iT Series

3rd Edition

A Practical Approach to

WEB PROGRAMMING

Complete Projects Included

- Complete Coverage of:
  - HTML
  - CSS
  - JavaScript
  - VBScript
  - FrontPage
  - ASP
- More than 250 Practices
- More than 300 Short Questions
- More than 425 Multiple Choice
- More than 125 Fill in the Blanks
- More than 175 True / False
- MS Access & SQL Included

Tariq Mahmood
Ahsan Raza
Imran Saeed
Muhammad Naeem

Reviewed by
Tasleem Mustafa

WWW.ITECVIEW.COM©

IT Series

A Practical Approach to

WEB PROGRAMMING

(3rd Edition)

2016-09-B145

Written by

Tariq Mahmood
MSc (CS), JCP, MCP

Ahsan Raza Sattar
MS (CS), MSc (CS), MBA, OCP

Imran Saeed
MSc (CS), JCP, MCP

Muhammad Naeem
MCS

Reviewed by
Tasleem Mustafa
MS (CS), M.Sc. (CS), M.Sc. (Math), MIS (USA), OCP
Chairman, Deptt. of Computer Science,
University of Agriculture, Faisalabad.

24 SEP 2019

Kitab Markaz, Faisalabad

WWW.ITECVIEW.COM©

All Rights Reserved with IT Series

No part of this book may be reproduced in any form or any means without the written permission from **IT Series.**

Ch. Ashfaq Shahid
Advocate, High Court

Published by: IT Series (0321-6615656)

Edition: 3rd Edition

Price Rs. 335/-

Can be had from:

Kitab Markaz
Aminpur Bazar, Faisalabad.
Ph: 041-2642707

Imtiaz Book Depot
Urdu Bazar, Lahore.
Ph: 042-37235944

WWW.ITECVIEW.COM©

Preface

Internet and the Web have fascinated the people in all fields of life. The use of Internet has become popular in commerce, education, banking and industry. This increase has created a demand for the professionals who can develop effective websites for organizations in the country to provide interaction between the industry and the customers.

The basic aim of this book **"A Practical Approach to Web Programming"** is to provide the knowledge of Internet and the Web with strong practical expertise. The book provides detailed background knowledge of the history and infrastructure of Internet. It also covers all technologies that are used to develop effective, productive and dynamic websites. The book is written for all levels of students i.e. beginners to professionals. It is effectively helpful for all who wish to acquire proficiency in web programming.

An important aspect of this book is its effectiveness for examinations of all types. We have included hundreds of short questions, multiple choice, fill in the blanks and true/false questions. These questions will enable the students to get high marks in any type of examination.

There are twenty one chapters in the book. Each chapter contains a large number of pictures and tables to explain the concepts. We have included four comprehensive **ASP projects** with database connectivity. The projects will be very helpful for the students to develop effective websites.

Readers are welcome to send suggestions for improvements of the book by sending email to us at **comments@itseries.edu.pk**. You can also visit our website **www.itseries.com.pk** for any interaction and latest information about the book.

Authors

WWW.ITECVIEW.COM©

Part 1: Introduction

1. Internet and World Wide Web 3

1.1 Internet 3
1.1.1 Uses of Internet 3
1.1.2 Working of Internet 3
1.1.3 Impacts of Internet on Society 4
1.2 History of Internet 4
1.3 Connection to the Internet 5
1.4 Addressing Scheme 6
1.4.1 IP Addressing 6
1.4.2 DNS Addressing 6
1.5 Services of Internet 7
1.5.1 World Wide Web (WWW) 7
1.5.2 Email 8
1.5.3 Social Networking 9
1.5.4 VoIP 9
1.5.5 Mailing List 9
1.5.6 News Groups 9
1.5.7 E-Commerce 9
1.5.7.1 Role of E-Commerce 10
1.5.7.2 Types of E-Commerce 11
1.5.7.3 Advantages of e-commerce 11
1.5.7.4 Functions of e-commerce Web sites 11
1.5.8 TELNET 12
1.5.9 FTP 12
1.5.10 Internet Relay Chat (IRC) 12
1.5.11 Instant Messaging (IM) 12
1.6 Basic Terminologies of Internet 12
1.7 Advantages of Internet 14
1.8 Disadvantages of Internet 16
1.9 Benefits of Internet for Students 17
1.10 Benefits of Internet for Business Organization 17
1.11 Web Browser 18
1.11.1 Functions of Web Browser 18
1.11.2 Types of Browsers 19
1.12 Web Server 19
1.13 Microsoft Internet Explorer 19
1.13.1 Main Window of Internet Explorer 19
1.13.2 AutoComplete Feature of Internet Explorer 20
1.13.3 Saving a Web Page 21

