as network of networks. On the platform of the Internet, a distributed information system exists, which is called World Wide Web, WWW or Web in short. The notion of the Web was conceived in 1991 by Tim Berners-Lee (figure 1.1), while consulting at CERN (a European Organization for Nuclear Research, <http://cern.web.cern.ch/CERN/>) in Switzerland. The Web is a repository of multimedia information on the internet platform. The web content in form of web pages is explored using browsers (special applications to retrieve and view web information). On these web pages, links are placed pointing towards different locations. These links are known as hyperlinks. Clicking on such hyperlink, one can redirect himself to an intended location. This operation is known as following the hyperlink. Any content such as text, picture, graphics, etc. can be embedded with such hyperlink.

The content and hyperlinks cannot be directly expressed on the Web. Hyper Text Markup Language (HTML), is needed to describe how a web page should be displayed by a web browser. Thus the HTML is considered as a language for describing web pages. The HTML is a documentation language to mark content of web pages such as heading, title, table, image, etc. It is machine independent and all Internet browsers accept the content written using HTML code.



Figure 1.1 : Tim Berners-Lee

HTML is a kind of markup language. A markup language is a set of tags that enables additional information (besides the content) on how to present the web content. HTML files are text files that contain additional formatting markup information in form of tags along with its content. The HTML is the most popular markup language; and it offers fixed set of tags. HTML is derived from of SGML (Standardized General Markup Language), which was developed by the International Organization for Standards (ISO) in 1986 to facilitate the sharing of machine-readable documents.

An HTML code is thus a combination of content to be displayed on a web page using browser and tags that helps in guiding the presentation of the content. Without such building block codes, it is impossible to display content on web pages. This makes HTML coding compulsory utility for web page creation, interpretation and presentation.

A Simple HTML Document

Let us create a simple web page that discusses about rainbow using HTML. The contents of the web page are shown in table 1.1.

RAINBOW

Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

Table 1.1 : Text to be displayed on web page using HTML

HTML code to display the contents shown in table 1.1 is given in code listing 1.1.

```
<html>
  <head>
    <title> About Rainbow
  </title>
</head>
<body>
  <h1> RAINBOW </h1>
  <p> Rainbow consists of seven colours. These colours are
      Violet, Indigo, Blue, Green, Yellow, Orange and Red.
      They are also acronymed as VIBGYOR.
  </p>
  <p> Rainbow is caused by reflection of light in water
      droplets in the Earth's atmosphere, resulting in a
      spectrum of light appearing in the sky. It takes the
      form of a multi coloured arc.
  </p>
</body>
</html>
```

Code Listing 1.1 : Sample HTML Code

Observe that besides the content about rainbow, code listing 1.1 also displays several sets of angular brackets with words or letters within them. These brackets and words inside them are known as tags.

A tag is made up of letters, words and numbers enclosed between a left and right angular bracket.

A combination of opening and closing tag along with some content between the two tags forms an element. An HTML element may be empty or can have some attributes to specify the additional formatting and publishing instructions. Figure 1.2 illustrates structure of tags and elements with an example.

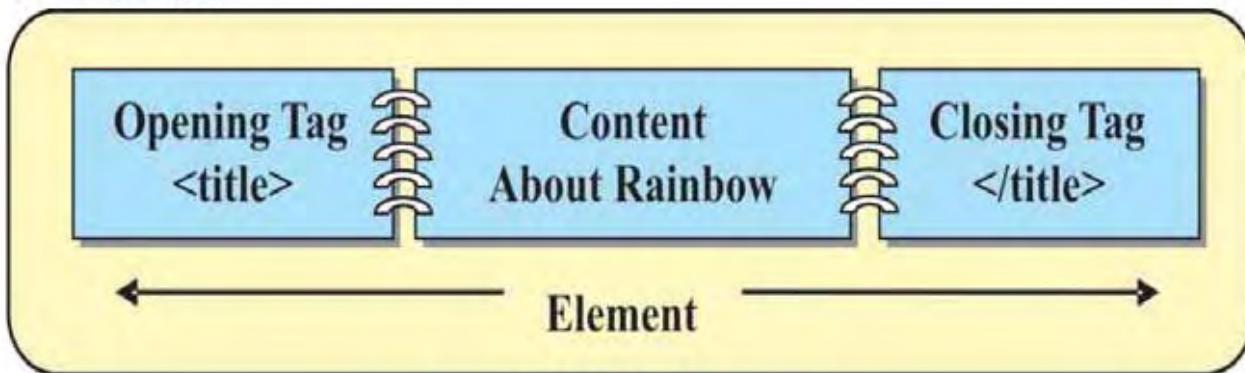


Figure 1.2 : An HTML element

Figure 1.2 indicates that **<title> About Rainbow </title>** forms an element of the HTML code. Observe that a closing tag contents are preceded with a forward slash (**</title>**). It marks the end of an element. Together, the pair of tags and the content within them forms an HTML element. The title tag here defines title of the web page generated by the HTML code.

Another example of such element is as follows :

<h1> RAINBOW </h1>

Here h1 tag refers to heading.

As stated above, content available between **<h1>** and **</h1>** is identified as heading and presented as heading. Similarly, content available between **<p>** and **</p>** is identified as paragraph and presented as paragraph text. The whole document is embedded between opening **<html>** and closing **</html>** tags.

To view how this page will look in a browser, follow the given steps :

Step 1 : Open gedit editor using Applications → Accessories → gedit. The gedit is a general purpose text editor for the GNOME (part of a project called GNU, free software by MIT) desktop environment, Mac OS X and Microsoft Windows. Alternatively you may use a shortcut available for the gedit editor at the header row of the screen.

Step 2 : Type the HTML contents of code listing 1.1 in the empty gedit Window. Figure 1.3 show the look of the gedit editor after you have typed the code. Save the code as "p1.html", by selecting save option shown at the header row of the editor. Note that the HTML file can be saved with html or htm extensions. Figure 1.4 shows method to save the code.

The screenshot shows the gedit text editor window. The title bar reads "*Unsaved Document 1 - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. Below the menu is a toolbar with icons for Open, Save, Undo, and others. The main text area contains the following HTML code:

```
<html>
  <head>
    <title> About Rainbow
    </title>
  </head>
  <body>
    <h1> RAINBOW
    </h1>
    <p> Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.
    </p>
    <p> Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.
    </p>
  </body>
</html>
```

The status bar at the bottom shows "Plain Text" and "Tab Width: 8". The cursor position is "Ln 19, Col 1". There is an "INS" indicator.

Figure 1.3 : The HTML code written in gedit editor

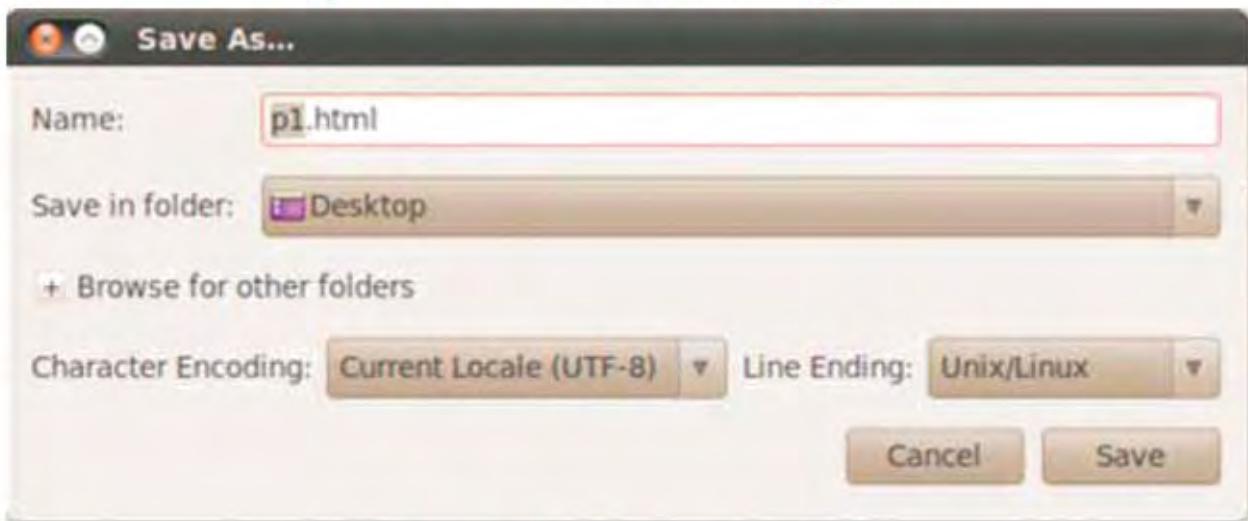
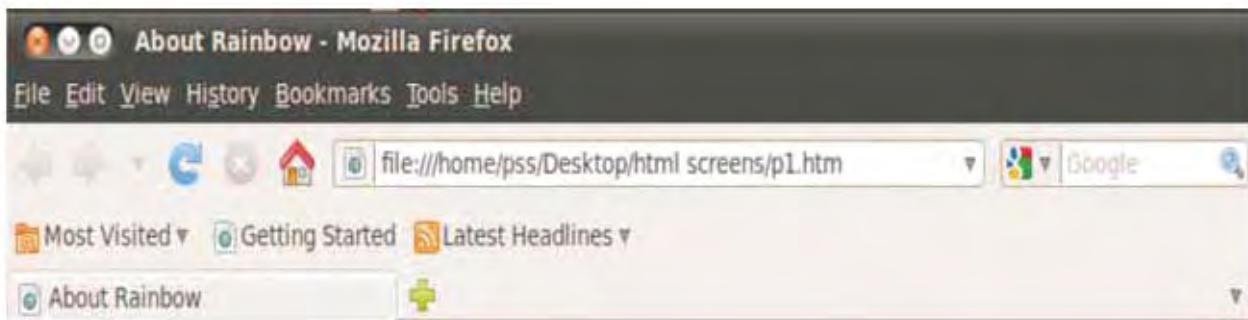


Figure 1.4 : Save As... dialog box

Step 3 : Open a browser such as Mozilla Firefox or any other browser that is installed on your computer. Select File → Open File, you will see an open file dialog box. Select the file that you want to open and click on Open button. Alternatively, you may double click on the file. Figure 1.5 shows the look and feel of the file when opened in Mozilla Firefox.



RAINBOW

Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

Done

Figure 1.5 : The web page generated through the HTML code in a browser

It is to be noted that the tags used in HTML code are not case sensitive. The tags may be written in uppercase letters, lower case letters or mixture of upper and lower case letters. Further, multiple elements can be written in a single line. However, it is advisable to write each element in a new line with proper indentations for the sake of readability.

Structure of an HTML Document

HTML document is structured into two major parts. The first part is head section and second part is body section. They are also known as head element and body element. The head section contains information about the page such as title and description of the page. All these information should be embedded within the `<head>` and `</head>` tags.

The body element is embedded within the `<body>` and `</body>`. This is the content which can be seen within the browser. Both the head and body elements are embedded within the `<html>` and `</html>` tags.

HTML Title

The title of a web page is specified by the `TITLE` element, which should be placed in the head section of the document. It is to be noted that a document should have only one title element. It is used to identify the document content in a general way. Further, the content of title is not a part of the document text. Because of this, it should be simple text and cannot contain special commands such as hyperlinks. The title appears as a label of the window displaying the text. The title also holds a place in a browser's history or bookmark list. It is therefore recommended that title should be short. In the example HTML code given in code listing 1.1, the title is "About Rainbow". It appears at the top of the windows displayed as shown in figure 1.5.

HTML Heading Style

HTML document generally begins with heading. In the example shown in code listing 1.1 heading style 1 (`h1`) tag is used. Observe the heading RAINBOW shown in the figure 1.5 to visualize how the heading style 1 looks. There are five more heading styles available in HTML. Heading can be

created in total six inbuilt sizes named as h1, h2, h3, h4, h5 and h6. These six levels of headings are described in the HTML code given in figure 1.6.



The screenshot shows the Mozilla Firefox browser window with the title bar "Source of: file:///home/pss/Desktop/p2.html - Mozilla Firefox". The menu bar includes "File", "Edit", "View", and "Help". The main content area displays the following HTML code:

```
<html>
  <head>
    <title> About Rainbow
    </title>
  </head>
  <body>
    <h1> Rainbow </h1>
    <h2> Rainbow </h2>
    <h3> Rainbow </h3>
    <h4> Rainbow </h4>
    <h5> Rainbow </h5>
    <h6> Rainbow </h6>
  </body>
</html>
```

Figure 1.6 : HTML code for different levels of heading in HTML

The code is entered using the gedit editor. Save the code as "p2.html". When we see the code in a browser, it presents a view as shown in figure 1.7.

Most browsers display the contents of the <h1>, <h2>, and <h3> elements larger than the default size of text in the document. The content of the <h4> element is similar to the default content size. However, one can always redefine the sizes of these headings.

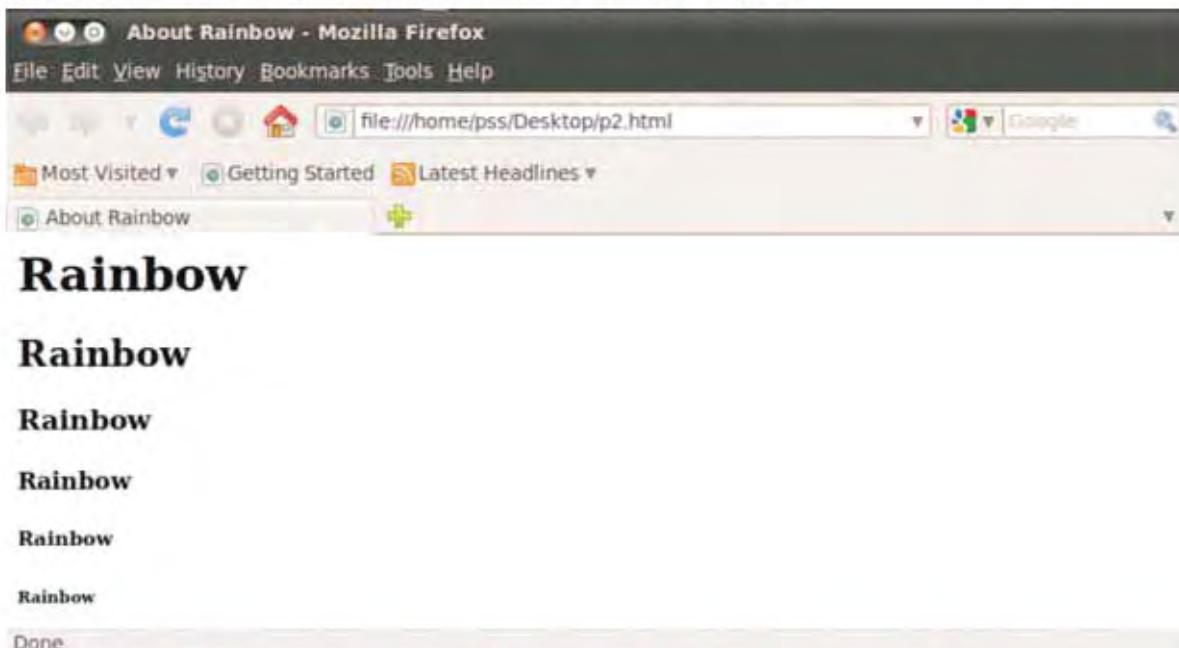


Figure 1.7 : Output of different levels of heading in HTML

Basic Text Formatting Tags

Besides the heading styles such as h1 to h6, there are more basic formatting tags available in HTML, which can be used in body section of an HTML code. Some of the useful tags are as follows :

Paragraph tags : <p> and </p>

The <p> tag structures the content into a paragraph. Each paragraph of text should go in between an opening <p> and closing </p> tag. Following are three valid examples of the same :

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

<p> The second paragraph is here. This is about multiple colours of rainbows.

Though it is a temporary event it leaves a great impact on human mind. </p>

<p> Best of Luck! </p>

Consider the second example from the above listed examples. You may notice the multiple blanks and carriage returns (where enter key is pressed and text appears on a new line) in the second paragraph. It is to be noted that such white space will be considered as a single space. When an HTML code is displayed into a browser, the text will automatically take up the full width of the screen. If you resize the browser window, the browser will wrap the text onto new lines. Such white space management allows developer of HTML code to indent the code and add extra spaces to maintain readability of the code.

**Line Break:
 or
**

Unlike all other tags, the
 element does not have an opening and closing tags.
 is an abbreviated form of break. Such tags are known as empty tags. Advanced versions like XHTML use
 tag. Note that there should be a space between 'br' and '/'.

The
 just pushes the coming text into next line. In case you need multiple lines, simply use multiple
 tags. Examples demonstrating use of the
 are as follows:

**First example

**

Second example

Using
 in paragraph forces compulsory break into the content and disturbs text wrapping while presentation and resizing the browser window.

Preformatted Text

Many a times we want to display text with multiple white spaces and in multiple lines without wanting to be changed it by the browser. For that, we may embed the content into pre-formatted tag set using <pre> and </pre>. Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the given content. Example is as follows :

```
<pre>
```

This is first line.

This is second line.

This is third line.

```
</pre>
```

In a browser the text appears along with given indentations. Also try following examples.

**Bold : and **

This tag is used to display given content into bold letters. The example can be given as follows :

```
<p> This is the <b> first </b> paragraph. </p>
```

Here the "first" word is displayed in bold letters.

Underline : <u> and </u>

This tag is used to display given content with underlined letters. The example can be given as follows :

```
<p> This is the <u> first </u> paragraph. </p>
```

Here the "first" word is displayed in underlined manner.

Italics : <i> and </i>

This tag is used to display given content into italics letters. The example can be given as follows :

```
<p> This is the <i> first </i> paragraph. </p>
```

Here the "first" word is displayed in italics letters.

Strike Through : <s> and </s>

The content of an <s> or <strike> element is displayed with a strikethrough a thin line through the text. Here the 's' is an abbreviated form of 'strike'. Example of presentation of strikethrough content is given as follows.

```
<p> This is the <s> cancelled </s> paragraph. </p>
```

Here the "cancelled" word is displayed in strikethrough manner.

Type writer font: <tt> and </tt>

The content of a <tt> element is written in typewriter type of fonts, which is also identified as mono-spaced font (like that of a teletype machine). Example is as follows :

```
<p> This is the <tt> first </tt> paragraph. </p>
```

Here the "first" word is displayed in mono-spaced fonts.

There are some other elements which are described in table 1.2.

Elements	Description
<small> and </small>	The content is displayed one font size smaller than the rest of the text surrounding it.
<big> and </big>	The content is displayed one font size bigger than the rest of the text surrounding it.
^{and}	The content is displayed in superscript.
_{and}	The content is displayed in subscript.
<acronym> and </acronym>	It defines the content as an acronym.
<dfn> and </dfn>	It defines a special term.
<q> and </q>	It defines a quote.

Table 1.2 : Some other formatting tags

Anchor Tag

When text is displayed within an HTML document, besides the content and format specification, some extra information or reference to other entity is needed. Many times further explanation is also required. Set of such words or text that appears in different colour (generally blue and underlined) are called hyperlink. A hyperlink is created using an <a> element, where the 'a' stands for an anchor. Let us modify the file "p1.html" as shown in code listing 1.2.

```

<html>
  <head>
    <title> About Rainbow
  </title>
  </head>
  <body>
    <h1> RAINBOW </h1>
    <p> Rainbow consists of seven colours. These colours are Violet, Indigo, Blue, Green, Yellow, Orange and Red. They are also acronymed as VIBGYOR.
    </p>
    <p> Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.
    </p>
    <p>
      <a href= "p4.html" > Click here to visit Theory of Rainbow. </a>
    </p>
  </body>
</html>

```

Code Listing 1.2 : HTML code for showing use of hyperlink

Save the file as "p3.html". Figure 1.8 shows the output of this code when viewed in browser.



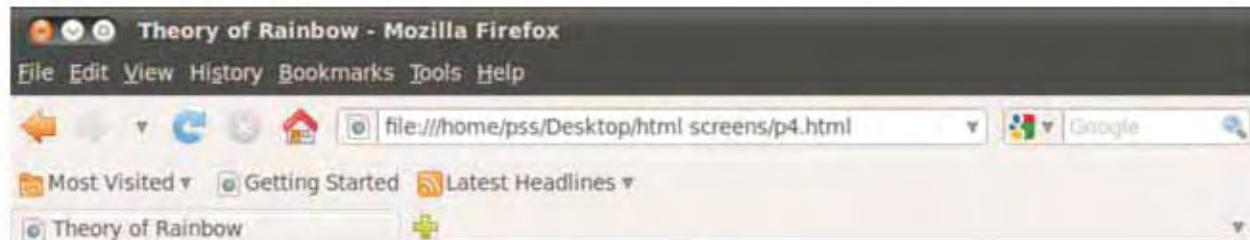
Figure 1.8 : Output of p3.html

When user clicks on the hyperlink, an intended file describing the 'Theory of Rainbow' must be opened. Let us create an HTML code for the file referred by the hyperlink as shown in code listing 1.3.

```
<html>
  <head>
    <title> Theory of Rainbow
    </title>
  </head>
  <body>
    <h1> How Rainbow Developed </h1>
    <p> Rainbow is caused by reflection of light in water droplets
      in the Earth's atmosphere, resulting in a spectrum of light
      appearing in the sky. It takes the form of multi coloured arc.
    </p>
  </body>
</html>
```

Code Listing 1.3 : HTML code linked to hyperlink

Save the code shown in code listing 1.3 as "p4.html". When you click on the linked (anchor) text ([Click here to visit Theory of Rainbow.](#)) shown in figure 1.8. The contents of p4.html will be loaded in the browser, see figure 1.9.



How Rainbow Developed

Rainbow is caused by reflection of light in water droplets in the Earth's atmosphere, resulting in a spectrum of light appearing in the sky. It takes the form of a multi coloured arc.

Done

Figure 1.9 : Contents of p4.html

The hyperlink is a way to link two HTML documents by creating a hyper text in a document and giving reference of the other document to it. A website, which is a collection of many web pages, manages links through such hyperlink management. Here it is to be noted that, the web pages we have created are just presenting given content in a formatted way as we have specified them into the HTML code. Such web page contains only static (fixed) information, hence known as static web pages.

Absolute or Relative Address

Observe the line ` Click here to visit Theory of Rainbow. ` shown in code listing 1.2.

Instead of giving a full address such as `http://www.somedomain.com/p4.html`, we have given only the file name "p4.html". Giving just a file name will work only when you have the calling file (p3.html, also called parent file) and called file (p4.html, also called referred file) in the same directory. The location of the called file is relative to the calling file. Hence it is known as a relative address. While the complete address is known as an absolute address. If the referred file is located one directory above, we may prefix `../` to the filename.

Note :

If no path is provided, the browser will understand that the referred file is located in the same directory where the parent file is stored.

Attributes to the Tags

To specify more information along with tags, additional attribute accompany the tags. In other words, attributes tell more about the elements. Attributes always appear on the opening tags of the elements that carry them. An attribute is made up of two parts. The first part is a name and the second part is a value.

The name of an attribute indicates the property to be set. In case of <a> tag demonstrated in code listing 1.2 has name href. The value is a value to be set to the property. In case of the href, the value is p4.html (the reference to the link). The values should be in double quotation marks. Between the name and the value there should be an equal (=) sign. See next section for example showing how to define a tag with an attribute.

Align Attribute

The align attribute indicates whether the heading appears to the left, center, or right of the page. By default, the content is aligned to the left of the page. It can take three values as follows.

Left : The content is aligned to the left of the page.

Right : The content is aligned to the right of the page.

Center : The content is aligned to the center of the page.

Following are some examples demonstrating use of the align attributes :

<p align="right"> This content will be displayed in right aligned form </p>

<p align="center"> This content will be displayed in center position of the page </p>

There are some attributes which can appear along with every tag. Such attributes are called universal attributes. Align is such universal attribute. Being a universal attribute, the align attribute can also go with heading as shown below.

<h1 align="center"> Centered Heading </h1>

When content given in a paragraph is aligned, some spaces are automatically added for adjustment. The spaces which are inserted automatically are known as soft spaces. If users himself (manually) inserts some spaces, such hard spaces will be automatically deleted unless the content is written using <pre> and </pre> tag pairs.

Working with other Editor - SciTE

SciTE is a text editor based on a free source code editing component called Scintilla [<http://www.scintilla.org>]. It comes with complete source code and a license that permits use in any free project or commercial product. The interface of the SciTE is shown in figure 1.10.



Figure 1.10 : Interface of the SciTE editor

The HTML code written above that prints information about rainbow (See example in figure 1.3) can be written in the SciTE editor as shown in figure 1.11.

The screenshot shows the SciTE editor interface. The title bar says "rainbow.html - SciTE". The menu bar includes File, Edit, Search, View, Tools, Options, Language, Buffers, and Help. Below the menu is a toolbar with various icons for file operations like new, open, save, cut, copy, paste, etc. The main window displays the following HTML code:

```
- <html>
-   <head>
-     <title> About Rainbow </title>
-   </head>
-   <body>
-     <h1> RAINBOW </h1>
-     <p> Rainbow consists of seven colours.  
These colours are Violet, Indigo, Blue, Green, Yellow, Orange  
and Red. They are also acronymed as VIBGYOR.
-     <p> Rainbow is caused by reflection of light in water droplets in  
the Earth's atmosphere, resulting in a spectrum of  
light appearing in the sky. It takes the form of a multi  
coloured arc.
-   </body>
</html>
```

Figure 1.11 : HTML code in SciTE editor

Notice the menu bar items at the top of the screen shown in the figure 1.11. The menu items include basic facilities for file operations, editing facilities, searching, viewing, and other tools, options, etc. Next line beneath the menu items represents some icons for basic operations such as new, open, save, edit, search, etc. You may open any existing code in the SciTE editor as follows.

Step 1 : Locate the file you want to open.

Step 2 : Right click on it and choose Open With. You will see interface as shown in figure 1.12.

Step 3 : Select SciTE Text Editor. It will open the file in the SciTE editor. You may note the indicators for indentations and colour of tags shown by the SciTE. Having such indentations and tags in different colour separates the content from the tags and increases ease of reading the code.

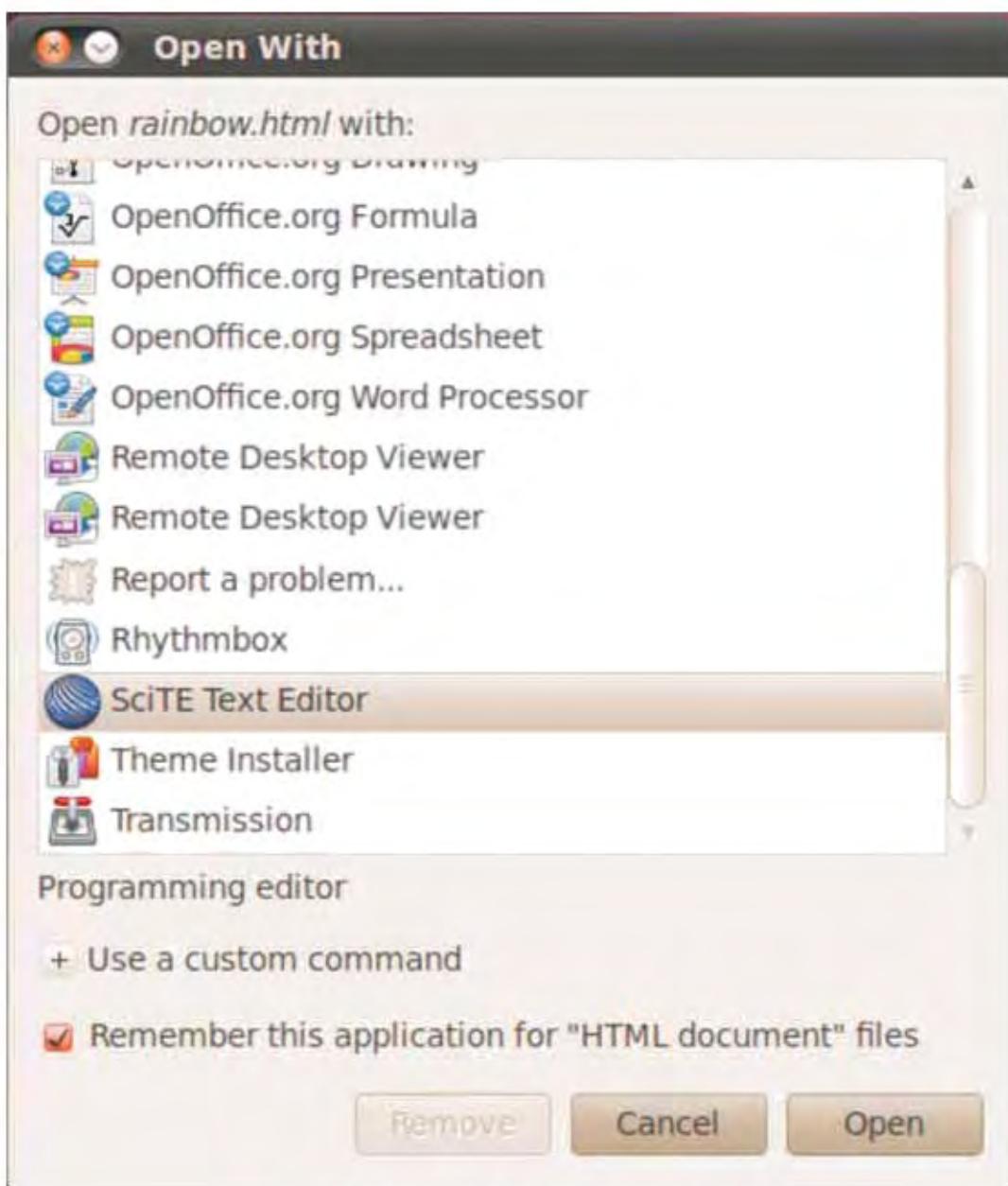


Figure 1.12 : The Open With dialog box

Step 4 : To view the html output in browser select Tools → Go or press F5.

Current Version of HTML

During the process of its evolution the HTML kept its focus on its purpose so that the composing and publishing content remains effective. With the progress of time, more and more functionalities were added to the HTML so that it can be useful and compatible with new browsers, new technologies and ever increasing developer demands. Latest version of the HTML is called HTML 5.0 and is becoming popular now.

Tags Covered in this Chapter

In this chapter we have discussed the tags mentioned in table 1.3.

Tag	Description
<code><a href> ... </code>	Anchors parent file to the referred file through hot text (link)
<code> ..</code>	Displays text in bold fonts.
<code><body>...</body></code>	Defines body of the HTML document. Appears within the <code><html></code> tag pair.
<code>
 or
</code>	Defines line break. It is an empty singular tag.
<code><h1>...</h1></code>	Defines a first level heading.
<code><h2>...</h2></code>	Defines a second level heading.
<code><h3>...</h3></code>	Defines a third level heading.
<code><h4>...</h4></code>	Defines a fourth level heading.
<code><h5>...</h5></code>	Defines a fifth level heading.
<code><h6>...</h6></code>	Defines a sixth level heading.
<code><head>...</head></code>	Defines the head section of an HTML document. Appears within <code><html></code> tag pair.
<code><html>...</html></code>	Covers the entire HTML document.
<code><i> ..</i></code>	Displays text in italics fonts.
<code><p>...</p></code>	Defines a paragraph
<code><pre>...</pre></code>	Displays preformatted text.
<code><s> ..</s></code>	Displays text in strikethrough manner.
<code><title>...</title></code>	Defines title of the document. Appears within the <code><head></code> tag pair.
<code><tt> ..</tt></code>	Displays text in typewriter fonts.
<code><u> ..</u></code>	Displays text in underlined fonts.

Table 1.3 : HTML Tags Covered in Chapter 1

Summary

In this chapter we learnt how to design a web page using HTML. HTML is a fundamental utility that describes how the web content is composed, published and retrieved using web browsers. In this chapter, two important components of the structure of an HTML document such as head and body are described with some basic tags. Using the information you can create simple web pages. You may also link multiple web pages created using the anchor tag discussed in this chapter. Besides providing fundamental concepts of the HTML and history of the HTML; the chapter also provides information about editors such as gedit and SciTE to build HTML documents.

EXERCISE

(9) Which type of information can be incorporated in an HTML document ?

- (a) Multimedia information
- (b) Text information
- (c) Address and path of filename
- (d) All of these

(10) Which of the following is an editor to edit an HTML document ?

- (a) SciTE
- (b) BrnTE
- (c) Light
- (d) SprITE

LABORATORY EXERCISE

1. Develop a web page that provides introductory information about your school. Give heading of this page as 'My School'. Use necessary formatting and presenting tags.
2. Develop a web page that introduces your class. Include information such as your class teacher, other course teachers and subjects you are learning. Give heading of this page as 'My Class'. Use necessary formatting and presenting tags.
3. Modify the web page created in question 1 of this exercise; to create a reference to another web page you have created in question 2 of this exercise. Set hot text in such a way that, when it is clicked, from the 'My School' page it will jump to the 'My Class' page.



Head and Body Sections

Head Section

An HTML document is divided into two sections called Head and Body. The head section of the HTML document provides necessary information about the HTML document. The content of the head section is included within `<head>` and `</head>` tags. Combination of these head tags and content between these is known as a head element.

The first obvious thing that can be included into the Head section is the title. It is to be noted that the title will not be displayed as a content of the web page. It is displayed as a title of the browser window that shows the web page.

It is to be noted that if a title is not specified, most browsers display the URL path or file name. Further, the title tag must have its closing pair. Failing to use the closing title tag, the whole content will be considered as title and it may be possible that the HTML document may not be loaded into the browser window.

The head section also contains additional information about the content and the HTML document. The tag that provides additional information is known as meta-tag. Meta-tags are used to store information usually relevant to browsers and search engines. Addition of appropriate meta-tags describes nature of the web page precisely and makes it easy for a search engine to search the web page efficiently.

Description

Most search engines will display the description when they put the results of a search to the users. In absence of such description, the search engine will display only first few words. An example of the description attribute of a meta-tag is as follows :

```
<meta name="DESCRIPTION" content="About rainbow and its colours">
```

Keywords

Keywords provided in this tag will be used by the search engine. Names of important characteristics, objective of the web page and important topics may be enlisted as keywords. Example of the meta-tag attribute that describes keywords is as follows.

```
<meta name="KEYWORDS" content="Rainbow, VIBGYOR">
```

Author

The following meta-tag attribute provides information about the author of the web page.

```
<meta name="AUTHOR" content="M K Gandhi">
```

Comments

Comments allow you to provide additional information in the HTML code. The comments are not meant to be displayed. That is no one can see the comments unless they look at the HTML source code. To add a comment we use '<!--' and '-->' tags. The '<!--' tag represents beginning of comment while the '-->' tag represents end of comment. Comments are also referred as prologue. One example of comment is <!-- This is a comment -->.

Other meta-tags

There are some meta-tags that tell the web page to load a specific URL after some seconds or tags telling it that a certain page should not be cached. The following example refreshes the web page (by reloading it) after every 5 seconds :

```
<meta http-equiv="REFRESH" content="5">
```

Following example refreshes the content of the given URL, <http://test.com/> every 5 seconds :

```
<meta http-equiv="refresh" content="5; URL='http://test.com/'>
```

HTML base

The `<base>` tag specifies the base URL/target for all relative URLs in a page.

```
<head>
<base href="http://test.com/" >
</head>
```

Table 2.1 summarizes important HTML head elements.

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

Table 2.1 : Head elements of HTML document

Consider the HTML code as shown in figure 2.1 to experiment with the head elements.

The screenshot shows a text editor window with a menu bar (File, Edit, Search, View, Tools, Options, Language, Buffers, Help) and a toolbar with various icons. The file name '2_2.html' is displayed in the title bar. The code in the editor is:

```
- <html>
  - <head>
    <meta name="description" content="About rainbow and its colours">
    <meta name="keywords" content="Rainbow, VIBGYOR">
    <meta name="author" content="M K Gandhi">
    <!-- This is a comment -->
  </head>
  - <body>
    -   <p> <h1>Nothing will be printed but this line... </p>
  </body>
</html>
```

At the bottom of the editor, status bars show 'l=22 co=1 INS (CR+LF)'.

Figure 2.1 : Experimenting head elements of HTML

Implement the above code and see its output in a browser. It will look similar to figure 2.2.

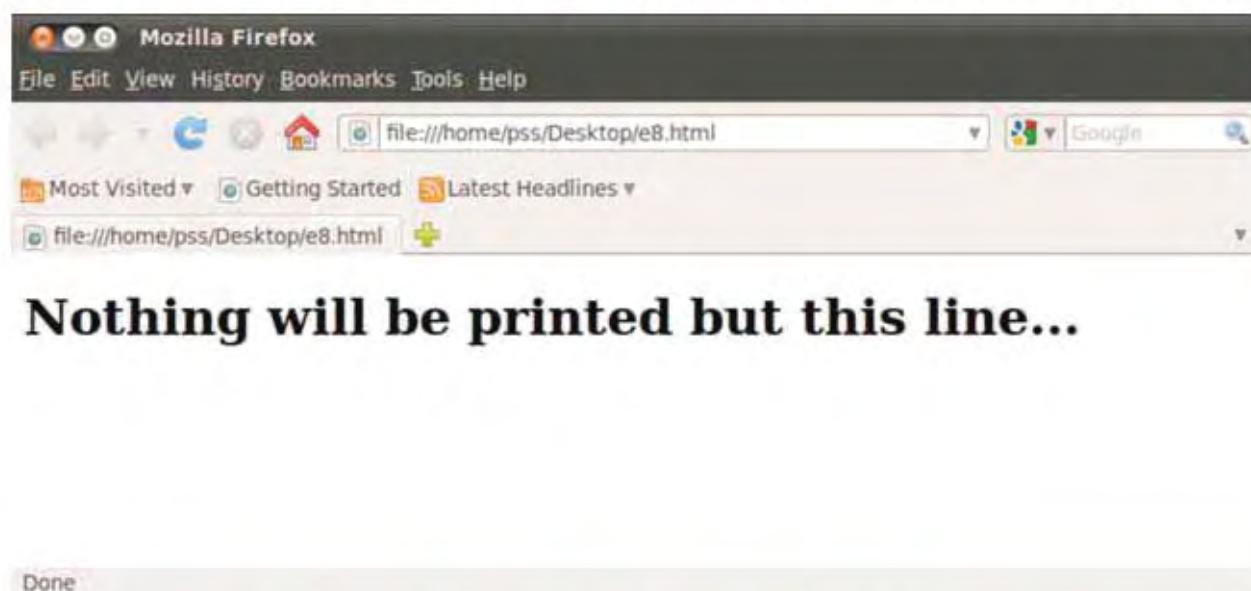


Figure 2.2 : Output of the HTML code shown in figure 2.1

Body Section

The body section of an HTML document is written between `<body>` and `</body>` tags. The HTML body element acts as a container of the items which have to be displayed within a browser.

In previous chapter we have seen some tags related to heading, paragraph and formatting text that can appear in a body section. These tags usually require some content between them. There are some tags, which do not require any content; and hence are called empty tags. Example of such empty tag is `
`. Let us see some more tags and attributes used in the body section of HTML document.

Background Image

So far, we have seen HTML code that presents content on simple white background. To present the content on decorative and colourful background, we can use the background attribute with body tag. For the text background, we may use any image available with us in our computer. To set a background of HTML page we use the following tag:

```
<body background = "rainbow.jpg">
```

Here the term background is an attribute. The attribute has to be provided with a value. This value is a name of a file (normally an image file) that we want to display as background. In our case it is `rainbow.jpg`. We can provide any valid image formats like jpg, bmp, png, and tiff.

```
File Edit Search View Tools Options Language Buffers Help  
2_3.html  
- <html>  
-   - <head>  
-     <meta name="description" content="About rainbow and its colours">  
-     <meta name="keywords" content="Rainbow, VIBGYOR">  
-     <meta name="author" content="M K Gandhi">  
-     <!-- This is a comment -->  
-   </head>  
-   - <body background="rainbow.jpg">  
-     <p>  
-       <h1> Nothing will be printed but this  
-           line in a rainbow background... </h1>  
-     </p>  
-   </body>  
- </html>  
ll=25 co=1 INS (CR+LF)
```

Figure 2.3 : HTML code to add a background image

Let us write HTML code as shown in figure 2.3 that sets background with an image called `rainbow.jpg`. Note that you may replace this file name with an image file name of your choice.

When we use a file within the HTML code, we have to be sure that the filename (along with the right path) and file extension are specified correctly. In case there is mistake in specifying the file name nothing will be visible in the background of the page.

Write the code as shown in figure 2.3 and save it with your desired file name (here it has been saved as e7.html). View the code in browser. Figure 2.4 illustrates the result of adding background image.



Figure 2.4 : Adding an image as a background

Many times, the background image is much more attractive than its content. We must take care that the background image must not take reader's focus from the content. The background image must be sober and adding an appeal to the content, instead of overriding it. Further, if the background image is too big, it may require lot of time to load the page in browser. Note that if the image used for the background is smaller in size than the screen size, the image may be replicated until it fills the entire screen. Generally the background image will scroll when you scroll down the page, unless you have set it to be fixed as follows:

```
<body background="rainbow.gif" bgproperties="fixed">
```

Background Colour

We can make the web page attractive by using background colour instead of using a background image. Consider if we want a background with yellow colour; following tag will work.

```
<body bgcolor="#FFFF00">
```

Here bgcolor is an attribute defining background colour. Note that we can also set both background image and colour together as shown:

```
<body background="rainbow.jpg" bgcolor="#FFFF00">
```

In such cases the background colour will be displayed till image is completely loaded in the browser. This effect will be generally visible on slow computers; on fast computers we may not be able to see any effect.

Any colour in an electronic media (such as television and computer screens) is considered as a combination of three basic colours namely red, green and blue. They are also acronymed as RGB. Recall, in your childhood, you were mixing two or more colours to get a new colour! Often you may have mixed blue and red colours to get purple colour !

Colours in computer are coded as degree from 00 to FF in hexadecimal. Hence, to represent a colour we need to build a code representing some red, some green and some blue (RGB) colour making it six digit code. That is, two digits for each colour. The string that defines the red colour is "FF0000". Here the red colour is assigned value FF (hex equivalent of decimal value 255), green is assigned a value 00, and blue is assigned a value 00. All these three sets of digits have ranges from 00 to FF (that is from 0 to up to 255; making total of 256 values). The combination of Red, Green, and Blue values from 0 to 255, gives more than 16 million different colors ($256 \times 256 \times 256$). By mentioning colour strings in hexadecimal format, we can specify various colours. Some examples of colours are shown in figure 2.5 :

FFFFFF→	White colour	→
FF0000→	Red colour	→
FFFF00→	Yellow colour	→
000000→	Black colour	→
FF00FF→	Pink/Magenta colour	→
0000FF→	Blue colour	→
00FF00→	Green colour	→

Figure 2.5 : Some examples of colour

Alternatively you may use the colour names also as shown below.

```
<body background="rainbow.jpg" bgcolor="Green">  
<body background="rainbow.jpg" bgcolor="Chocolate">
```

You may produce various colourful backgrounds by changing just the colour string. Try the HTML code shown in code listing 2.1.

```
<html>  
  <body bgcolor="#FF00FF">  
    <h1>Everything is pink here, if you call this pink .....</h1>  
  </body>  
</html>
```

Code Listing 2.1 : HTML code for adding background

When viewed in the browser it will look similar to figure 2.6.

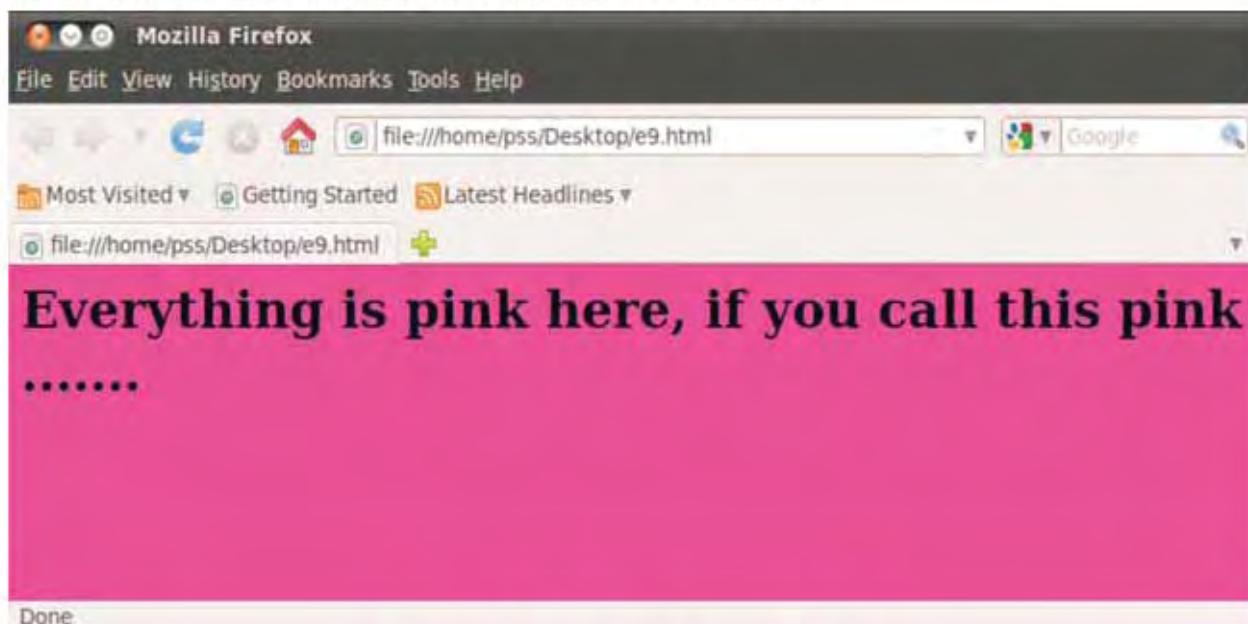


Figure 2.6 : Use of background colour

Text Colour

As we have defined background colour and images using attributes along with the body tag, we can also define text colour with attribute text along with the body tag. Assume that we write the following example.

```
<body text="#FF0000">
```

The use of text attribute with value "FF0000" will set the text colour as red. Suppose, you want to set background colour as pink and text colour to be yellow, you can do this in single instruction as follows.

```
<body bgcolor="#FF00FF" text="#FFFF00">
```

Link Colour

Generally you might have seen a blue coloured text pointing to different locations/pages within a web page being displayed. When such a text is clicked, it takes you to another page/resource. When you come back to the original page, you might observe that the colour of this text changes. As you have visited the link, it is identified by the web page as a 'visited link'. Such a visited link is identified as vlink. While the link that is being visited is called an active link. Active link is identified as alink. To distinguish visited links and active links, different colours are used. See the example below:

```
<body alink="#00FF00">  
<body vlink="#FF0000">
```

Alternatively, you may write

```
<body vlink="pink">
```

The `alink` attribute used in the above example sets the colour for active links within the document. Active link represent the state of a link as it is being clicked. The `vlink` attribute used in the above example sets the colour for hyperlinks within the document that have already been visited.

Horizontal Line

The `<hr />` element displays a horizontal line across the page. It is an empty element like `
` element. This is used to separate content by breaking it into sections. It is also called horizontal rule. Consider example shown in figure 2.7.



The screenshot shows the SciTE code editor interface. The title bar says "e10.html - SciTE". The menu bar includes File, Edit, Search, View, Tools, Options, Language, Buffers, and Help. A single buffer tab labeled "1 e10.html" is open. The code in the buffer is:

```
- <html>
- <head>
    <title>About Rainbow</title>
</head>
- <body>
    <hr/>
    This line
    <hr/> <hr/>
    is broken in
    <hr/>
    three parts.
</body>
</html>
```

Figure 2.7 : Illustrating a horizontal rule

The output of code written in figure 2.7 is shown in figure 2.8.

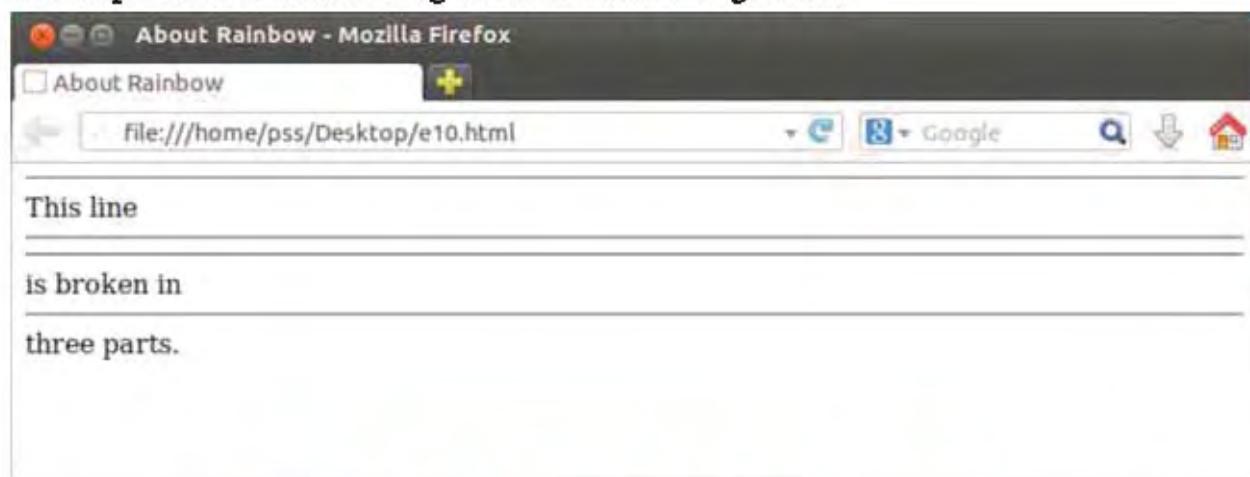


Figure 2.8 : How browser displays the `<hr>` command

There are five attributes which can be used with the horizontal line. These are given in the table 2.2.

Attributes	Description
size	Defines the thickness of the line. You may use pixel size 3, 5 or even 10.
color	Defines the line colour in explorer such as the Mozilla Firefox. The line colour may be defined in hexadecimal.
width	Defines the length of the line. The length can also be defined as in percentage of the width of the browser's window. The default value of the width is 100%.
align	Defines alignment of line in RIGHT, LEFT, or CENTER manner.
noshade	Displays a solid plain line instead of shaded line.

Table 2.2 : Attributes of a horizontal rule

Assume that the user wants to display a horizontal rule of size 2 that uses only 25% of the screen size and is aligned to the right side of the screen then following tag can be used.

<hr size=2 width="25%" align="right" >

Note that in HTML closing of **<hr>** tag is not compulsory. In addition to the above arguments, we can also fill the horizontal line with some images. Embed this line into a valid HTML code and view it in a browser.

Example of Preformatted Text

Consider a scenario where we would like to display a notice about a forthcoming event. The content of the notice is as shown in table 2.3.

Practicing Rainbow Colours

Attention Please !

The rainbow is made up of seven colours. When we mix all seven colours they will become a white colour. To demonstrate this practically, we have arranged a laboratory session. The detail of the laboratory session is as follows:

Date	:	6 August
Time	:	9 : 30 am
Place	:	First Floor, Lab-1
Instructor	:	I M Patel

Principal

Table 2.3 : Contents of notice

Let us first make a few observations from the information given to us in form of table 2.3.

1. The text shown in table 2.3 must be included in body section of the HTML document.
2. The first line ('Practicing Rainbow Colours') is heading line and appears in center. We may use **<h1>** tag with attribute align="center" to display it appropriately.
3. The second line ('Attention Please!') is normal text. We may use **<p>** tag with attribute align="center" to display it appropriately.
4. There is a normal text informing us about the experiment going to take place in a laboratory. We may display the content using simple paragraph tag.
5. The next four lines that give information about date, time, venue and instructor are presented in different way. This is an example of preformatted text. We can show it exactly in the same way by using tag pair **<pre>** and **</pre>**.

The HTML code that publishes content shown in table 2.3 is given in code listing 2.2.

```
<html>
<head>
<title> Notice </title>
</head>
<body>
<h1 align= "center "> Practicing Rainbow Colours </h1>
<p align =center> Attention Please! </p>
<p> The rainbow is made up of seven colours. When we mix all seven colours
    they will become a white colour. To demonstrate the experiment,
    we have arranged a laboratory session. The detail of the laboratory
    session is as follows:
</p>
<pre>
    Date      : 6 August
    Time     : 9:30 am
    Place    : First Floor, Lab-1
    Instructor : I M Patel
    Principal
</pre>
</body>
</html>
```

Code Listing 2.2 : HTML code for setting tabs

The output of the code is shown in figure 2.9.



Figure 2.9 : Preformatted text in HTML

Formatting Characters

In the previous chapter we have seen that how text can be made bold, italics, and underlined. When we physically indicate the formatting style, browser will follow the instructions strictly and publishes the content in the said manner. Such tags are known as physical style tags. Table 2.4 summarizes the physical style tags.

Tag	Description
 ... 	Displays the content in bold fonts.
<i> ... </i>	Displays the content in italics.
<strike> ... </strike>	Displays the content with a line strike through it.
_{...}	Displays the content as a subscript.
^{...}	Displays the content as a superscript.
<tt> ... </tt>	Displays the content in a fixed typewriter-like font.
<u> ... </u>	Displays the content as underlined text.

Table 2.4 : Physical style tags

There is another way to format text. Instead of physically mentioning the formatting instructions in the browser, we just have to tell the browser what we want. Tags that manage such formatting are known as logical style tags. Following are example of some logical style tags.

** and **

The content is displayed in emphasized manner using this tag pair. Important things such as "Must be done" and "Important" can be written using this tag. Example of this tag is as follows.

What a beautiful rainbow...!

** and **

The content is displayed in much emphasized manner using this tag pair. Example of this tag is as follows.

** What a beautiful rainbow...!**

Observe the difference between the emphasized and strong text content in the above examples. Some other tags that manage the logical formatting are summarized at table 2.5.

Tag	Description
<dfn>	To publish the text content in defined fashion.
	To give emphasis on specific text.
<cite>	For citing important text such as titles of books, films, etc.
<code>	To demonstrate computer programming code segments. The text is displayed in a fixed-width font.
	This tag prints the content in strong emphasis manner. The content is typically displayed in bold.

Table 2.5 : Logical style tags

Font Tag

The font tag is used to set a specific font and size. Some examples are shown below:

```
<p><font size="3" color="red">This is some text!</font></p>
<p><font size="2" color="blue">This is some text!</font></p>
<p><font face="verdana" color="green">This is some text!</font></p>
```

Here you may notice that instead of a hexadecimal number, we have used colour names such as green, blue and red. The above examples have also used attributes along with the font tags. These attributes are size and colour; which specify size of the content and colour of the content respectively.

At the beginning of the HTML code, it is possible to decide the default font and its attributes for the page. This can be done as follows.

```
<basefont face="Arial" size="16">
```

The above line can be tested by writing the full HTML code as shown in code listing 2.3.

```
<html>  
  
<body>  
  
    <basefont face = Arial size=16>  
  
    <p><font color="red">This is some text!</font></p>  
    <p><font color="blue">This is some text!</font></p>  
    <p><font color="green">This is some text!</font></p>  
  
</body>  
  
</html>
```

Code Listing 2.3 : Sample HTML code for testing font tag

Try to write and view this code in a browser. The output will look similar to the one shown in figure 2.10.

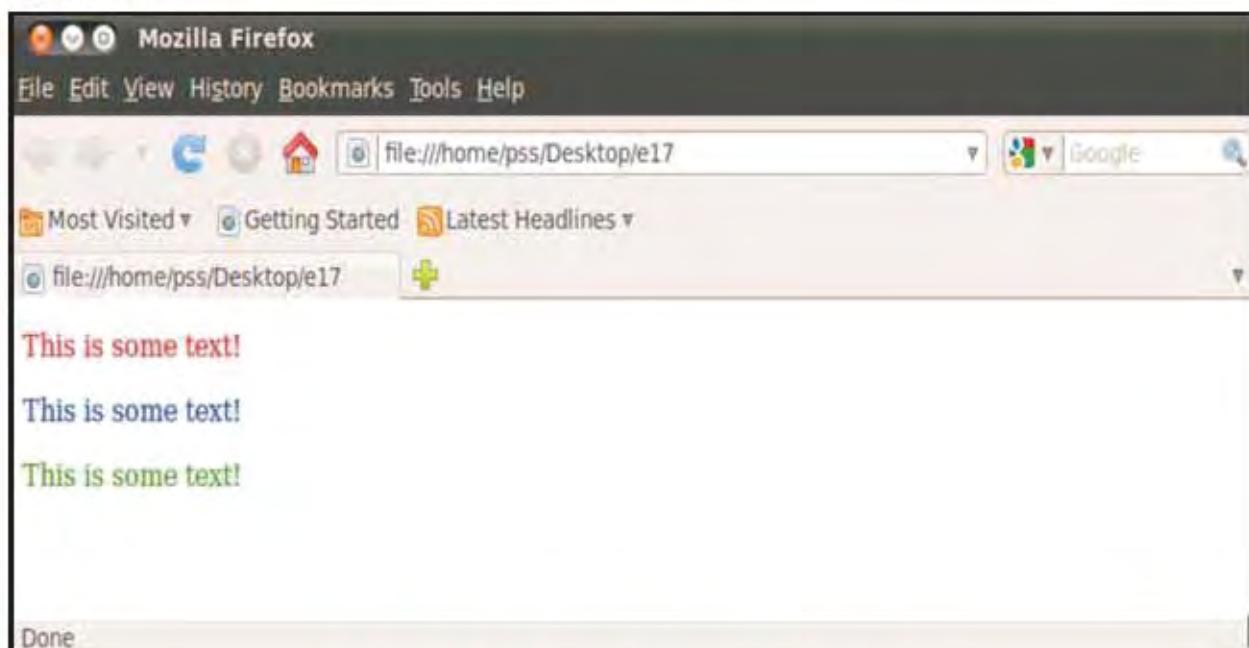


Figure 2.10 : Base font and font colour

Note :

Observe that the file name visible in figure 2.10 is e17 instead of e17.html. Linux generally does not use the file extensions. When an extension is used it indicates the content or usage of that file. For example in our case the HTML extension tells us that it is a HTML file. To view the desired output user on Linux does not need a file extension but needs to make sure that a proper program is used to open the file.

Special Characters

Special characters such as <, &, ©, etc. can be included on the web page along with & (ampersand) as a prefix of the mnemonic code (symbol) of the letter. Instead of such symbolic code, we can also give the ASCII value for the symbol. This is required because the characters like '<' and '>' cannot be used directly in an HTML document, as they can be mixed with tags. Use the special character shown in table 2.6.

Symbol	Description	Mnemonic code	ASCII value
<	Less than	lt	60
>	Greater than	gt	62
&	Ampersand	amp	38
©	Copy right	--	169
¼	One quarter	--	188
½	Half	--	189
¾	Three quarters	--	190
®	Registered trademark	erg	174

Table 2.6 : Mnemonic codes and ASCII values for special characters

Here is an example demonstrating use of less than (<) and greater than (>) characters within an HTML code.

```
<p align="center"> &lt Basic Information &gt;</p>
```

The above line will print following text on a web page.

```
< Basic Information >
```

Tags Covered in this Chapter

In this chapter we have discussed the tags mentioned in table 2.7.

Tag	Description
<!-- ---->	To embed text of comments in an HTML code.
	Displays the content in bold fonts.
<base>	Specifies the base URL/target for all relative URLs in a page.
<basefont>	To set default font for the web page.
<body>	Defines body section of an HTML document. It has attributes such as background image, background colour, text colour, link colour, etc.

Tag	Description
<cite>	For citing important text such as titles of books, films, etc. the text is typically displayed in italics.
<code>	To demonstrate computer programming code segments. The text is displayed in a fixed-width font.
<dfn>	To publish the text content in defined fashion. Typically the text is displayed in italics.
	The content is displayed in emphasized manner.
	Font tag is used to set specific fonts and size.
<head>	Head section of an HTML document.
<hr>	Defines a horizontal rule. It has attributes such as size, colour, width, alignment, no shade etc.
<i>	Displays the content in italics.
<link>	Defines the relationship between a document and an external resource.
<meta>	Provides additional information to search engine and other utility programs about author, keywords, description, purpose etc.
<script>	Defines a client side script.
<strike>	Displays the content in a strikethrough manner.
	The content is displayed in much emphasized manner.
<style>	Defines style information for a document.
<sub>	Displays the content as a subscript.
<sup>	Displays the content as a superscript.
<tt>	Displays the content in a fixed typewriter-like font.
<u>	Displays the content as underlined text.

Table 2.7 : HTML tags discussed in Chapter 2

An Example

Let us consider a real life example. Suppose you want to build a simple website of your school. The website is compilation of school's resources and activities. The main page of the website of the school may be known as index or homepage. On the first web page, name of the school is published. We may have additional information about school's affiliation to a trust, registration number, full address and contact information on this home page itself. Besides this, the home page also has links to various other pages such as teaching activities, photo gallery, and programmes and events.

Description of these web pages is given as follows :

- Home page of my school: The main page of the schools website include the school name, brief information about the school trust, school registration number, affiliation of the school to some government body and school contact details besides menu for other pages.
- Staff and teaching activities: This page shows brief introduction of the staff members including full name, qualification, classes they normally teach and their specialization along with their email address.
- Photo gallery: This page shows photos of classes, library, laboratory and recent events. You may develop this page later when you learn about image handling.
- Circular and Notices: This page contains information about forthcoming programmes and events. It also displays circular and notice for students and parents. For example, notice for sports day, essay competition, results and announcement of scholarship programme.

Consider HTML code given in code listing 2.4.

```
<html>
<head>
    <title>About my school</title>
</head>
<body vlink="purple">
<h1> My School </h1>

<p> <h3>
    <a href="Home.html">Home</a>
    <a href="Activities.html">Activities</a>
    <a href="Staff.html"> Staff </a>
    <a href="Gallery.html">Photo Gallery</a>
    <a href="Event.html">Events</a>
    <a href="Notice.html">Notice</a>
    <a href="Contact.html">Contact Us</a>
</h3>
</p>
<h3>About My School</h3>

<p> My school is one of the best and most reputed school of this city. My school offers education right from kindergarten to 12th standard. My school is equipped with all sorts of modern day facility to enhance education. My school has a play ground, an assembly hall, library, and activity lab. </h3>
</p>

<p> My school offers teaching in English and Gujarati medium and is affiliated to Gujarat Secondary Education Board. The teachers in my school make learning easy and interesting by providing presentations and demonstrations. </h3>
</p>
</body>
</html>
```

Code Listing 2.4 : HTML code for the main page of My School website

Write the code given in code listing 2.4 in an editor. Save it as Home.html. View the code in a browser. The output will look similar to the one shown in figure 2.11.



Figure 2.11 : Home page of school website

If you try to click on any link other than Home, you will get a message that file could not be found. This message comes because we have not yet created any other file besides Home.html.

To develop the files perform the steps mentioned :

- Write HTML code for other files such as activities, staff, photo gallery, notice and contact. Name them appropriately and save them in the same directory where you have stored home.html file. You may skip the photo gallery page. In next chapter when you learn about embedding images in to HTML documents, try to modify this project. Then you can also add your school's logo on main page. At this stage, you are able to add some images as a background of page.
- You have already created a notice following directions given in code listing 2.2 about "Practicing Rainbow Colours". You can rename the file and save it as Notice.html.
- Use necessary images, backgrounds and links beside the formatting tags. You may modify the text also. For example, on the home page, you may write your school's name, name of school trust, foundation date of school, contact information of school etc.

Summary

In this chapter we have learnt about meta-tag used in the head section of an HTML document. The meta-tags themselves may not display anything; however they often provide information regarding the author, keywords, description about the page. Such information is helpful to the search engine and other utility programs. Like head section, we have also discussed tags and attributes used in body section of an HTML document. HTML body section also provides physical and logical formatting style tags along with tag settings and displaying special symbols. We have also seen a way to add comment, default fonts and colours in an HTML document. At the end, we have seen a small website having many web pages describing activities of a typical school.

EXERCISE

1. Write a short note on head sections and meta tags used in it for an HTML document.
2. Explain how an image can be added on a web page background using HTML tag.
3. Explain colour representation in electronic media. Also explain colour coding schemes in brief.
4. Explain horizontal line `<hr />` element in HTML.
5. Explain font tag in HTML.
6. Explain how special characters will be presented using HTML.
7. Choose the correct option from the following :
 - (1) Which of the following form basic two sections of an HTML code ?

(a) Head and body	(b) Physical and logical
(c) Code and browser	(d) Meta-tags and body
 - (2) The meta-tags in an HTML document are written in which of the following section ?

(a) Body	(b) Code
(c) Head	(d) Special
 - (3) Which of the following about a web page is described when a meta-tag is used in HTML page ?

(a) Author, purpose and keywords	(b) Layout
(c) Style	(d) Size
 - (4) Title of a web page is embedded within which of the following tag ?

(a) <code><p></code> and <code></p></code>	(b) <code><body></code> and <code></body></code>
(c) <code><title></code> and <code></title></code>	(d) <code><h1></code> and <code></h1></code>
 - (5) In which of the following sections can we add comments in HTML document ?

(a) Head	(b) Body
(c) Both head and body	(d) Either head or body
 - (6) Which of the following is used to specify a colour in HTML code ?

(a) Colour code in hexadecimal	(b) Colour code in decimal
(c) Colour mixing model	(d) Pixel in percentage
 - (7) Which of the following is used to set a visited link in HTML code ?

(a) <code>alink</code>	(b) <code>vlink</code>
(c) <code>before link</code>	(d) <code>after link</code>

LABORATORY EXERCISE

1. Complete 'My School' web site development project explained in the chapter.
 2. Explore different websites of educational institute and study their design. Write a comment on these designs.
 3. Develop a page about yourself. Describe about you, your family, your friends, your education goals, and contact address. You may consider the design of page as follows.

I, me and myself...!

Here all the underlined words are hyperlinks that link you to a different page. Also set active link and visited link property, light colour background. This should be a complete website as we have developed earlier website for 'My School'.

4. Go to <http://www.w3schools.com/>, find out HTML section from the menu and experiment "try it yourself". This page is an editor that takes your html code and presents a view of the web page as if it is viewed in a browser.



Handling Images in HTML

It is said that a picture is better than thousand words. Pictures or images attract majority of persons and convey an important message to the audience. If a web page shows a picture, the user will immediately pay attention to it. Further, it is observed that pictures increase degree of understanding and acceptance in general.

Images are added to a web page using the `` tag in HTML. We have already added an image as a background in the previous chapter. There are other ways to include an image on a web page. For example, the following line would add the image called `rainbow.jpg` into the page.

```

```

The above example uses `src` attribute which is described below along with other attributes that may be used with the `image` tag.

The `image` element (image tag with the image content and attributes) does not cause a line break, hence it is known as an inline element.

The `src` attribute

The `src` attribute provides information about location of the image. The location here refers to the source of the image. It tells the browser where to find the image. Many times we use URL as an image location. Without mentioning the source of the image, it is impossible for browser to find and display an image. To include an image from an URL given as "`http://pritisajja.info/images/img1.jpg`" we may write following tag.

```

```

The image file specified should have a proper image format such as `bmp`, `gif`, `tiff` or `jpg`.

Consider a scenario where you are surfing the Internet and liked an image. How would you get its complete URL to include it in an HTML code? Answer is simple. The first option is to right click on the image and copy the image link. Another option is to save the image into the local memory of your computer. While opting for the second option you must be careful about the copyrights for the image.

When tags like `<p>` or `<h1>` are used with the content, the tags just have to display the content in a particular style. While using the `image` tag, we need not have to provide any content, but the source of image with some attributes. That is why the `image` tag is called an empty tag. However, when an HTML document containing image is displayed in a browser, the browser needs to retrieve the image. The image must be available to your local computer or server. If you have used an URL, check that your internet connection is working.

It is a good practice to create a separate folder/directory for images. This approach is better, especially when the website is large and uses multiple images. Code to add three images one by one in an HTML is shown in code listing 3.1.

```

<html>
<head>
    <title>My favourite food ....!</title>
</head>

<!-- ----- -->
<body>
    <h1> My favourite food ....!</h1>

    <!-- ----- -->
    <h1> <u> Chocolates </u></h1>

    <p>
        
        <h1>
            Chocolates are good for health, better for hunger and best for mood !
            Do not forget to clean your teeth until it is too late !
        </h1>
        </p>

    <!-- ----- -->
    <h1><u> Fruits and Dry Fruits </u></h1>
    <p>
        
        <h1>
            Do not think dry fruits are dry, though they are dried fruits !
            They are really really very interesting !
        </h1>
        </p>

    <!-- ----- -->
    <h1> <u> Ice creams </u> </h1>
    <p>
        
        <h1>
            Ice creams are really cool ! These become coolest when you taste them !
        </h1>
        </p>

    <!-- ----- -->
</body>
</html>

```

Code Listing 3.1 : Adding multiple images on a web page

Code listing 3.1 contains some dotted lines. This dotted line will not be displayed in the web page as they are embedded within comment tags (<!-- and -->). The line simply divides the code section into separate blocks for ease of reading and better understanding. As browser will accept only valid HTML statements, we cannot directly enter a dotted line. We have to embed the line within comments tag.

Figure 3.1 shows output of the HTML given in code listing 3.1.

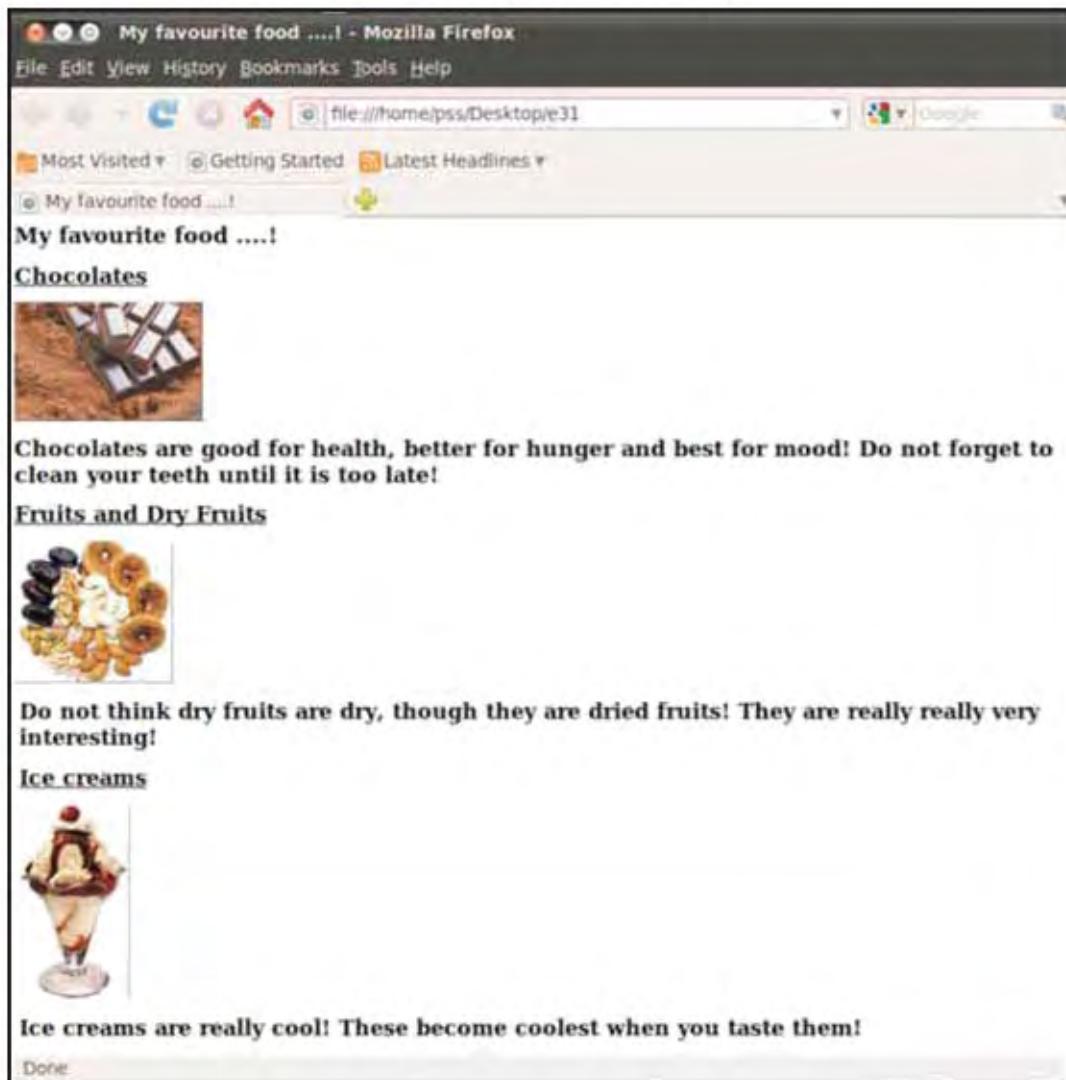


Figure 3.1 : Multiple images on a web page

The browser will get all the images from the given source one by one and display them in the order in which they are mentioned in the HTML code. Users generally see a page display along with all its images.

The alt attribute

The alt attribute along with an image tag describes the source image. It is always advisable to define this attribute, as it describes the image. The example of the alt attribute is as follows:

```

```

This attribute is quite useful. The first reason is it describes the image, which is many times necessary. Another reason is due to some reasons if the browser does not display the image, the reader will at least know, what to expect as a picture. The third reason is, when the web page is read by

the software such as screen reader (special software designed for visually challenged people) and search engine, the interpretation of the image becomes easy.

The height and width attributes

The height and width attributes specify the values of height and width of the image in pixels as illustrated below.

```

```

Here, the width attribute indicates the browser how wide the image should appear on the page. The height attribute specifies how tall the image should appear on the page. Values of both width and height attributes can be specified using the number of pixels. It is not compulsory to use both the width and height arguments together; you can use any one of them. However, using only one attribute will not be much useful.

By providing additional information about the height and width of an image to the browser, presentation and loading of the image in a browser become smooth and efficient. Here, the browser knows the amount of space to be allocated to the image. It is to be noted that your image should not be greater than your screen. However, it is possible to accommodate larger image into an HTML code. It is also not advisable to magnify or shrink your images by using much higher or lower height and width values. Doing this will ruin quality and visibility of the image.

The align attribute

The align attribute is used to align the given image within the page or any element that contains the image. Example for the same is as follows.

```

```

The align attribute can take one of the values shown in table 3.1.

Value	Purpose
Top	The image is aligned at the top of the current line of text.
Middle	The image is aligned in such a way that the middle of the image appears at the current text.
Bottom	The bottom of the image is aligned with the baseline of the current line of text.
Left	The image is aligned to the left side of the containing window or element.
Right	The image is aligned to the right side of the containing window or element.

Table 3.1 : Possible values for the align attribute

The border attribute

An image on a web page can be highlighted with a border. By default, images do not have borders. To create a border around an image, the following attribute can be used.

The border attribute was created to specify the width of the border in pixels:

```

```

Some browsers (such as the Internet Explorer) highlight image when link is given to the image; that is when an image is used as a link, it is highlighted with a border.

The id attribute

With the help of the id attribute, you can specify an identifier (name) for the image. Later the image can be referred by the identifier in a script written in a programming language such as java. The tag below shows how to use name attribute.

```

```

Adding space around image

To add space around an image vspace and hspace attributes are used. To add space over and under the image the vspace attribute is used. In a similar way, to add space to the left and right of the image the hspace attribute is used. The example given below illustrates the use of these attributes.

```

```

This attribute will be helpful when the image is inline with the text leaving no gaps. Leaving the vertical and horizontal gaps makes the image clearer and the page more readable. Code listing 3.2 shows an example of HTML code that leaves some gap surrounding the image of chocolates.

```
<html>
  <head>
    <title>My favourite food ....!</title>
  </head>
  <!-- ----- -->
  <body>
    <h1> My favourite food ....!</h1>
    <!-- ----- -->
    <h1> <u> Chocolates </u></h1>
    <p>
      
    <h1>
      Chocolates are good for health, better for hunger and best for mood !
      Do not forget to clean your teeth until it is too late !
    </h1>
    <p>
    <!-- ----- -->
  </body>
</html>
```

Code Listing 3.2 : HTML code illustrating hspace and vspace

When you see the HTML code illustrated in code listing 3.2 in a browser, it will look like figure 3.2. You may notice the horizontal and vertical gaps introduced by hspace and vspace attributes. Add some more images within the HTML code and see how your browser treats the images.

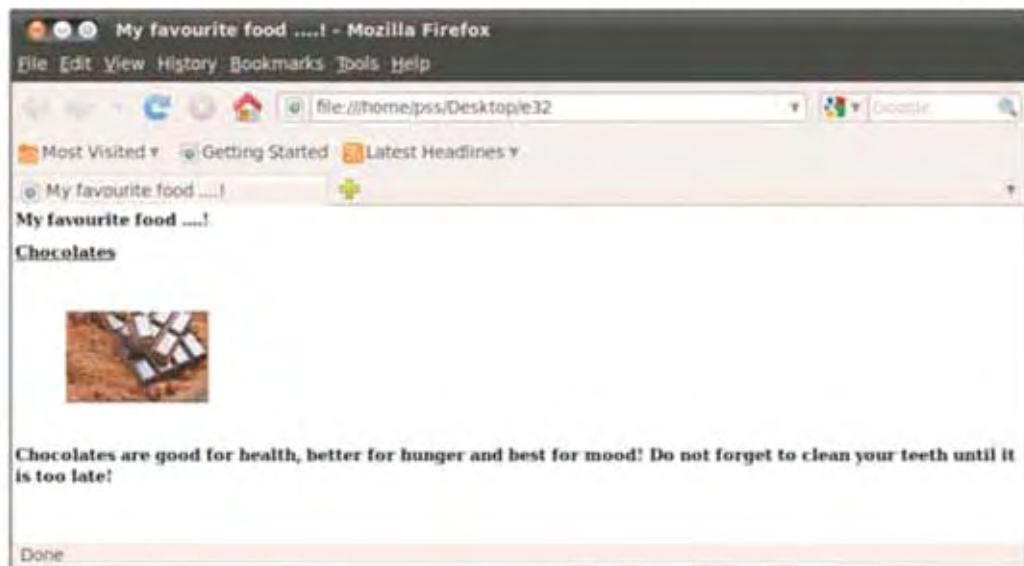


Figure 3.2 : Illustrating hspace and vspace

Now you have observed that the hspace and vspace attributes add spaces on both the sides of the image. That is, when hspace is used, it adds space to the left as well as to the right sides of the image. Similarly, when vspace is used, it adds space to the top as well as to the bottom of the image. To add image at only one side, we need to modify image appropriately. An alternative is to print a blank image beside the original image. However, the spacing and alignment will be difficult to manage in the second alternative.

Try to remove the "chocolate.png" file from your computer memory and see what happens. Will it print the alternative description of the image such as "Here comes the sweet image" ? You may find the output as shown in figure 3.3.



Figure 3.3 : Displaying alternative description in absence of image

Low Resolution Image

High resolution image takes lot of space and time for loading it on web page. Till the high resolution (original) image is loaded, we may temporarily publish a low resolution image on the page as illustrated below.

```

```

To get low resolution image, you can use resize, crop or resample the image using techniques supported by a suitable image editing tool.

By adding an attribute managing the alternative low resolution image, we may solve problems of browser speed as well as size of the image; however, we have to create and store an alternative image. Further, the low resolution image may appear fuzzy and vague.

Embedding Images of Various Formats

So far, we have incorporated bmp (Bit Mapped Picture) file into the image tag. But there are some other formats of image that can be embedded into the image tag. Among the various image formats, formats such as bmp, jpeg, png, tiff and gif are very popular. The description of these image formats are shown in table 3.2.

Image file format	Description
BMP	Bitmap graphic files for Windows and OS/2
GIF	Graphics Interchange File
JPEG	Joint Photographic Expert Group file
JPG	JPEG/JIFF Image file
PNG	Portable Network Graphics bitmap graphic file
TIFF	Tag Image File Format bitmap file

Table 3.2 : Image formats

Digital cameras and web pages normally use jpg/jpeg files; as these formats drastically compress the data in the files. The image formats jpg and jpeg are good for continuously toned images such as photographs. Commercially tiff format images are used very much, as they offer highest quality and good amount of compression of the image with minimum loss. Tiff format is actually one of the good lossless image formats. Other file formats such as gif and png also use the lossless compression technology. However, png format is comparatively slower in reading and writing.

Image as a Hot Spot

Many times the browser will take some time to load the big image. Above this, when we have multiple images on one or more pages, it will be confusing for users that which image should be considered first. Have you noticed some online shopping sites for books, jewelry, mobile phone and other such items? On a single page you might have seen small images of many items displayed with brief information of each. When you find a particular item interesting, you may click the small image to go to the page containing detailed information as well as bigger, good quality image of the selected item. This can be done by applying a link to each of the small image on a page. You can say that here the image is considered as hot text. Since there is no text here, it is identified as hot spot. The small image itself has a link to the other suitable page. Code Listing 3.3 illustrates HTML code that uses such two small images and leads to two different web pages.

```

<html>
  <head>
    <title> Image Hot Spot </title>
  </head>
<!-- ----- -->
  <body>
    <h1> Either you can go to office or go to temple!</h1>

    <p> <h2>Click on any image below to see its larger version </h2> </p>

    <br>
<!-- ----- adding links to the images ----- -->

    <a href="big_office.html"></img></a></p>

    <a href="big_temple.html"></img></a></p>
<!-- ----- -->
  </body>
</html>

```

Code Listing 3.3 : HTML code for image as hot spot

Save this code as main.html. When you see the code in a browser, it will look as shown in figure 3.4.

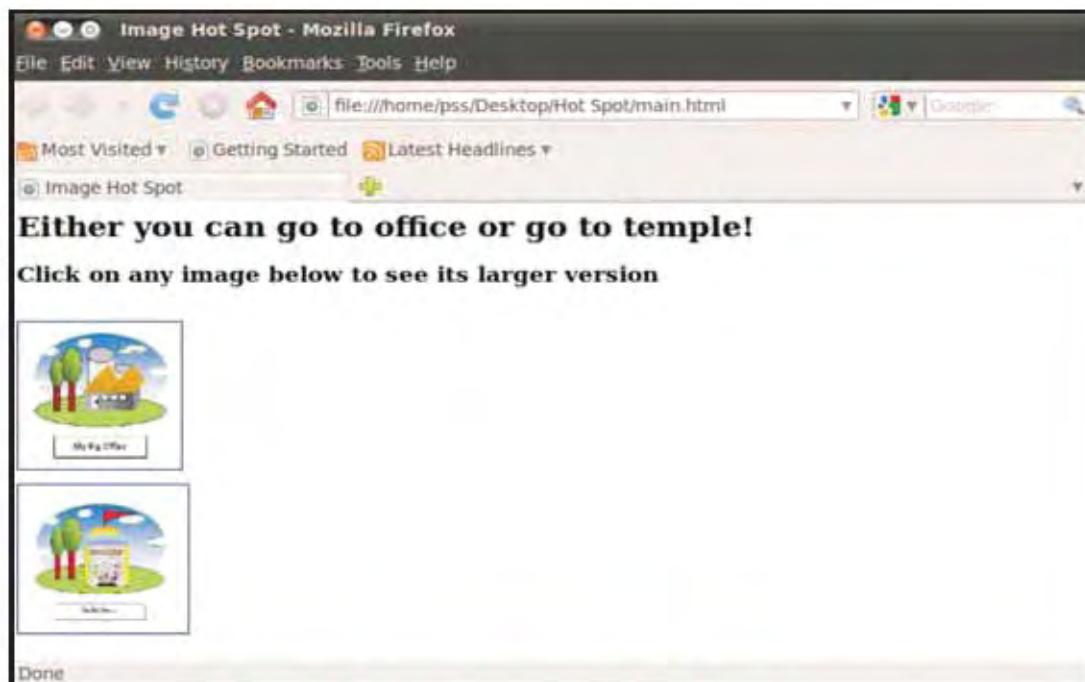


Figure 3.4 : Using image as hot spot

Besides the HTML code shown in code listing 3.3, we also need four images. We need one small image for office called small_office.bmp and another image called big_office.bmp. Similarly, we also need one small image for temple called small_temple.bmp and another image called big_temple.bmp. You can create the images or you may use the existing images by renaming them.

When you see the main.html file in a browser, you can see the output as shown in figure 3.4; provided two small figures for an office (small_office.bmp) and a temple (small_temple.bmp) are available.

Once you correctly visualize both the images in the main.html page in a browser, you may try clicking on images. When you click on first image the href tag (given below) redirects you to another page (big_office.html) considering the small image as hot spot.

```
<a href="big_office.html"></img></a></p>
```

Note that your big_office.html file must be ready with big office image embedded in it.

By clicking on the first image you will see the screen as shown in figure 3.5.

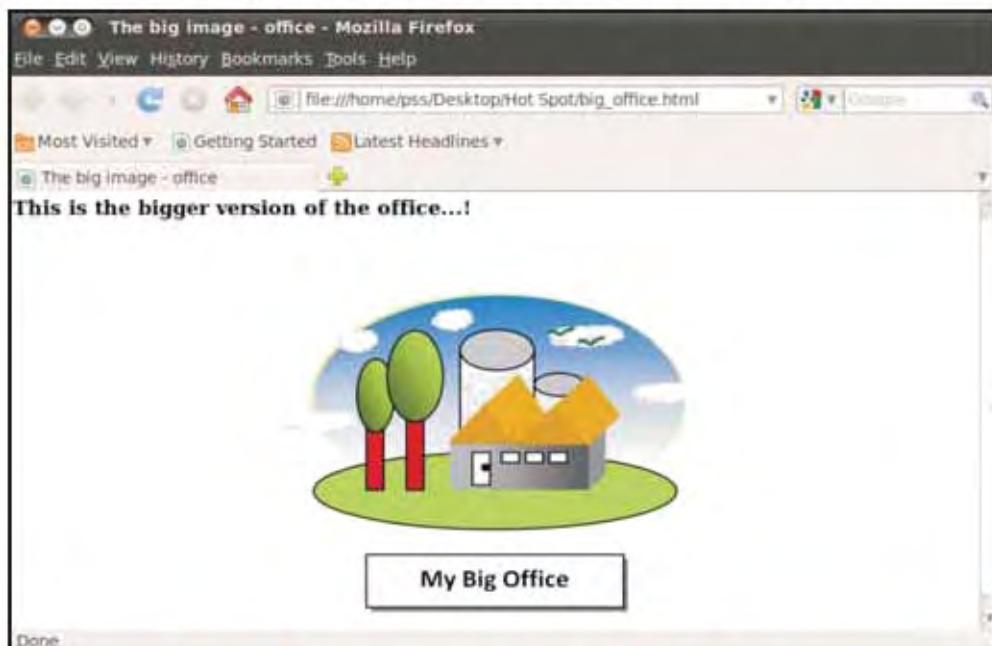


Figure 3.5 : When you click on the first image

HTML code for creation of big_office.html is given in code listing 3.4.

```
<html>
    <head>
        <title> The big image - office </title>
    </head>
    <!-- ----- -->
    <body>
        <h1> This is the bigger version of the office...! </h1>
        <br>
        <br>
        <br>

        </img>
    <!-- ----- -->
</body>
</html>
```

Code Listing 3.4 : Code for the first image

Similarly, when you click on the second image shown in figure 3.4, you will see output as shown in figure 3.6.

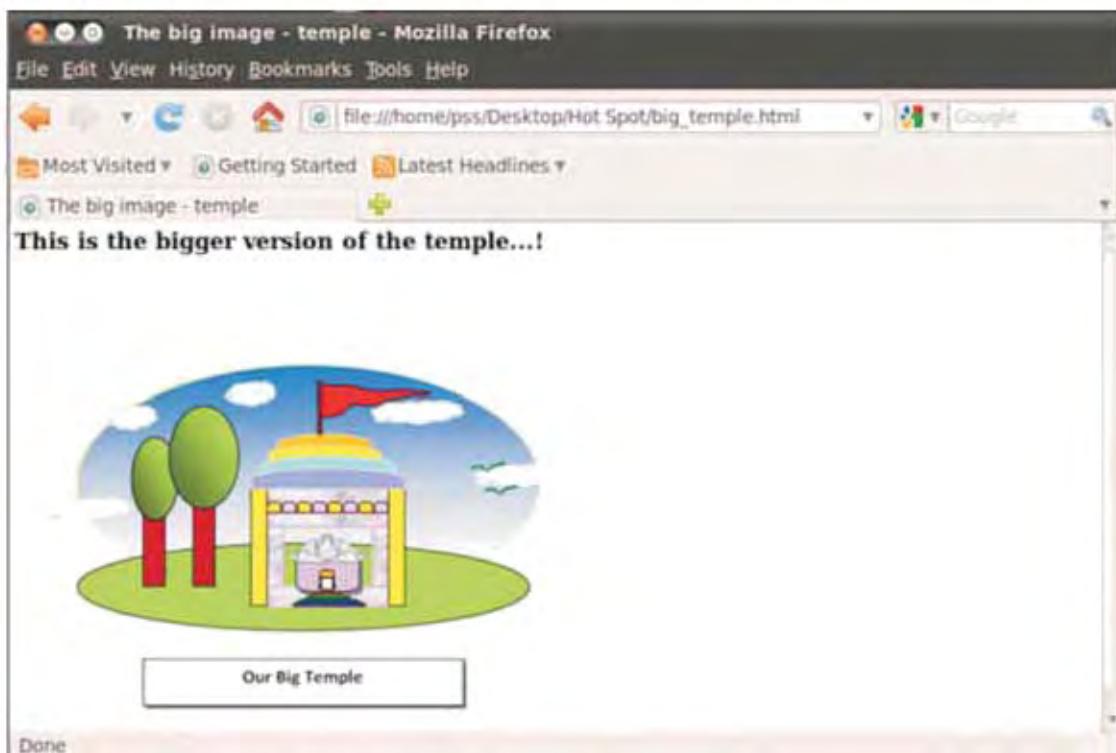


Figure 3.6 : When you click on the second image

HTML code (big_temple.html) that generates figure 3.6 is illustrated in code listing 3.5.

```
<html>
  <head>
    <title> The big image - temple </title>

  </head>
  <!-- ----- -->
  <body>
    <h1> This is the bigger version of the temple...! </h1>
    <br>
    <br>
    <br>
    </img>
  <!-- ----- -->
  </body>
</html>
```

Code Listing 3.5 : Code for the second image

Observe that to complete the above example you require four images and three HTML files as follows :

1.	<u>Small_office.bmp</u>	→	a small image of office in bmp format (or any valid image format).
2.	<u>Small_temple.bmp</u>	→	a small image of temple in bmp format.
3.	<u>Big_office.bmp</u>	→	a big image of office in bmp format.
4.	<u>Big_temple.bmp</u>	→	a big image of temple in bmp format.
5.	<u>Main.html</u>	→	a file HTML code with two small images, reference to the other two HTML files, and some text in it.
6.	<u>Big_office.html</u>	→	a file HTML code with bigger image of the office.
7.	<u>Big_temple.html</u>	→	a file HTML code with bigger image of the temple.