WWW.ITECVIEW.COM©

1.14 Web Directories 21
1.15 Search Engines 21
1.15.1 Searching the Web 21
1.16 Types of Websites 22
1.17 Multimedia on Web 24
1.18 Netiquette 25
1.19 Web Page Program Development 25
1.19.1 Roles in Web Site Development Team 26
1.20 Scripting Languages 27
1.21 Web Hosting 29
1.21.1 Web Hosting Services 29
1.21.2 Types of Web Hosting 29
1.21.3 Web hosting technologies 29
1.22 Web Publishing 30
1.23 Cookie 30
1.23.1 Types of Cookies 30
1.23.2 Uses of Cookies 30
1.23.3 Browser Setting for Cookies 31
1.23.4 Privacy Concerns about Cookies 31
1.24 TCP/IP 31
1.25 Intranet and Extranet 31
1.26 Web 2.0 32
1.27 Web 3.0 32

Short Questions 33
Multiple Choices 40
Fill in the Blanks 45
True/False 46

Part 2: Hyper Text Markup Language

2. Introduction to HTML 49

2.1 HTML 49
2.1.1 Tags and Attributes 49
2.1.2 Benefits of HTML 50
2.1.3 Limitations of HTML 50
2.2 Web Page Creation using HTML 50
2.2.1 Basic Structure of HTML Document 50
2.3 Headings 52
2.4 Paragraphs 53
2.5 Line Breaks 54
2.6 Horizontal Line 55
2.7 Text Formatting 55
2.7.1 Fonts 56

WWW.ITECVIEW.COM©

2.7.2 <BIG> and <SMALL> Tags....57
2.7.3 Preformatted Text....58
2.8 Spaces....58
2.9 <DIV> Tag....59
2.10 <SPAN> Tag....60
2.11 <MARQUEE> Tag....60
2.12 HTML Comments....62
2.13 XML....62
2.14 XHTML....62
2.14.1 Changes in XHTML from HTML....62
2.14.2 Flavors of XHTML 1.0....62
2.14.3 Validating XHTML Documents....63
2.14.4 Differences between XHTML and HTML....64

Exercise Questions....64

---

3. Lists, Images and Tables....65

3.1 Lists....65
3.1.1 Ordered Lists....65
3.1.2 Unordered List....67
3.1.3 Definition Lists....68
3.1.4 Nested Lists....69
3.2 Images in HTML....73
3.2.1 Using Images in HTML....74
3.3 Tables....76
3.3.1 The <TR> Tag....77
3.3.2 The <TD> Tag....77
3.3.3 The <CAPTION> Tag....87

Exercise Questions....88

---

4. Hyperlinks and Image Maps....90

4.1 Hyperlinks....90
4.1.1 Types of Hyperlinks....90
4.2 Creating Hyperlinks in HTML....90
4.2.1 Links to another Web Page....91
4.2.2 Links to the Page of other Website....93
4.2.3 Links to other Parts of the Same Page....94
4.2.4 Email Hyperlinks....96
4.2.5 Images as Hyperlinks....97
4.2.6 Hyperlinks in Lists....100
4.2.7 Hyperlinks in Tables....100
4.3 Image Maps....101

WWW.ITECVIEW.COM©

Exercise Questions 103

5. Frames and Forms 104

5.1 Frames 104
- 5.1.1 Advantages of Frames 104
- 5.1.2 Disadvantages of Frames 104
- 5.1.3 Nested Frames 110

5.2 Forms 113
- 5.2.1 Uses of Forms 113

5.3 Form Elements 114
- 5.3.1 Text Field 114
- 5.3.2 Password Field 115
- 5.3.3 Radio Button 115
- 5.3.4 Checkbox Field 116
- 5.3.5 Select Field 117
- 5.3.6 Text Area Field 119
- 5.3.7 Button 120
- 5.3.8 File Upload Field 120
- 5.3.9 Submit Button 121
- 5.3.10 Reset Button 121