You may try to add some more small images in a main page. Small images are also identified as thumbnails. When you click on such small images arranged in a systematic way within a web page, each image will lead to the detailed (or bigger) version of it.

Image Map

In previous section we have added a link to an image. It is also possible to add multiple links to an image. Here each link points to a different reference. To create multiple links within a single image, we need to create multiple clickable locations within the image. Clicking on each such location, a particular reference (page) can be opened. Each such location is called a hot spot. Consider you have some kind of drawing of an area showing different places such as a corporate office, shopping mall, temple, police station, and fire brigade. On each of this location, we may create a link. For example on temple location, we may set a link that leads to a page containing information about the temple. The linked page may show some photos, news, brief history about the temple, and directions for how to reach the temple. In case you have a map (suppose map of India), then some regions of the map (states like Gujarat, Maharashtra, etc.) can be defined as hot spots. When one clicks on such hot spot, new web page can be opened.

Consider the image as shown in figure 3.7. The image contains a temple, an office, a mall, a fire brigade and a police station. These utilities are located on two main roads called "Shri M K Gandhi Marg" and "Shri Netaji Marg". On each of these five buildings and two roads a link is set. That is, on seven different areas seven different links are set. As stated earlier, these areas on which the links are to be set are known as hot spots. The hot spots must be big and visible enough; so that users can easily identify them and click. Otherwise, users will find it difficult to select the hotspot and follow the link. Further, the image should convey information that by clicking on each such area/hot spot, the user may be redirected to a new web page showing detailed information about the selected area. For example, "click on temple to know more ..." message can be given to the user when the image shown in figure 3.7 is displayed in a browser. Generate the image shown in figure 3.7 using appropriate image creation tool and name it as city.bmp.

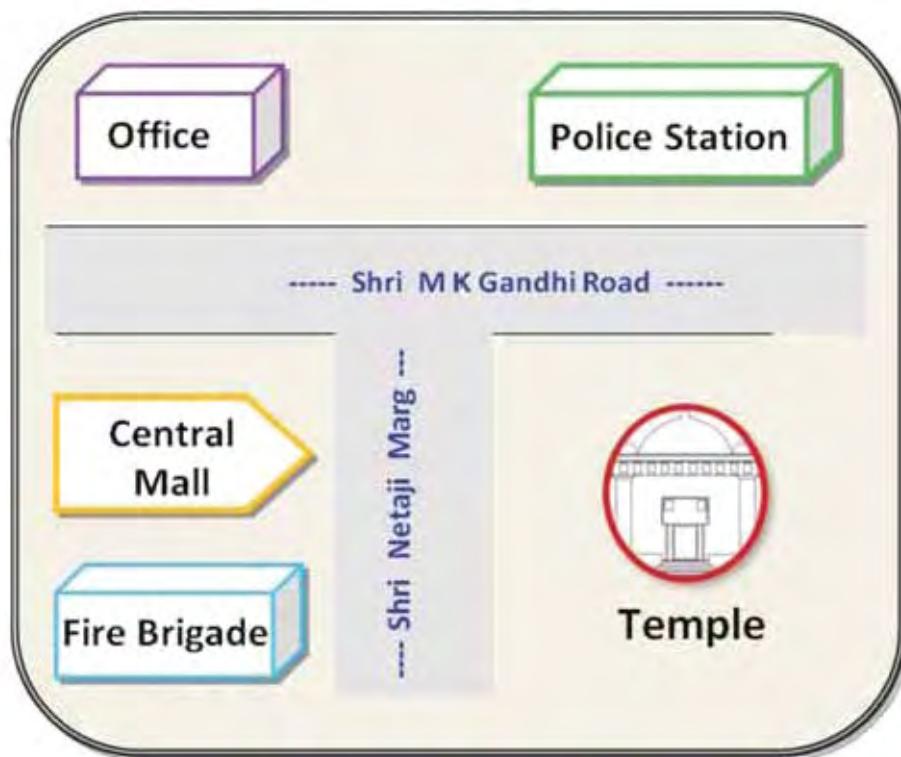


Figure 3.7 : A scenario of a city market place

The location of a hot spot is mentioned using x and y co-ordinates measured from the top left hand corner of the image. These coordinates are used in two ways :

- To specify where the hot spots are
- To compare the coordinates provided by user through a click.

The browser at the user's side identifies the coordinate of the user's click and takes user to the referred web page. For this, `<map>` and `<area>` tags are used along with a `usemap` attribute into its image tag. The map tag has an attribute called `map`. The `map` attribute value must be matching with the `usemap` attribute value. See the following example.

```

```

Here, the "#roadmap" is the name of identifier of `usemap` we have created. Within the `<map>` and `</map>` tags, the hot spot co-ordinates and their corresponding links are to be specified. Here, the temple co-ordinates are (518,378,70) and its shape is circular. We can embed this information in `map` tags as follows.

```
<map name="roadmap">
<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">
<!-- --About co-ordinates of other hot spots ----- -->
</map>
```

The co-ordinates given in the second line of the above HTML segment represents temple in the `city.bmp` file shown in figure 3.7. As the temple is specified in circular area, we have used "circle" value to its shape. Suppose, the temple is in a rectangular shape, we need to use shape value as "rect". The `area` tag must mention the shape of a hotspot. The valid possible shapes are circle,

rectangle and polygon. The rectangle is specified by rect; circle is specified by circle; and polygon is specified by poly. Alternatively, full names such as rectangle are also used. See example below illustrating area tag with different shapes.

- `<area shape="poly" coords="32,301,183,301,239,352,188,399,32,399" alt="Central Mall" href="CentralMall.html">`
- `<area shape="rect" coords="32,432,233,532" alt="Fire Brigade" href="FireBrigade.html">`
- `<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">`

Code listing 3.6 shows a complete HTML code to generate an image map using the scene of city shown in figure 3.7.

```
<html>
<body>
<p>Click on the location presented on map to look in detail:</p>
<!-- ----- -->


<!-- ----- -->
<map name="roadmap">

<area shape="rect" coords="46,37,219,141" alt="Office" href="Office.html">

<area shape="rect" coords="407,38,632,142" alt="Police Station" href="PoliceStation.html">

<area shape="poly" coords="32,301,183,301,239,352,188,399,32,399" alt="Central Mall" href="CentralMall.html">

<area shape="rect" coords="32,432,233,532" alt="Fire Brigade" href="FireBrigade.html">

<area shape="circle" coords="518,378,70" alt="Temple" href="Temple.html">

</map>
<!-- ----- -->
</body>
</html>
```

Code Listing 3.6 : HTML code for image map

The output of code listing 3.6 will look as shown in figure 3.8 in a browser.

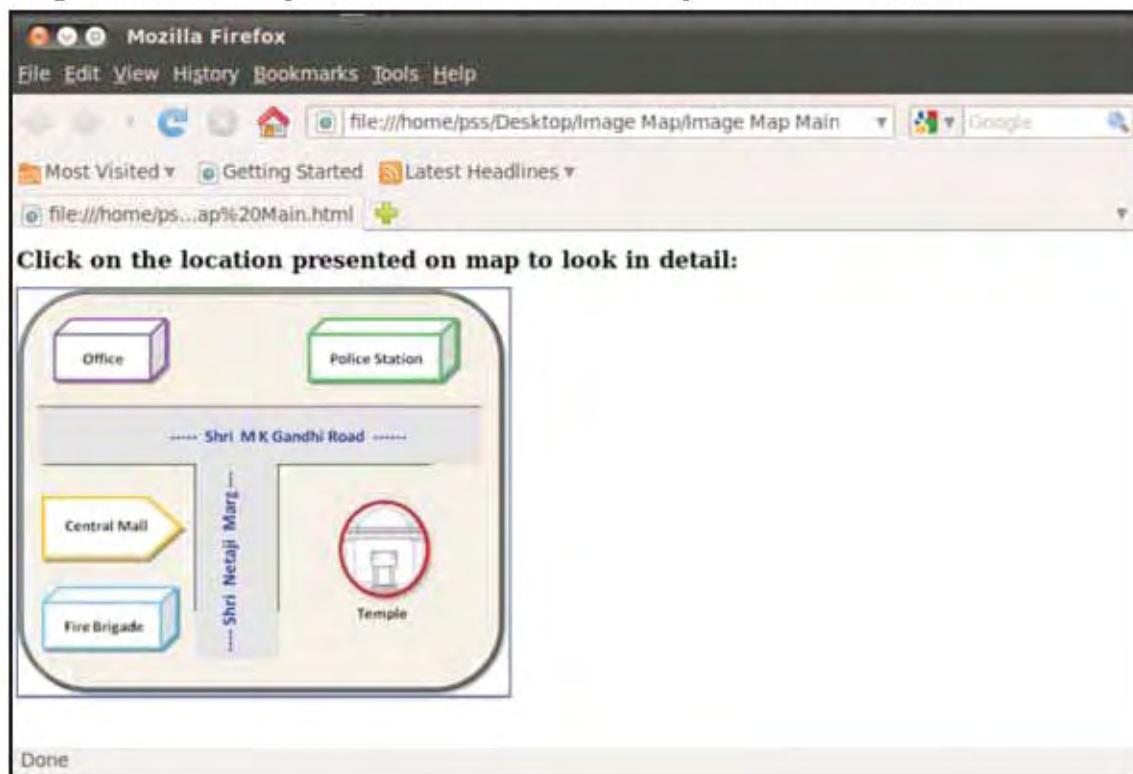


Figure 3.8 : Image map in a browser

When you click on the circular image of temple shown in figure 3.8, you will be redirected to a new page showing details about the temple. See figure 3.9 illustrating another file showing temple details.

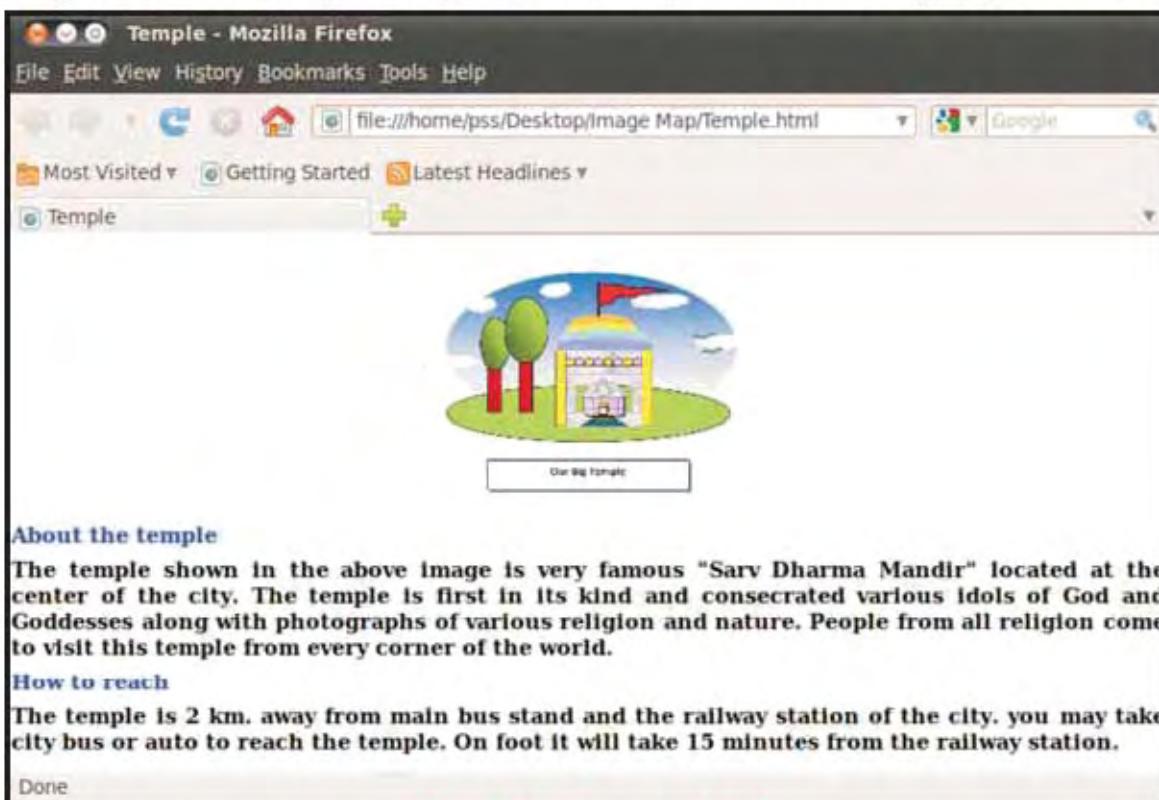


Figure 3.9 : Temple details

Code listing 3.7 shows the code which is required to generate the web page shown in figure 3.9.

```
<html>
<head> <title> Temple </title> </head>

<body>
    <center>
         </img>
    </center>

    <!-- ----- -->

    <h1> <font color="blue">About the temple </font> </h1>

    <p align="justify"> <b>
        The temple shown in the above image is very famous "Serv Dharma Mandir"
        located at the center of the city. The temple is first in its kind and
        consecrated various idols of God and Goddesses along with photographs
        of various religion and nature. People from all religion come to visit
        this temple from every corner of the world.
    </b>
    </p>
    <!-- ----- -->
    <h1> <font color="blue"> How to reach </font> </h1>

    <p align =justify> <b>
        The temple is 2 km. away from main bus stand and the railway station of the city.
        you may take city bus or auto to reach the temple. On foot it will take 15 minutes
        from the railway station.
    </b>
    </p>
</body>
</html>
```

Code Listing 3.7 : Code for the temple detail

Similarly, you can prepare HTML files for office (Office.html), police station (PoliceStation.html), fire brigade (FireBrigade.html), and central mall (CentralMall.html). Verify that all the files work individually and try them from clicking the main HTML file showing overall scene of the city.

Linking Multimedia Files

A video as well as an audio file can be integrated into an HTML document using an anchor tag as illustrated below.

[follow this recipe and enjoy delicious food...!](food.mp4)

The example shown in previous line includes a movie file called "food.mp4". You should have a video/movie file with you. When the user clicks the words "Enjoy this... !" displayed on the web page, the video file will be shown in appropriate software. If the referred movie file is available on some other website or remote place, first it will be transferred to the user's computer. Such files are stored in local directory of user's computer as temporary internet files. If the required software is not found while executing the movie file, browser will ask you to choose alternative software from the computer or from the Internet.

Following is an example which adds a sound file in an HTML code.

[Enjoy this song...!](song.mp3)

Besides the mov file format, there are other formats for a video file. To name a few, these formats are avi, wmv, meg or mpeg, and swf. Some of these require downloading a video player that executes the files.

Summary

In this chapter we have learnt how to add image in an HTML document. Besides addition of one or more images into an HTML document, we have seen how to arrange image on the monitor screen using attributes like align, hspace and vspace. We have also used an image as hot spot. Not only the whole image, but part of image (such as a geographical map) can also be linked to different web pages using image map features. At the end, information about how to add video and audio files to an HTML is given. Using the information provided in this chapter, users are able to accommodate not only plain text and numbers into an HTML document, but video, animation, images and audio can also be included in a web page.

EXERCISE

1. Explain how images can be added into an HTML document by giving suitable example.
 2. Write a short note on various attributes of the image tag in HTML.
 3. Define hot spot. Also explain how an image can be defined and used as a hot spot in HTML.
 4. Write a short note on image map.
 5. Choose the correct option from the following :
 - (1) Which of the following tag is used to insert a video file into an HTML document ?
(a) img (b) image (c) href (d) imc

LABORATORY EXERCISE

- 1.** Consider the example discussed in code listing 3.1 of this chapter. It prints images of chocolate, dry fruit and ice-cream. Extend the HTML code in such a way that if you click on chocolate image, it will lead you to a new HTML page describing chocolates. You may include more images of chocolates, some facts about chocolates, history of chocolates and recipe of making chocolates at home.

If you click on dry fruits image, it will lead you to a new HTML page describing the dry fruits. Similarly, if you click on ice-cream image, it will lead you to a new HTML page describing various ice-creams.

2. Create a page called index.html. Put some smaller three images of your favourite personalities on the index page. They can be your school teachers, your friends, sports persons or great authors. Also prepare three web pages that display some information about each of these three persons including a big photograph. Set link on each of the smaller photograph on the index page in such a way that it leads to the full biography of the selected person. You need to set three links on the three small photographs. (Hint: To create a link from an image, add an element, and put the link of bigger picture in the href attribute of the element.)
3. You may use the approach discussed in the above example 1 to develop your family tree. Create a web page with one or more images of your grandfather and grandmother. Develop other pages in such a way that, when you click on the image of your grandfather, a new page will appear with some more related pictures.
4. Implement the image map showing city scenario as discussed in this chapter.
5. Complete the school website project you did in Chapter 2 by adding the photo gallery. You may add logo of your school on the main page too.
6. Use an existing video file and embed it into a web page. Alternatively, you may create a video file through your mobile or any other device and embed it into a web page.



List and Table Handling in HTML

List Handling

While presenting information, sometimes we need to list different individual entities. Instead of writing such entities in continuous paragraph form, these individual items are presented as list with or without numbers. By presenting the entities in such way increases clarity in representation and aids readers in understanding them.

Let us see an example. When you want to shop some items from the market, you need to remember them. Figure 4.1 illustrates the list of some of such items in a classic way which can be taken with you while shopping.

Items to be purchased

- Chocolates
- Notebook
- Practice book
- Ball pen
- Markers

Figure 4.1 : An example of list

There are three basic types of HTML lists. These types are unordered list, ordered list and descriptive list. All the types are discussed in forthcoming sections of this chapter.

Unordered List

As illustrated in figure 4.1, an unordered list contains item along with a symbol. This symbol is also called a bullet. The list shown in figure 4.1 is an unordered list with bullet as '•'. To create an unordered list in HTML, the `` tag and `` tag pair is used. The items of the list are enclosed within the `` and `` tag pair. HTML example that creates the unordered list is shown in code listing 4.1.

```
<html>
<!DOCTYPE html>
<body>
<font color="Blue">
    <h1>An Unordered List:</h1>
</font>
<ul>
<font size="6">
```

```
<ul>
    <li>Chocolates</li>
    <li>Notebook</li>
    <li>Practice book</li>
    <li>Ball pen </li>
    <li>Markers </li>
</ul>
</li>
</ul>
<!-- ----- -->
</body>
</html>
```

Code Listing 4.1 : An unordered list

Write the code given in code listing 4.1 in the SciTE editor and check its output in a browser. The output will look similar to the one shown in figure 4.2.

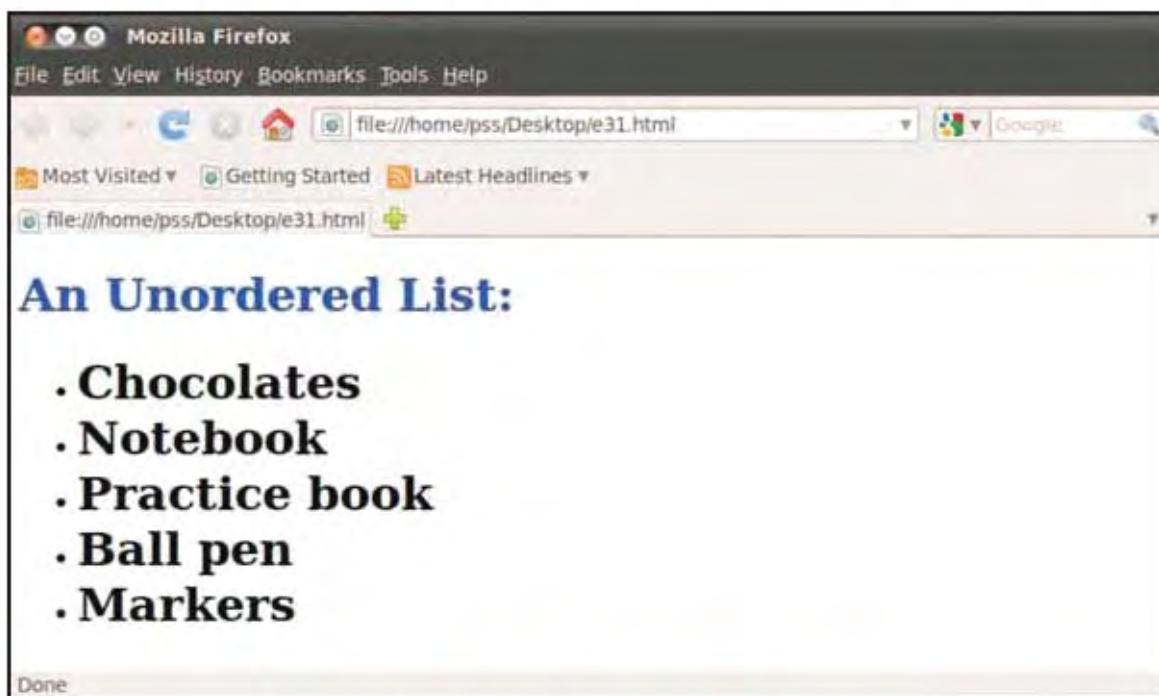


Figure 4.2 : An unordered list in a browser

Changing Bullets

The default shape of bullet is a filled circle. If you want to change the bullets, you can use an attribute called type. The types can be circle, square and disc. See the following example.

```
<ul type="square">
```

You can also change the bullets to unfilled circle by using the tag shown below :

```
<ul type="circle">
```

Modify the HTML code given in code listing 4.1 and experiment with different type of bullets.

List without Bullets

In case you do not want any bullet, you may use description list using tag pairs `<dl>` and `</dl>`. Here 'dl' is an abbreviated form of the word description list. Within the description list, we need to define description terms using `<dt>` and `</dt>`. That is, an item say Chocolates can be defined as follows:

```
<dl>
  <dt> Chocolates </dt>
</dl>
```

To define sub-items we may use description tag pair given as `<dd>` and `</dd>`. That is, we can add different types of chocolates and ice-creams. Figure 4.3 demonstrates different types of bullets on a web page.

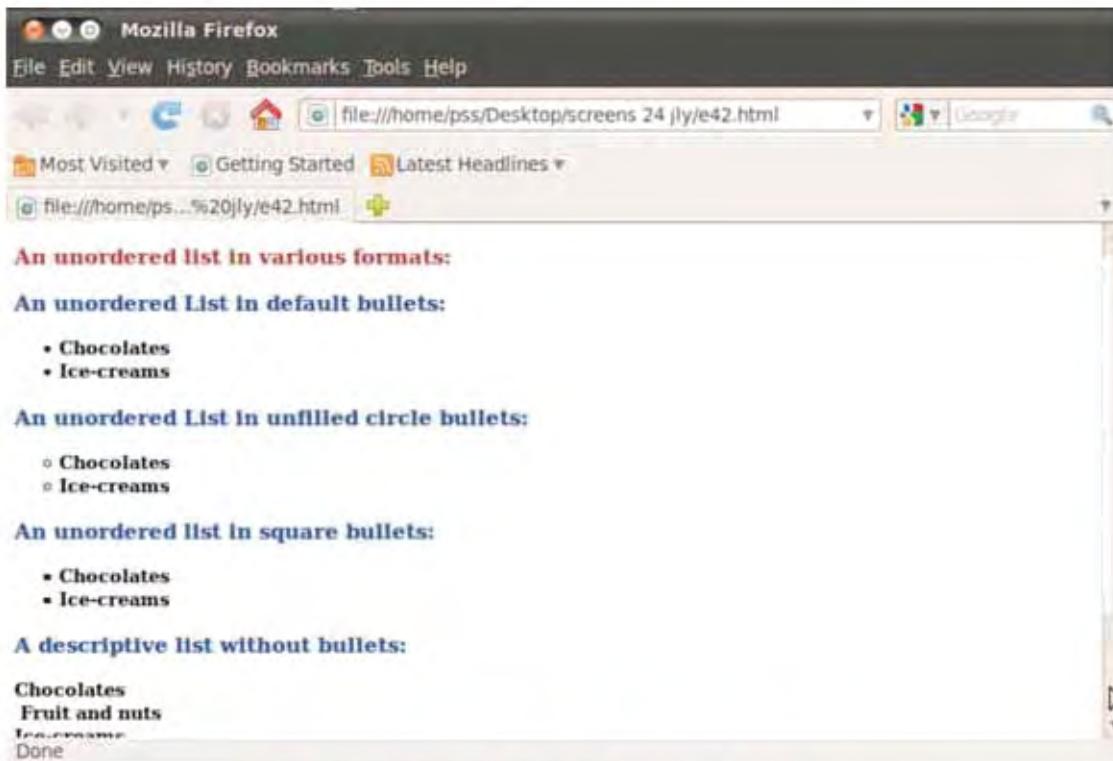


Figure 4.3 : Different types of bullets

HTML code to generate figure 4.3 is shown in code listing 4.2.

```
<html>
<body>
<font color = "Brown">
  <h1>An Unordered List in various formats: </h1>
</font>
<!-- ----- -->
```

```

<font color = "Blue">    <h1>An unordered List in default bullets: </h1>    </font>
<ul>
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
</b>
</font>
<ul>
<!-- ----- -->
<font color = "Blue">  <h1>An unordered List in unfilled circle bullets: </h1> </font>
<ul type="circle">
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
</b>
</font>
<ul>
<!-- ----- -->
<font color = "Blue">  <h1>An unordered list in square bullets: </h1> </font>
<ul type="square">
<font size="6">
<b>
    <li>Chocolates    </li>
    <li>Ice-creams    </li>
    <li>Dry fruits    </li>
</b>
</font>
<ul>
<!-- ----- -->
<font color = "Blue">
    <h1>A descriptive list without bullets: </h1>
</font>
<ul>
<font size="6">
<b>
    <dt>Chocolates    </dt>
        <dd> Dark </dd>
        <dd> Fruit and nuts </dd>
        <dd> Milk  </dd>

```

```
<dt>Ice-creams </dt>
    <dd> Vanilla </dd>
    <dd> Chocolate chips </dd>
<dt>Dry fruits </dt>
    <dd> Almonds </dd>
    <dd> Cashew nuts </dd>
</b>
</font>
</db>
<!-- ----- -->
</body>
</html>
```

Code Listing 4.2 : HTML code for different types of bullets

Heading of List

To provide heading of a list, we have to simply use `<lh>` tag. Obviously, the heading must appear before the list. Hence, before starting the `` and `` tag pair, you need to use the `<lh>` tag with the required heading. See the example given below.

<lh> My Shopping List </lh>

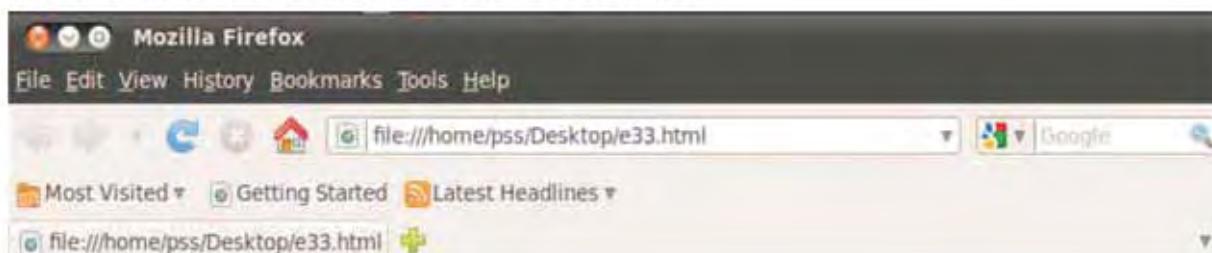
Ordered List

An ordered list contains items along with numbers or alphabets instead of bullets. To create an ordered list in HTML, the `` tag and `` tag pair is used. As usual, the items of the list are enclosed within the `` and `` tag pair. Code listing 4.3 shows the HTML example that creates an ordered list.

```
<html>
<body>
<font color = "Blue">
    <h1>An ordered list: </h1>
</font>
<!-- ----- -->
<ol >
<font size="6">
<b>
    <li>Chocolates </li>
    <li>Ice-creams </li>
    <li>Dry fruits </li>
</b>
</font>
</ol>
<!-- ----- -->
</body>
</html>
```

Code Listing 4.3 : HTML code for an ordered list

Output of the code listing 4.3 is shown in figure 4.4.



An ordered list:

1. **Chocolates**
2. **Ice-creams**
3. **Dry fruits**

Done

Figure 4.4 : Ordered list

Generally the ordered list always starts with number 1. You may start the ordered list with a specific number. Say, you want to start your list with a number 6; you may use start attribute with `` tag as shown below.

```
<ol start ="6">
```

Modify the HTML code given in code listing 4.3 in order to start the list with number 6.

Ordered List with Alphabets

Instead of numbers, we may use alphabets such as 'A', 'B', 'C' or 'a', 'b', 'c'. We may also use Roman numbers. This can be done by using type attribute with the `` tag. Table 4.1 illustrates possible values for the type attribute.

Value	Description
1	Numbers
A	Uppercase alphabets
a	Lowercase alphabets
I	Uppercase Roman numbers
i	Lowercase Roman numbers

Table 4.1 : Values of type attributes

An example of HTML code is provided in code listing 4.4. The code will print two ordered lists one with alphabets and another with small Roman numbers starting with number 10.

```

<html>
<body>
<font color="Blue">
    <h1>An ordered list: </h1>
</font>
<!-- ----- -->
<font size="6">
</b>
<ol type=A>
    <li>Chocolates </li>
    <li>Ice-creams </li>
    <li>Dry fruits </li>
</ol>
</b>
</font>
<!-- ----- -->
<font size="6">
</b>
<ol type='i' start="10">
    <li>Chocolates </li>
    <li>Ice-creams </li>
    <li>Dry fruits </li>
</ol>
</b>
</font>
<!-- ----- -->
</body>
</html>

```

Code Listing 4.4 : HTML code to print Roman numbers as bullets

Output of the HTML code listing 4.4 is given in figure 4.5. You may try working with different values of the type attribute.



Figure 4.5 : Ordered lists with alphabets and Roman numbers

Nested List

It is possible to have a list within a list. This is called nesting of list or nested lists. Types of both the lists may differ. An example HTML code is given in code listing 4.5.

```
<html>
<body>
<font color = "Blue">
    <h1>An ordered list: </h1>
</font>
<!-- ----- -->
<ul >
<font size= "6 ">
<b>
    <li>Chocolates    </li>
    <font size= "4">
        <ol type="a">
            <li> Dark chocolates      </li>
            <li> Fruit and nuts Dark </li>
        </ol>
    </font>
</b>
</font>
<!-- ----- -->
<li>Ice-creams   </li>
<ol type="a">
    <font size="4">
        <li> Vanilla           </li>
        <li> Chocolate chips   </li>
    </font>
</ol>
<!-- ----- -->
</b>
</font>
</ul>
</body>
</html>
```

Code Listing 4.5 : HTML code for nested lists

Output of code listing 4.5 is shown in figure 4.6.

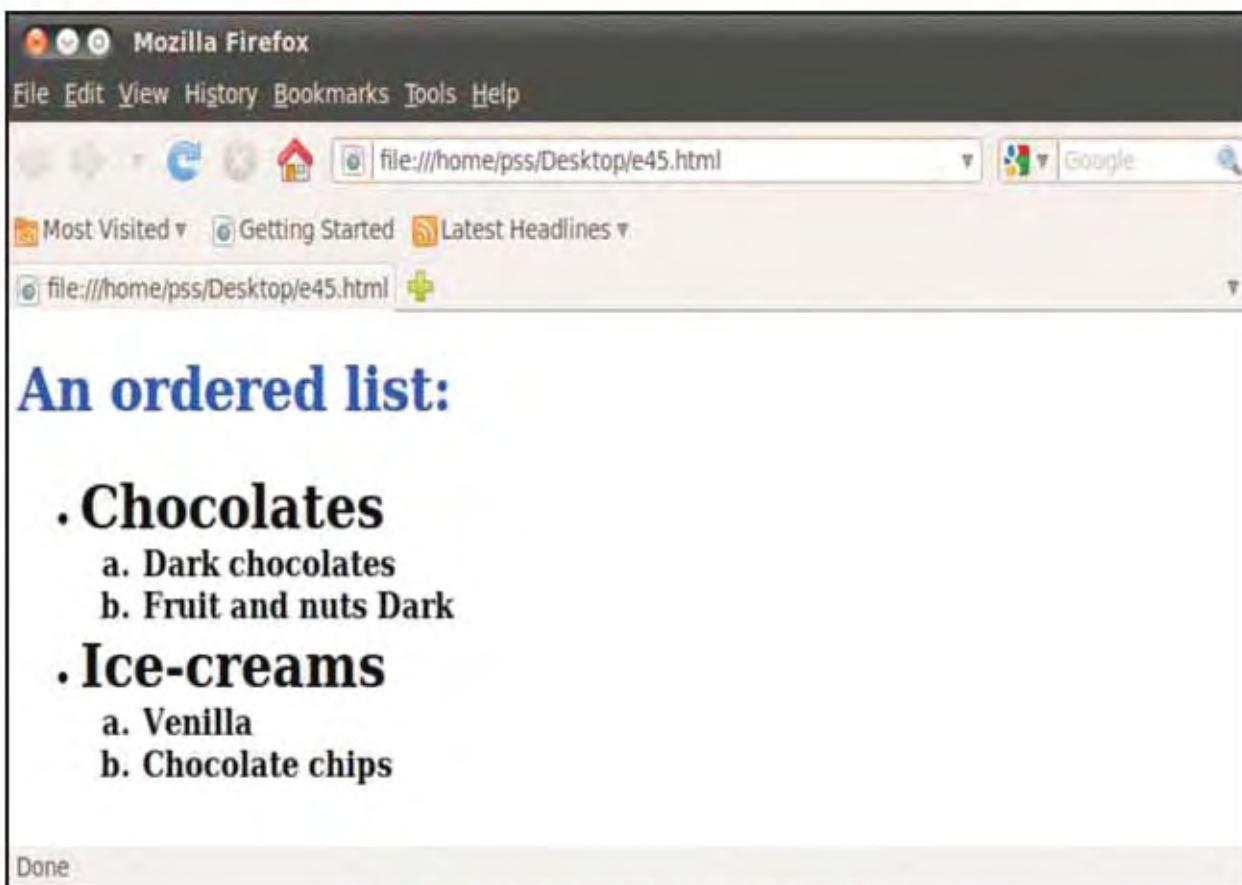


Figure 4.6 : Nested lists in HTML

There are other structures that publish the content in more structured way. One of such structure is called table. Following section presents tables in HTML.

Table Handling in HTML

Representation of information in proper format increases degree of understanding and facilitates ease of its use. Can you imagine a shopping bill printed in continuous paragraph manner ? What about your results sheet showing marks of different subjects? If it looks like the figure 4.7, it will be certainly very difficult for you to read. Not only it is effort taking and time consuming, but it also leads to misinterpretation of information.

Modern School Affiliated to Star Education, Gujarat Mr. Arvind B Patel Seat No 123
March 2013 Subjects: English, Gujarati, Science, Maths, SS*
Internal Marks (40): 20, 30, 25, 35, 25 External Marks (60): 35, 32, 48, 40, 35
Class: First Class SS*: Social Science

Figure 4.7 : Weird mark statement example

Now look at figure 4.8. Obviously there is no confusion at all. Name of the school, name of the student, date, all the marks, and result are clearly visible.

Modern School					
<i>Affiliated to Star Education, Gujarat</i>					
Mr. Arvind B Patel		Seat No 123		March 2013	
Subject:	English	Gujarati	Science	Maths	SS *
Internal Marks (40):	20	30	25	35	25
External Marks (60):	35	32	48	40	35
Class: First class					
SS*: Social Science					

Figure 4.8 : A sample mark sheet in proper format

Tables in HTML are very efficient in presenting structured information. A table contains information separated in form of grids. You might have used special note book for your mathematics calculation having rows and columns. Similarly, information can be presented in form of row and column in a table. The next section illustrates how a small table can be displayed on a web page using HTML tags. Later, we will see each tag that generates a table in detail.

Creating a Small Table

To create table in HTML, `<table>` and `</table>` tags are used. That is, starting and ending of a table are marked by these tags. Within the `<table>` and `</table>` tags, we may use attributes such as caption (`title`) of the table, table border, table rows and columns.

Main content of a table is formed by rows and columns. A row in a table is defined by `<tr>`, the table row tag. The first row of a table is heading row, which is denoted by the `<th>`, table heading tag. Entries of remaining rows are entered using `<td>` tag. See code listing 4.6 that shows an example HTML code.

```

<html>
<body>
<h1>My First Table </h1>
<p>My first table is as follows.</p>
<table border="2">
<tr>
    <td> This is Row 1, Column 1 </td>
    <td> This is Row 1, Column 2 </td>
</tr>
<tr>
    <td> This is Row 2, Column 1 </td>
    <td> This is Row 2, Column 2 </td>
</tr>
</table>
</body>
</html>

```

Code Listing 4.6 : HTML code for simple table

You can write the code in your editor and see the output in a browser. It will resemble the output as shown in figure 4.9.



Figure 4.9 : A simple HTML table displayed on a web page

The HTML code shown in code listing 4.6 uses table tag with some attribute and component. The first attribute used within the table tag itself is the border attribute to print border of pixel size 2. The table further defines a row with <tr> tag. The table shown in figure 4.9 has two rows; hence two sets of <tr> tags must be used.

Within a row, with the help of <td> tag pairs, cells are defined. The attributes and tags used in this example are discussed later in this chapter. The objective of the example is to understand the row and column formation of the table. This arrangement is also known as grid type arrangement. Let us take a detailed view on other table tags.

Table Tag and its Attributes

The first tag that is used to define a table is table tag. The <table> tag can carry many attributes. Some of them are not much popular now-a-days; however, you may use them. The attributes are shown in table 4.2.

Attribute	Description
align	This attribute indicates alignment of table.
bgcolor	This attribute specifies background of the table.
border	This attribute specifies table border.
cellpadding	This attribute leaves specified gap between edges of the cells and their content.
cellspacing	This attribute manages space between each cell of the table.
dir	This attribute specifies the direction of text that is displayed in the table.
frame	This attribute controls the outermost border of the table.
rules	This attribute controls the presentation of inner borders of table.
summary	This attribute presents description of the table.
width	This attribute specifies width of the table.

Table 4.2 : Table tag attributes

Some commonly used attributes are discussed below:

Border Attribute

In the example discussed in code listing 4.6, a border with pixel size 2 is created using the border attribute. This attribute creates a border around the table as well as around each individual cell. The width of the border is given in form of pixel. Use of the border attribute is optional. If you use the attribute with value 0 (zero), no border will be visible. Consider the HTML code that we have created earlier using the code listing 4.6. Let us modify the value of the border attribute border by changing the value of border attribute as shown below.

```
<table border="4" align="left">
```

Now once again check the appearance of the table. It should look similar to the one shown in figure 4.10.

You might have noticed one more attribute we have used in the example given in previous paragraph; that is align attribute. Did you notice anything? You might not have noticed any change. Try to change the alignment of the table to the right and now you may able to notice change in the table alignment. We will see other alignment values later.

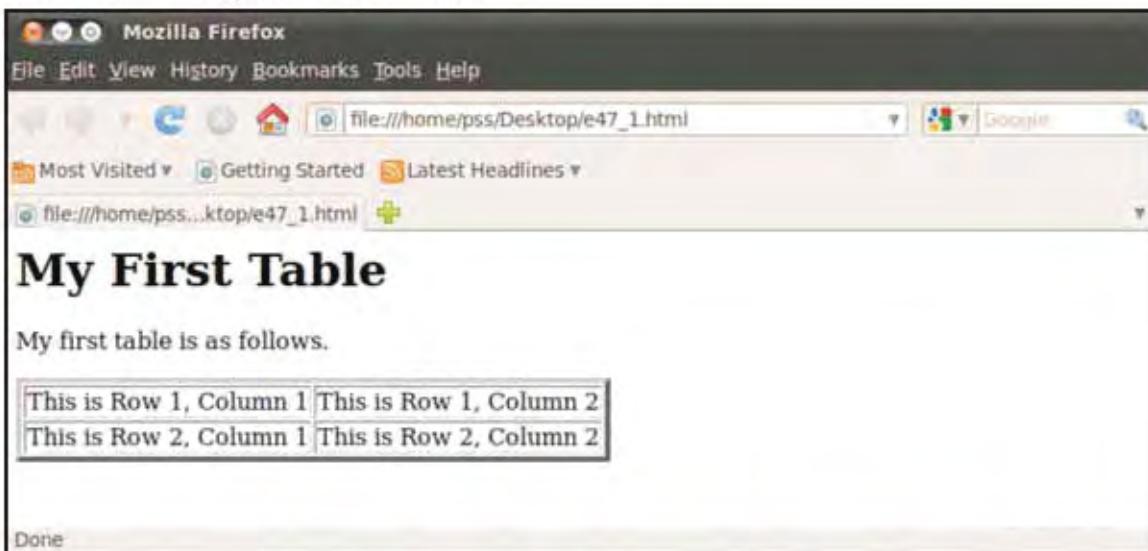


Figure 4.10 : Table with changed border value

Bgcolor Attribute

Let us make the table more attractive and colourful by adding background colour. To do this, we have to use bgcolor attribute. The bgcolor attribute sets the background colour for the table. The value of this attribute is either a colour name or a six digit hexadecimal code. Modify the first line of the HTML code you are currently experimenting (as given in code listing 4.6) as follows.

```
<table border="4" align="right" bgcolor="yellow">
```

Check the output by viewing the code in a browser.

Cellpadding Attribute

If two table cells are placed next to each other and both contained text, there may be a problem. If there is not enough space between the edges of the cells and the text, the words would combine with each other, making them hard to read. Similarly, if there is a border around each cell and

the text touches the border, it will be hard to read. By adding some space as padding to the cell makes their contents easier to read.

The cellpadding attribute is used to create a space between the edges of a cell and its contents. The cellpadding attribute pads some spaces inside each wall of the cell of the table. The value can be specified either in pixels or as a percentage value. Here, the value specified in percentage refers to percentage of the width of each cell of the table. See the following example.

```
<table border="4" align="center" bgcolor="pink" cellpadding="25">
```

Alternatively, you may use percentage value as follows.

```
<table border="4" align="center" bgcolor="yellow" cellpadding="20%">
```

You might have noticed that we have also changed values of align and bgcolor attributes. Check the changed appearance of the table in a browser.

Cellspacing Attribute

The cellspacing attribute is used to create a space between the cells of the table. The amount of the space can be specified either in pixels or as a percentage value. Here, the percentage is a percentage of the width of each cell of the table. See the following example.

```
<table border="4" align="center" bgcolor="yellow" cellspacing="20%">
```

Or

```
<table border="4" align="center" bgcolor="yellow" cellspacing="25">
```

Width Attribute

The width attribute is used to specify the width of the table. The value as usual is given either in pixels or in percentages of the available space as follows.

```
<table border="4" align="center" bgcolor="pink" cellspacing="25" frame="box" rules="cols" width="50%">
```

Or

```
<table border="4" align="center" bgcolor="pink" cellspacing="25" frame="box" rules="cols" width="50%">
```

Align Attribute

The align attribute specifies the position of the content of all of the cells in the row. Just like simple text the cell content can also be aligned. Table 4.3 lists the possible values for the align attribute.

Value	Description
left	Content is left aligned. This is the default case for normal text.
right	Content is right aligned.
center	Content is centered horizontally within the cell. This is the default case for headings.
justify	Text within the cell is justified to fill the cell.
char	Cell contents are aligned horizontally around the first instance of a specific character (for example, numbers could be aligned around the first instance of a decimal point).

Table 4.3 : Possible values for the align attribute

If the align attribute has a value of char, then the contents of each cell of the table within the row will be aligned around the first instance of a specific character. The given character is known as an axis character. The default character for this attribute is the decimal place. By using the decimal point as a char for alignment (char=".") , the existing decimal figures on the page would be aligned by the decimal point as shown as mentioned:

	1	2	3	.	5	6	
		4	6	.	1	2	7
3	8	1	6	.	4	5	3

After having a detailed look at the table tag and its attributes, let us now see table row tag in following section:

Table Row Tag

The <tr> tag is used to display a row in a table. Everything appears within a <tr> tag should appear on the same line. It can carry three attributes as shown in table 4.4.

Attribute	Description
align	Content of the row is aligned as specified.
bcolor	Background colour for selected row.
valign	Specifies the vertical alignment of the contents of each cell in a row.

Table 4.4 : Attributes of the table row tag

Let us now see how these attributes can be used in the HRML code.

Align Attribute

The align attribute indicates position of the content of all of the cells in the row. The general form of an alignment attribute is as follows :

align="alignment"

The align attribute takes values such as left, right, center, justify and char as shown in table 4.3. Use these attributes in any valid HTML code displaying table.

Bgcolor Attribute

Earlier, we have seen the bcolor attribute to paint background of the table by giving colour code in hexadecimal or name of colour. What if, only a row of a table is required to be painted with a specific colour? The answer is again the bcolor attribute with either colour name or colour code in hexadecimal. The bcolor attribute sets the background colour for the row.

The bcolor attribute is commonly used on the <tr> element to paint alternate rows of a table with different colours. This will make it easier to read across each row. Try the HTML example given in code listing 4.7.

```

<html>
<body>
<table border="1">
<tr >
    <th>Name of player</th>
    <th>Points earned</th>
</tr>
<tr bgcolor="lightGreen">
    <td>Disha</td>
    <td>110</td>
</tr>
<tr >
    <td>Sweety</td>
    <td>100</td>
</tr>
<tr bgcolor="lightGreen">
    <td>Gayatri </td>
    <td>90</td>
</tr>
</table>
</body>
</html>

```

Code Listing 4.7 : HTML code for painting alternative rows with light green colour

The output of code listing 4.7 will look as illustrated in figure 4.11.

The screenshot shows the Mozilla Firefox browser window. The title bar says "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The address bar shows the URL "file:///home/pss/Desktop/e412.html". Below the address bar are several icons and links: "Most Visited", "Getting Started", "Latest Headlines", and a link to "file:///home/pss...sktop/e412.html". The main content area displays a table with three rows. The first row has a green background and contains the column headers "Name of player" and "Points earned". The second row has a white background and contains the data "Disha" and "110". The third row has a green background and contains the data "Sweety" and "100".

Name of player	Points earned
Disha	110
Sweety	100

Done

Figure 4.11 : Table with rows with colourful background

Valign Attribute

The valign attribute specifies the vertical alignment of the contents of each cell in a row. This can be done by using the general syntax as follows.

valign="Position"

Table 4.5 shows the possible values of the valign attribute.

Value	Description
top	Aligns content with the top of the cell.
middle	Aligns content in the center of a cell.
bottom	Aligns content with the bottom of the cell.
baseline	Aligns content so that the first line of text in each cell starts on the same horizontal line.

Table 4.5 : Possible values for the valign attribute

An example of tr tag is <tr width="150" valign="bottom">, embed it in the HTML code you are practicing and check it in a browser. Try HTML code given in code listing 4.8.

```
<!DOCTYPE html>
<html>
<head>
<title>My Timetable</title>
</head>
<body>
<table border="1" style="width: 100%; border-collapse: collapse;">
| Subject | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Period | 1 to 2 | 2 to 3 | 3 to 4 | 4 to 5 | 5 to 6 | 6 to 7 | 7 to 8 |
| 1 to 2 | Maths | Maths | Science | Gujarati | Maths | Science | Gujarati |
| 2 to 3 | Maths | Maths | Gujarati | Maths | Gujarati | Assembly |  |
| 3 to 4 |  |  |  |  |  |  |  |
| 4 to 5 |  |  |  |  |  |  |  |
| 5 to 6 |  |  |  |  |  |  |  |
| 6 to 7 |  |  |  |  |  |  |  |
| 7 to 8 |  |  |  |  |  |  |  |


</body>
</html>
```

Code Listing 4.8 : An HTML code to create timetable

The code shown in code listing 4.8 creates two rows of a simple timetable for a class. If you see this code in a browser it will look as shown in figure 4.12. Do you think the table is incomplete? In that case, complete the timetable by creating some more rows.

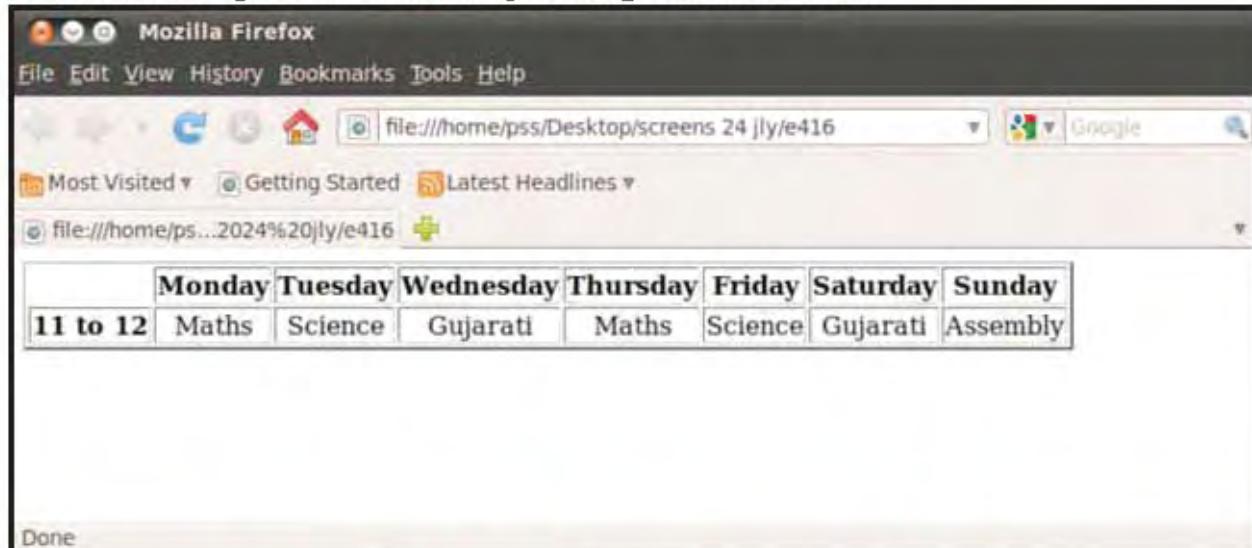


Figure 4.12 : Time table generation using HTML code

Cell Representation using **th** and **td**

Each cell in a table is represented by either a **<td>** or **<th>** tag. In a way they facilitate breaking of a row into multiple columns. If a heading is to be specified, a **<th>** tag is used. If table data is to be presented, **<td>** tag is used.

By default, content of a **<th>** tag is usually displayed in a bold font, horizontally aligned in the center of the cell. The content of a **<td>** element is displayed as left aligned normal text. Both the **<th>** and **<td>** tags can hold the same set of attributes. Effect of these attributes is limited to a single cell which carries them. Any effect these attributes cause, will override settings for the whole table or any upper level containing element such as a row. Effect caused by the **<th>** and **<td>** attributes is final.

In addition to the universal attributes and the basic event attributes, the **<th>** and **<td>** tags can also have the attributes shown in table 4.6.

Attribute	Description
abbr	Provides an abbreviated version of content of the cell.
align	Aligns content of the cell.
bgcolor	Adds background to the cell.
char	Manages the cell content to be aligned around the first instance of the specified character.
colspan	Indicates number of columns that the cell spans across.
headers	Indicate corresponding headers to the cell.
height	Specifies height of the cell.
nowrap	Stops text from automatically wrapping into a new line within the cell.
rowspan	Indicates number of rows that the cell spans across.
valign	Specifies vertical alignment of the cell.
width	Specifies width of the cell.

Table 4.6 : Attributes of **<td> and **<th>****

We will learn about these attributes in detail as and when we proceed with the chapter.

Adding Caption to a Table

To give name to the table <caption> tag is used. This tag is required when you display a specific table along with the table name. Table name indicates what the table is for; for a result purpose, for timetable or for a diet chart. Most of the browsers display the contents of the caption above the table on a centered fashion.

Addition of the following line before the first row of any valid HTML encoded table generates table captions as "This is our time table".

<caption> This is our timetable </caption>

Code listing 4.9 shows how to create a time table using HTML.

```
<html>
<body>
    <table border="2">
    <!-- ----- -->
        <caption> <h1> <font color="Brown" >
            This is our time table
            </font> </h1> </caption>
    <!-- ----- -->
        <tr>
            <th> </th>
            <th> Monday </th>
            <th> Tuesday </th>
            <th> Wednesday </th>
            <th> Thursday </th>
            <th> Friday </th>
            <th> Saturday </th>
            <th> Sunday </th>
        </tr>
    <!-- ----- -->
        <tr align = "middle">
            <th> 11 to 12</th>
            <td> Maths</td>
            <td> Science </td>
            <td> Gujrati</td>
            <td> Maths</td>
            <td> Science </td>
            <td> Gujrati</td>
            <td> Assembly </td>
        </tr>
    </table >
</body>
</html>
```

Code Listing 4.9 : Adding caption to the table

Adding caption to the table increases the degree of understanding and user friendly presentation of table on web page. Figure 4.13 shows the output of code listing 4.9.

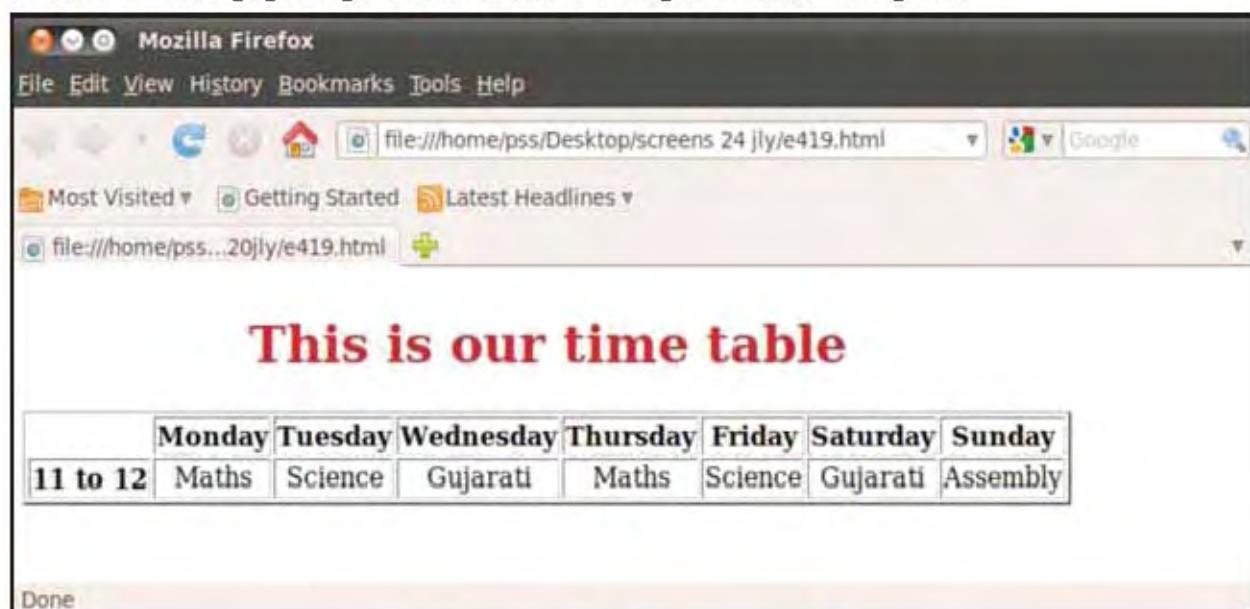


Figure 4.13 : Table with caption

Observe that the time table shown in figure 4.13 is incomplete. Students may complete the same and observe the output.

Nested Tables

Many times we need to add table within a table. The example shown in code listing 4.10 shows a nested table.

```
<html>
<body>
<!-- ----- -->
<table border="4">
<caption> <h1>Conference Activities </h1></caption>
<tr>
<th> </th>
<th width = "40%"> Morning </th>
<th> Afternoon </th>
</tr>
<tr>
<th> Day 1</th>
<td> Inauguration </td>
<td> Key-note Address</td>
</tr>
<!-- ----- -->
```

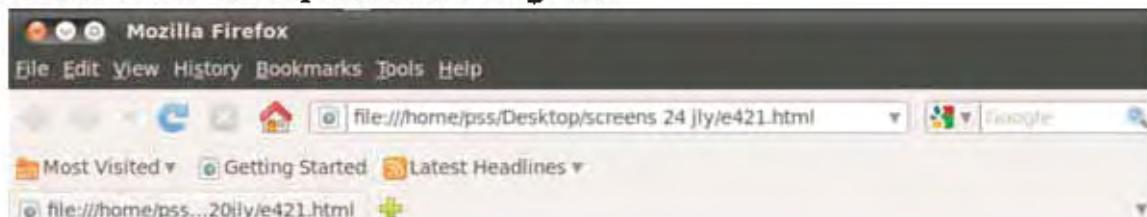
```

<!-- -->
<table border="1" bordercolor="pink" frame="box" align="center">
    <tr>
        <td> Day 2 </td>
        <td> Paper Presentations</td>
        <td>
            <table border="1" bordercolor="pink" frame="box" align="right">
                <caption> <h3> Inner Table </h3></caption>
                <tr>
                    <th> Teachers</th>
                    <th> Students </th>
                </tr>
                <tr>
                    <td> Meeting </td>
                    <td> Quiz </td>
                </tr>
            </table>
        </td>
    </tr>
</table >
<!-- ----- -->
</body>
</html>

```

Code Listing 4.10 : Nested table

Figure 4.14 shows the output of code listing 4.10.



Conference Activities

	Morning	Afternoon	
Day 1	Inauguration	Key-note Address	
Day 2	Paper Presentations	Inner Table	
		Teachers	Students
Teachers	Students		
Meeting	Quiz		

Done

Figure 4.14 : Nested tables

You might have noticed that the inner table has background colour and border. Both the outer as well as inner tables have their captions.

An Example of a Table

Figure 4.15 shows an example of a table that you have practically observed. It is about a menu of a typical Punjabi restaurant containing items of your choice along with prices.

Star Restaurant					
Starters		Roti and Bread		Curries	
Item	Price	Item	Price	Item	Price
Paneer Tikka Dry	100.00	Plain Roti	10.00	Paneer Bhurji	100.00
Panner kabab	100.00	Tandoori Roti	20.00	Mixed Veg	100.00
Drinks		Desserts		Water	
Item	Price	Item	Price	Item	Price
Tea	20.00	Gulab Jamun	30.00	Regular	---
Coffee	20.00	Rasgulla	35.00	Mineral	15
Koko juicy	25.00	Pudding	40.00	Sparkling	28
Mengo delight	25.00	Ice-cream	50.00		
The prices are in Indian rupees. Customers have to pay extra taxes. Please wait for 20 minutes after giving order.					

Figure 4.15 : Restaurant Menu card

Code listing 4.11 shows partial HTML code to generate restaurant menu card shown in figure 4.15.

```
<html>
  <head> <title> Star Restaurant Menu </title> </head>
  <body>
    <center>
      <table width="75%" border="1" bordercolor="#000000" bordercolordark="#000000"
             cellspacing="0">
        <tr align="center">
          <th align="center" colspan="6"> <h1> Star Restaurant </h1> </th>
        </tr>
        <tr>
          <td colspan="2" align="center"> <h2> Starters </h2> </td>
          <td colspan="2" align="center"> <h2> Roti and Bread </h2> </td>
          <td colspan="2" align="center"> <h2> Curries </h2> </td>
```

```

</td>
</td>
<td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
<td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
<td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
<td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
<td align="center"> <strong> <font color = "Blue" > Item </font> </strong> </td>
<td align="center"> <strong> <font color = "Blue" > Price </font> </strong> </td>
</td>
<tr bgcolor="#CCCCCC">
    <td align="center"> <strong> Paneer Tikka Dry </strong> </td>
    <td align="center"> 100.00 </td>
    <td align="center"> <strong> Plain Roti </strong> </td>
    <td align="center"> 10.00 </td>
    <td align="center"> <strong> Paneer Bhurji </strong> </td>
    <td align="center"> 100.00 </td>
</td>
</td>
<td align="center"> <strong> Paneer Kabab </strong> </td>
<td align="center"> 100.00 </td>
<td align="center"> <strong> Tandoori Roti </strong> </td>
<td align="center"> 20.00 </td>
<td align="center"> <strong> Mixed Veg </strong> </td>
<td align="center"> 100.00 </td>
</td>
<!-- Remaining part of the table may be completed using similar code... -->
<tr align="center">
    <td align="center" colspan="6"> The prices are in Indian rupees. <br>
    Customers have to pay extra taxes. <br>
    Please wait for 20 minutes after giving the order.
</td>
</td>
</table>
</center>
</body>
</html>

```

Code Listing 4.11 : Displaying restaurant menu card using HTML tables

Note that the HTML code given in code listing 4.11 displays a partial table. Students can complete the remaining part of the table by adding similar HTML code.

Frames in HTML

Frame in an HTML document is used to combine multiple web pages and display them as a single web page. Frames divide a browser window into several parts or sub windows, each containing an independent web page. By dividing the browser window in many frames, you can handle different HTML codes individually and manage loading and reloading of them. A collection of frames in the browser window is known as a frameset.

It is to be noted that some browsers do not support the frameset. To create a frameset document, first we have to create a <frameset> element, which is treated as container of different frames. The frameset defines the division of the browser window. Within the frameset, each frame is represented by a <frame> and </frame> tag pair.

Within the frameset, you may add <noframes> element, which provides an alternative message for users, when the browser used does not support frames. See the example given in code listing 4.12.

```
<html>
<head>
<title>Creating example</title>
</head>

<frameset rows="20%,60%,20%">
<frame src="top.html" />
<frame src="main.html" />
<frame src="bottom.html" />
<noframes>
    <body>
        Your browser does not support frames.
    </body>
</noframes>
</frameset>
</html>
```

Code Listing 4.12 : HTML code to create frames

Write the HTML code shown in code listing 4.12 and save it as frame.html. Create the required files such as top.html, main.html and bottom.html. Add appropriate contents of your liking within these files. Once you create these files we are ready to test them in a browser. Our output looks similar to the one shown in figure 4.16. Note that the output that you will get will differ based on the contents that you have added in the files.

From figure 4.16 it may be observed that the browser window is divided into three parts. The top portion of the window is painted with blue colour and it shows the contents of HTML file, top.html. The middle portion prints simple table for a restaurant menu. The middle portion is displayed through the HTML file called main.html. The bottom portion, painted with light yellow colour is displayed through the HTML file called bottom.html.

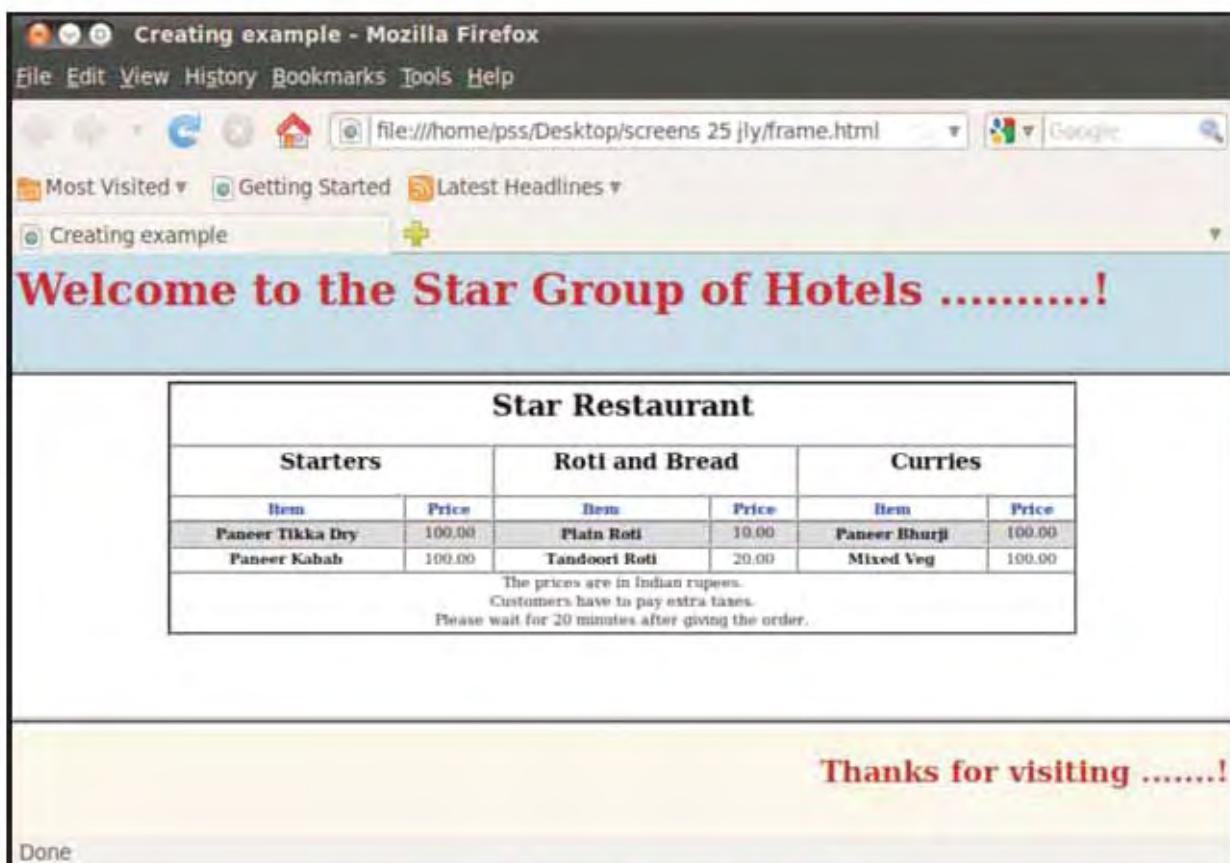


Figure 4.16 : Frames in HTML

Note that it is also possible to divide the browser window in vertical parts. We may use frameset element with columns (cols) as follows to split browser windows vertically

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

Here we have tried to divide the browser window in three vertical parts. Change the code `<frameset rows="20%,60%,20%">` in code listing 4.12 to `<frameset cols="25%,*,25%">` and try to see the output again.

Summary

In this chapter we have seen different ways to create lists and tables in HTML. We have seen the ordered as well as unordered lists with different attributes. The nested lists are also introduced in this chapter. We learnt to create simple tables as well as nested tables. The chapter has discussed attributes that can be applied to a whole table, attributes for individual table rows and attributes for individual cells.

EXERCISE

1. Explain how list can be defined in HTML document by giving suitable example.
2. Explain various types of lists in HTML.
3. Write a short note on nested lists in HTML.
4. Write a short note on table handling in HTML.

5. Choose the correct option from the following :

LABORATORY EXERCISE

1. Write an HTML code to define a table showing name of your friends and score in percentages they obtained in test of a school. Give appropriate caption and column heading for the table.
2. You may have seen your parent managing household expenses and incomes by writing accounts in a diary. Write an HTML code to prepare a table showing expenses and income for household activities. Include sources of income, major expenses and balance remained.
3. Write an HTML code to prepare the statement of marks (mark sheet) as shown in the first section of this chapter.
4. Write an HTML code to prepare time table of your class. Take hints from the time table shown in the chapter.
5. Write an HTML code to prepare a table for a multiplex cinema showing movies for its display as well as advertisement. The multiplex is having three screens and four shows on 9:00 am, 1:00 pm, 5:00 pm and 9:00 pm for all the screens. The late night show on third screen offers three different regional movies: one in Gujarati, second in Marathi and third in Telugu. Tentative look of the table is as follows. Add movies of your choice in each cell. Also give caption to the inner table as "Regional Movies".

Timings→	9:00 am	1:00 pm	5:00 pm	9:00 pm
Screen 1				
Screen 2				
Screen 3				Regional Movies

6. Write an HTML code to create list of the items that you want to shop. Categories items in groups such as books, vegetables, and food items. See the following sample framework which shows how the list should be.

List of items to be purchased

- I. Books
 - a. Textbook of science
 - b. Practice book of maths
 - c. Notebook
- II. Vegetables
 - a. Tomatoes
 - b. Spinach
 - c. Peas
- III. Food items
 - a. Chocolates
 - b. Butter
 - c. Bread



Introduction to Calc

Calc is an electronic spreadsheet package from the OpenOffice suite. Spreadsheet is a type of package that is used for fast and accurate calculations as well as formatting of data in a document. The spreadsheet package is used for managing financial and accounting documents, creating data reports, generating invoices, analysing data from scientific and statistical researches, and for doing variety of calculations on data. A spreadsheet program can also store, manipulate and create graphical representations of data.

A spreadsheet allows entering data in row and column fashion. You may recall; in your childhood, you might have used special notebook with small cells marked with rows and columns to practice mathematics. Spreadsheet is a long sheet of rows and columns on computer screen to do data analysis and calculation. In other words, a spreadsheet is a grid which interactively manages and organizes data in rows and columns. To facilitate the operations, spreadsheet packages also allow formulas besides data and formatting mechanisms. User can enter data interactively into a spreadsheet page, format it and calculate/analyse them for decision making. Besides user's data and formulas, spreadsheet packages also provide built-in formulas/functions for common mathematical, financial, statistical, and logical operations in very sophisticated manner. Because of these abilities the spreadsheet packages are used as a universal programme for structured data preparation and processing. LibreOffice Calc [www.libreoffice.org/], which closely resembles the Calc is also a free and open-source spreadsheet package.

Beyond a spreadsheet

As spreadsheets became larger, they became more difficult to manage. To handle with the increasing size of the spreadsheet, a concept of a workbook was identified. Main purpose of a workbook is to manage collections of spreadsheets. Beside a workbook, online spreadsheets are also becoming popular.

Typical Applications of Spreadsheet Packages

Spreadsheet packages are widely used for data analysis and accounting applications. Table 5.1 presents a list of typical activities that can be generally done through a spreadsheet package.

Activity	Description
Balance sheet	Statements of financial position and summary in typical account format.
Result analysis and merit list preparation	Calculating, sorting and filtering results of some activities such as examinations.
Statistical data analysis	Sorting, arranging, and applying statistical operations such as finding mean, median, probability, etc.
Financial activities	Financial activities such as loan instalment calculation, interest calculation, etc.
Personal activities	Personal activities such as tracking personal weights, managing list of activities, items and events; such as preparing list of guests for a party or a forthcoming event.

Table 5.1 : Popular applications of spreadsheet

Getting Started with Calc

The first thing you require to know is how to open the Calc. Follow the command sequence Applications → Office → OpenOffice.org Spreadsheet as shown in figure 5.1.

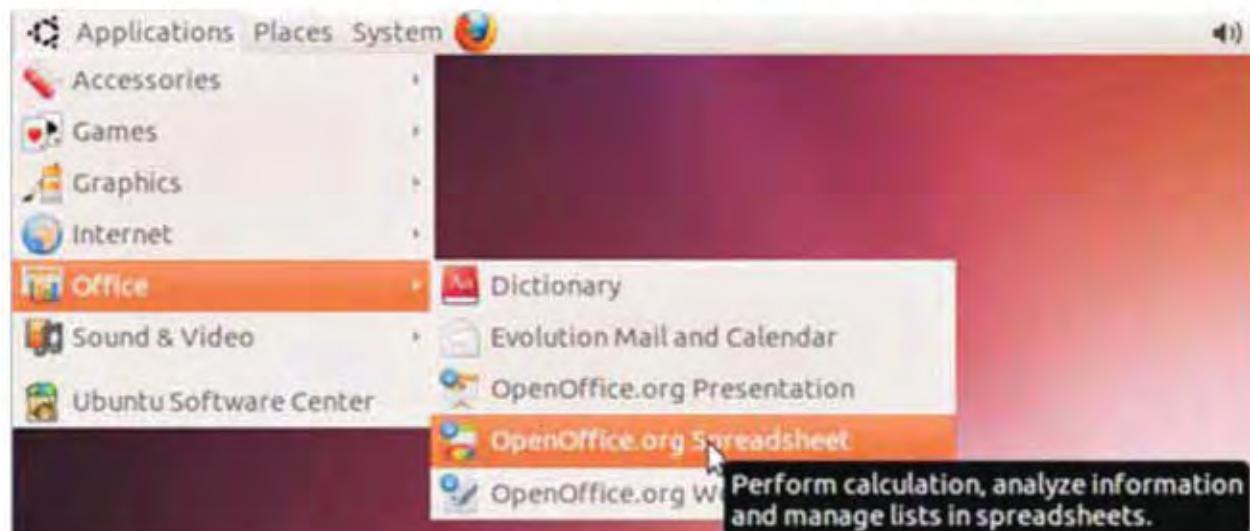


Figure 5.1 : Getting started with Calc

It will open an application document called a spreadsheet. A spreadsheet consists of multiple sheets; also called worksheets. By default, a new spreadsheet contains three worksheets, but the number of worksheets can be added or removed as per requirement. There are maximum 256 worksheets per Calc spreadsheet document.

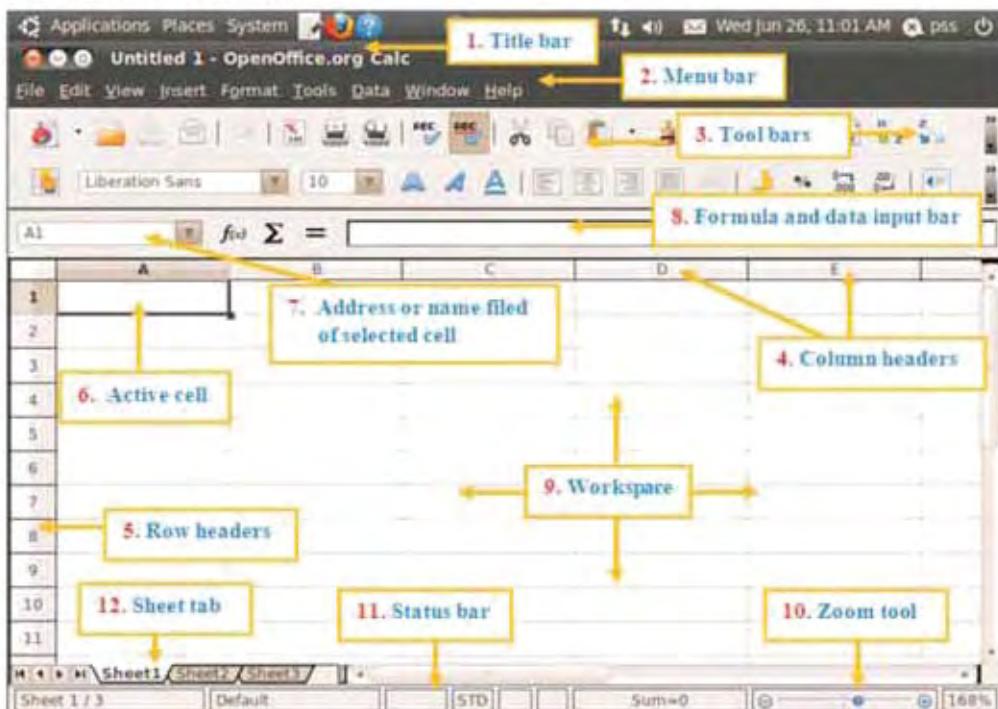


Figure 5.2 : An empty worksheet

Figure 5.2 presents an empty spreadsheet that you will see when you open a new document in the Calc. This is also known as a user interface of Calc. It shows different items with the number tags. Let us discuss them in detail.

Title Bar

The title bar is located at the top. It shows name (title) of the current spreadsheet. If you have just opened a new file, and not given name of the spreadsheet, it is an untitled spreadsheet. It is denoted as Untitled X, where X is a number. When you save a spreadsheet for the first time, you are asked to enter a name of your choice. It will let you know which workbook you are currently working with. Figure 5.3 shows the title bar.

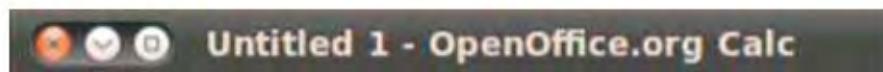


Figure 5.3 : Title Bar

Menu Bar

The menu bar contains menus with commands for various tasks. Each menu item represents a separate submenu. This submenu is also called pull down menu. To view the menu, just put the mouse cursor over it, click once and submenu appears. Figure 5.4 shows a typical menu bar.



Figure 5.4 : Menu Bar

The main menu commands are listed in table 5.2 with brief description of each.

Menu	Description
File	Commands to operate the entire document such as creating a new document, saving file, printing, print preview, etc.
Edit	Editing commands such as copy, paste, find & replace, etc.
View	Adding or removing elements of the user interface, page break preview, etc.
Insert	Insertion of rows, columns, worksheets, elements, objects, etc.
Format	Formatting cells, grouping of elements, sorting, conditional formatting, etc.
Tools	Additional tools such as spellchecking, document protection, formulas, error correction, etc.
Data	Data processing, data sorting, data filter, etc.
Window	New window, freezing cells, list of open OpenOffice.org documents
Help	Help about function, information about applications and version of software, etc.

Table 5.2 : Menus in Calc

If you select particular menu, then its corresponding submenu appears. You may have noticed a triangle (►) shape along with options available in the submenu. When such sign of triangle (►) appears, there are more options available; from which you can select the suitable one. Figure 5.5 shows this situation. In figure 5.5 you can see the 'New' choice under the File menu with the triangle (►) shape showing more options. These options are presented in form of another black column, as a vertical pop down menu.

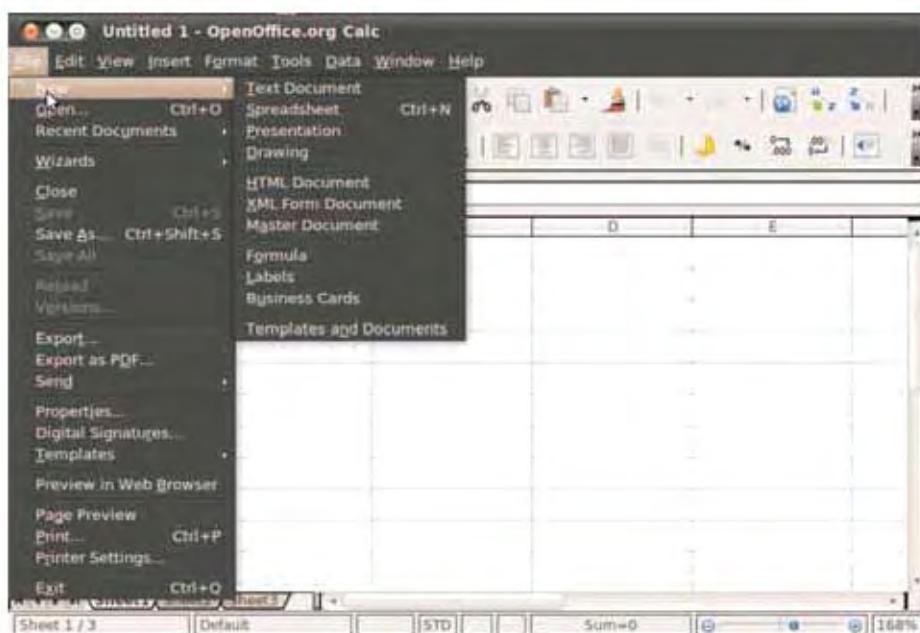


Figure 5.5 : Available options

If additional information is required to follow user's instructions, a dialog box appears indicated by (...) sign. Suppose you select **Save as** option; the Calc will further ask you about the filename and location where the file is to be saved as illustrated in figure 5.6.

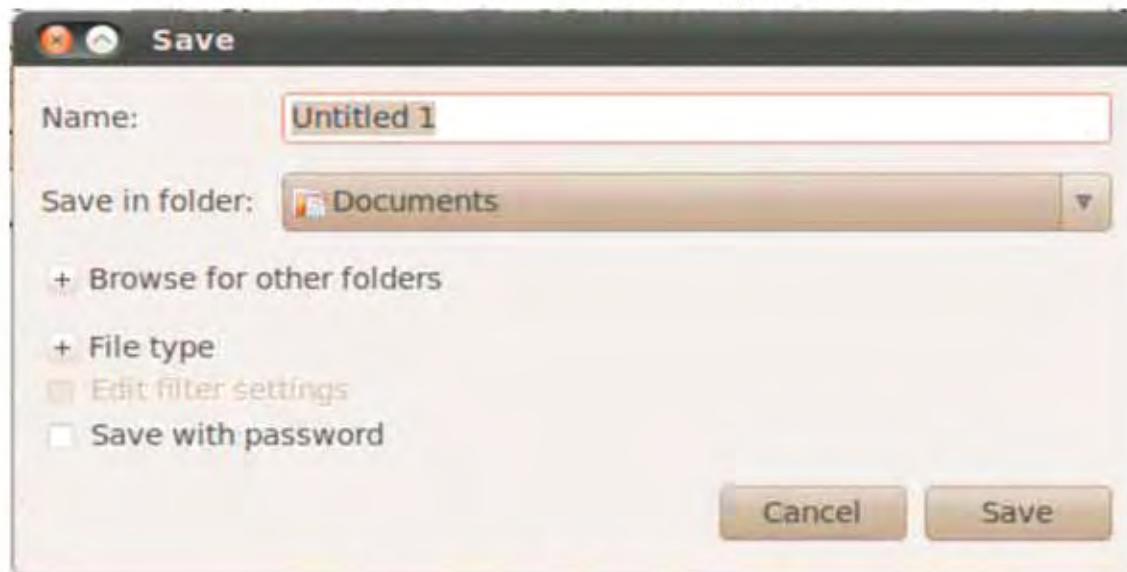


Figure 5.6 : Save dialog box

Similarly if you see "**Print**" option, you may also see the sign of dots (...) at the end of "**Print**" option; which indicates that a dialog box will appear with various print options. We will see about these menus later in detail.

Toolbars

A toolbar presents the most common commands in form of onscreen buttons. Just by clicking the mouse pointer over a button, you can select the utility. In case, you do not know the purpose of the button, you have to hover (put) the mouse pointer on it. Immediately it will display name of the corresponding function. Toolbars can be turned off and on as per the need by selecting **View** → **Toolbars**. The default (system given) toolbars are as follows.

- **Standard Toolbar** : As the name suggests, this toolbar contains most frequently used standard commands from the File and Edit menus. Recently used icons are reflected in this bar. Most commonly you will see icons for creation of a new Calc document, opening an existing document, checking spelling within the document, cutting and pasting content and printing document etc. On this standard toolbar, Figure 5.7 displays a standard toolbar.



Figure 5.7 : Standard Toolbar

- **Formatting Toolbar** : The formatting toolbar presents the most frequently used commands for formatting content of a cell. This toolbar allows you to do formatting quickly. The icons in this bar will vary as you use the application. If any icon is not available, you may go to far end of the formatting toolbar and select the option to add an icon. Figure 5.8 illustrates a typical formatting toolbar.



Figure 5.8 : Formatting Toolbar

- **Formula Toolbar** : Formula toolbar presents facilities for entering and editing formula within a cell. A typical formula toolbar is shown in figure 5.9.

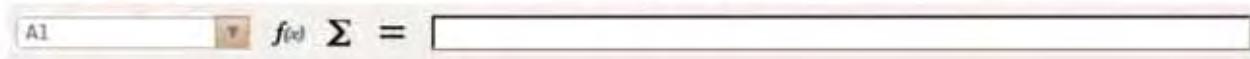


Figure 5.9 : Formula Toolbar

Buttons in toolbars can be modified. User can add buttons to the toolbar, remove them and adjust their sequence. You can add or remove toolbars as shown in figure 5.10.

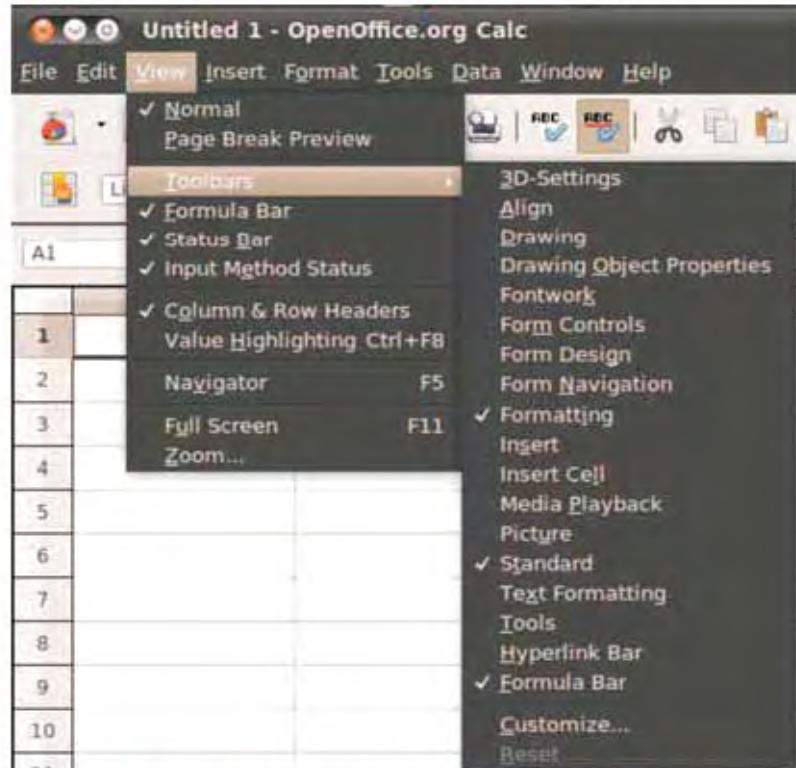


Figure 5.10 : Adding/Removing Toolbars

Rows, Columns and Cells

Each worksheet is divided into vertical columns and horizontal rows, forming cells. Both the rows as well as columns are numbered; columns with alphabet (and their combinations), and rows with numbers. Figure 5.11 highlights 3rd row in a worksheet.

Each **Column** in Calc is vertical series of cells. A column as a whole can be selected by clicking on the corresponding letter on the spreadsheet. Figure 5.11 shows 1st column in a worksheet.

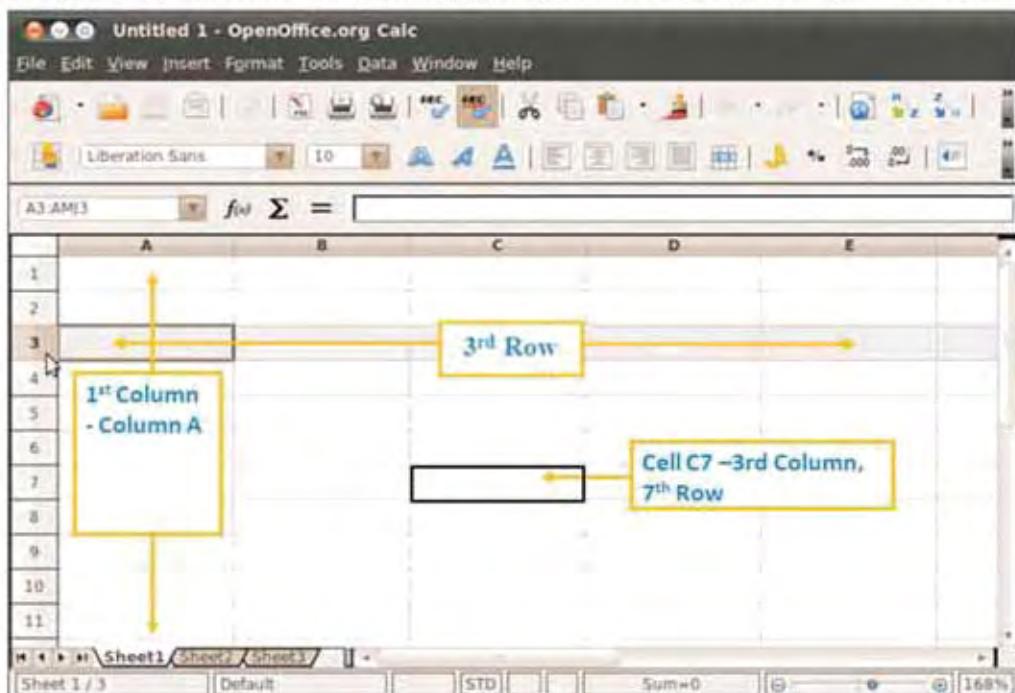


Figure 5.11 : Row in Calc

A cell is an intersection of a row and a column. It is identified by its column letter (letters) and row number, e. g. C7. Figure 5.11 displays cell C7. The cell which is currently selected is known as an active cell. Here C7 is shown to be an active cell. (Note that there will be a plus marker on bottom right side of the active cell in the actual screen). The square box appears here is an indicator of an active cell. The corner of the active cell is highlighted with filled small rectangle which is known as an autofill handle, which will be discussed in next chapter.

As we have a few alphabets (26), and number of rows is more, we need to use combination of alphabets as AA, AB, ..., AZ, BA, ..., BZ, CA, ...etc. The Calc contains 1,024 columns and 1,048,576 rows.

A cell is the basic element of a spreadsheet. Data, formulas are entered in a cell. That is, a cell holds individual elements such as text, numbers, and formulas. You can select multiple rows, columns or cells by clicking and dragging the cursor over the letters.

Formula and Data Input Bar

Check the big empty line on the top area of the worksheet, immediately below the toolbars. You have seen this in previous section as figure 5.9. This space is provided for user to enter data and formula. The data once entered, will appear in a particular cell.