Exercise Questions 126

Short Questions 128
Multiple Choices 134
Fill in the Blanks 140
True/False 141

Part 3: Cascading Style Sheets

6. Cascading Style Sheets 145

6.1 DHTML 145
- 6.1.1 Benefits of DHTML 145
- 6.1.2 Disadvantages of DHTML 146

6.2 Cascading Style Sheets 16
- 6.2.1 Cascading Order 147
- 6.2.2 Benefits of Style Sheet 147
- 6.2.3 Parts of CSS 147

6.3 Types of Style Sheets 148
- 6.3.1 External Style Sheets 148
- 6.3.2 Internal Style Sheets 149

WWW.ITECVIEW.COM©

6.3.3 Inline Styles 150
6.4 CSS and Text 151
6.4.1 Special Effects 153
6.4.2 Text Layout 153
6.5 Lists 155
6.6 CSS and Hyperlinks 156
6.7 CSS and Backgrounds 157
6.7.1 Background Colors 157
6.7.2 Background Images 157
6.7.3 Background Control 157
6.8 CSS and Borders 159
6.8.1 Parts of Border 160
6.9 CSS and Tables 161
6.10 CSS and Scrollbars 162
6.11 CSS and Cursors 163
6.12 CSS Class 165

Exercise Questions 166

Short Questions 167
Multiple Choices 168
Fill in the Blanks 170
True/False 171

Part 4: JavaScript

7. Introduction to JavaScript 174

7.1 Scripts 174
7.1.1 Server-Side Scripts 174
7.1.2 Client-Side Scripts 174
7.2 JavaScript 175
7.2.1 Advantages of JavaScript 176
7.2.2 Uses of JavaScript 176
7.3 Running JavaScript 177
7.4 Data Types 177
7.5 Variables 178
7.5.1 Variable Declaration 178
7.5.2 Variable Initialization 179
7.6 Literals 180
7.6.1 Floating-Point Literals 180
7.6.2 Integer Literals 180
7.6.3 String Literals 180
7.7 Type Conversion 182

WWW.ITECVIEW.COM©

7.8 Expressions 182
7.9 Operators 183
7.9.1 Assignment Operators 183
7.9.2 Arithmetic Operators 183
7.9.3 Operator Precedence 187
7.9.4 Operator Associativity 188
7.10 Type Casting 189
7.10.1 Implicit Type Casting 189
7.10.2 Explicit Casting 189

Exercise Questions 189

---

8. Conditions and Loops 191

8.1 Conditional Statements 191
8.1.1 Relational Operators 191
8.1.2 Relational Expression 192
8.2 if Statement 192
8.3 if...else Statement 193
8.4 Nested "if" Statement 196
8.5 Logical Operators 197
8.6 switch Statement 200
8.7 Conditional Operator 202
8.8 Loops 204
8.9 while Loop 204
8.10 do-while Loop 206
8.11 for Loop 208
8.12 "continue" Statement 211
8.13 "break" Statement 212
8.14 Nested Loop 213

Exercise Questions 214

---

9. Functions and Arrays 215

9.1 Functions 215
9.1.1 Function Declaration 215
9.1.2 Function Call 216
9.1.3 Passing Parameters to Functions 217
9.1.4 Returning Value 218
9.2 Local Variables 219
9.3 Global Variables 220
9.4 Built-in Functions 221
9.4.1 Dialog Boxes 221

WWW.ITECVIEW.COM©

9.4.2 Conversion Functions 223
9.5 Arrays 227
9.5.1 Array Declaration 228
9.5.2 Assigning Values to Array 228
9.5.3 Retrieving Values from Arrays 229
9.5.4 Passing Array to Function 229
9.6 Manipulating Arrays 230
Exercise Questions 231

10. JavaScript Object Model 232
10.1 JavaScript Object Model 232
10.2 Objects 233
10.2.1 Browser Objects 233
10.2.2 Arrays of Objects 233
10.3 Form Elements 234
10.4 Events 234
10.4.1 Event Handlers 234
10.5 Forms and Events 235
10.5.1 Button Element 235
10.5.2 Text Element 237
10.5.3 Password Element 239
10.5.4 Checkbox Element 239
10.5.5 Radio Element 241
10.5.6 TextArea Element 242
10.5.7 Select and Option Elements 242
10.6 Form Validation 244
10.6.1 Common Validation Types 244
10.7 Advanced JavaScript 252
10.7.1 Date Object 252
10.7.2 Image Object 253
10.7.3 Window Object 254
10.7.4 History Object 257
10.7.5 String Object 257
Exercise Questions 258

Short Questions 259
Multiple Choices 268
Fill in the Blanks 276
True/False 278

WWW.ITECVIEW.COM©

Part 5: VBScript

11. Introduction to VBScript 281

11.1 VBScript 281
- 11.1.1 Advantages of VBScript 281
- 11.1.2 Uses of VBScript 282
- 11.1.3 Structure of VBScript 283
- 11.1.4 The Document Object 283