The formula bar shows the cell, which is currently selected, on its left. The box indicated by fx on the right provides an area in which you can enter data or formulas into the cell.

Workspace

The empty grids of cells form an area where user's data appears. While entering data, you may directly click on a specific cell (to make it active) and enter data as well as formula. The cell may not show a formula, but displays its result in form of data.

Zoom Tool

The zoom tool can be used to zoom in/out a worksheet. You may zoom out your worksheet for better view. Figure 5.12 highlights the zoom tool from which you can observe that the worksheet is zoomed to 168%.

Scroll Bar

As the worksheet size is more than the computer screen (monitor) size, the Calc automatically provides you scroll bars on either sides of the sheet as shown in figure 5.12. That is, you can see vertical or horizontal scroll bar on screen. It is convenient to visualize data using the scroll bar, especially when the data do not fit into the screen.

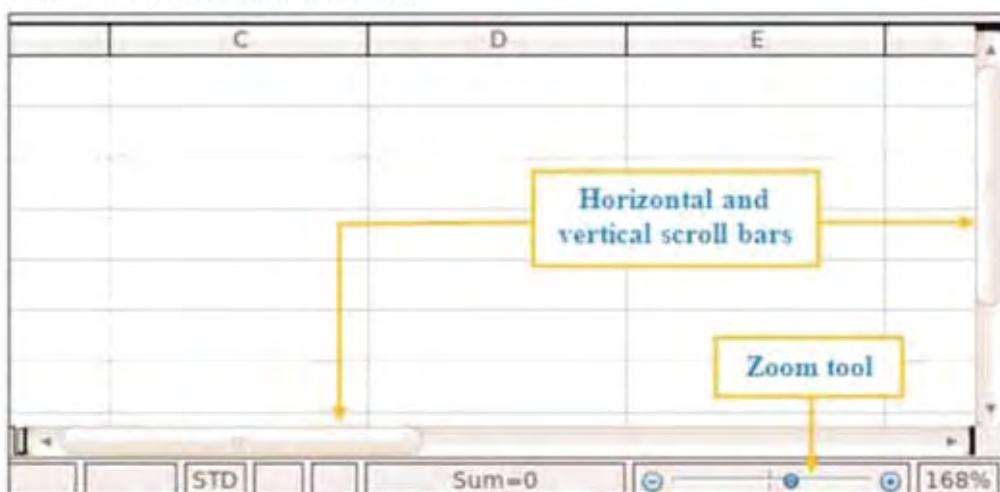


Figure 5.12 : Vertical and horizontal scroll bars with zoom tool

Status Bar

This bar shows present status of the document along with other data, such as the sum of the selected cells, page style, selection mode and unsaved changes. Figure 5.13 represents a typical status bar.



Figure 5.13 : Status Bar

Sheet Tab

Using the sheet tabs you can move between worksheets (also called sheets) of the document. The sheet tab serves as a navigation tool for the worksheets. The current worksheet tab name has the white background and other sheets have the grey background. You may just click on required worksheet in order to select it. You can also change name of any sheet just by right clicking on the text "Sheet1", choosing the "Rename sheet..." option, and typing the required name. Figure 5.14 displays a sheet tab. Alternatively, in specific situations, the arrows shown in figure 5.14 can also be used to move between the worksheets. You can jump to next (right), last (right most), previous (left) and first (left most) worksheet using the arrows.



Figure 5.14 : Sheet tab to select a worksheet

Creating a Calc Document

Let us create a simple document containing a shopping bill. The bill enlists company's name and address, items purchased, units of items purchased, price per unit, and total amount for the items. The bill also required to have grand total of prices, amount of tax, discount (if any), and total amount payable. The steps to generate the document are as mentioned:

Step 1 : Choose Applications → Office → OpenOffice.org Spreadsheet. This will open the Calc. You can see the user interface of Calc.

Step 2 : Observe that the Calc has created three worksheets automatically. However, as stated earlier, you can delete worksheets or add additional worksheets on need by just right clicking it. These three default spreadsheets are named Sheet1, Sheet2 and Sheet3. To give specific name to any of the worksheets, do the following :

- Select the worksheet that is to be renamed by clicking on the worksheet tab (figure 5.14) located just above the Status Bar.
- Click the Format menu, select Sheet from the menu options and select Rename from the submenu options available here as displayed in figure 5.15.
- Give appropriate name (say "bill") and click on OK to complete the operations. Notice the change in the worksheet name.

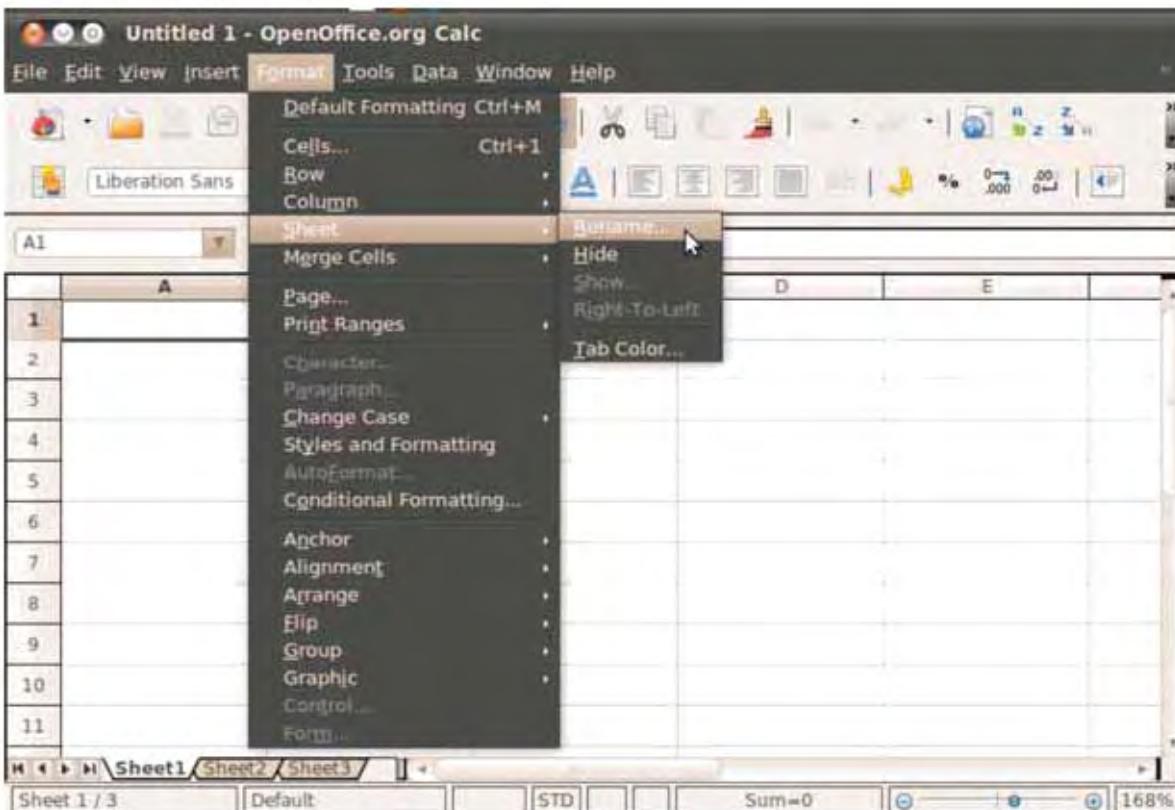


Figure 5.15 : Renaming sheet

Step 3 : Enter data given in table 5.3 into the worksheet named 'bill'.

Sr. No.	Item	Quantity	Unit price
1	Pen	5	10
2	Pencil	6	2
3	Pencil box	1	50
4	Notebook	10	20
5	Notebook cover	10	2

Table 5.3 : Shopping bill data

To enter the above data into the 'bill' worksheet do the following :

- Left-click on the cell **A1**. Enter the word "**Sr. No.**" and press the enter key.
- Left-click on the cell **B1**. Enter the word "**Item**" and press the enter key.
- Left-click on the cell **C1**. Enter the word "**Quantity**" and press the enter key.
- Left-click on the cell **D1**. Enter the word "**Unit price**" and press the enter key.

You may drag the column to resize it. It may be required to resize a column when you are entering name of items. To do so, put the mouse at the edge of the column, which you want to resize, and drag it to the required length. You may see that the first heading line is prepared. You may select the cells and make them bold using formatting toolbar.

Now enter the number "1" in cell A2, the word "Pen" in cell B2, the number "5" in cell C2 and the number "10" in cell D2 respectively. Similarly, enter the remaining lines. The worksheet will now look as shown in figure 5.16.

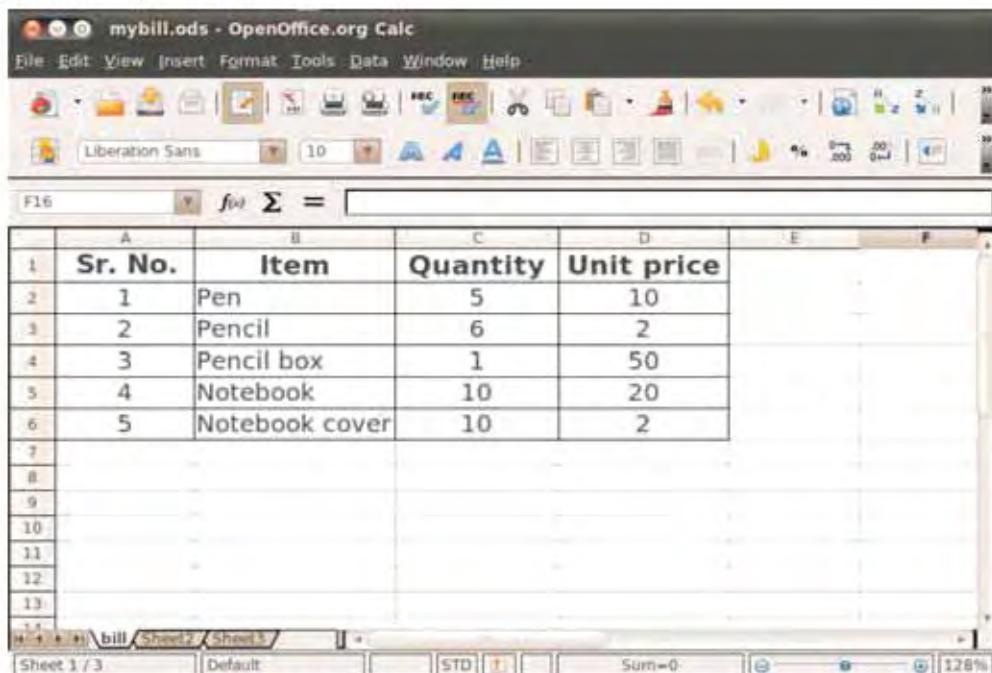


Figure 5.16 : Data entered in a worksheet bill

Step 4 : Save the worksheet with appropriate name so that the entered data is not lost accidentally. Perform the following command to save the worksheet.

- Click on **File → Save**, a dialog box as shown in figure 5.17 will appear; provide filename and path (where you want to store the file).
- Click on the Save button.

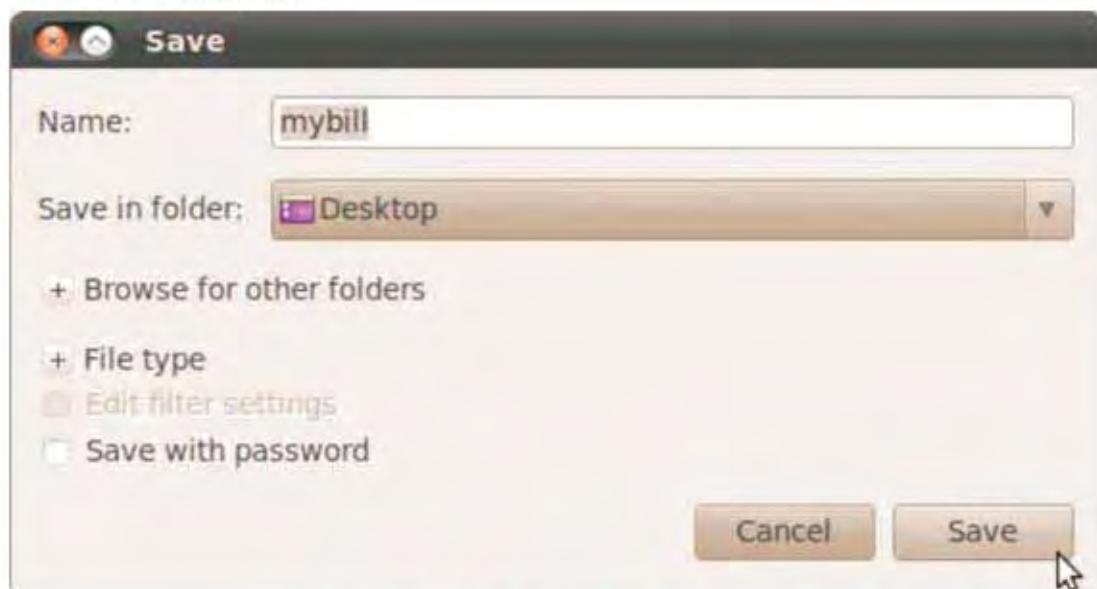


Figure 5.17 : Saving Calc document

The **OpenOffice.org Calc** saves a spreadsheet in a file format with the extension ods. You need to remember the location of the file, so that you can retrieve it in future.

We have just entered the necessary data and saved the file. We have not yet calculated the item-wise bill, and total bill for all the items, taxes and net amount payable. To calculate amount of one particular item (such as Pen), we need to multiply the item price (cost of a single Pen) and quantity (number of Pens) purchased. According to the first row, 5 pens of 10 rupees each were purchased. Total amount for the first item is 50 rupees. Similarly amount payable for other items are as follows :

1 st row → 5 pens 10 ₹ each	= 5*10 ₹	= 50 ₹
2 nd row → 6 pencils 2 ₹ each	= 6*2 ₹	= 12 ₹
3 rd row → 1 pencil box 50 ₹ each	= 1*50 ₹	= 50 ₹
4 th row → 10 notebooks 20 ₹ each	= 10*20 ₹	= 200 ₹
5 th row → 10 notebook covers 2 ₹ each	= 10*2 ₹	= 20 ₹

According to the above calculation, we can deduce a general rule that if x units of an item are purchased, each having unit price y , the amount payable is $x*y$. To insert this result in last column entitled as "**Amount**", we may manually multiply figures and write it in a cell. However, the better alternative is that we may enter a generic formula.

To enter a formula in cell we may provide direct data (such as $=5*10$) or we may provide cell reference where these data are available. That is if 5 is available at cell C2 and 10 is available at D2, in this case value of ($=5*10$) is equal to ($=C2*D2$). Typing '=' is necessary in beginning of any formula, failing to do so, the content is identified as regular text in a cell (even if a number is entered) and does not perform calculations.

Using cell reference instead of the direct value has some advantages. The first advantage is that we need not have to manually perform the arithmetic operation; which may lead to mistake. Second advantage is that when we change values in the cell, result of the formula will also automatically change. Third advantage is, when the data block is moved, or some rows or columns are inserted/deleted, the reference in formula will automatically change.

Let us add one column in our previous worksheet 'bill' to calculate item-wise amount to be paid. Do the following :

- Select cell E1 and enter the word "**Amount**". Press enter key. You may make the word bold.
- Select Add formula in cell E2 by typing =C2*D2 as displayed in figure 5.18.

The screenshot shows a spreadsheet application window titled "mybill.ods - OpenOffice.org Calc". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar below the menu bar includes various icons for file operations, cell selection, and data processing. The font toolbar shows "Liberation Sans" and a font size of 10. The formula bar at the top displays "SUM" and the formula "=c2*d2". The main spreadsheet area contains the following data:

	A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount	
2	1	Pen	5	10	=c2*d2	
3	2	Pencil	6	2		
4	3	Pencil box	1	50		
5	4	Notebook	10	20		
6	5	Notebook cover	10	2		
7						
8						
9						
10						
11						
12						
13						

The status bar at the bottom shows "Sheet 1 / 3", "Default", "INSRT", "STD", "Sum=0", and "128%".

Figure 5.18 : Adding simple formula

- After entering formula in cell E2, when you press the enter key, you will see the result of multiplication in the cell.
- You may add formulas in cells respectively in cells E3 ($=C3*D3$), E4 ($=C4*D4$), E5 ($=C5*D5$), and E6 ($=C6*D6$).
- Another alternative is just drag the content of E2 to the remaining cells of the columns upto E6 cell. Automatically the formula will be copied. To drag the content of the cell, click on the cell. Drag the corner of the cell to the required number of cells.
- You may see the output as displayed in figure 5.19.

The screenshot shows a spreadsheet application window titled "mybill.ods - OpenOffice.org Calc". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains various icons for file operations, cell styling, and data processing. The formula bar displays the range E2:E6 and the formula $\Sigma = =C2*D2$. The main table has columns labeled "Sr. No.", "Item", "Quantity", "Unit price", and "Amount". The data rows are as follows:

Sr. No.	Item	Quantity	Unit price	Amount
1	Pen	5	10	50
2	Pencil	6	2	12
3	Pencil box	1	50	50
4	Notebook	10	20	200
5	Notebook cover	10	2	20

The "Amount" column for row 5 is currently selected. The status bar at the bottom shows "Sheet 1 / 3", "Default", "STD", "Sum=332", and "128%".

Figure 5.19 : Amount calculated for one cell and dragged to other cells

We also need to make grand total of all the payable amounts for different items. To make total of the amount column, SUM function is used. The SUM function button is located in the Function Bar at the top region of the screen. Using the SUM function button, you can automatically add the numbers in the cell range you select. Figure 5.20 shows the sum function.

mybill.ods - OpenOffice.org Calc

File Edit View Insert Format Tools Data Window Help

Liberation Sans 10 A A A | % 0,- .00 0,-

SUM =sum(E2:E6)

A	B	C	D	E	F
1	Sr. No.	Item	Quantity	Unit price	Amount
2	1	Pen	5	10	50
3	2	Pencil	6	2	12
4	3	Pencil box	1	50	50
5	4	Notebook	10	20	200
6	5	Notebook cover	10	2	20
7					
8					
9					
10					
11					
12					
13					

=sum(E2:E6) SUM(number 1 > number 2, ...)

bill Sheet2 Sheet3

Sheet 1 / 3 Default INSRT STD Sum=0 128%

Figure 5.20 : Sum function

Sometimes the symbol available in the top formula bar (denoted as ' Σ ') as displayed in figure 5.21 can be used.

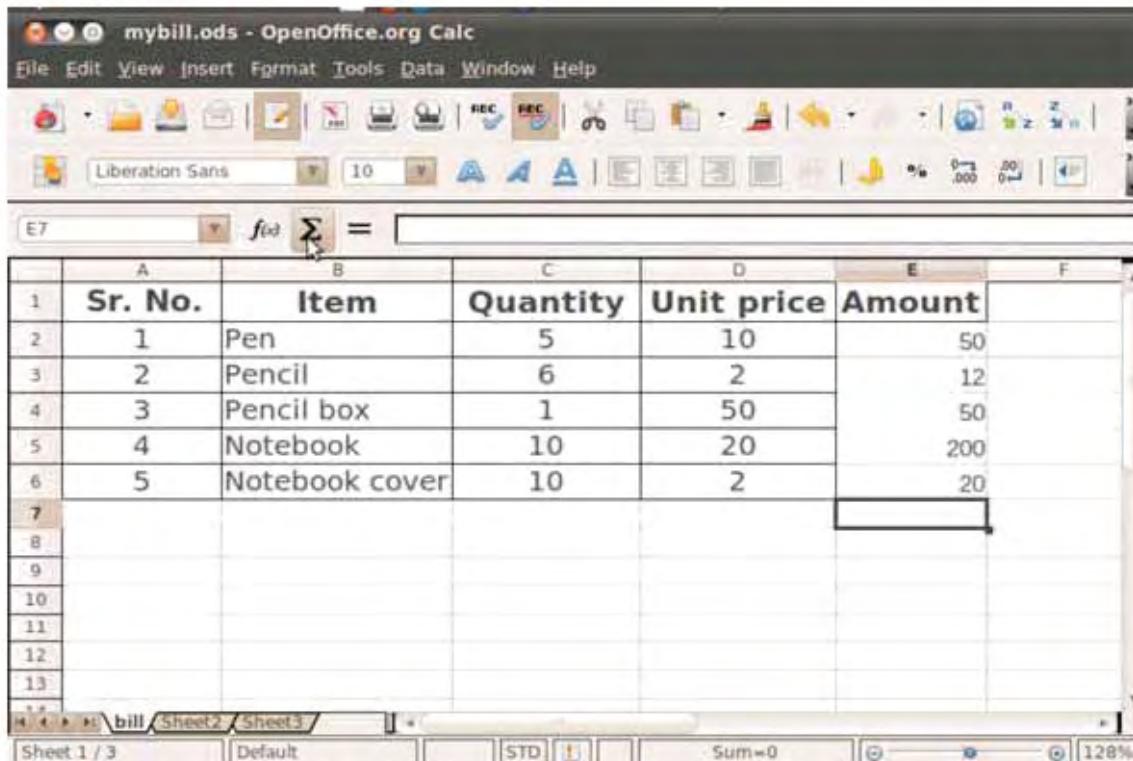


Figure 5.21 : Using autosum symbol

In spreadsheets, a cell range is denoted by the address of the first cell and the address of the last cell, separated with a colon; e. g. E2:E6 refers to cells from E2 to E6. Alternatively you can also select the cell E7 and type formula =E2+E3+E4+E5+E6. Select cell D7 and write caption "Total" and press enter. You will see screen as shown in figure 5.22.

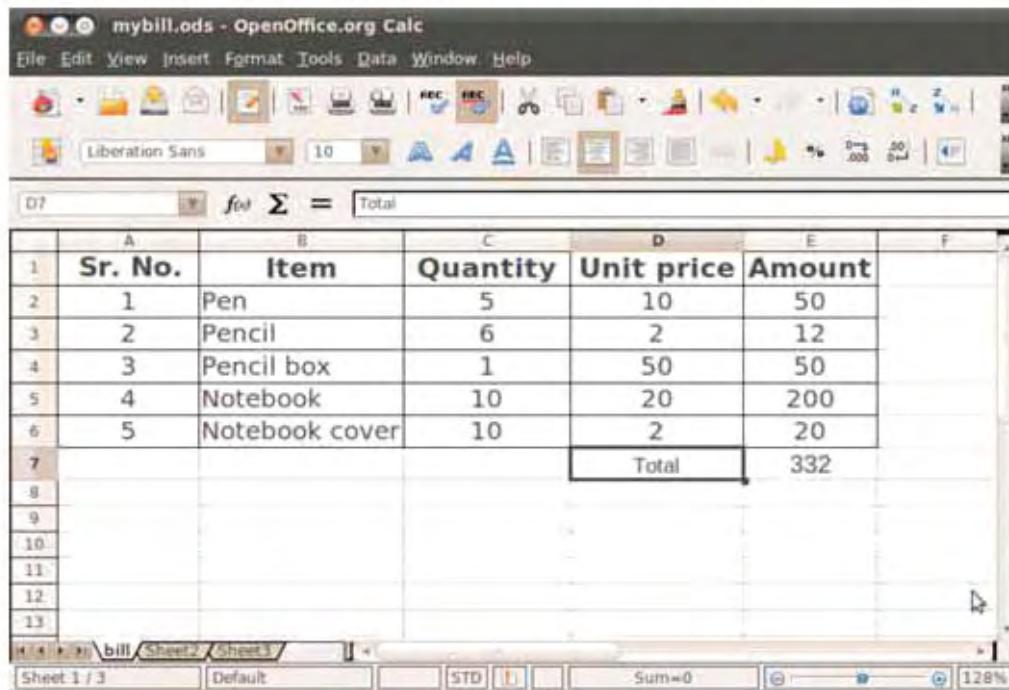


Figure 5.22 : Calculation of total amount

Let us now calculate the tax to be paid on the total amount calculated above, we need to calculate taxes. If we consider the tax is 5% of the total bill amount, then the formula to get tax amount is as follows.

$$\text{Taxes} = \text{Total amount} * 5/100$$

- Select cell D8 and write caption as "Taxes".
- Select cell E8 and write the formula =E7*5/100

To calculate the net amount to be paid, we need to add taxes to the total amount. That is the content of the cell E7 and E8 is to be added and displayed in the cell E9. You may use sum function as discussed above or may write direct formula (=E7+E8) by selecting cell E9.

Select the cell D9 and write caption "Net Amount" and press enter. You can see the result as displayed in figure 5.23.

Sr. No.	Item	Quantity	Unit price	Amount
1	Pen	5	10	50
2	Pencil	6	2	12
3	Pencil box	1	50	50
4	Notebook	10	20	200
5	Notebook cover	10	2	20
			Total	332
			Taxes	16.6
			Net Amount	348.6

Figure 5.23 : Calculating net amount

You can see that the net amount calculated here 348.6. To make it more familiar, we may add a decimal point by clicking on toolbar as shown in the figure 5.23. Adding one decimal point will make the Net Amount value as 348.60.

You may want to move the whole content in such a way that you can insert name of the company and date. Perform the following steps.

- Using mouse, select all the cells containing data. In the above example, the cells that contain data are A1 to E9.
- Cut the selected data by selecting edit menu and choosing "Cut" operation.
- Go to cell B3 and paste the data.
- Ensure that your data is not changed.

You might have noticed the change in formulas. Total, Taxes and Net Amount now refer to F column instead of E column. Calc automatically changes the column references if the cells are moved from one location to another location unless specifically told not to do so. In next chapter we will learn more about such relative or absolute (fixed) addressing techniques.

Saving and Re-opening File

While working with a spreadsheet (or any other computer application), frequently you need to save your work. There might be a power fluctuation, or any other such problem; because of which you may lose the data you have entered. Saving the file frequently is a good practice and prevents accidental loss of data.

Once operations of files are finished, you need to close the file. You may click on the cross button on the top of the window or select File menu and choose Close operation. To reopen the file again, you need to select File menu and choose Open operation. You may take help of standard tool.

Meet the Developers of the OpenOffice Suit via Calc

Now you know how the spreadsheet in OpenOffice works. Do the following. Open a workbook in Calc. Select any cell and type =starcalcteam() in any of the cell, which will display the picture of OpenOffice development team members. Figure 5.24 displays the Calc development team.

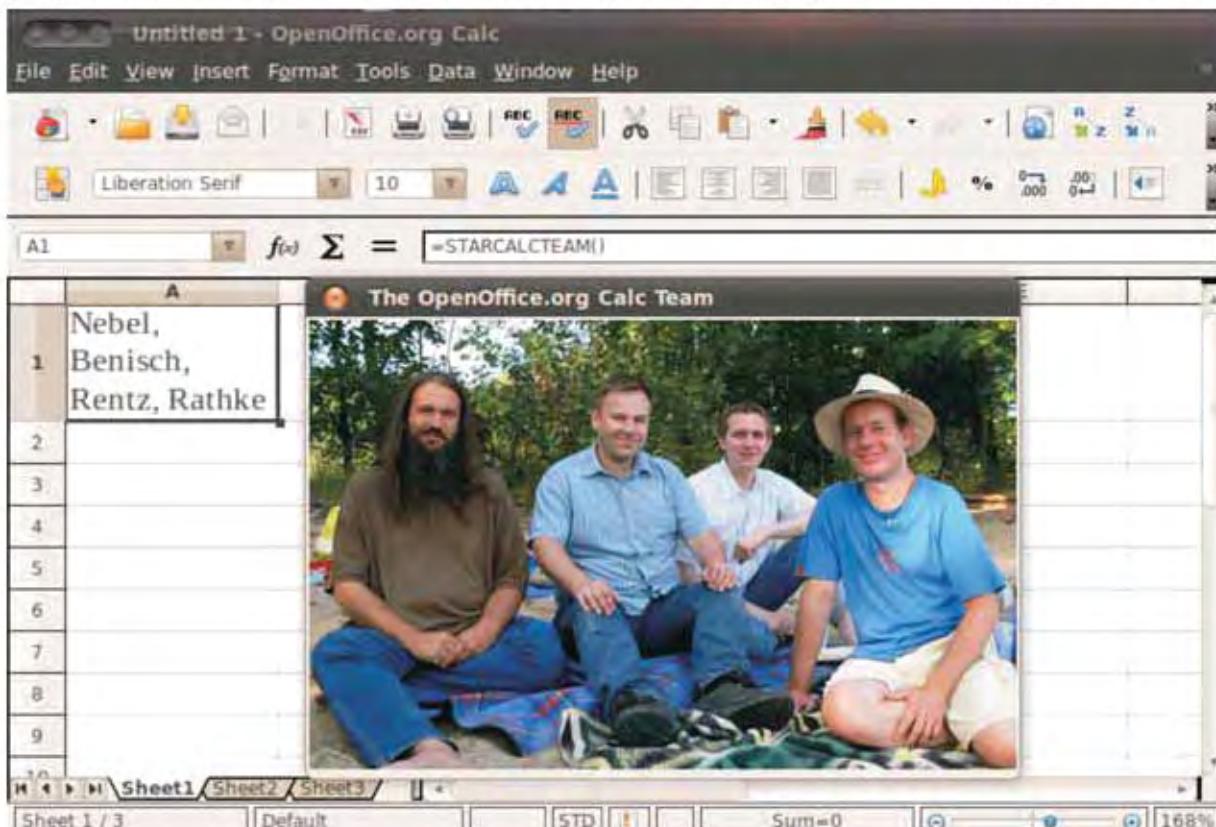


Figure 5.24 : Calc developer's team

Summary

In this chapter we have learnt what is spreadsheet and typical applications that can be developed using such packages. The chapter has focused on introduction and basic functionalities of a spreadsheet package from OpenOffice suit. The features and functionalities discussed have been illustrated with help of a real life example in step by step manner. With the concepts learnt in this lesson you may create a new spreadsheet, edit data in it, analyse and process the data with some basic formulas, save it and reopen it whenever required.

EXERCISE

1. List suitable applications of spreadsheet packages.
2. Explain file saving and reopening in Calc.
3. Explain working of formulas in Calc worksheet.
4. Can we rename an existing sheet of Calc ? How ?
5. What would you do if a standard toolbar is not visible in a Calc document ?
6. Choose the correct option from the following :
 - (1) Which of the following type of package does Calc refer to ?

(a) Spreadsheet	(b) Multi-sheet
(c) Double sheet	(d) Cannot be determined
 - (2) Which of the following applications are not suitable for Calc ?

(a) Balance sheet preparation	(b) Result analysis
(c) Presenting an idea about a product	(d) All of these
 - (3) Which of the following is the extension of a worksheet created in Calc ?

(a) .ods	(b) .odd	(c) .xls	(d) .obj
----------	----------	----------	----------
 - (4) Which of the following will be inserted in a worksheet if =starcalcteam() is inserted in a Calc cell ?

(a) Stars	(b) Photograph of Calc developer team
(c) Calc licence information	(d) Calc version information
 - (5) How can one calculate total of values entered in a worksheet in a Calc document ?

(a) By manual entry	(b) By autosum
(c) By formula	(d) All of these
 - (6) If we move a cell containing a formula having reference to another cell in the worksheet what will happen to the cell numbers used in formula ?

(a) The cell row and columns are changed at destination	(b) The cell row numbers are changed at destination
(c) The cell column numbers are changed at destination	(d) Nothing will be changed

LABORATORY EXERCISE

1. Implement the example of the shopping bill discussed in this chapter.
2. Study any bill you get from a super store and implement it in Calc.
3. Generate a Calc document having your marks of six different subjects. Make total of the marks, find out average and percentages from the data.
4. Extend the third example given in this exercise by adding marks of your friends.



Data Editing and Formatting in Calc

As discussed in the earlier chapter, basic unit of data storage in a spreadsheet is a cell. A spreadsheet is made up of rows and columns intersecting each other forming multiple cells. All the data, formulas and functions are to be written within these cells. This chapter gives you brief outline about the basic operations regarding data editing and formatting.

Basic Worksheet Operations in Calc

Operations such as opening a new or an existing spreadsheet, saving the spreadsheet, renaming the whole spreadsheet with save as option; renaming work sheets, adding/deleting worksheets etc. are frequently used operations on spreadsheet or work sheet level. This section discusses the commands for the same.

Creating a New Document

When you open the OpenOffice.org Calc application, it will automatically open a new (empty) Calc document. If you have already used the document and wrote some data in it, you may open another new spreadsheet document. To open a new Calc document, that is a Calc spreadsheet, perform the following procedures.

- Select **File → New → Spreadsheet**; from the Menu Bar.
- Press **CTRL + N**; or
- Click on the **New Document Icon** on the toolbar at the top of screen and select Spreadsheet. The icon (encircled) will look as shown in figure 6.1.

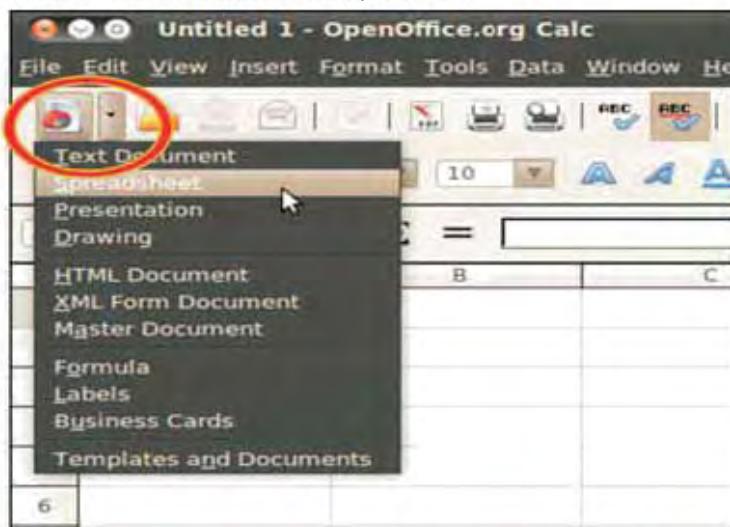


Figure 6.1 : An icon for opening new spreadsheet

When a new spreadsheet is opened, it will give you three separate worksheets; just like you have purchased a notebook for mathematics with only three pages having grid of rows and columns. In Chapter 5 you have learnt how to open a spreadsheet with three worksheets, for entering data of a shopping bill.

An existing Calc document can be opened in multiple ways. The first way is to open the Calc first and go to file menu and select open. See the following command sequence.

- Select **File → Open**;
- In the **Open** dialog box, select the required file from a default folder or any other specified folder;
- Complete the action by clicking on the **Open** button.

Another alternative is first to find the required file from computer and then open it within the Calc software as follows.

- Open the folder **Documents** by choosing **Places → Documents**;
- Select the required spreadsheet file and open it.

Still if you cannot find your file by performing the steps mentioned above; you can find the file through search utility of Ubuntu. This utility is helpful, especially when you do not remember the location of the required file. To do so, perform following actions.

- Select **Places → Search for Files** as shown in the vertical menu in figure 6.2;
- When you click **Search for Files** (see mouse arrow in figure 6.2), a dialog box appears asking for file details. The details include name of file, folder in which the file is to be searched and some content within the expected file. We can also search file by specifying some dates like last modified date. See block arrow pointing to the dialog box in the figure 6.2;
- In the **Name contains:** field of the **Search for Files** dialog box, enter a full or a part of the file name;
- Click on the Find button.

These steps are illustrated in figure 6.2. It will show a list of files as a result of the search operation. To open a file, double click on the required filename in the list using mouse.

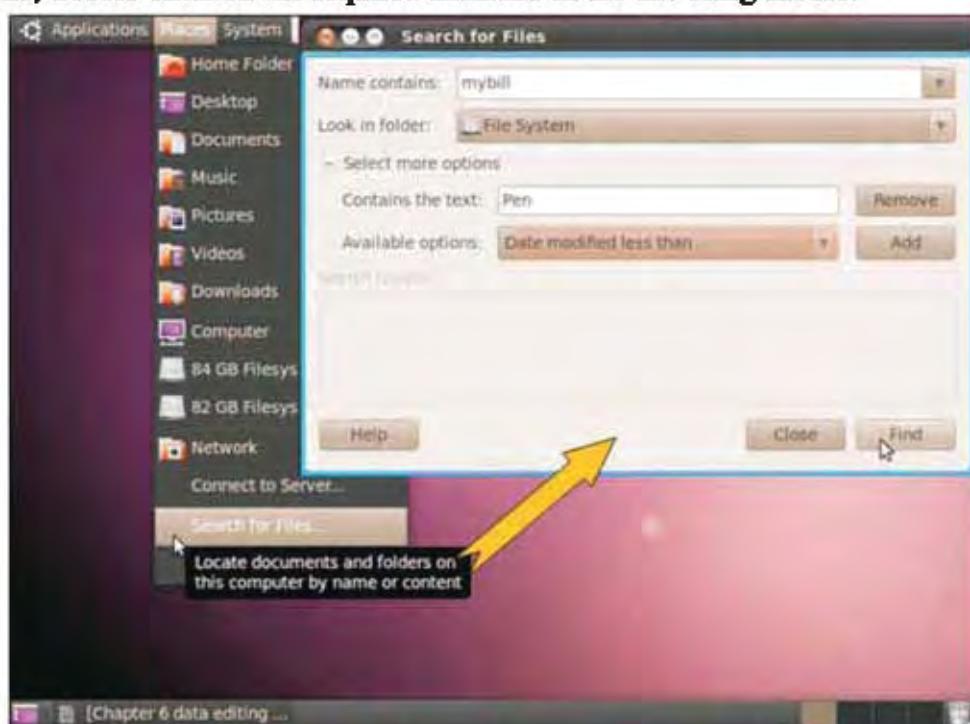


Figure 6.2 : Result of search operation to find a file

Saving and Closing Spreadsheet

You have already opened a new or an existing spreadsheet document for viewing and editing in the previous section. Once you finish working with the opened spreadsheet, you need to save the changes. It is advisable to frequently save the changes to prevent any loss of data.

We have already discussed how to save a worksheet in Chapter 5. To save spreadsheet choose, **File → Save**. You can also choose Save as option; if you want to save the document with different name or different type. As stated earlier, if you are saving the spreadsheet first time, you will be prompted to give file name and path.

To close the current file, choose **File → Close**. If you have not saved the latest changes, the Calc will give you an opportunity to save the file before closing it.

Saving Worksheet in Different Format

By default, the Calc saves the spreadsheet in file format with extension ods. Some of the other formats are listed in table 6.1.

Format	Extension	Description
ODF Spreadsheet	ods	OpenOffice.org Calc format.
ODF Spreadsheet Template	ots	Calc spreadsheet template format.
dBASE	dbf	Database file format.
Text CSV	csv	Text file containing comma-separated values; such files are typically used for data exchange among various programmes.
HTML Document (OpenOffice.org Calc)	html	Web page format.
Portable Document Format	pdf	Most frequently used format; this is a universal Adobe format of a portable document.
Microsoft Excel 2007 XML	xlsx	Spreadsheets of MS Office 2007/2010.
Microsoft Excel 2003	xls	Spreadsheets of MS Office 2003.

Table 6.1 : File formats in which a worksheet can be saved

To save a spreadsheet in other file format :

- Go to File and choose Save (for unsaved document) or the Save As option (for opened and already saved document);
- Open the file type menu (encircled) as shown in figure 6.3;

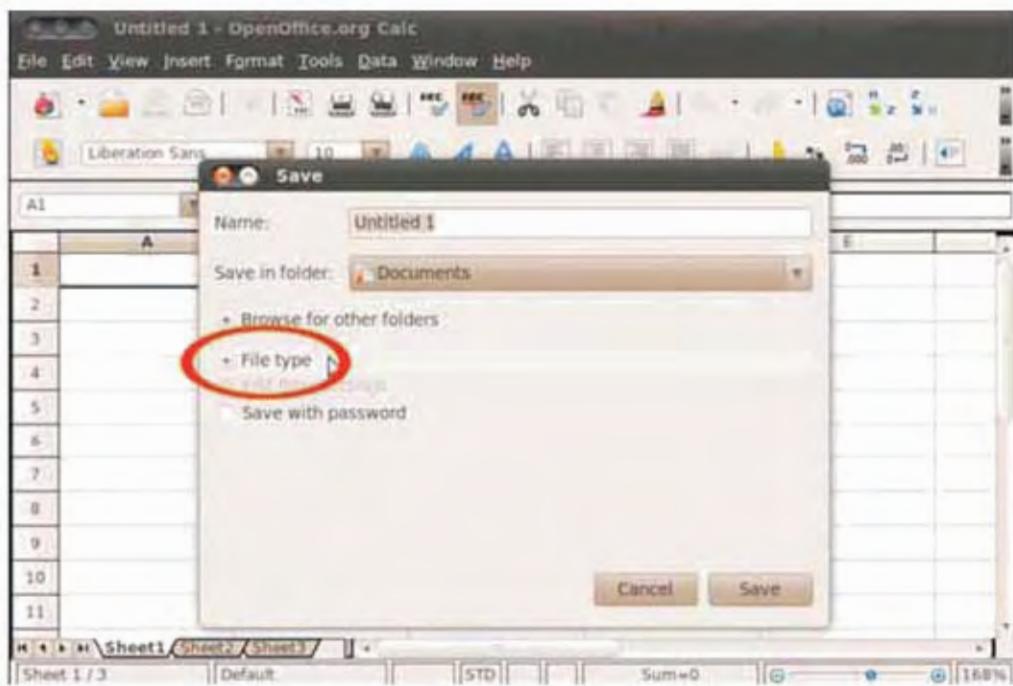


Figure 6.3 : Opening file type menu

- Choose required file format as shown in figure 6.4.

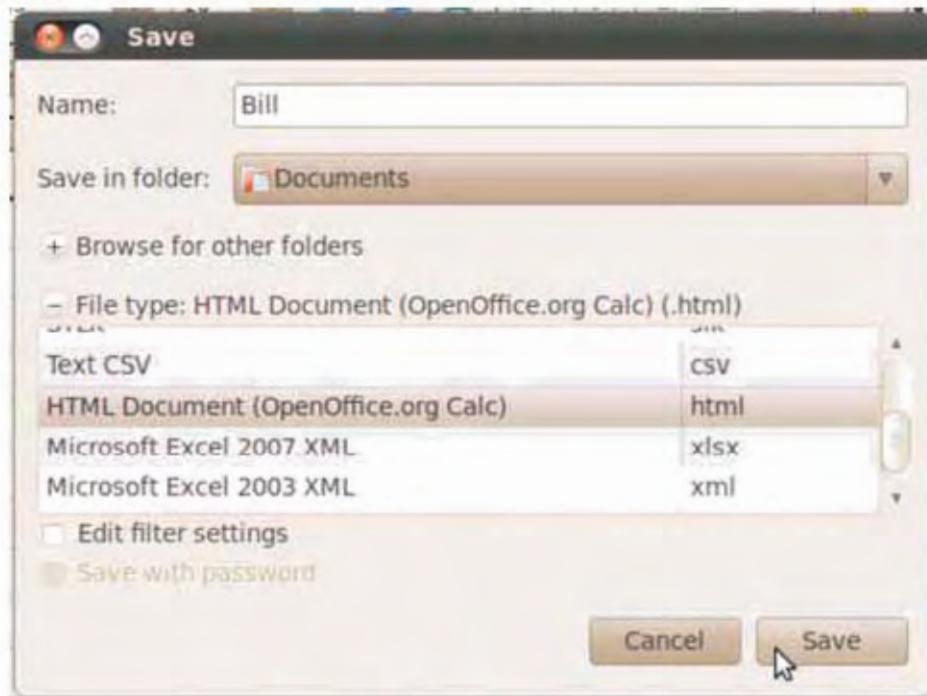


Figure 6.4 : Choosing file type

- Save the changes by clicking on the **Save** button.

Renaming and Re-colouring Worksheet

In the previous chapter we have changed the name of a worksheet by selecting a worksheet from status bar (right click) and changing its name. If you have many worksheets in a spreadsheet document, better to assign a suitable name (and colour) for easy management of the document. To change the name of a worksheet, perform following actions.

- Select any cell in an opened worksheet.
- Select **Format → Sheet → Rename;**
- In the dialog box **Rename Sheet**, enter a new name;
- Save changes by clicking on the **OK** button.

Alternatively, click the right mouse button on the name of a worksheet shown on the sheet tab and select **Rename sheet** option as follows. You have already learnt this option. The method is given below :

- Go to appropriate sheet, which you would like to rename. The selected sheet must be highlighted (with white background);
- Right click on the sheet. A vertical menu appears as displayed in figure 6.5. From the menu, select **Rename Sheet** option; A new dialog box appears as shown in form of rectangle in middle of the figure 6.5;
- Provide new name of the sheet.