11.2 Variables 283
- 11.2.1 Rules for Variable Names 284
- 11.2.2 Variable Declaration 284

11.3 Mathematical Operators 285
- 11.3.1 Operator Precedence 286

11.4 Concatenation 287

11.5 Comments 287

11.6 Input and Output 287
- 11.6.1 Input Box 287
- 11.6.2 Message Box 288

Exercise Questions 294

---

12. Conditions and Loops 295

12.1 Control Structure 295
- 12.1.1 Types of Control Structures 295

12.2 Relational Operators 297

12.3 Conditional Statements 297
- 12.3.1 If...Then Structure 298
- 12.3.2 If...Then...Else Structure 299
- 12.3.3 Extended If...Then...Else Structure 302
- 12.3.4 Compound Comparison 304
- 12.3.5 Logical Operators 304
- 12.3.6 Nested If...Then...Else Structure 306
- 12.3.7 Select...Case Structure 308

12.4 Loops 310
- 12.4.1 While...Wend Loop 310
- 12.4.2 Do...While...Loop 312
- 12.4.3 Do...Until...Loop 317
- 12.4.4 Do...Loop...While 318
- 12.4.5 Do...Loop...Until 320
- 12.4.6 For...Next Loop 321
- 12.4.7 Early Termination of Loops 325
- 12.4.8 Nested Loops 326

WWW.ITECVIEW.COM©

Exercise Questions 227

13. Procedures, Functions and Arrays 328

13.1 Procedures 328
13.1.1 Passing Parameters to Procedures 329
13.2 Functions 329
13.2.1 Returning a Value from Function 329
13.3 Built-in Functions 330
13.3.1 Date & Time Functions 330
13.3.2 Numeric Functions 332
13.3.3 Informational Functions 333
13.3.4 Conversion Functions 334
13.4 Arrays 336
13.4.1 Advantages / Uses of Arrays 336
13.4.2 Declaring One-Dimensional Arrays 337
13.4.3 Initializing Array 337
13.4.4 Searching in Arrays 341

Exercise Questions 341

14. Events and String Handling 342

14.1 Document Object Model 342
14.2 Objects 343
14.2.1 Browser Objects 343
14.2.2 Arrays of Objects 343
14.3 Form Elements 344
14.4 Events 344
14.4.1 Event Handlers 344
14.5 Forms and Events 345
14.5.1 Button Element 346
14.5.2 Text Element 351
14.5.3 Password Element 352
14.5.4 Checkbox Element 354
14.5.5 Radio Element 354
14.5.6 TextArea Element 355
14.5.7 Select and Option Elements 355
14.6 String Handling 357
14.7 Form Validation 365
14.7.1 Common Validation Types 365

Exercise Questions 373

WWW.TECVIEW.COM©

Short Questions 374
Multiple Choices 376

Part 6: Microsoft FrontPage

15. Web Designing using Microsoft FrontPage 379

15.1 Microsoft FrontPage 379
15.1.1 FrontPage Environment 379
15.1.2 Screen Layout 379
15.2 Designing the Documents 380
15.3 Hyperlinks 387
15.3.1 Types of Hyperlinks 387
15.4 Tables in FrontPage 389
15.5 Frames 392
15.5.1 Hyperlinks in Frames 395
15.6 Graphics in FrontPage 396
15.7 Creating Web in FrontPage 400
15.8 Web Site Publishing in FrontPage 401
15.9 Forms 409
15.10 Cascading Style Sheets in FrontPage 416
15.10.1 Inline Styles 416
15.10.2 Embedded Style Sheets 416
15.10.3 Linked Styles Sheets 418

Part 7: Active Server Pages

16. Web Applications with ASP 423

16.1 Web Applications 423
16.1.1 Web Application Requirements 424
16.2 Web Application Development Cycle 425
16.3 Client Server Model 426
16.3.1 Components of Client/Server Model 426
16.3.2 Single-tier 426
16.3.3 Two-tier 426
16.3.4 Three-tier 427
16.3.5 N-tier 428
16.4 Static and Dynamic Pages 428
16.5 Active Server Pages 429
16.5.1 Advantages of ASP 430
16.5.2 Requirements for ASP 430