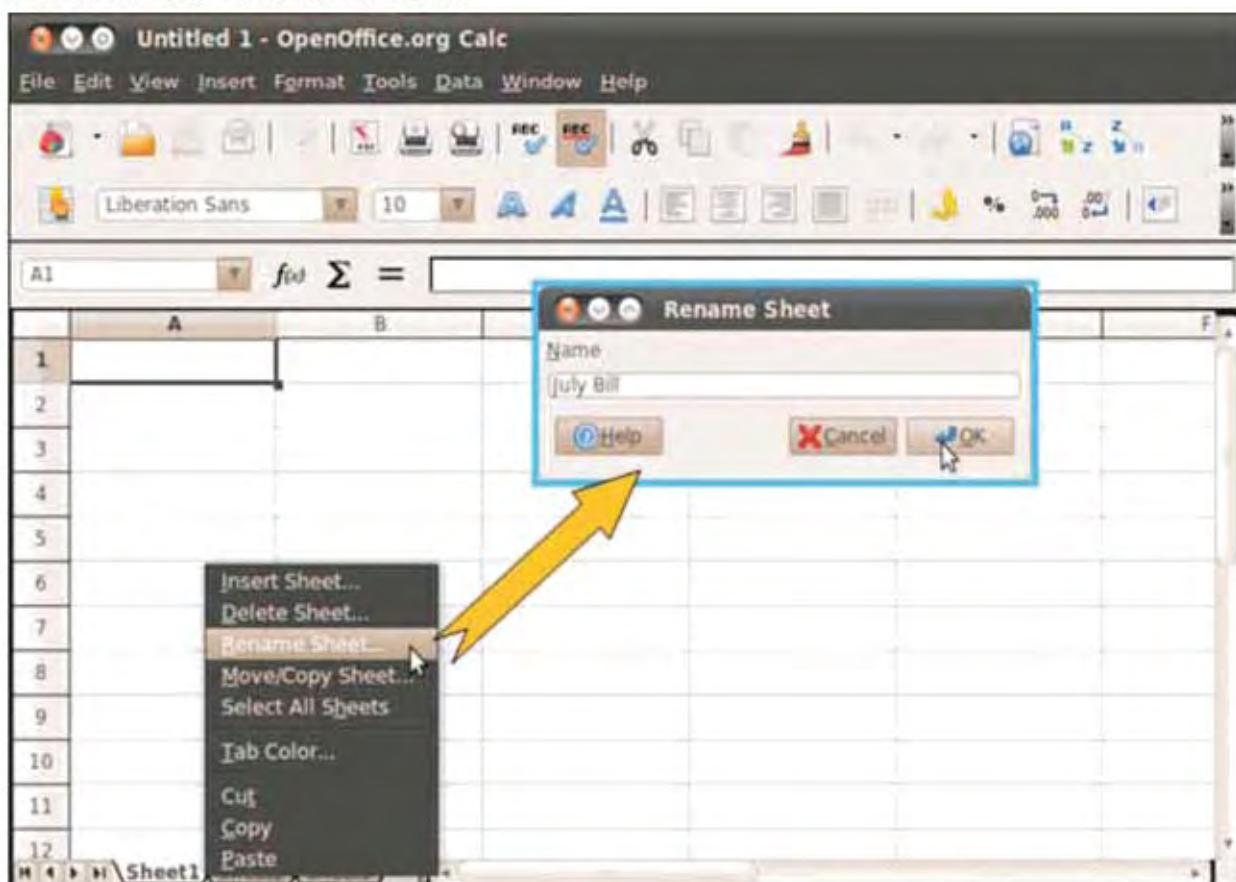


Figure 6.5 : Renaming worksheet with sheet tab

Changing Colour of the Worksheet Tab

To change the tab colour of a worksheet, we just need to right click the worksheet name and choose the command **Tab Color** from a vertical menu appears as shown in figure 6.6. We need to select an appropriate colour for the tab from the colour choices presented to us.

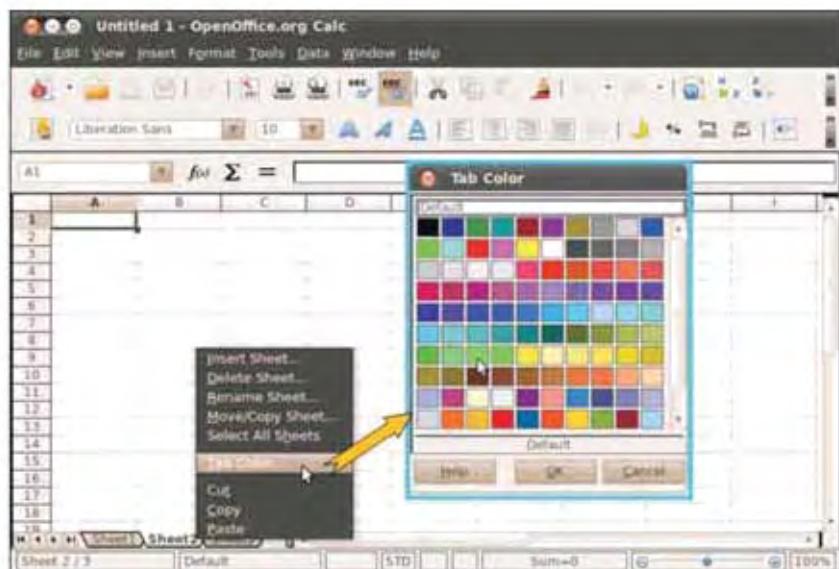


Figure 6.6 : Changing colour of a worksheet tab

Inserting and Deleting Worksheets

As you know, many times three worksheets provided by the Calc may not be sufficient to enter all the data related to the application. In such cases we may need to insert additional worksheets. For example, there are five groups of students in an activity, say sports. For each of the group, an attendance report is to be made. In this case, it is better to have five worksheets; each having an attendance report of an individual sport. Similarly, rows, columns can be added to a formatted spreadsheet, width can be changed and merging cells can be executed.

To insert a new worksheet do the following actions.

- Select any cell in a worksheet before or after which a new worksheet is to be added;
- Select **Insert → Sheet**. An Insert Sheet dialog box will appear;
- In the Insert Sheet dialog box, specify additional options.
- At the end, click **OK** button to confirm action.

Figure 6.7 illustrates above actions for inserting an additional worksheet.

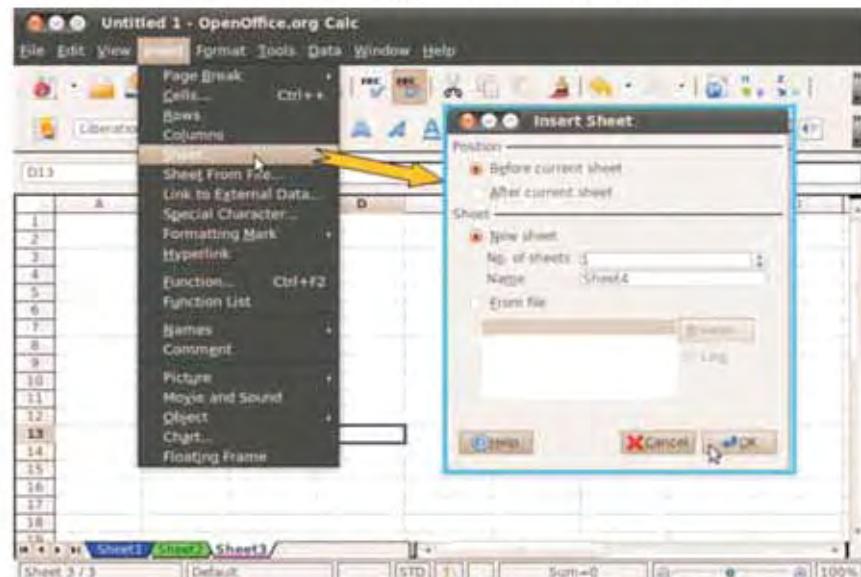


Figure 6.7 : Adding new worksheet in a document

You might have also thought the possibility of inserting a worksheet through the worksheet tab. To insert a worksheet using the sheet tab, perform the steps mentioned.

- Select a worksheet and right click on it;
- A vertical menu will open;
- Select **Insert Sheet**;
- You will see a new Insert Sheet dialog box;
- Select required options and click OK.

Deleting Worksheet

You can delete one or more worksheets in Calc. To delete a single worksheet, right click on the sheet tab of the worksheet you want to delete. It will give you a list of possible actions. Select **Delete Sheet** from these choices. Alternatively you can select **Edit → Sheet → Delete**.

Row and Column Level Operations

Sometimes we need to add some columns or rows in a worksheet. Though the worksheet provides many rows and columns, it may be possible that we would like to add some blank rows or columns in the middle of the data already entered. This section discusses how to add columns and rows in a worksheet along with other operations at row or column level. To insert a row do the following:

- Select a cell in the row above which a new row is to be added;
- Select **Insert → Rows**.

To insert a column do the following:

- Select a cell in the column besides which a new column is to be added;
- Select **Insert → Column**.

Width of a Column and Height of a Row

Calc presents all rows and columns with same height and width. When you write some text in a cell, often the cell content becomes invisible due to the adjacent cell content. Also at times result of some formula may not fit into the cell. To change the width of a column or height of a row, do the following.

- Click on the dividing line between columns (rows) headers;
- Without releasing the left mouse button, drag it in the direction needed.

To provide specific size to a column or a row, do following :

- Select a cell from a column (row);
- Select **Format → Column → Width** or **Format → Row → Height**;
- In the dialog box that appears, set the width of the column or height of the row in inches;
- Click on **OK** button.

If you are not comfortable with inches, go to **Tools → Options → General**. Modify the options provided by the category as per your need as illustrated in figure 6.8.

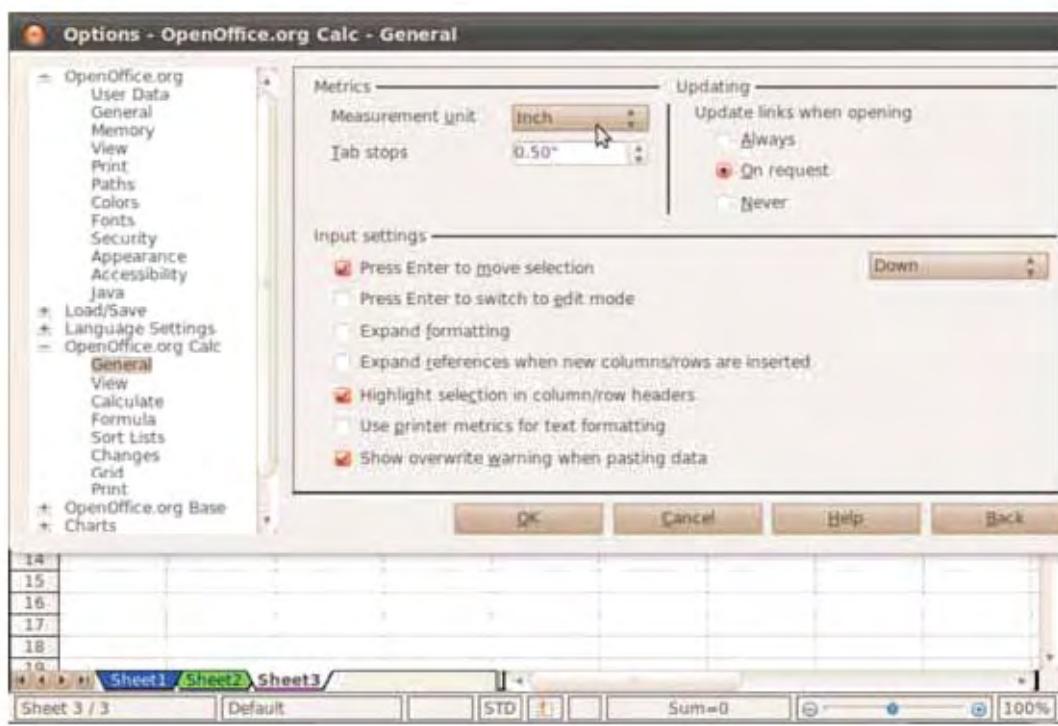


Figure 6.8 : General options of Calc

Deleting Rows and Columns

To delete a row, perform following steps:

- Go to a specific row which you want to delete;
- Right click on the row;
- Select **Delete Rows**.

In a similar way you can delete a column also. Figure 6.9 demonstrates delete operations for rows and columns.

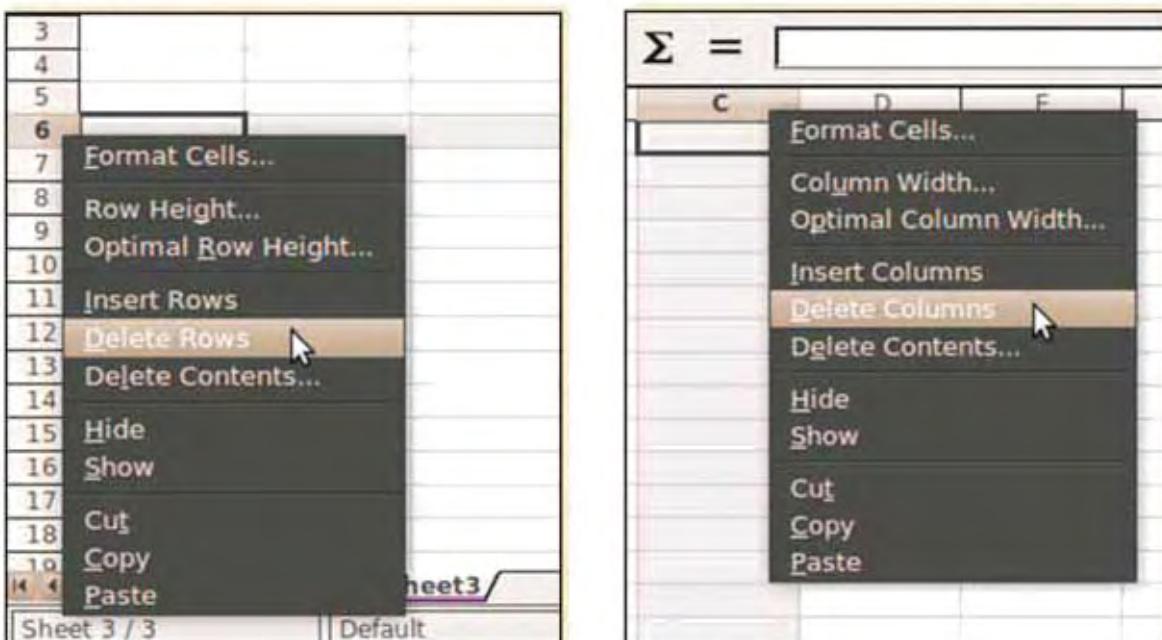


Figure 6.9 : Deleting rows and columns

Alternatively, to delete a row or a column, you may perform following steps:

- Select a column or a row;
- Select **Edit → Delete Cells**.

This is illustrated in figure 6.10. It is possible to delete entire column or row and shift cells as per our need using this option.

Selecting Rows and Columns

To select an entire column, click on the column header; you may see that the column is highlighted. Similarly, to select an entire row, click on the row header; and the row will be highlighted.

Freezing a Pane

When many rows and columns are accommodated in a single worksheet, the row or column headers do not remain visible; which makes handling the data difficult. To make headers of rows and columns always visible, do the following:

- Select a cell below the row containing headers. In case of column, select the right of the column that must always be visible;
- Select **Window → Freeze**.

To unfreeze cells, deselect the Freeze option by selecting **Window → Freeze** again.

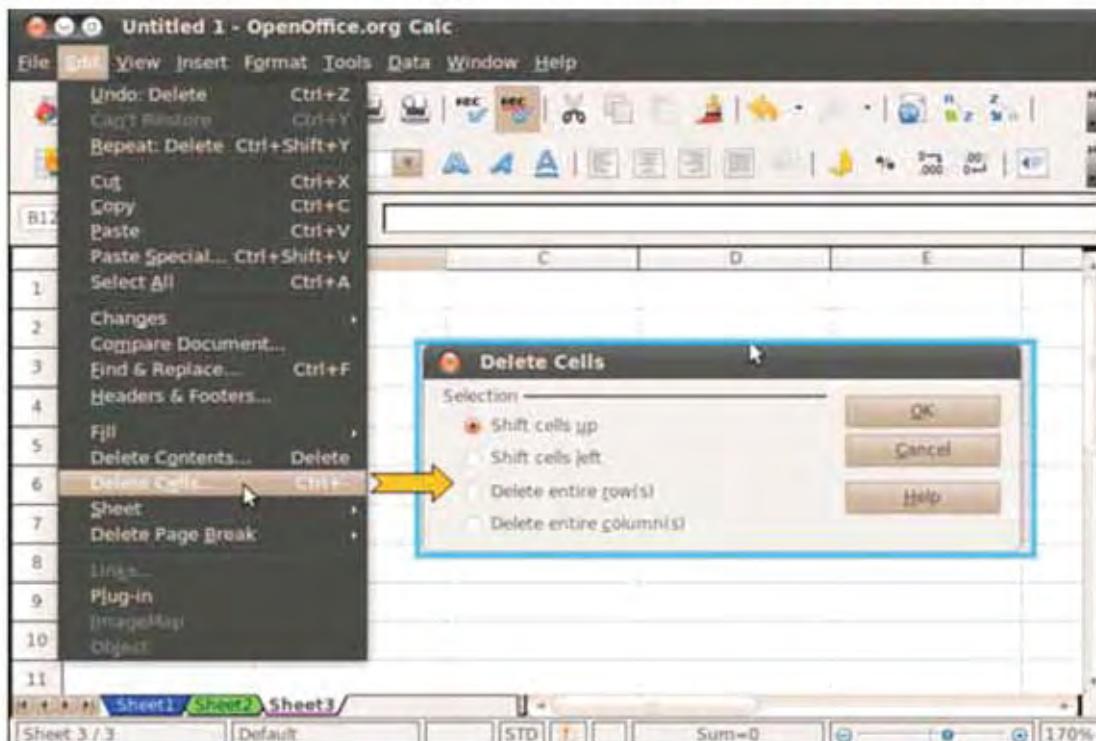


Figure 6.10 : Deleting cells

Cell Level Operations

This section gives you an idea that how content can be written in a cell. The content may be text, number, functions, formulas and reference to other cell. You will also learn to copy and move content of a cell/multiple cells.

Selecting Cells

As stated above, text, numbers, formulas and references to other cells can be entered in cells. But before we enter anything into the cell, we need to select the cell, otherwise the content may not be written in desired cell. To select the cell, you need to just left click on it. You may enter your content in the cell.

Selecting a Range of Cells

To select a cell range, go to the first cell using mouse and left click on it. After selecting the first cell with left mouse click, without releasing the mouse button, drag the mouse pointer over the cells to be included, until the last cell of the desired range is reached. Once the cell in this range is highlighted properly, then release the mouse button.

This is applicable to adjacent cells only. If cells are not physically adjacent to each other, you need to hold down the **CTRL** key. To select all the cells in a worksheet, you need to just click the button in the beginning of the row and column headings. The selection range can also be specified by giving row and column reference such as A1:B12 in the address box.

To cancel the selection of cell or a range of cells, all you have to do is a left click on any cell using mouse.

To Delete the Content of a Cell

To delete the content of a cell or cell range, perform following steps.

- Select a cell or a range of cells;
- Press the **Delete** key on the keyboard, a dialog box will appear;
- Give your choice by clicking. If you want to delete text, then click on **Text**. You may select more than one choice.
- Confirm your choice by pressing **OK** button in the Delete Contents dialog box as shown in figure 6.11.

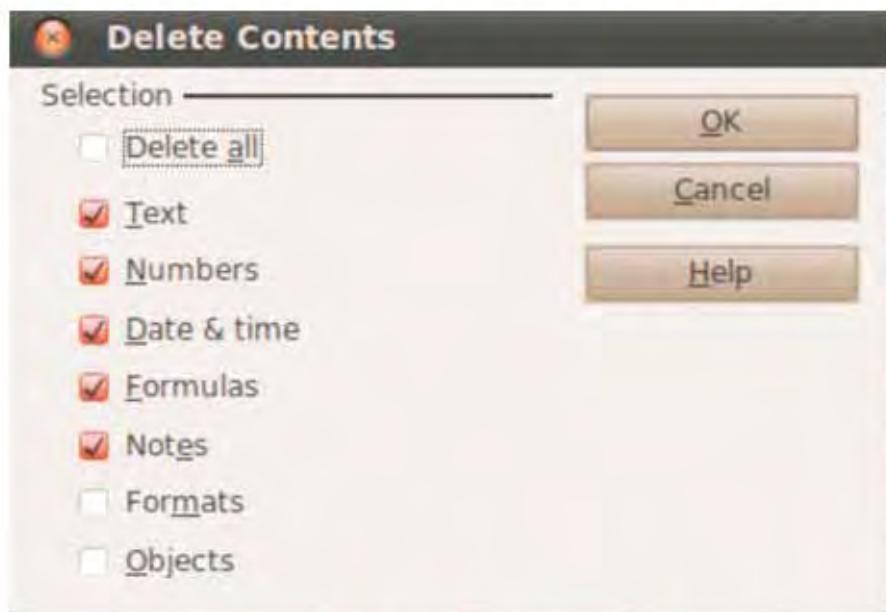


Figure 6.11 : Delete options for cell

The dialog box presents options about deleting rows and columns. Table 6.2 lists the options available in the **Delete Contents** dialog box.

Choice	Description
Delete all	Deletes the entire content of a cell/cell range.
Text	Deletes text from the cell/cell range selected.
Numbers	Deletes numbers from the cell/cell range selected.
Date & time	Deletes only date and time from the cell/cell range selected.
Formulas	Deletes formulas and outcomes from the cell/cell range selected.
Notes	Deletes notes for a cell if added.
Formats	Deletes cell formatting but keeps data.
Objects	Deletes additional elements of a cell such as images.

Table 6.2 : Options available in the Delete Contents dialog box

Editing the Cell Content

To edit the content of a cell, just select the cell, and double click with mouse and edit content. When you finish, press enter key.

Copying and Moving Cells

Cells in spreadsheets can contain many things such as numbers, text, formulas, format, and references to other cell. The content of a cell can therefore be complex, consisting of many values. To copy a cell, a worksheet or a range of cells, do the following.

- Select the cell/part of cell/ range of cells/worksheet that you want to copy;
- Select **Edit → Copy**. You may select the cells and right click on it; from which you can select **Copy** option. The copied data will be highlighted with dotted rectangle;
- Select the paste location and perform the **Paste** command.

For additional paste options, you select **Edit → Paste Special** as shown in figure 6.12.

The Paste Special dialog box provides various parameters for pasting contents. These choices are listed in table 6.3.

Label	Description
1	Pastes only text.
2	Pastes only numbers.
3	Pastes date and time.
4	Pastes only formulas.
5	Pastes notes only.
6	Pastes formatting of a cell or value.
7	Pastes objects only.

Table 6.3 : Paste Special choices

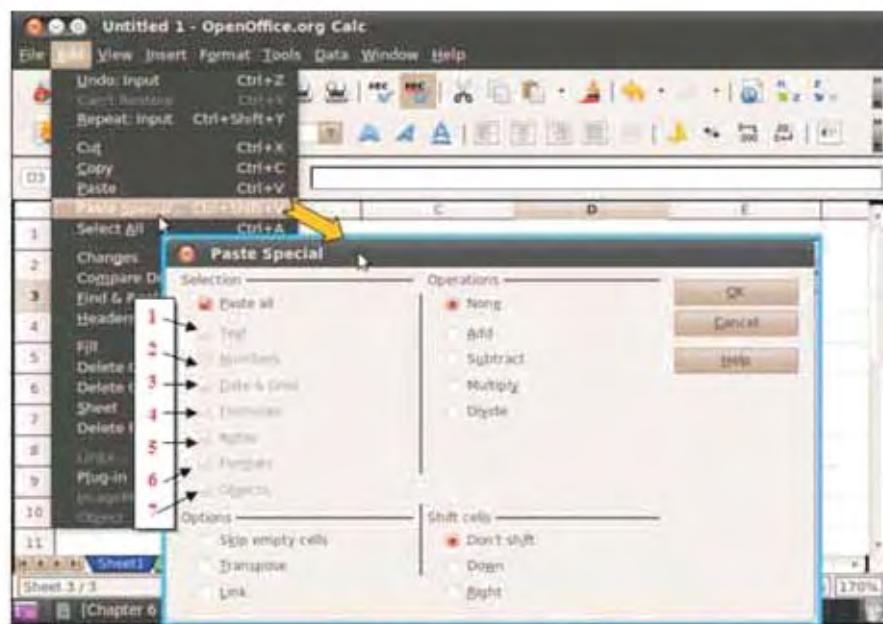


Figure 6.12 : Paste Special options

With the paste special operation you can control the way how content is pasted on destination cells.

Autofill Tool

To automatically fill data into the cells, the autofill tool is used. Perform following steps.

- Enter a number in the first cell;
- Enter the next number in the next cell (row or column);
- Select both the cells;
- Left click on the handle in the cell;
- Without releasing the mouse button, drag across the desired cell range;
- Release the mouse button.

You may see screen as shown in figure 6.13.

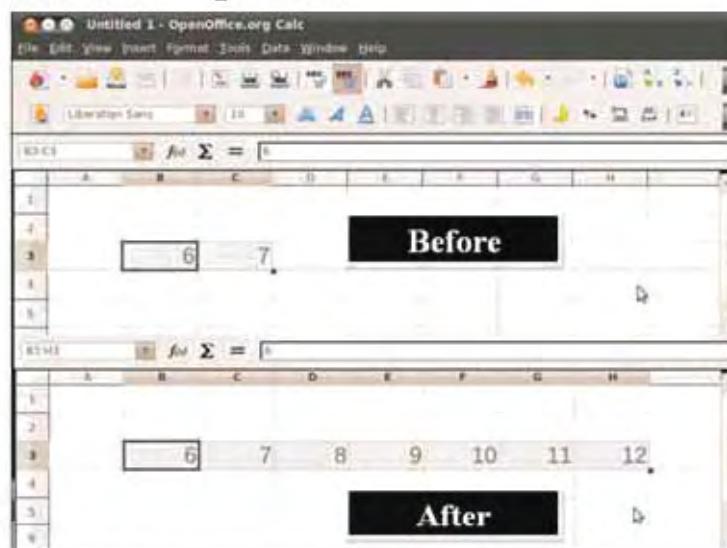


Figure 6.13 : Autofilled numbers

Auto fill of numbers as mentioned above is also known as an arithmetic progression. Try the arithmetic progression in a different way as follows.

Write an odd number (say 1) in a cell. In its adjacent cell, write successive odd number (here, the next odd number is 3). Select both the cells together and try autofill tool to fill next ten cells. Observe the numbers filled automatically. Are they all odd numbers?

Try different series of numbers such as:

- 10, 20, 30,
- 25, 50, 75,....
- 2, 4, 5,....

You can create number tables ($1*1=1$, $2*1=2$, ...) using this tool.

If you have only single number in a cell and you try autofill tool with it, you may have list of numbers increased by 1.

Just think of this. Instead of a number, what if an address or reference is there in a cell? Write some cell address in a cell and try to autofill some adjacent cells.

Relative and Absolute Address

Earlier we have copied a formula from one cell to many other cells. If a formula written in a cell is copied to other cell, the Calc automatically changes the addresses of cells included in a formula (as we have seen in the example of shopping bill in Chapter 5).

A cell address in the cell A1 is =C1. That is, the cell A1 refers content of the cell C1; third element from the cell (A1) itself. If you copy the content of the cell A1 to B1; the B1 should refer to content of the third cell from the cell B1; that is D1. Such address mechanism is called relative. When a relative address is copied from one cell to another, it will be automatically changed. To avoid this we need to write address such as =\$A\$1 (with the dollar sign (\$) added before the row and/or column) to make it as fix address. This is called an absolute address. The absolute address does not change when the formula is copied or moved. Absolute address always indicates the value of a specific cell. For example, the address =\$A\$4 will always be replaced with the value of the cell A4 in an expression; wherever it is copied or moved.

You may fix a column and vary rows by putting \$ sign against column and vice-versa. For example, if you write address as =A\$4, and try to copy it to different location, only the column will be changed, not the row. It will always remain row 4.

To autofill cells with a formula, do the following.

- Select a cell that contains a formula;
- Left click on the handle in the cell;
- Without releasing the mouse button, drag across the desired cell range;
- Release the mouse button.

Formatting a Cell

We may change property of a cell. We may apply colours to a cell, add image within a cell, add border to a cell, and change fonts of content of the cell. We can also apply different formats to content of a cell; such as changing format of dates and numbers. That is, formatting can be applied to numbers, content, alignment, cell contents, cell border and cell background. For this purpose, we may use a dialog box Format Cells or formatting toolbar. Figure 6.14 displays formatting toolbar.

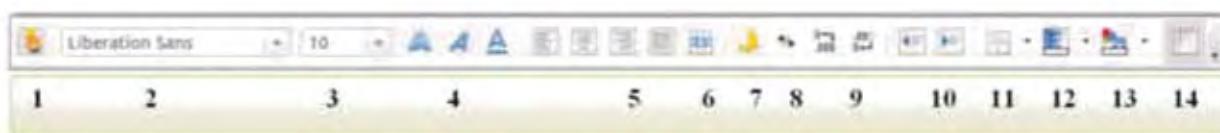


Figure 6.14 : Formatting toolbar with labels

The description of labels used in figure 6.14 is given in table 6.4.

Label	Description
1	Apply style.
2	Font.
3	Font size.
4	Bold, italic, and underline fonts.
5	Alignment as: left, centred, right, and justified.
6	Merge selected cells.
7	Currency.
8	Percent.
9	Add/delete decimal place.
10	Decrease/increase indent.
11	Cell border format.
12	Background colour of a cell.
13	Font colour in a cell.
14	Add/remove unformatted cell border.

Table 6.4 : Formatting toolbar options

Let us see some formatting styles. Later you may experiment the above formatting styles on some cells of a worksheet.

Number Format

The number format affects the appearance of numbers in cell. The number format can be applied to a single cell, multiple selected cells and range of cells. Choose Format cells, open tab Numbers and set the choices through the dialog box appear which is illustrated in figure 6.15. The description of labels used in figure 6.15 is given in table 6.5.

Label	Description
1	Selection of appropriate format such as numbers, dates etc.
2	Typical appearance of the format selected.
3	Regional settings.
4	Preview of the selected cell; inserting a thousand separator.
5	Number of decimal places.
6	Number of leading zeros.
7	To make negative numbers red.
8	Thousand separators.

Table 6.5 : Number formatting options for cells

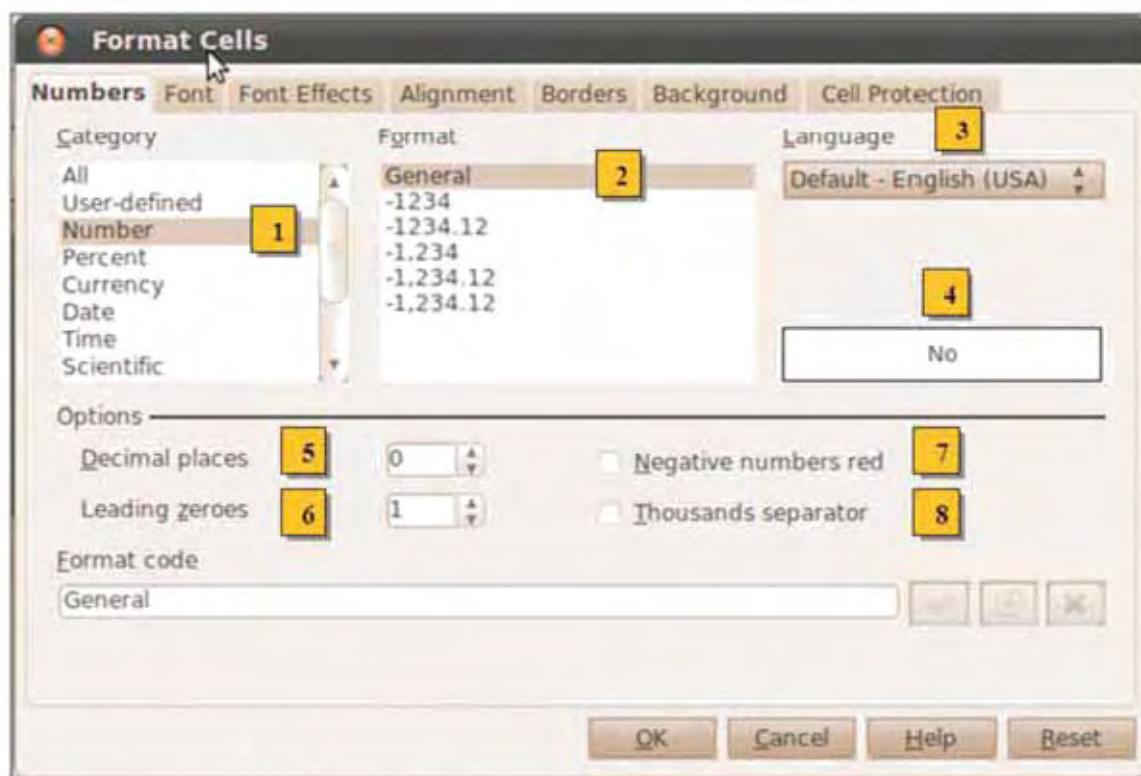


Figure 6.15 : Number formatting options for cells

Figure 6.15 shows typical number formatting options for cell containing numbers. To change date format, currency format or percentage format, appropriate choice is to be selected.

Fonts can also be changed through the **Font** tab of the **Format Cells** dialog box; effects can be changed in the **Font Effects** tab as illustrated in figure 6.16.

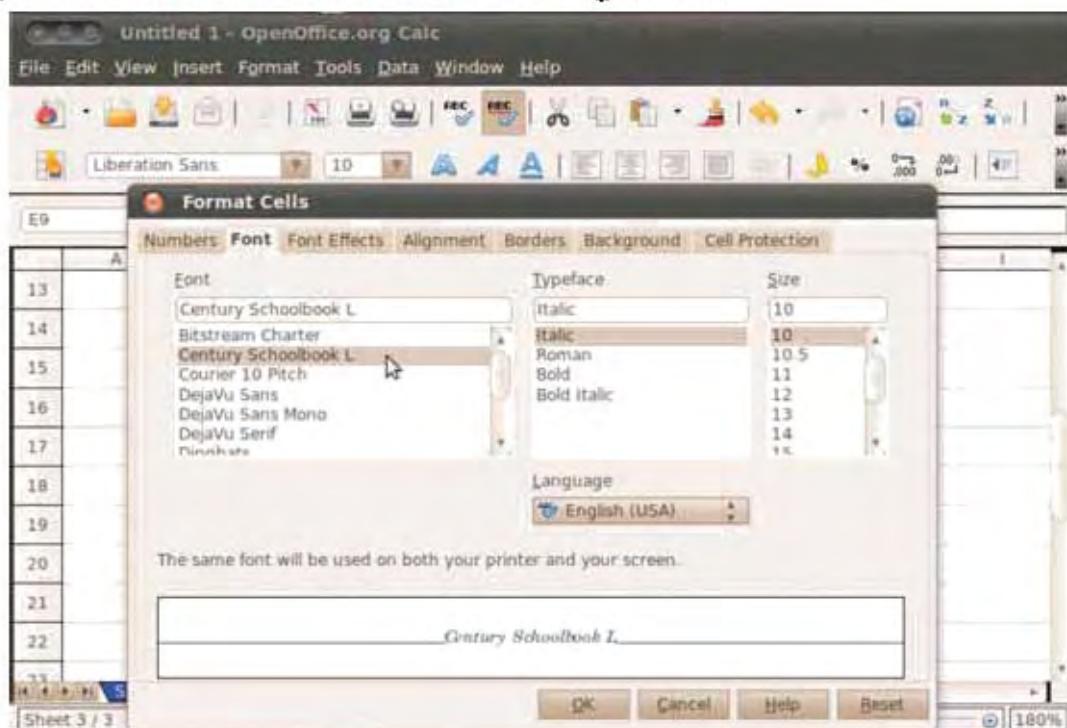


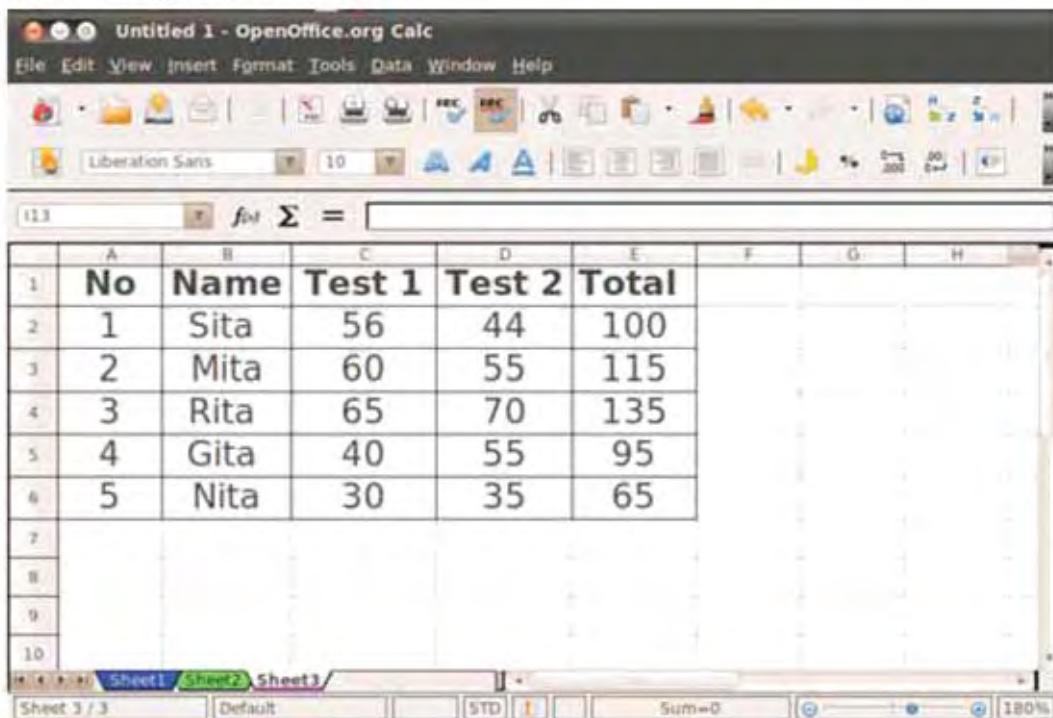
Figure 6.16 : Changing fonts

You might have applied some of these font effects; such as bold, italic, underline, and font size. Some of these effects are presented as icons on formatting toolbar. Just selecting the proper icons, these effects can be directly applied.

Data Sorting and Filtering

It is easy to understand and use data when it is arranged in some order. Arrangement of data in particular order is known as sorting. Operations to sort data in ascending or descending order are available in Calc. If there is a single column or row of data, it is sorted in desired fashion. However, if data is spread in more than one row and column, then all the columns or rows are shifted and sorting is done.

Consider a set of marks of students as an illustration. To find out who has secured maximum total marks and to prepare merit list, we need to do sorting in descending (largest value comes first) order. To experiment this, open a new spreadsheet and enter the data as shown in figure 6.17. Save the file with appropriate name.



The screenshot shows a spreadsheet titled "Untitled 1 - OpenOffice.org Calc". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar below has various icons for file operations, cell selection, and data manipulation. The spreadsheet area contains the following data:

	A	B	C	D	E	F	G	H
1	No	Name	Test 1	Test 2	Total			
2	1	Sita	56	44	100			
3	2	Mita	60	55	115			
4	3	Rita	65	70	135			
5	4	Gita	40	55	95			
6	5	Nita	30	35	65			
7								
8								
9								
10								

Figure 6.17 : Data entered in worksheet

To sort the data as shown in figure 6.17 based on values of cell E i.e. **Total**, we need to do following.

- Click on a cell in the column by which data are required to sort;
- Click the **Sort Descending** button in the **Formatting** toolbar as shown in figure 6.18.



Figure 6.18 : Sorting using formatting toolbar

There is another way to do the same operation. All you need is to select **Data → Sort**. A dialog box will appear. In the sort criteria of the dialog box, select **Descending**. Figure 6.19 illustrates a typical view of sort dialog box.

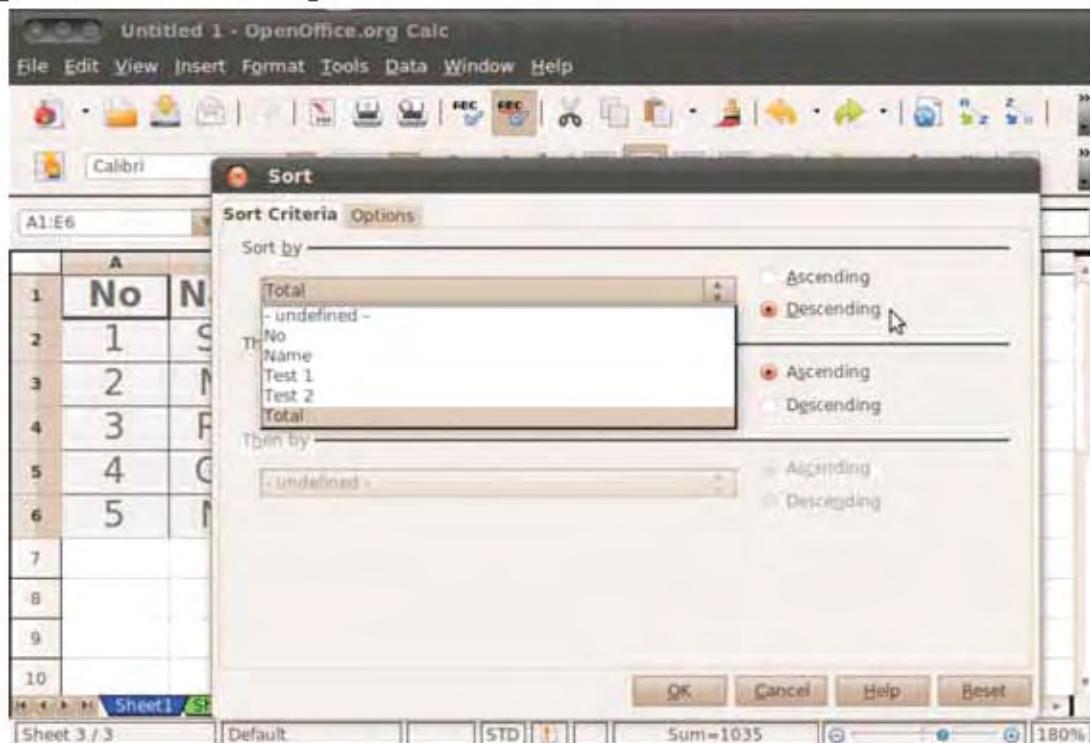


Figure 6.19 : Sort dialog box

After sorting, you will find that the total marks **135** will come at first position in the column. Obviously these marks are scored by **Rita**, and her name should come first in the student's name column. If you see the result, along with Rita's total marks all other data related to her have been moved to the first row. That means, if you sort by a column, many rows may be moved as illustrated in figure 6.20. Note the changed order of the records (data) in the figure 6.20.

f13 Σ = []					
	A	B	C	D	E
1	No	Name	Test 1	Test 2	Total
2	3	Rita	65	70	135
3	2	Mita	60	55	115
4	1	Sita	56	44	100
5	4	Gita	40	55	95
6	5	Nita	30	35	65
7					
8					
9					
10					

Figure 6.20 : Sorted data in ascending order

Not only on values but you can sort items according to alphabets too. Take an example of an English language dictionary, in which words are sorted in alphabetical order. Look at the student's data we are experimenting right now, and try to sort the data set on the students' name. Do the following to sort the above data according to students name in ascending order.

- Select **Data → Sort**;
- Give column name as **Name**;
- Check the sort order, by default it is **Ascending**.

You may sort data on two or more fields. Consider a scenario, you have list of students with information such as their full names, their locations and their marks. You may want to sort the data first by location and then by name. Consider the example given below :

1. Enter some sample data provided in table 6.6 in a Calc worksheet.

Name	Location	Marks
Nita	Vallabh vidyanagar	65
Gita	Anand	95
Sita	Baroda	100
Mita	Ahmedabad	115
Rita	Gandhinagar	135
Kavita	Vallabh vidyanagar	110
Punita	Ahmedabad	105
Sangita	Anand	85
Sunita	Ahmedabad	70
Babita	Anand	90

Table: 6.6 : Sample data for sorting

2. Select the data including headings.
3. Choose **Data → Sort**.
4. Select "**Location**" in the **Sort by** field.
5. Select "**Marks**" in "**Then by**" filed.
6. Click **OK** button.

These operations are shown in figure 6.21.

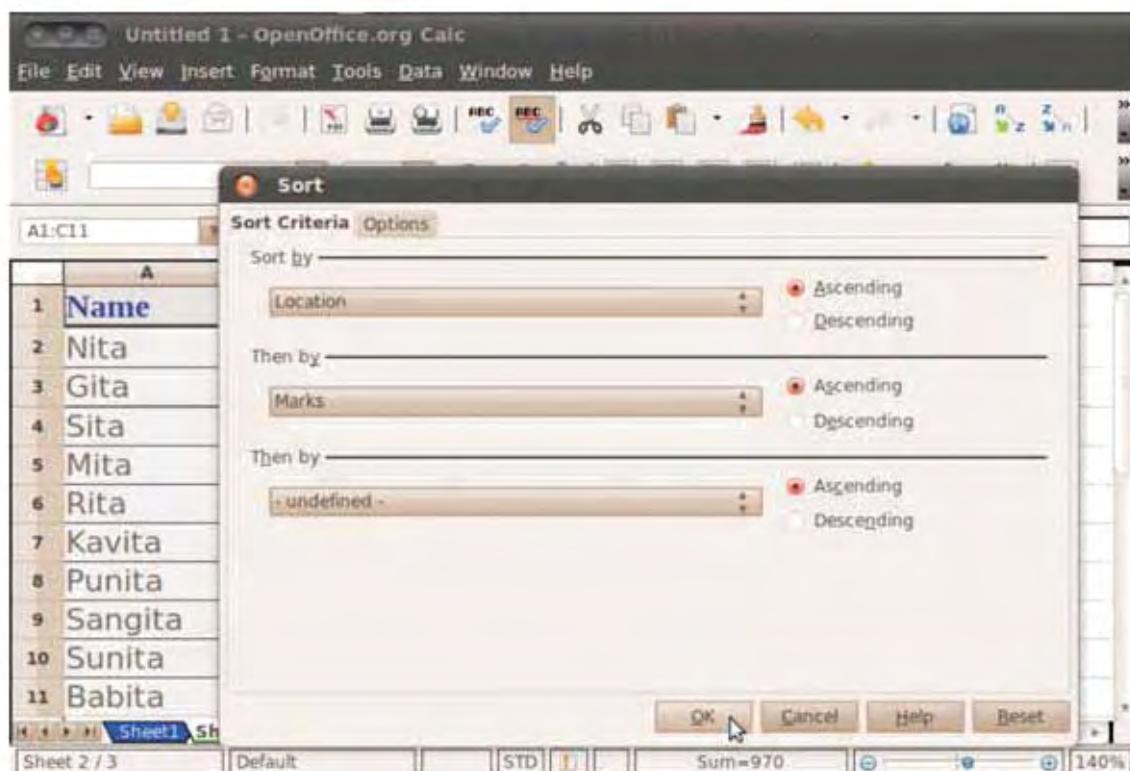


Figure 6.21 : Sorting options

Performing these steps will sort the selected data first by location and then by marks. That means, all Ahmedabad students are listed together in ascending order (lowest first) of their marks. As location is sorted in ascending order, the Ahmedabad location will come first and Vallabh vidyanagar will come at end. The sorted data will look like as shown in figure 6.22. You may add third level of sorting, if required.

The screenshot shows the sorted data in OpenOffice.org Calc. The data is now organized by location and marks. The 'Location' column has been moved to the second column. The data is:

	A	B	C
1	Name	Location	Marks
2	Sunita	Ahmedabad	70
3	Punita	Ahmedabad	105
4	Mita	Ahmedabad	115
5	Sangita	Anand	85
6	Babita	Anand	90
7	Gita	Anand	95
8	Sita	Baroda	100
9	Rita	Gandhinagar	135
10	Nita	Vallabh vidyanagar	65
11	Kavita	Vallabh vidyanagar	110

Figure 6.22 : Sorted data

Data Filtering

Data filtering help to filter out unnecessary data and presents only those which you want to see. Consider the data about the students name and location, which we have just sorted in previous section. Let us add a filter on the data. Suppose we want to see details of students who scored 100 or more than 100 marks. To do so, we may add automatic filters as follows:

1. Select **Data → AutoFilter** as shown in figure 6.23. You may use the exiting worksheet, in which you have already entered the data;

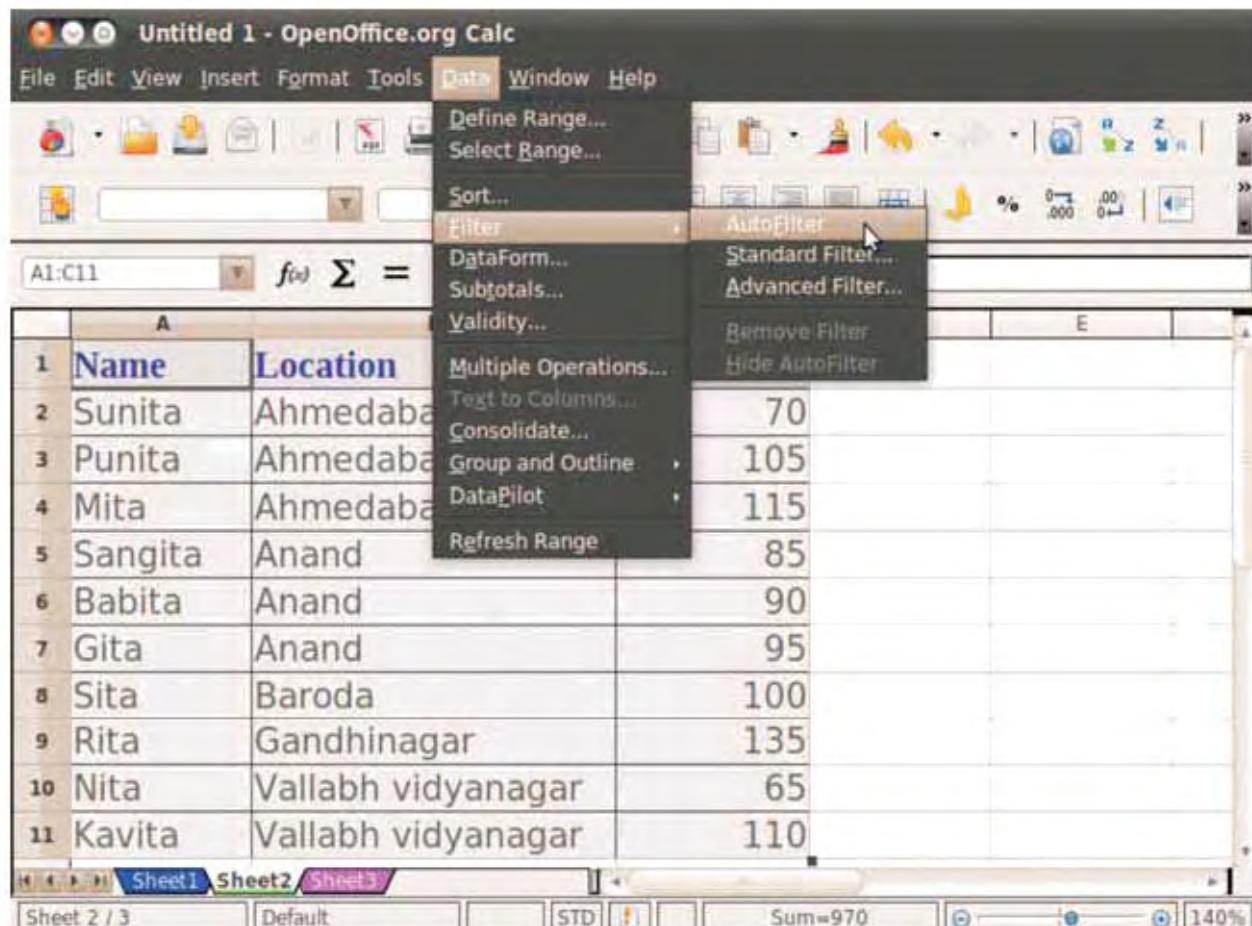


Figure 6.23 : Applying filters to the selected data

2. You will see dropdown arrows at the top of each column;
3. Click and hold down the arrow on particular column (here Marks) and select a value. You may choose from values you have already entered or phrases such as top 10 etc. If you require specific option, you may select standard filter and choose values from it as shown in figure 6.24.
4. You may see that the spreadsheet has filtered out student's data with marks less than 100. You may have noticed the changed colour of the arrow indicating that some data have been filtered out as shown in figure 6.25.

Choose **Data → Filter → AutoFilter** again to turn off the filter.

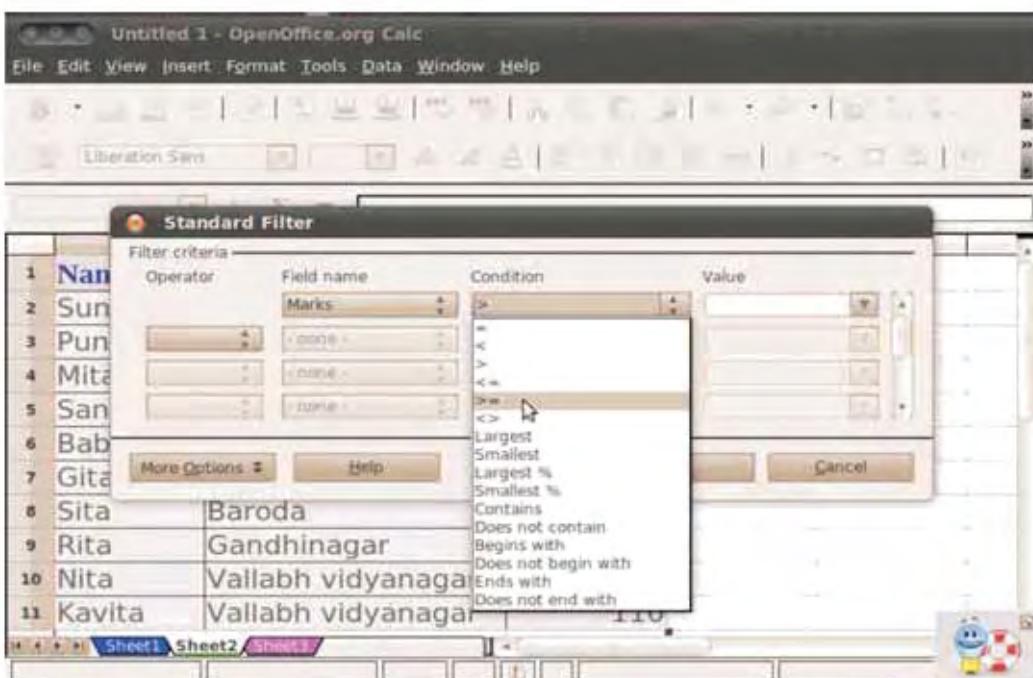


Figure 6.24 : Standard filter

A1:C11					f(x)	$\Sigma =$	Marks
	A	B	C	D	E		
1	Name	Location	Marks				
3	Punita	Ahmedabad		105			
4	Mita	Ahmedabad		115			
8	Sita	Baroda		100			
9	Rita	Gandhinagar		135			
11	Kavita	Vallabh vidyanagar		110			
12							
13							
14							
15							
16							
17							

Figure 6.25 : Filtered data

Data Validations

As stated earlier, the spreadsheet packages are used for data analysis such as balance sheet preparation and financial analysis, merit list preparation and analysis of students' results. In this case the formula and functions you have used will output correct result provided valid data are given. However, one cannot make sure that the users of the system always enter valid data. To avoid errors in entering data we may use data validations. To validate cell contents select the cells and select **Data → Validity**. This will open a Validity dialog box as shown in figure 6.26.

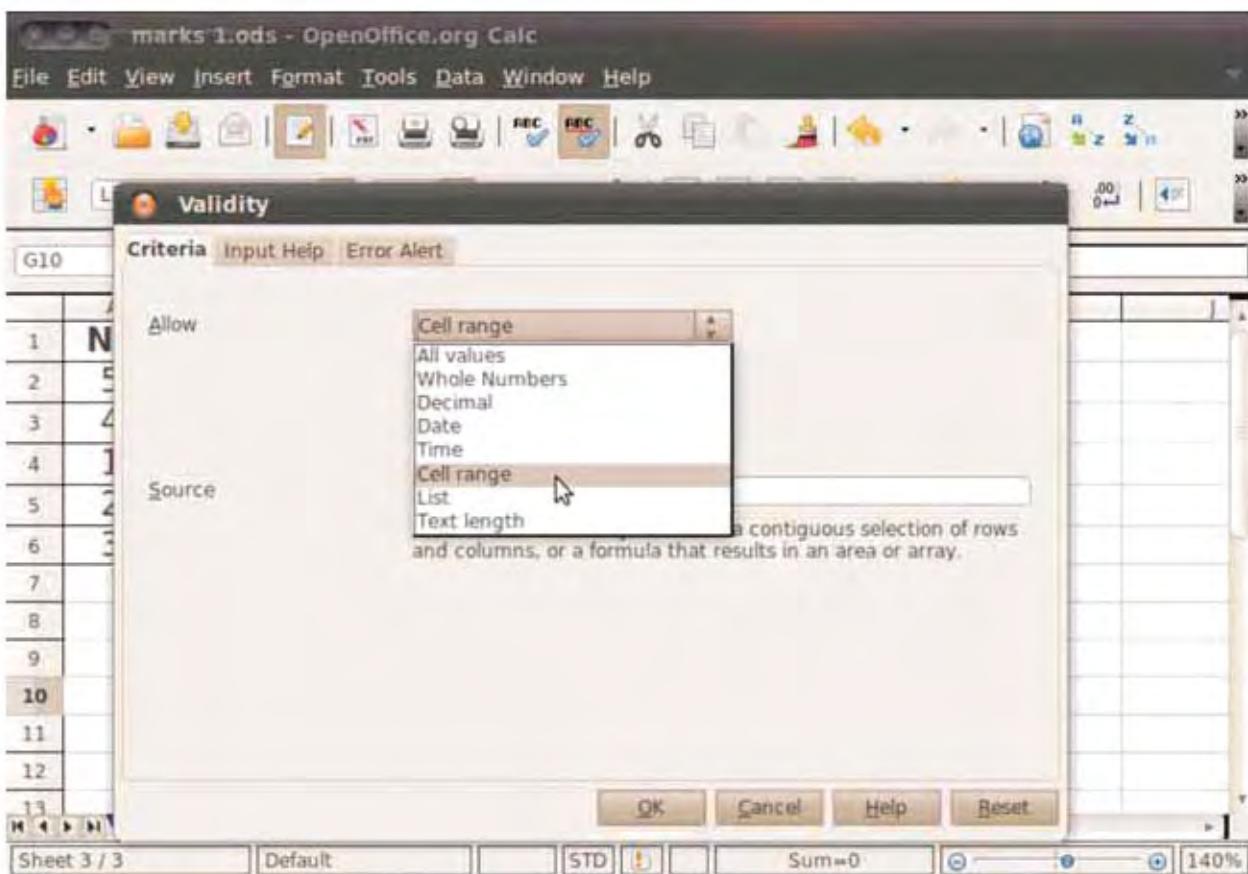


Figure 6.26 : Data Validity options

You may now set different criteria for data which you expect from users. To control invalid data entry we may set some conditions here. These conditions include cell range, all values, only whole numbers to be entered, only date values to be allowed etc. We may do such setting with **Criteria** tab shown in the figure 6.26. The second tab is about **Input Help**, where we can provide help to users at the time of data entry.

There is one more tab, **Error Alert** to provide alert on error, if any. It is to be noted that if you select **Format** or **Delete All** from the **Delete Contents** window, then the validity options you have set will be removed.

Other Operations in Calc

Calc provide many other useful operations checking of spelling, finding and replacing text, printing the worksheet and many more. Some of these options are discussed in this section.

Spelling Check

Users while typing the contents in Calc worksheet generally make errors in spelling. Calc gives us a facility to check spellings. The words spelled incorrectly will be underlined with red line, provided the spelling check toggle button on the toolbar is turned on. As you type something wrong (with incorrect spelling), the word is highlighted with red underline. All you need to do is just a right click and select appropriate option from given choices. To check spelling the Calc uses a dictionary. If the word you have entered is found in the dictionary with the same spelling, the word is considered as correct word. Figure 6.27 illustrates possible option for spelling checking and correction.

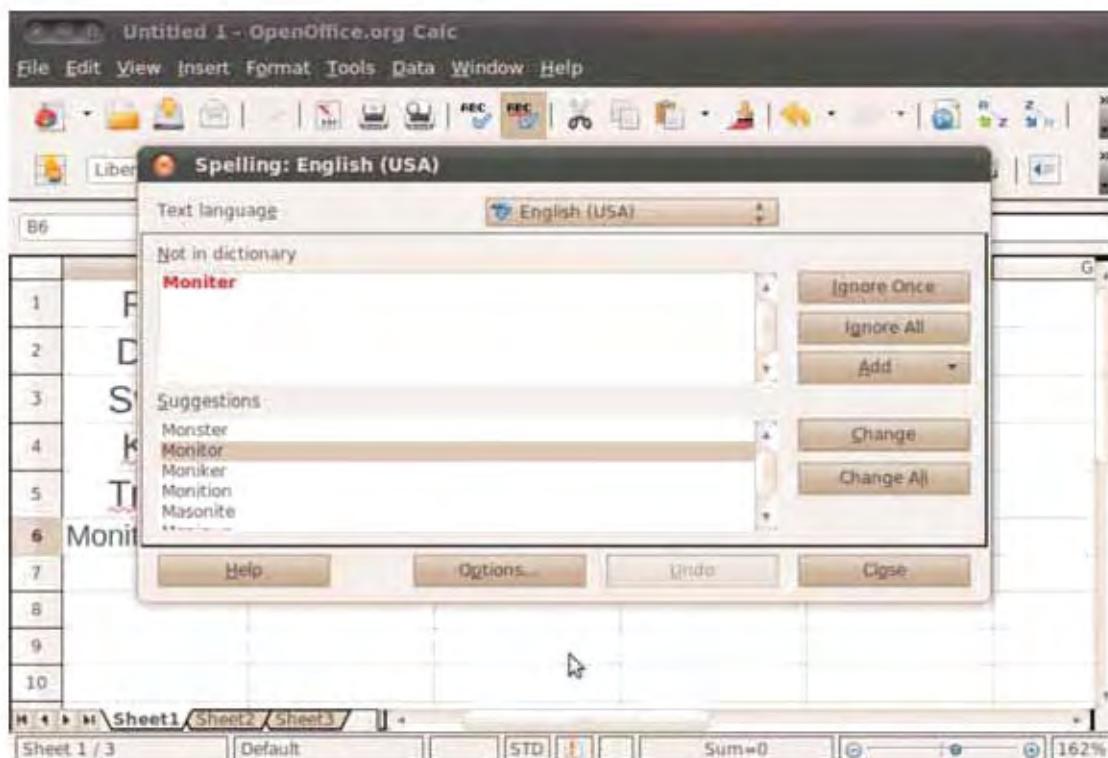


Figure 6.27 : Spelling check options

It may so happen that the word is correct, such as your name, but it may not be there in the standard dictionary. In that case, your name may have red underline too! To avoid this, you may add your name to the dictionary using option as "Add to dictionary". Now your name will not be highlighted as incorrect anymore. One simplest way to go for spelling check is just to press a function key F7. To avoid spelling check automatically, you may choose **Format → Cells → Font language=[None]**

Find and Replace

The commands Find and Replace are used to quickly find data in a worksheet and replace text or numbers in cells fully or partially. To do so, perform following steps:

- Select **Edit → Find & Replace;**
- Perform necessary actions as asked in the Find & Replace dialog box.

Printing

Printing the worksheet requires special care because of the size of the worksheet. Unlike the word processor, the size of a document of a spreadsheet application is not fixed. Multiple rows and columns you have created might not fit in a page. Moreover, some information will be truncated. Further, it is not required to print the whole worksheet. Therefore, before printing a document, it is advisable to use the print preview option. The print preview option provides a view of spreadsheet along with page breaks and margin. To preview a worksheet following steps can be carried out.

- Select **File → Page Preview;**
- At top of the computer screen, a bar similar to the one shown in figure 6.28 will appear. It shows possible actions that user can take to preview the document.

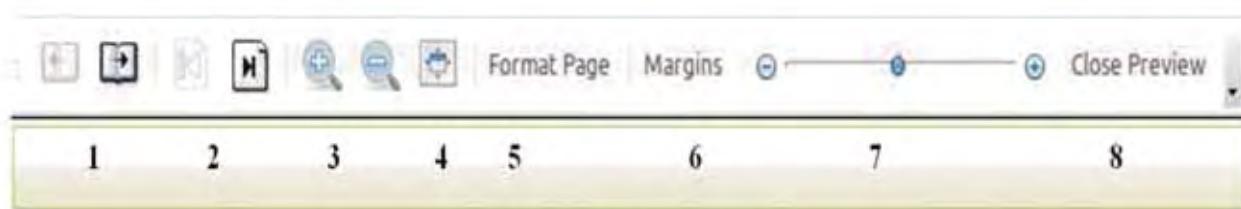


Figure 6.28 : Preview bar

By choosing necessary icons, we can carry out necessary actions. Labels shown along with the figure 6.28 describes the actions, which are listed in form of table 6.7.

Label	Description
1.	Turning of pages
2.	Jump to last page
3.	Zoom tool
4.	Full screen preview
5.	Leads to a dialog box for page formatting
6.	Makes margins visible
7.	Scaling
8.	Close preview

Table 6.7 : Preview options

Page preview can also be used to set page formatting values. To carry out page formatting, perform following steps :

- Select **File → Page Preview**;
- Click the **Format Page** button on the horizontal dialog box appear; it will open a rectangle dialog box called Page Style.
- Select the second tab entitled "**Page**" in the **Page Style** dialog box as shown in figure 6.29;
- Select required choices and save changes by clicking on **OK** button.

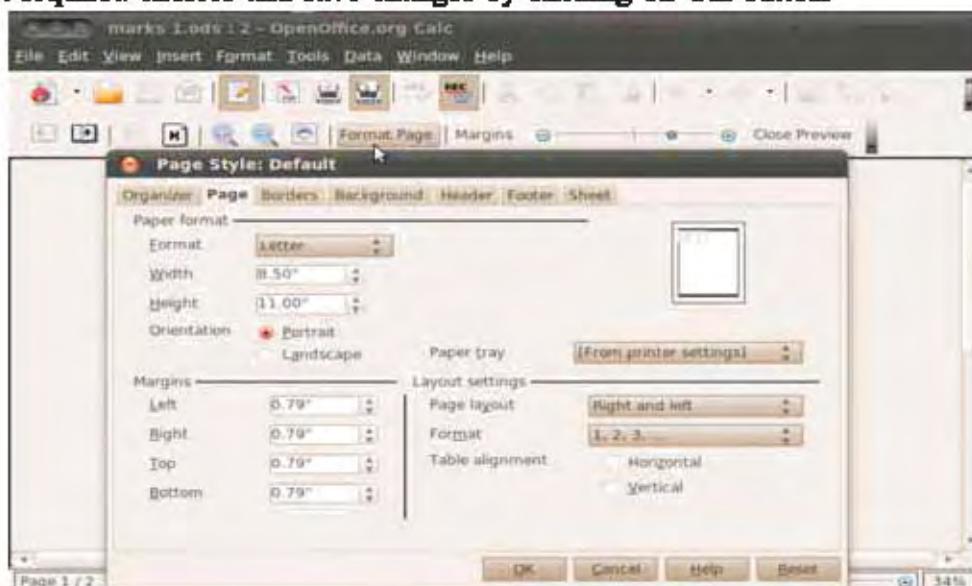


Figure 6.29 : Page formating options

You may choose paper size (**Format**) such as A4, legal or any custom size page. You can also set left, right, top and bottom margins using **Margins**. Beside these options, the other page formatting option you may experiment are **Paper Tray** option (specifying from where your papers comes to the printer; from upper tray or lower tray), and **Page Layout** if you are printing more than one pages on a physical page (single paper). The format choice is also used to specify the paper **Orientation** such as portrait or landscape orientations.

What if you want to print the cell grids (borders surrounding the cells) as well as the column and row headers? Just do the following things :

- Select **Format → Page**; a dialog box will appear.
- Alternatively, select **File → Page Preview** and click the **Format Page** button;
- Open the **Sheet** tab;
- Tick the **Grid** checkbox as illustrated in figure 6.30;
- Tick the **Column and row headers** checkbox;
- Close the dialog box.

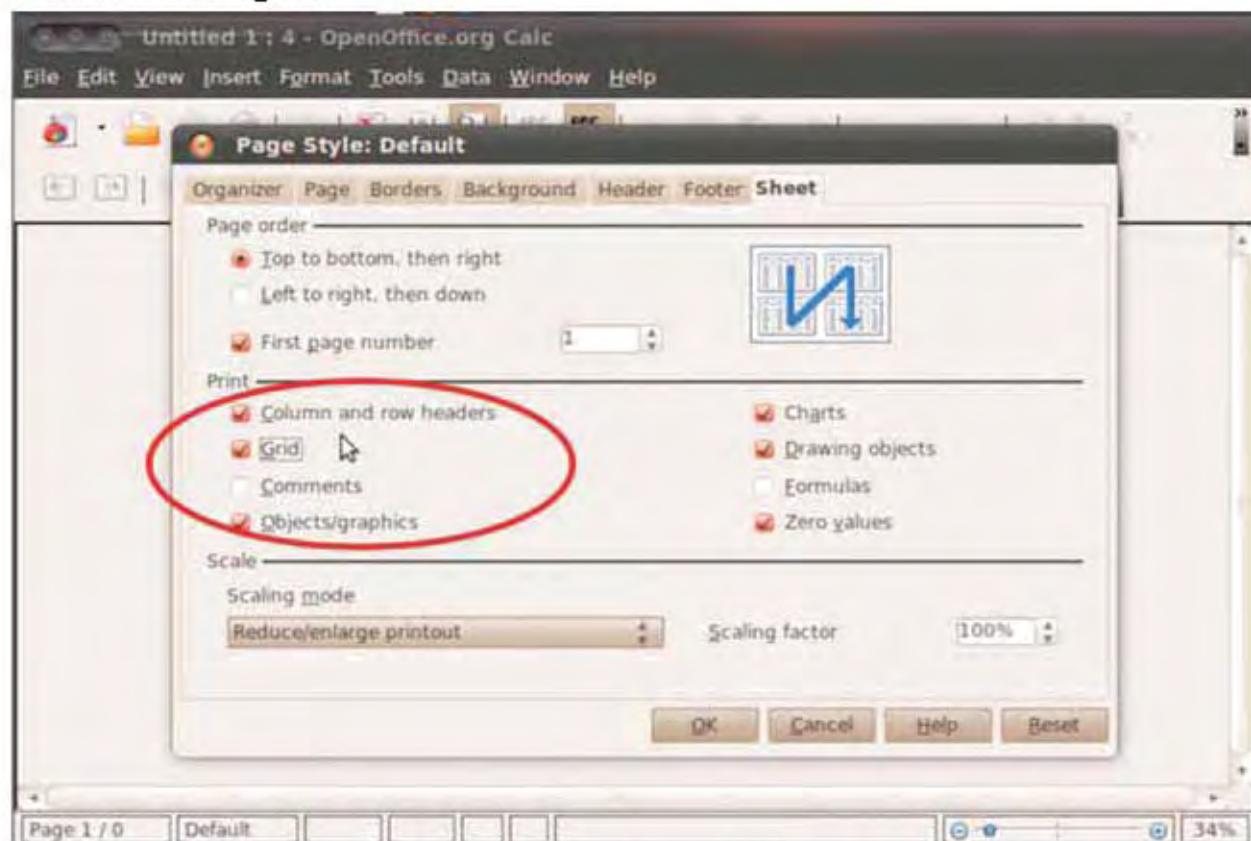


Figure 6.30 : Setting column and row headers as well as cell grids

You may also print colourful borders pressing the tab **Borders**. You may select border type, border colour and border thickness along with other options to set borders. Figure 6.31 illustrates options for setting borders to the cells. Once you set parameters click on **OK** button. Then you may preview the page. Figure 6.31 also shows a preview of data (blue rectangle indicated by block arrow) with red border as well as yellow background for the data entered. However, to see preview you have to go for Page Preview after setting the borders, cell headers and background.

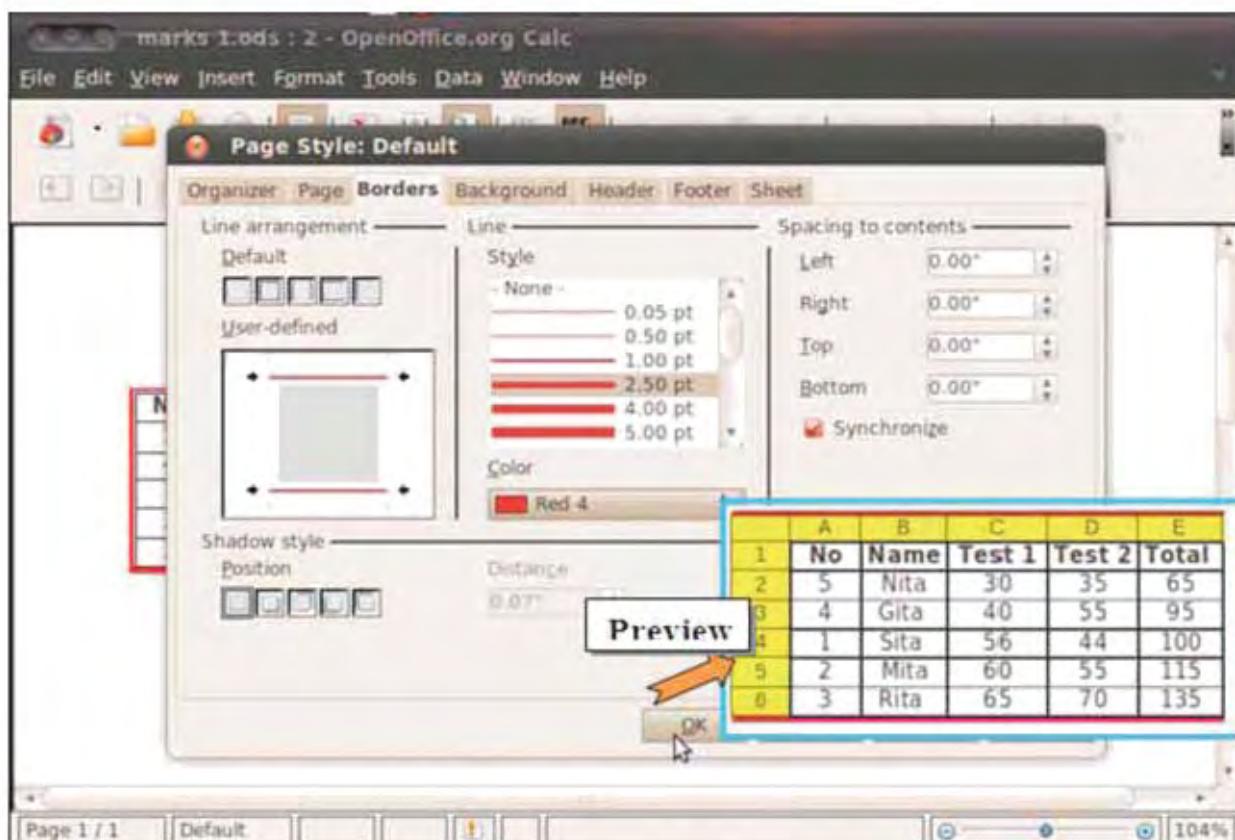


Figure 6.31 : Setting Borders and background

When every option and format is set, you may print the document just by clicking the printer icon at the top line.

Modifying Page Breaks

To see and modify page breaks, do the following:

- Select **View → Page Break Preview**;
- If you are happy with the layout, you may print it else change the page breaks by dragging the lines as per your requirement.

By performing the menu command **View → Normal** you will get the normal view of the page.

Header and Footer

At top of every page and at bottom of every page an area is kept reserved. These areas at top and bottoms are known as header and footer respectively.

To create the header do following.

- Select **File → Page Preview → Format**;
- Open the Header tab and click Edit and apply the changes needed.

The header and footer areas in Calc are divided in three areas each, called left area, centre area and right area. Whatever you write in these areas, will be automatically inserted into the document. Inserting header is illustrated in figure 6.32.



Figure 6.32 : Inserting Header

Getting Help

Calc has a built-in manual to provide help on different functions. To search required function, Calc provides a search function. All you need to do is just select **Help → OpenOffice.org Help**. Figure 6.33 shows a typical screen that appears when you call for help. The simplest alternative way is to press the **F1** function key on the keyboard.



Figure 6.33 : Getting Help

Summary

In this chapter we have seen data formatting, sorting and validating operations in Calc spreadsheet. We learnt spreadsheet level operations which facilitate opening a new or existing document, saving document in Calc as well as other format, other operations on file such as closing and searching documents. We also learnt sheet level operations such as re-naming and re-colouring sheet, inserting and deleting sheet, etc. in various ways. The cell level as well as row and column level operations also we have seen in this chapter. We learnt data sorting, data validation, and other miscellaneous utilities such as autofill operations, setting header and footer, printing documents, spelling check and getting help.

EXERCISE

- 1.** Explain how to create a new document in Calc.
- 2.** List how to rename and recolor the Calc worksheet.
- 3.** Explain insertion and deletion of extra worksheets in Calc.
- 4.** Write a short note on deleting cell content.
- 5.** Explain working of autofill tool in Calc.
- 6.** Explain relative and absolute address in Calc.
- 7.** Write a short note on data filtering.
- 8.** Write a short note on data sorting.
- 9.** What is data validation? Why is it necessary ? Also explain how Calc does it ?
- 10.** Explain how to add header and footer in a Calc worksheet while printing.
- 11.** Choose the correct option from the following :
 - (1) Which of the following technique arranges the data set in a particular order such as ascending or descending ?

(a) Data formatting	(b) Data validations
(c) Data filtering	(d) Data sorting
 - (2) Which of the following techniques can be used to allow only date values in a cell ?

(a) Data formatting	(b) Data validations
(c) Data filtering	(d) Data sorting
 - (3) Which of the following options when selected deletes all data validations ?

(a) Delete formatting	(b) Delete all
(c) Delete formula	(d) Delete me

LABORATORY EXERCISE

1. Save your Calc document in other format using Save as option. Open it from other software in which you have saved it.
 2. Open a Calc document, add 7 sheets using Sheet tab, colour them as rainbow colour such as "Violet", "Indigo", ...etc. Make sure that "Indigo" sheet is re-coloured with indigo colour and so on.
 3. Try autofill tool with negative numbers.
 4. Try autofill tools with two dates. Write 15-08-2013 in a cell and in its adjacent cells write a successive date. Drag the content to next ten cells. Using this technique try to create a monthly calendar.
 5. Add your name in to the dictionary so that red line will not be displayed under it.
 6. Enter your friends list with their birth dates and sort the list according to City and then by names. Take columns headings as shown below.

Name	Address in one line	City	Mobile no	Email	Date of Birth	Month of Birth	Year of Birth
Krushi	1, Gokul Park	Anand	01	July	1995
....							

7. Filter the list of friends you have created in problem 6 of this exercise, so that it will display only those friends information whose birthdates are in August month.
8. Use help to find information about sorting and filtering. Prepare notes on these topics using the help.
9. Use your notes on data validations and also use the data you have entered in the problem 6 of this exercise, to validate the data in the date of birth column so that nobody can enter date such as 33. You have to make data validation in such a way that it will accept only 1 to 31 numbers as date of birth values.





Functions in Calc

Entering only data and simple mathematical expressions may not be sufficient for real life applications involving complex decision making operation and analysing large amount of data. To carry out such activities, spreadsheet programmes such as Calc provide built-in standard functions for mathematical, logical, statistical, date and time, financial and other calculations on numbers and text.

A function, like an expression generally begins with an equals (=) sign, followed by a function name and one or more function argument specified in its brackets. Sometimes, operators such as "+" may also be used with functions and formulas. However, it is a common practice to use '=' symbol as a prefix to avoid a function being simple text. The arguments can be a value, an address of a cell, text, a constant, or one or more functions. In Chapter 5, we have used the SUM function from the tool bar (shopping bill example). The SUM function, directly applied from the tool bar (indicated by symbol sigma - Σ), is a kind of shortcut. Systematically the same function can be written as =SUM(A1:A10) where SUM is a name of function, which specifies that summation procedure is called; and A1 and A10 are references to the Calc cells in the same worksheet. Together A1 to A10 (A1:A10) specify a range of cells containing 10 different cells. The content written in the brackets is known as argument of the function. Instead of the cell address or cell range, we can also write a value as =SUM(23,25).

Similarly, you may also wish to try =SUM(A1:A10) function using ';' (comma) or ';' (semi-colon) instead of ':' (colon) in between the cell address A1 and A10. You may observe that just by using ';' or ';' one can sum up only two values at the cell A1 and A10 respectively. To sum all the values between A1 and A10 including content within them; we must use either ':' or individually all cell addresses as =SUM(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10).

Figure 7.1 illustrates the three ways to perform addition operation. You can see the status of selected cell having value (SUM=55) and the formula bar showing full summation expression. You can also see when A1 is referred in the formula, its colour is blue (as shown in the figure 7.1) and the colour of the cell A1 is also blue. This is really useful to cross check the entered formula.

The process of inserting a function is similar for all functions. The arguments specific to the function and the correct spelling of the function are mandatory. That is, to use a function we need to enter a correct spelling of the function along with necessary arguments. This is known as syntax of the functions. If we do not know the exact spelling of a function and the possible attributes, it will result in error.

The screenshot shows a Microsoft Excel spreadsheet with data in columns A and B. Column A contains numbers 1 through 10. Column B contains the sum of the corresponding row values: 55, 55, and 55 respectively. The formula bar at the top shows three different ways to enter the sum of cells A1 to A10:

- SUM(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10)
- SUM(A1:A10)
- =SUM(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10)

Red circles highlight the formula bar and the status bar at the bottom, which displays "Sum=55".

	A	B	C	D
1	1	55	Using Σ	
2	2	55	SUM(A1:A10)	
3	3	55	=SUM(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10)	
4	4			
5	5			
6	6			
7	7			
8	8			
9	9			
10	10			
11				

Sheet1 Sheet2 Sheet3

Sheet 1 / 3 Default INSRT STD Sum=55 152%

Figure 7.1 : Three ways to perform addition operation

To enter a function in a cell, do the following :

- Select a cell in result is to be displayed;
- Start entering function with '=' sign; if you do not write '=' sign (or any other valid operator such as "+"), the function will be considered as normal text.
- Enter the name of a function;
- Enter function arguments; and
- Press the Enter key;

Function Wizard

It is not possible to place many functions on the toolbar (generally top line of the screen), or to remember them with their correct syntax; hence most of the functions are entered through the function wizard.

Use the function wizard in the following way.

- Select a cell in which result is to be displayed;
- Select **Insert → Function**. You can also press **Ctrl** and **F2** to do the same from an opened spreadsheet document;
- Another alternative is to press the **Function Wizard** key in the formula toolbar as shown in figure 7.2;

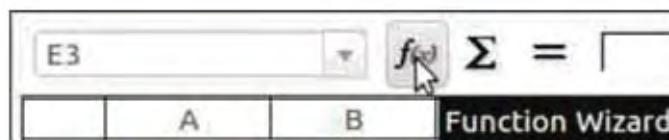


Figure 7.2 : Function Wizard

- A dialog box will appear. Choose required function from the dialog box. Figure 7.3 shows typical dialog box.

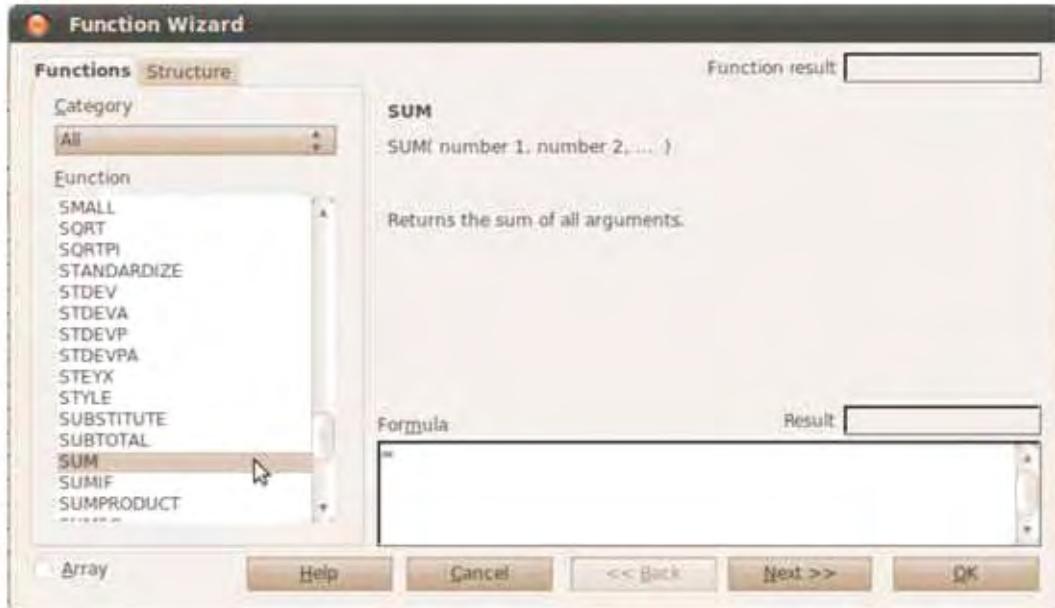


Figure 7.3 : Dialog box for function wizard

- Select the **SUM** function and select **Next >>** button;
- Enter values or select the function arguments from the worksheet. Here you can enter up to 30 arguments values, addresses (references) or ranges one by one. You may see the list of the arguments indicated by number 1, number 2, ... etc. As shown in figure 7.4;

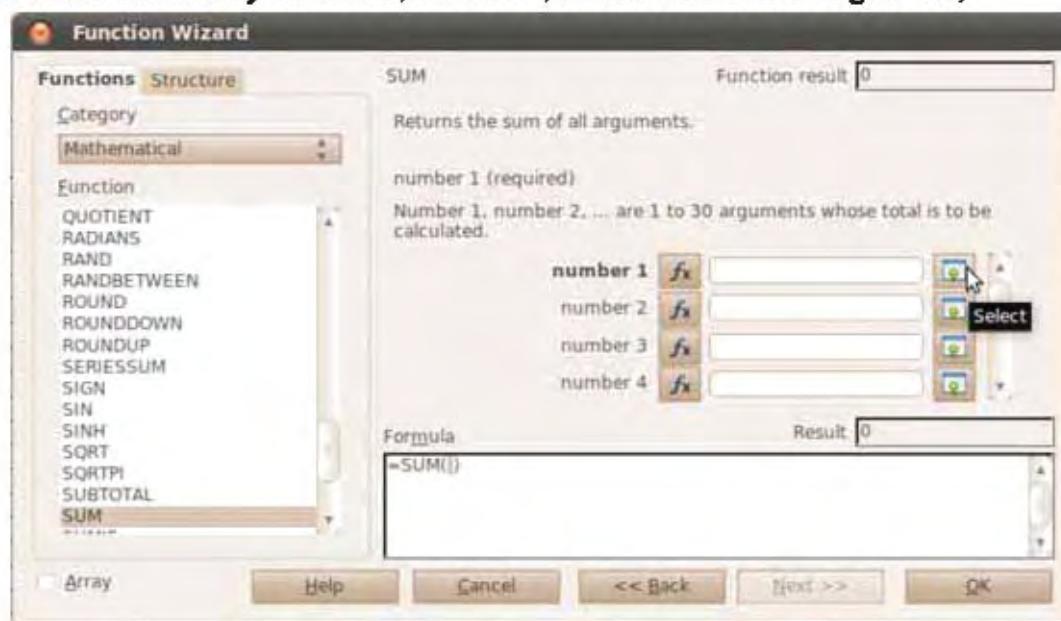


Figure 7.4 : Entering arguments of a function

- If you do not know, what is the value of a number and where it lies within the spreadsheet, you may wish to select it from the worksheet directly;
- To select values use the **Select** button as shown with the mouse arrow in the figure 7.4;
- When you click on the select button, it will lead you to the worksheet area. The whole dialog box for the function wizard is converted into a small toolbar type rectangle. Now you can see the worksheet area as well as the minimized function wizard. You are free to use mouse and select the cell or cell range which you want. Figure 7.5 indicates the minimised function wizard (rectangle outlined with blue line) and the worksheet area. If the data in which you are interested are not available on the same sheet, you may go to other sheet also. Verify data on the function wizard. It is **Sheet1:A1:A10** in our case;

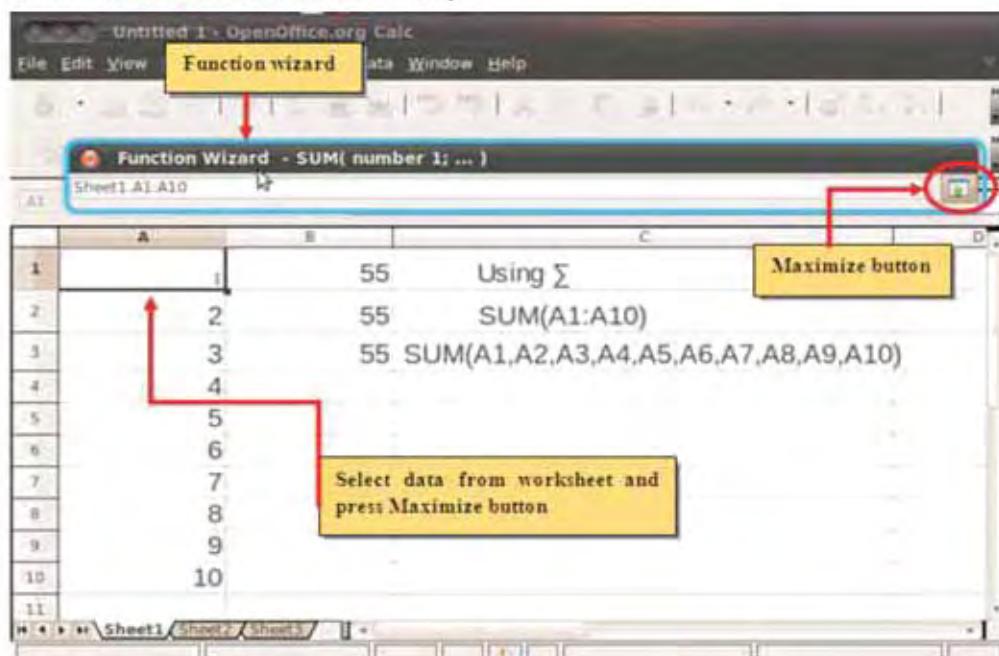


Figure 7.5 : Selecting a cell range using Select button

- After doing this, press **Enter** key or **Maximize** button (encircled in figure 7.5). Again full function wizard similar to the one shown in figure 7.5 will appear. Observe that the data range you have selected is visible in number 1 argument box.
- Click on **OK** button to finish and close.

Calc provides many built-in functions in various categories. The major categories of functions are mathematical, statistical, logical, and text formatting function. When you want to use another function as an argument, place the text cursor inside the original function's parentheses and choose the new function from the list.

Mathematical Functions

Math functions are very helpful when we work with numbers in Calc. Let us now see usage of some math functions.

Absolute Values of Numbers

The ABS function removes the minus sign (-) from a negative number and makes the number positive. It does nothing to 0 (zero) or positive numbers. The syntax is **ABS (n)** where n is a number. Figure 7.6 illustrates working of ABS function. You can also see the formula of the function in the formula bar (encircled).

	EXP	Find	X	=ABS(B1)
1	Value of n:			-13
2	Function used:			=ABS(B1)
3	Value of ABS(n):			13

Figure 7.6 : Working of an ABS function

The ABS function is useful when you want only positive number. Consider that you are using an expression within another expression. The Calc first calculates inner expression and uses the inner expression value as argument of the outer expression. But the outer expression (such as square root) allows only non-negative (positive and zero) values. What if the inner expression returns a negative value? The negative value of the inner expression makes the outer expression invalid. To avoid this, we just take absolute value of the inner operation and send it to the outer expression.

Exponential Function

The EXP function returns the values of the exponential function e^x for the given number x where e is approximately equal to 2.718281828. The syntax is **EXP(Number)**. For example if we write **=EXP(1)** in a cell it will be considered as e1, which returns 2.72. Figure 7.7 illustrates this function. You can see the formula of function in formula bar (encircled).

	EXP	Find	X	=EXP(B1)
1	Value of n:			1
2	Function used:			=EXP(B1)
3	Value of EXP(n):			2.71828182845904

Figure 7.7 : EXP function

Factorial Function

The FACT function returns the value of the factorial function for the given number (the product of all integers from 1 to the given number say n). The general form of the function is **FACT(n)** where n is an integer number.

For example if we write **=FACT(3)**, the Calc will consider this as an expression ($1 * 2 * 3$) and return 6. Similarly FACT(6) returns multiplication of 1 to 6 numbers (that is $6 * 5 * 4 * 3 * 2 * 1$); which is 720. Figure 7.8 illustrates the factorial function with value 6.

	FACT	<input type="button" value="f(x)"/>	X	✓	=FACT(B1)
1	Value of n:		B		6
2	Function used:				=FACT(B1)
3	Value of FACT(n):				720

Figure 7.8 : FACT function

Natural Logarithm

The **LN** function returns the value of the natural logarithm (the logarithm base of e) for the given positive number. The syntax is **LN(Number)** where number is a positive number. For example, if we write =LN(8) in a cell it will return 2.08 (log_e8) as shown in figure 7.9.

	LN	<input type="button" value="f(x)"/>	X	✓	=LN(B1)
1	Value of n:		B		8
2	Function used:				=LN(B1)
3	Value of LN(n):				2.0794415417

Figure 7.9 : LN function

Logarithm Base 10

The **LOG10** function returns the value of logarithm base 10 for the given positive number. The general form is **LOG10(Number)** where number is a positive number. For example, LOG10(6) is considered as LOG106 and returns 0.77 as shown in figure 7.10.

	LOG10	<input type="button" value="f(x)"/>	X	✓	=LOG10(B1)
1	Value of n:		B		6
2	Function used:				=LOG10(B1)
3	Value of LOG10(n):				0.7781512504

Figure 7.10 : LOG10 function

Power

The POWER function returns the number raised to the given power. The general form **POWER(Number, Power)**. If we write =POWER(10,3); it will return 1000 (10^3). It is equivalent to the notation 10^3 . See figure 7.11.

POWER		f(x) X ✓	=POWER(B1,B2)
	A		B
1	Value of n:		10
2	Value of power:		3
3	Function used:		=POWER(B1,B2)
4	Value of POWER(n, power):		1000
5			

Figure 7.11 : POWER function

Product of Many Arguments

The PRODUCT function multiplies up to 30 arguments (x_1, x_2, \dots, x_{30}). Each argument can be a single value or a range of cells. The general form is **PRODUCT(n1,n2,n3,...,n30)**. Consider that we have entered some data in a worksheet as displayed in figure 7.12.

A4		f(x) Σ	=	=PRODUCT(A1:A2,B1:B2,D1:D2)
	A	B	C	D
1	10	1		3
2	20	2		1
3				
4	1200			
5				

Figure 7.12 : Product of many arguments using PRODUCT

Suppose we write =PRODUCT(A1:A2, B1:B2, D1:D2) in a cell. It will return 1200. Therefore, we can say that the product function considers all the values in a cell, a cell range or more than one cell range; and calculates its product. In figure 7.12 a total of 6 values are present in column A, column B and column D. Note that the column C is empty.

Square Root

The SQRT function returns the square root of a positive number. The general form of the SQRT function is **SQRT(Number)**. To find a square root of 100 =SQRT(100) is to be written simply in a cell. It will return 10 as shown in figure 7.13.

	A	B
1	Value of n:	100
2	Function used:	=SQRT(B1)
3	Value of SQRT(n):	10
4		

Figure 7.13 : SQRT function

Integer

The INT function rounds the given number down to the nearest integer. The general form of the function is **INT(Number)**. That is, if you write **INT(15.3)**, it will return 15 as shown in figure 7.14.

	A	B
1	Value of n:	15.3
2	Function used:	=INT(B1)
3	Value of INT(n):	15
4		

Figure 7.14 : Integer number using INT

Round

This function rounds the given number to the specified places. Many times a formula results in a number having lot many figures after decimal place. Example of such a number is 12.6546. If you want only two digits after the decimal point; obviously, you can not cut the number. You have to round it. The number 12.6546 is rounded to 12.65. Student result (total marks or percentage) is often rounded in such a way that it will result a complete number. That is, if a student gets percentages as 76.66, then it is considered that the student has secured 77 percentage. The general form of the ROUND function is **ROUND (Number, Places)**. Some examples of the round function are as mentioned :

=ROUND(76.6633,2) results in 76.66

=ROUND(45.1634,0) results in 45

=ROUND(-56.5467,3) results in -56.547

In the second example listed above, we have rounded a number with 0 decimal place in order to generate a whole integer. Figure 7.15 illustrates the ROUND function.

	A	B	C	D
1	Value of n:	76.660000	45.560000	-56.547890
2	Decimal places:	2	0	3
3	Function used:	ROUND(B1,B2)	ROUND(C1,C2)	ROUND(D1,D2)
4	Value of ROUND(n,decimal places):	76.66	46	-56.548

Figure 7.15 : Examples of ROUND function

Just like ROUND, there are other two functions, ROUNDDOWN and ROUNDUP with same general form. Experiment these functions and identify their differences.

Truncate

Truncate function (TRUNC) takes two arguments; first argument is a number and the second argument is number of digits to be cut down from the number from its fractional part. The truncate function chops off the fractional part of a given number leaving some digits after decimal point, if specified. If there is no fractional part, leaving y digits after the decimal point (if the second argument is omitted), the TRUNC function cuts off the whole fractional part from the given number. The general form of the TRUNC function is **TRUNC(Number,Places)**.

See the following examples.

=TRUNC(1.239,2) returns 1.23 and digit 9 is lost.

=TRUNC(12.5) returns 12 and digit 5 is lost.

=TRUNC(-15.72) returns -15 and digits 7 as well as 2 are lost.

In the last example instead of TRUNC function try to use INT function and see the difference. You will find that the TRUNC function simply cuts off the specified part from the given number. It does not round the number. These examples are demonstrated in figure 7.16.

	A	B	C	D
1	Value of n:	1.239	12.5	-15.72
2	Decimal places:	2	--	--
3	Function used:	TRUNC(B1,B2)	TRUNC(C1)	TRUNC(D1)
4	Value of TRUNC(n,decimal places):	1.23	12.00	-15.00

Figure 7.16 : Examples of TRUNC function

See the examples mentioned in the ROUND function. Try to use the same examples with TRUNC function.

Statistical Functions

Besides typical mathematical functions Calc also offers you a variety of statistical functions on a series or range of values. Such statistical functions help in collection, analysis, explanation, and presentation of data in order to support forecasting, predicting as well as decision making activities. Let us see some popular statistical functions.

Average

The AVERAGE function returns the average of numbers given to it. The average is also considered as arithmetic mean. You have to sum up all the numbers in a list and divide the sum by the total count (size of the list). You may consider up to 30 such values separated with the semicolon sign (;). Instead of values, as you know, you may enter cell address or a valid cell range. The general form of the function is **AVERAGE(x₁,x₂,...,x₃₀)**. For example if you write =AVERAGE(10,7,6,8,9,5) in a spreadsheet cell, it will return 7.5. Figure 7.17 shows the average operation.

	B3	f(x)	Σ	=	-AVERAGE(B1:G1)	
1	Numbers:			10	7	6
2	Function used:				AVERAGE(B1:G1)	
3	Average of Numbers:			7.5		

Figure 7.17 : Example demonstrating AVERAGE function

Mean

The GEOMEAN mean function returns the geometric mean of numbers given to it. The geometric mean of say n non-negative numbers is obtained by multiplying them all together and then taking the nth root. You may use up to 30 arguments (values, address or range). The general form of the function is **GEOMEAN(x₁,x₂,...,x₃₀)**. For example if you write =GEOMEAN(10,7,6,8,6,5) in a spreadsheet cell, it will return 6.8.

There is another type of mean function supported by the Calc, which is HARMEAN (harmonic mean). The harmonic mean has similar general form as **HARMEAN(x₁,x₂,...,x₃₀)**. Try to find harmonic mean of the same data using the above function.

Median

The median is defined as middle number of the group when they are ranked in order. If you can not find the exact one middle number (in case the total number of arguments in group is even), take two middle numbers and average them. The MEDIAN function returns the median up to 30 arguments given to it. The general form of the function is as **MEDIAN(x₁,x₂,...,x₃₀)**. For example if you write =MEDIAN(10,7,6,8,6,5) in a spreadsheet cell, it will return 6.5.

Mode

The MODE function returns the most common value in a data set from at most 30 arguments, which can be single values or ranges of cells. The general form is **MODE(x₁,x₂,...,x₃₀)**. For example, if you write MODE function as =MODE(10,7,6,8,6,5) it will return 6. Figure 7.18 illustrates use of different statistical functions.

	A	B	C	D	E	F	G
1	Numbers:	10	7	6	8	6	5
2	Function used:	AVERAGE(B1:G1)					
3	Average of Numbers:	7					
4							
5	Function used:	MODE(B1:G1)					
6	Mode of Numbers:	6					
7							
8	Function used:	GEOMEAN(B1:G1)					
9	Geo.Mean of Numbers:	6.822					
10							
11	Function used:	MEDIAN(B1:G1)					
12	Median of Numbers:	6.5					

Figure 7.18 : GEOMEAN, MEDIAN and MODE functions

CountA

The COUNTA function returns the total number of cells or arguments which have some value in it. You may give up to 30 arguments (values, address or range) separated with the semicolon sign (;). Some of the arguments may contain some value (of any type) some may be empty. The COUNTA function returns the total number of cells having some values in it. The empty cells will not be considered. The general form of the function is **COUNTA(x1, x2, ..., x30)**. For example if you write =COUNTA(10,7,6,8,6,5) it will return 6.

Largest Value

The LARGE function returns the k^{th} largest numeric value found in the given set of values x ; The general form is =LARGE(x, k) where x denotes the range of cells containing some numeric values; and k is the position.

This function considers the numbers specified in the range in a descending order and number at k^{th} position is returned. For example, with set of values as 10, 7, 6, 8, 6 and 5 the LARGE function with value 2 returns value 8. The 8 number is 2nd largest value in the given set.

You can set the k argument to any integer value you like, but it must not be zero or larger than the number of elements in the given set of numbers.

Smallest Value

The SMALL function returns the k^{th} smallest numeric value found in the given set of values x . The general form is =SMALL (x, k) where x denotes the range of cells containing some numeric values; and k is the position.

This function considers the numbers specified in the range in an ascending order and number at k^{th} position is returned. For example, with set of values as 10, 7, 6, 8, 6 and 5; the SMALL function with value 4 returns value 7 value, which is the 4th smallest value in the given set. See figure 7.19 illustrating examples of the LARGE, SMALL and COUNTA function.

	A	B	C	D	E	F	G
1	Numbers:	10	7	6	8	6	5
2	Function used:	COUNTA(B1:G1)					
3	Count of Numbers:	6					
4	Function used:	LARGE(B1:G1;2)					
5	Second Largest Value:	8					
6	Function used:	SMALL(B1:G1;4)					
7	Fourth Smallest Value:	7					
8							
9							
10							

Figure 7.19 : Examples of the LARGE, SMALL and COUNTA functions

Rank

The RANK function returns the rank of a given number in the given set of numbers. You need to provide a value (or address of the value) for which you need to find the position, the set of values (cell range) and order of the sorting; 1 for ascending and 0 for descending. The function will sort the values from the indicated set of values in ascending or descending order and returns the position (1st, 2nd, 3rd, and so on) of the given number.

The general form is **RANK(number, set, order)**. Suppose you have secured 67 marks in an examination. Your five friends have secured marks such as 47, 56, 78, 59, and 66. You know that among your friends, your rank is second; considering highest marks first, that is in descending order. This information you can find just by seeing the data; as you consider only your friends data; only six items. What if, there are so many items? Answer is simple. Use the Rank function. For example, to know your position (from the top, obviously!), you can use the Rank function =Rank (67, A1:F1, 0) it will return the position 2. Figure 7.20 illustrates RANK function.

Maximum

The MAX function can take up to 30 arguments (individually or as a range) and simply returns the maximum number within the arguments.

The general form is **MAX(x1, x2, ..., x30)**. From the example mentioned earlier regarding students marks, to find out maximum (highest) marks you may use the MAX function as =MAX(67,47,56,78,59,66). It will return 78 as illustrated in figure 7.20.

	A	B	C	D	E	F	G
1	Numbers:	67	47	56	78	59	66
2	Function used:	RANK(67,B1:G1,0)					
3	RANK of 67:	2					
4	Function used:	MAX(B1:G1)					
5	Largest Value:	78					
6	Function used:	MIN(B1:G1)					
7	Smallest Value:	47					
8							
9							
10							

Figure 7.20 : RANK, MAX and MIN functions

Minimum

The MIN function can take up to 30 arguments (individually or as a range) and simply returns the minimum number within the arguments. The general form is **MIN(x1, x2, ..., x30)**. From the example mentioned earlier regarding students marks, to find out minimum (lowest) marks you may use the MIN function as **=MIN(67,47,56,78,59,66)**. It will return 47 as illustrated in figure 7.20.

Calculation with Money

Calc also support other financial functions to aid calculation with money. Suppose one of your uncles comes to you and ask:

"Beta, just tell me using your computer, how many years should it take me to pay the money I borrowed from a bank, providing that I pay in equal amount of money every time and the interest rate is constant?"

You need not have to go for big calculation or to write a complex program. Your job is made simple by the Calc! You need to just write a function in a Calc cell as

=NPER(7.5%, -12000, 100000)

Here NPER is to calculate the period to repay your loan. The first argument (7.5%) is a constant interest rate. The second argument -12000 is the amount (in rupees) of annual instalment (it is negative, as we need to subtract this amount from the total amount borrowed every time an instalment is paid); here your uncle is paying 1000 per month and in a year he pays 12000 in total; and the third argument is total amount of loan. When you write this function in a Calc cell it will return 13.56 years ! This is what your uncle wants ! Figure 7.21 illustrates the same example.

E7			
	A	B	C
1	Total amount of loan taken:	100000	(Rs.)
2	Interest rate in %:	7.50%	(Constant rate)
3	Yearly total installment:	12000	(Rs. 1000 per month)
4	Function used:	NPER(B2, -B3, B1)	
5	Duration in years :	13.562227330178	(Years)

Figure 7.21 : Time to repay loan using NPER

The general form of the NPER function is **NPER(Interest, Instalment, Loan, Future, Type)**. The Future argument is used to enter the amount of money that remains after the last payment is made. Usually we repay whole amount of loans. The Type argument indicates if payments are made at the beginning of each accounting period or not. The Type argument value is equal to 1 when payments are made at the beginning of each accounting period else 0. The future as well as type arguments are optional and you may see that we have not used them in our example as we want to repay all our debts and each instalment we normally pay in the beginning of the account period; failing to which we have to pay more interest.

There are more such functions supported by Calc which deal with money management. Examples are interest calculation, net present value, future value, and time to maturity. Go to Calc help and explore them.

Making Decisions

The functions so far we have discussed are simply considering some arguments, process them and return some values. They really work well for us; however such functions are still not smart. A function is called smart if it is able to take some decision. For this, we need conditional functions that can make decisions based on a set of pre-defined conditions. A classic example is your result. Consider the following scenario discussing some predefined conditions about grading your performance.

If you get marks $\geq 35\%$ and $< 48\%$; you get pass class;

If you get marks $\geq 48\%$ and $< 55\%$; you get second class;

If you get marks $\geq 55\%$ and $< 60\%$; you get first class;

There are many such situations which need this type of descriptive logic and pre-defined conditions. Calc offers a set of such logical functions that simplifies the job. Let us have a look at some popular such functions.

Making decisions with IF

The prominent of all conditional functions used in the Calc is IF. It takes three arguments called Test, True and False. The Test expression/argument indicates the condition you want to test for. Calc calculates the expression and if result is true, it returns value indicated by True part; else False part.

Consider the following example and try it in a Calc worksheet.

- Write different values at cell A2 and B2.
- Go to cell C2 and write the following function.
=IF(A2>B2,"Value at A2 is greater", "Value at B2 is greater")
- Change values at the cell A2 and B2 and see what happens to the cell C2.

You can write a small formula to decide who the winner, of each game; is when yours and your friend's score are entered in two different cells as illustrated in figure 7.22.

B4	f(x)	Σ	=	=IF(B2>C2, "I win", "My friend wins")
1	A	B	C	
2	Score of a game	Me	My friend	
3		13	17	
4	Result of the game:	My friend wins		

Figure 7.22 : Who is the winner ?

You can see the formula we have used in the formula bar (encircled). In the case of student's result example mentioned above it is required to build multiple if functions. One of them is given below. Considering total marks of a student in percentages is stored in the cell A1. You may write If function as :

=IF(A2>=60,"First",IF(A2>=55,"Second",IF(A2>=48,"Third","Fail")))

In the above function, the first IF statement checks value of the cell A2; the entered marks of students in percentages. If value at the cell A2 is greater than or equal to 60, then the expression holds true and it returns the value indicated by the first expression; which is "First". If the percentages are less than 60, it leads to the second expression, which is another IF statements. You may go on adding many such If within Ifs; hence called nested if. Figure 7.23 explains it in an illustrated way. In the figure we have written multiple student names and their percentages to illustrate various possibilities of the IF statement. However, the formula is written once only (see encircled formula bar), and it is dragged (copied) for other students as shown in figure 7.23.

	A	B	C	D	E	F	G
1							
2	Name of student:	Mita	Sita	Rita	Gita	Nita	Babita
3	Student's marks in %:	77	69	59	49	44	30
4							
5	Class of the student:	First	First	Second	Third	Fail	Fail
6							
7							

Figure 7.23 : Example of IF function

When it comes to bigger formulas, it is possible to loose the grip on it and easy to introduce errors. To avoid this, you may take help of function wizard. One advantage of using function wizard is that you need not have to type the exact name of the functions. Once you see and experiment the function wizard, you may select the **Structure** tab on this and see the structure of the formula you have written as shown in figure 7.24; which graphically displays the structure of the function creation. If there are errors, they are shown as red spots on the list.

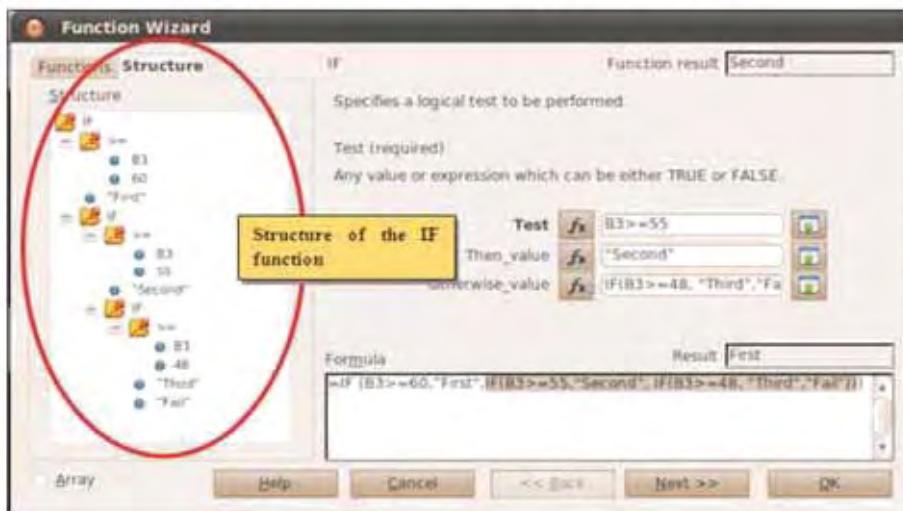


Figure 7.24 : Structure of IF function

Relational Operators

In the above example discussing IF statement, we have seen operators such as \geq . Such operators are known as the relational operators. Table 7.1 lists relational operators that you may need.

Relational Operator	Symbol	Description
Equal	=	Both the operands are (say A1 and A2) are equal Example:=IF(A1=A2, "A1 is equal to A2", "A1 and A2 not equal")
Less than	<	First operand is less than the second one Example:=IF(A1<A2, "A1 is less than A2", "A1 is not less than A2")
Greater than	>	First operand is greater than the second one Example:=IF(A1>A2, "A1 is greater than A2", "A1 is not greater than A2")
Less than or equal to	\leq	First operand is less than or equal to the second one Example:=IF(A1<=A2, "A1 is less than or equal to A2", "A1 is not less than or equal to A2")
Greater than or equal to	\geq	First operand is greater than or equal to the second one Example:=IF(A1>=A2, "A1 is greater than or equal to A2", "A1 is not greater than or equal to A2")
Different from	\neq	Both the operands are not equal, the first operand is not equal to the second one Example:=IF(A1 \neq A2, "A1 is different than A2", "A1 is not different than A2")

Table 7.1 : Relational operators

Logical Functions

The Calc support some more functions that support logical decision making. These functions are AND, OR, and NOT. These functions are used to build complex formulas and functions. This section describes the three logical functions mentioned above.

TRUE

This function returns the logical value TRUE. The general form of this function is **TRUE()**

This function returns the logical value FALSE. The general form of this function is **FALSE()**

The general form of NOT is **NOT(expression resulting in logic value)**. NOT negates logical values, so FALSE becomes TRUE and TRUE becomes FALSE. See the following examples.

=NOT(6<>6) returns TRUE.

=NOT(1=1) returns FALSE.

Can we write NOT(TRUE) and NOT(FALSE)? Try this. Figure 7.25 illustrates some examples of TRUE, FALSE and NOT functions.

	A	B
1		
2	Logical function	Value
3	TRUE()	TRUE
4	FALSE()	FALSE
5	NOT (TRUE)	FALSE
6	NOT (FALSE)	TRUE
7	NOT ($6 <> 6$)	TRUE
8	NOT ($1 = 1$)	FALSE
9		

Figure 7.25 : Examples of TRUE, FALSE and NOT functions

AND

This function is used to compare results of up to 30 (thirty) arguments. The result of the AND function is TRUE if all of the conditions /arguments results in TRUE as well. The general form is **AND(Cond_1,Cond_2,...,Cond_30)**.

Here the arguments of the AND function are conditions, each capable of resulting itself in either TRUE or FALSE; that is Boolean values. Consider this example. If we need to check whether content of three cells A1, A2 and A3 are equivalent or not, we need to check if A1 and A2 are equal and then A2 and A3 are equal. If so, we may conclude that all three are equal. That is two conditions need to be formed using relation operator "="; which are respectively $A1=A2$ and $A2=A3$. AND function can be written as =**AND(A1=A2,A2=A3)**. Figure 7.26 illustrates an example as discussed above.

C5	f(x)	Σ	=	=AND(A1=A2,A2=A3)
	A	B	C	
1	12			
2	12			
3	6			
4				
5	Are the above values same?			FALSE
6				

Figure 7.26 : Example of AND

OR

The general form is **OR(Cond_1,Cond_2,...,Cond_30)**.

The OR function is used to compare results of up to 30 (thirty) conditions. The value of OR is TRUE if at least one of the conditions tested is TRUE. For example, consider the formula we have used in the logical AND function described above. Let us change slightly the formula. Earlier we wanted all the three arguments A1, A2 and A3 must be same. Now we want to proceed if any two of them are equal. This can be written as =**OR(A1=A2,A2=A3,A1=A3)**.

The above function returns TRUE value in following conditions :

- if A1 is equal to A2;
- if A2 is equal to A3;
- if A1 is equal to A3; or
- If A1, A2 and A3 all are equal.

The function returns FALSE if any of them is not matching. That is in following conditions the function returns FALSE value.

- If A1 is not equal to A2; if A2 is not equal to A3; and if A1 is not equal to A3; and this happens altogether then only the formula is FALSE.

That is, OR function returns FALSE value only if all the conditions provided are FALSE. Figure 7.27 shows some examples of OR function while Table 7.2 illustrates some cases of logical function.

A4	f(x)	Σ	=	$=OR(A1=A2,A2=A3,A1=A3)$
	A	B	C	D
1	12	12	12	6
2	12	12	6	12
3	12	6	12	12
4	TRUE	TRUE	TRUE	FALSE

Figure 7.27 : Example of OR

Cases			Expression Values	Result
Value of A1	Value of A2	Value of A3		
12	12	12	A1=A2 → TRUE A2=A3 → TRUE A1=A3 → TRUE	TRUE
12	12	6	A1=A2 → TRUE A2=A3 → FALSE A1=A3 → FALSE	TRUE
12	6	12	A1=A2 → FALSE A2=A3 → FALSE A1=A3 → TRUE	TRUE
6	12	12	A1=A2 → FALSE A2=A3 → TRUE A1=A3 → FALSE	TRUE
6	12	31	A1=A2 → FALSE A2=A3 → FALSE A1=A3 → FALSE	FALSE

Table 7.2 : Cases of the logical function $OR(A1=A2;A2=A3;A1=A3)$

String Functions

Joining Two Strings of Text

The CONCATENATE function can join up to 30 strings of text. The general form of the function is **CONCATENATE(text1, text2, ..., text30)**.

You may use '&' sign instead of the function name. That is, to join two strings such as "ice" and "cream". You may write a function as shown below.

=**"ice"&"cream"** or

=**CONCATENATE("ice","cream")**

Both the above functions return a string "icecream". In case you want space between them, you may add a blank within any of the string as =**"ice "& "cream"** or =**"ice"& " cream"**.

If the strings to be joined are stored in the separate cells (say A1 and A2), you could also use the function as =**A1&A2** or =**CONCATENATE(A1,A2)**.

Perform the following exercise.

- Go to cell A1, write your first name.
- Go to cell A2, write your middle name.
- Go to cell A3, write your surname.
- Go to cell A4, write a blank as " ".
- Use & function in cell C3 as =**A1&A4&A2&A4& A3**

Figure 7.28 demonstrates some concatenation examples.

	A	B	C
1	ice		
2	+ cream		
3			
4		-----	icecream
5			
6	Use of & operator		
7	"ice"&"cream"		icecream
8	"ice" & " " & "cream"		ice cream
9	B1 & " " & B2		ice cream
10	B1 & B2		icecream

Figure 7.28 : Examples of CONCATENATE function

LOWER

This function turns all given characters into lowercase. The general form of the function is **LOWER(text)**. For example =**LOWER("School")** returns "school".

Upper

This function turns all given characters into uppercase. The general form of the function is **UPPER(text)**. For example =**UPPER("School")** returns "SCHOOL".

Proper

This function turns first letter of all words into uppercase and remaining alphabets in lowercase letters. The general form of the function is **PROPER(text)**. For example =**PROPER ("I love my school")** returns **I Love My School.**

The string functions lower, upper and proper are normally used for formatting the documents. A properly formatted document has good readability and clarity.

Roman and Arabic

The Roman function converts the number provided into its equivalent Roman number. The general form is **ROMAN(number, mode)**.

Please note that Roman numerals can only represent numbers from 0 to 3999; therefore the number argument you provide must be within this range. As you know a Roman number can be written in many ways. Suppose you want to write number 8 in Roman. You may write it as VIII; however, the number 8 can also be written as IIX. The second argument of the function is about such mode of representation, which used to simplify the R the resulting Roman number. The Mode operator can take values 0, 1, 2 or 3. TV and movie industries use Roman numerals to encode their film strips and other material.

The Arabic function is reverse of the roman function, it considers a roman number as its arguments and convert the number into an Arabic number.

See the example given below.

=**ROMAN (125)** which returns **CXXV**.

Find out what the function =**ARABIC("MMXI")** returns ? Is it **2013** ?

Figure 7.29 shows some examples of ROMAN and ARABIC functions.

	A	B	C
1	Function	Value	
2	ROMAN(125)	CXXV	
3	ARABIC("mmxiii")	2013	
4	ARABIC(ROMAN(2013))	2013	
5			
6	Also consider following examples to see how mode works		
7	ROMAN(999)	CMXCIX	
8	ROMAN(999,0)	CMXCIX	
9	ROMAN(999,1)	LMVLIV	
10	ROMAN(999,2)	XMIX	
11	ROMAN(999,3)	VMIV	

Figure 7.29 : Examples of ROMAN and ARABIC functions

Trim

This function removes extra spaces from strings. That is if you write **TRIM("I Love My School")**, it will return **I Love My School !**

The general form is **TRIM(text)**.

Comparing Strings

EXACT compares two strings and returns 1 if they are exactly the same or 0 if they are not. For example =**EXACT("Blue", "Blue")** returns 1, while =**EXACT("Blu", "Blue")** returns 0.

The general form is **EXACT(s1, s2)**.

Substring

There are two functions in Calc, one for left side substring and another for right side substring.

The **LEFT** function returns the first character from the string text, or a specified number of n characters starting from the left.

The general form is **LEFT(Text, n)**. That is to extract first 4 left characters from a string you may write =**LEFT("Ram Dhashrathbhai Patel", 4)** returns "Ram".

Similarly the **RIGHT** function returns the last n characters from the end (right side) of the string text. That is if you write =**RIGHT("Ram Dhashrathbhai Patel", 5)**, it will return "Patel". The general form is **RIGHT(Text, n)**.

Calc also provides utility to find middle of a string. The general form is **MID(Text, Start, n)**.

That is to extract middle name from the arguments given in the function; we need to write function as =**MID("Ram Dhashrathbhai Patel", 5, 13)** which returns "Dhashrathbhai".

Length of a String

The **LEN** function returns the length of the given string. The general form is very simple, which is **LEN(Text)**. That is, if you write **LEN("Ram")** it will return 3. Figure 7.30 displays some examples of string function.

	A	B
1	Function	Value
2	TRIM("I Love My School")	I Love My School
3	EXACT("Blue", "Blue")	TRUE
4	EXACT("My School", "New School ")	FALSE
5	LEFT("Ram D Patel", 3)	Ram
6	RIGHT("Ram D Patel", 5)	Patel
7	MID("Ram D Patel", 5,1)	D
8	LEN("Ram D Patel")	11

Figure 7.30 : Examples of string functions

Time and Date Functions

Calc offers functions for time and date calculations. For calculation that involve shorter period (such as hours, minutes, or seconds), you may use time functions and for bigger period (such as weeks, months, or years), you may use date functions.

Today's Date

The TODAY function returns values for the today's date stored in your computer. It is function without arguments. Suppose today is 11th July, 2013; then the today() function returns 11/07/2013 or 11. Jul. 2013 (as per the format set in your computer).

Date

The DATE function returns the valid date for the given numbers for year, month, and day. That is if you experiment =DATE(2013, 1, 1), it will return 1/1/13.

Day

The DAY function is used to convert a date given in any recognizable format into a day of the month. For example, today's date is 11th July, 2013; then =DAY(today()) returns 11.

We may also write the function as =DAY(DATE(2013,7,11)). The general form is DAY(Date).

To ensure the argument provided to a date type function is in the date format, you may use DATE function as we have used. The DATE function will first convert the numbers into a valid date format and then returns the month number of the date.

Weekday

The WEEKDAY function is used to convert a valid date given into a day of the week. Days are numbered from 0 (Sunday) to 6 (Saturday). For example, if you write

=WEEKDAY(DATE(2013,7,11)), it will return 5.

The general form is WEEKDAY(Date, n). Here date is any valid date and n is a number that indicates week starts from which day. You may use following values for n.

N=1 : week starts on Sunday (day 0).

N=2 : week start on Monday (day 1).

If you do not specify the argument, it will take value 1, that is, the days are numbered from 0 (Sunday) to 6 (Saturday) as shown in the example above.

Here are some more examples.

=WEEKDAY(DATE(2013,7,11),0) returns 3

=WEEKDAY(DATE(2013,7,11),1) returns 5 (default); which is equivalent to
=WEEKDAY(DATE(2013,7,11))

=WEEKDAY(DATE(2013,7,11),2) returns 4

Month

The MONTH function is used to convert a valid date into the month number. For example, if you write =MONTH(DATE(2013,7,11)) in an Calc cell, it will return 7.

The general form of the function is **MONTH(t)**.

Year

The YEAR function is used to convert a valid date into the year number. For example, if you write =YEAR(DATE(2013,7,11)) in an Calc cell, it will return 2013.

The general form of the function is **YEAR(t)**.

Number of Days in a Year

The DAYSINYEAR function returns the number of days in the year presented by given date within the function. Try following function in a spreadsheet cell.

=DAYSINYEAR(DATE(2013,7,11))

The function returns 365.

The general form of the function is **DAYSINYEAR(t)**.

Earlier you might have written a complex formula or a special program to check whether a given year is leap year or not. Instead of this, you may just check no of days in a given year.

Difference between Two Dates

The DAYS function calculates the number of days between two dates. Note that the name of function is DAYS instead of DAY. The earlier (DAY) function returns a day number of month. The DAYS function provides number of days between two valid dates.

=DAYS(DATE(2012,7,11), DATE(2013,7,11)) returns 366. As the year 2012 is a leap year containing 366 days in it !

Use this function in this way.

=DAYS(A1, "NOW")

Here A1 is a cell address where you have to write your birth date. The second argument is **NOW()**; which will return current date stored in your computer along with time. Guess what will it return? Yes, you are correct, it will return your age in number of days! With this you can exactly find out who is the eldest in your class!

The general format of **NOW()** function is the same; **NOW()**. It is a function without argument.

Note that when second date is subtracted from the first date, many times the function returns a negative value. To avoid this, you may use the ABS function. Figure 7.31 demonstrates some date and time functions.

	A	B
1	Function	Value
2	TODAY()	11. Jul. 2013
3	DATE(2013,7,11)	July 11, 2013
4	DAY(DATE(2013,7,11))	11
5	WEEKDAY(DATE(2013,7,11))	5
6	WEEKDAY(DATE(2013,7,11),0)	3
7	WEEKDAY(DATE(2013,7,11),1)	5
8	WEEKDAY(DATE(2013,7,11),2)	4
9	MONTH(DATE(2013,7,11))	7
10	YEAR(DATE(2013,7,11))	2013
11	DAYSINYEAR(DATE(2013,7,11))	365
12	DAYS(DATE(2012,7,11), DATE(2011,7,11))	366
13	WEEKS(DATE(2011,7,11), DATE(2012,7,11),0)	52
14	YEARS(DATE(2011,7,11), DATE(2013,7,11),0)	2

Figure 7.31 : Date and time functions

Number of Weeks Between Two Dates

If there are two functions called DAY and DAYS; Similarly the WEEKS function calculates the number of weeks between two dates.

=WEEKS(DATE(2011,7,11); DATE(2013,7,11),0) returns 52.

The function uses two DATE functions to convert the arguments given into valid date formats. Beside this, the WEEKS function also uses a '0'; which indicates a week type. The '0' value indicates weekly intervals. Instead of '0' you can use calendar weeks indicated by '1'.

The general form of the WEEKS function is **WEEKS(t1, t2, Week type)**. Just as DAY and DAYS are different, similarly WEEK and WEEKS are different functions.

Number of Years Between Two Dates

The YEARS function calculates the number of years between two valid dates, either in yearly intervals (indicated by 0) or calendar years (1). For example, if you write:

=YEARS(DATE(2011,7,11), DATE(2013,7,11), 0), it will return 2.

You may find out how old you are using this function. Just find out number of years from your birth date and current date!

Functions in Other Spreadsheet Packages

Most of the spreadsheet packages support similar functions. If you take example of Microsoft Excel, it also supports most of the functions discussed in this chapter. You may have to change some minor syntax, such as little difference in spelling or use of ';' (semicolon) or ',' (comma) between the arguments of a function.

Package like Google Spreadsheets [doc.google.com] also supports the similar functions. As you know, it can be useful on the Internet platform, mobile phones and any machine that supports the standard infrastructure (hardware/software) such as Android operating system.

In addition to these, the Google packages also offer you free spreadsheet templates for various jobs such as creating students' schedule, grade reports, project planner, attendance of students and employees, managing personal budget, wedding list, managing invoices, etc. These templates have ready framework such as sections, row & column headings, formulas and script required for necessary calculations. Applications such as Google spreadsheet are becoming popular because of platform independence and mobility.

Summary

In this chapter we learnt different mathematical, statistical, string, date & time and formatting functions. We have also seen that how the functions can be entered through functions wizards. Moreover, the function wizard will also be helpful in pointing and correcting errors while working with functions.

EXERCISE

1. Explain working of function wizard in brief.
2. List any three mathematical functions of your choice with proper example of each.
3. List any three statistical functions of your choice with proper example of each.
4. List any three decision making functions of your choice with proper example of each.
5. List any three time and date functions of your choice with proper example of each.
6. List any three string functions of your choice with proper example of each.
7. Explain If function in Calc in detail with suitable example.
8. List the three logical functions with proper example of each.
9. Choose the correct option from the following :
 - (1) In which of the following ways we can enter a function in Calc ?
 - (a) Directly typing function name in a cell
 - (b) Function wizard or selecting from tool bar
 - (c) A and B both
 - (d) Depends on functions
 - (2) A function can start with which of the following options ?
 - (a) '=' sign
 - (b) Alphabets
 - (c) Numbers
 - (d) Any of these
 - (3) Which of the following is not a logical function?
 - (a) OR
 - (b) AND
 - (c) NOT
 - (d) PROPER
 - (4) Which date and time function of Calc you will use to find out whether the given year is leap year or not ?
 - (a) DATE
 - (b) TIMESTAMP
 - (c) YEARS
 - (d) YEARDIFF

LABORATORY EXERCISE

1. Consider your marks of different subjects and prepare a simple mark sheet containing school name, student's number, student's name, standard and marks. Also find percentages and class (first class, pass, fail etc.) from the marks provided. You may use your last year school report card.
 2. Simulate the electricity bill of your house using Calc.
 3. Ask your family member or neighbour about the loan information, if any. Use NPER function to calculate the time to repay the loan.
 4. Write three numbers in three different cells. Use values in such a way that it forms a valid date. Use date function to convert these values into a valid date.
 5. Take a date and find out that whether the year is leap year or not.
 6. Try to find out information spreadsheets packages for mobiles, iphones and ipads. Find out name of the package/spreadsheet that offers this facility.





Charts in Calc

With the advancement of technology and the increased need of technical analysis, the use of charts has increased. Utility of chart is one of the factors that make the spreadsheet packages widely popular. Charts are illustrations of a business situation with ability to display a significant amount of information in attractive way. A chart is also considered as a graphical presentation of numerical data.

To prepare a chart, one needs to have basic knowledge about various charts, concept underlying the chart patterns and their applications. That is, just by selecting a chart type and creating a chart through a spreadsheet package will not contribute much in business. You must have understanding of the chart concepts such as how the chart would be helpful in your business. One must also know when to use a particular type of chart, what information it conveys, what support it provides and which type of decisions can be made using such chart.

Type of Charts

According to the nature of applications and requirements, a chart type should be selected. Charts are used for various purposes such as telling history; evaluate alternatives, presenting trends or find-out exceptional cases. That means, an incorrect choice of chart can lead to poor representation of the concept and generates misunderstanding. On the other hand, correct choice of chart can lead to right and faster decisions. Popular reasons why we should go for a chart are mentioned below :

- For comparisons;
- For demonstration of distribution;
- For understating of situation;
- For analysis of trend over the time;
- For investigating deviations; and
- For identifying and understanding the relationship between entities;

Follow the steps given below to prepare chart in any situation :

- Make it clear what you want to say;
- Collect and arrange data;
- Remove invalid data;
- Determine the best chart type; also check the data you have collected for the chart are sufficient or not;
- Prepare the chart; and
- Format the chart.

As stated, the very first step about the chart making after identifying its objective; is to prepare data. Once data are entered in a spreadsheet document, you can view it in a graphical manner.

Inserting a Chart

To insert a chart in a worksheet do the following.

- Select the data range in a spreadsheet;
- Select **Insert → Chart**;

Consider you have given some surprise quizzes and class tests from January to April in an academic year and obtained marks as indicated table 8.1.

	Quiz marks	Test marks
January	35	27
February	28	30
March	37	42
April	31	33

Table 8.1 : Marks of a student

Open a spreadsheet document and enter the data as shown in table 8.1. Do not change the order of the data, otherwise when we plot a chart, it will give different result. Also, save the data frequently.

Once you have entered all the data, select the data range. Here the data ranges from cell A1 to cell C5. Keep the data range selected, now choose option **Insert**. A vertical submenu appears. Select **Chart** option from it. The operation sequence is shown in figure 8.1.

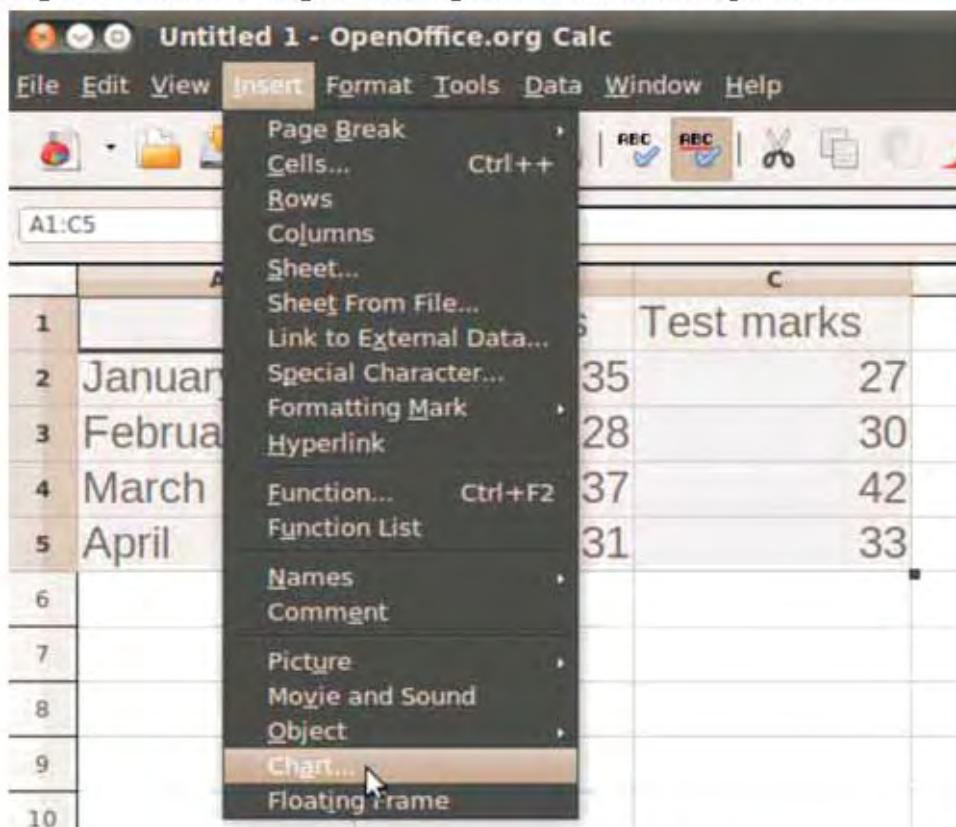


Figure 8.1 : Inserting a chart