WWW.ITECVIEW.COM©

16.6 Personal Web Server (PWS) 431
16.6.1 Installing PWS from Windows 98 CD 431
16.7 IIS 434
16.7.1 IIS Version 434
16.7.2 IIS Features 435
16.7.3 IIS Architecture 435
16.7.4 Protocol Listeners 436
16.7.5 World Wide Web Publishing Service (WWW service) 436
16.7.6 Windows Process Activation Service (WAS) 436
16.7.7 HTTP Request Processing in IIS 7 437
16.7.8 Installing IIS on Windows XP 438
16.7.9 Installing IIS on Windows Vista 438
16.7.10 Installing IIS on Windows 7 439
16.7.11 Installing IIS on Windows Server 2003 439
16.8 Embedding ASP in HTML 439
16.9 Variables 440
16.10 Mathematical Operators 442
16.11 Comments 443

17. Programming Constructs in ASP 444

17.1 Conditional Statements 444
17.1.1 If...Then Structure 444
17.1.2 Relational Operators 444
17.1.3 If...Then...Else Structure 446
17.1.4 Compound Comparison 447
17.1.5 Logical Operators 447
17.1.6 Nested If...Then...Else Structure 448
17.1.7 Select...Case Structure 449
17.2 Loops 450
17.2.1 While...Wend Loop 450
17.2.2 Do...While...Loop 451
17.2.4 Do...Until...Loop 451
17.2.5 For...Next Loop 452
17.2.6 Early Termination of Loops 453
17.2.7 Nested Loops 454
17.3 Procedures 454
17.3.2 Passing Parameters to Procedures 455
17.4 Functions 456
17.4.1 Returning a Value from Function 456
17.4.2 Exit Sub & Exit Function 457
17.5 Arrays 457
17.5.1 Searching in Arrays 459

WWW.TECVIEW.COM©

Exercise Questions 459

18. ASP Objects 460

18.1 Client-Server Communication 460
18.2 ASP Objects 460
18.2.1 Request Object 460
18.2.2 Response Object 462
18.2.3 The "global.asa" File 470
18.2.4 Application Object 470
18.2.5 Session Object 472
18.2.6 Server Object 476
18.3 Server Side Includes (SSI) 479

19. Microsoft Access and SQL 481

19.1 Database 481
19.1.1 Database Management System 481
19.1.2 Relational Databases 481
19.2 Primary Key 481
19.3 Foreign key 482
19.4 Referential Integrity 482
19.5 Database Creation in MS Access 483
19.6 Table Creation in MS Access 485
19.7 Data types 486
19.8 Views in MS Access 487
19.9 Opening a Table 487
19.10 Adding Data to a Table 488
19.11 Deleting a Record from Table 488
19.12 SQL 489
19.12.1 The SELECT Statement 489
19.12.2 The WHERE Clause 490
19.12.3 The INSERT INTO Statement 491
19.12.4 The Update Statement 492
19.12.5 The DELETE Statement 492
19.13 Functions 495
19.13.1 Types of Functions 495

20. Database Connectivity in ASP 496

20.1 Database Connectivity in ASP 496
20.2 Open Database Connectivity 496
20.3 OLE DB 497

WWW.ITECVIEW.COM©

497
20.4 ActiveX Data Object 497
20.5 The ADO Object Model 497
20.6 Connection Object 497
20.6.1 Properties of Connection Object 498
20.6.2 Collections of Connection Object 499
20.6.3 Methods of Connection Object 500
20.7 Recordset Object 500
20.7.1 Properties of Recordset Object 502
20.7.2 Collections of Recordset Object 502
20.7.3 Methods of Recordset Object 515
20.8 Command Object 516
20.8.1 Properties of Command Object 516
20.8.2 Collections of Command Object 516
20.8.3 Methods of Command Object 518

21. ASP Projects 518
Project 1: User Registration 521
Project 2: Online Survey Poll 524
Project 3: Online Result 527
Project 4: Login Window 534

Short Questions 537
Multiple Choices 542
Fill in the Blanks 543
True/False 543

WWW.ITECVIEW.COM©

Part - I

Introduction

Includes:

Chapter 1 Internet & World Wide Web

WWW.ITECVIEW.COM©

1

Internet & World Wide Web

1.1 Internet

Internet is a huge collection of computers all over the world that are connected to one another. It is a global network of computers. These computers are connected through different telecommunications links like:

- Phone lines
- Fiber optics lines
- Satellites and wireless connections

Internet is used to find information stored on the computers called **hosts** or **servers**. These computers use a common protocol called TCP/IP for communication. **TCP/IP** stands for **Transmission Control Protocol / Internet Protocol**. Each computer connected to Internet can act as host. A host computer provides information to the people. Today, more than 100 million users around the world connect to Internet for different purposes. No government or organization is the owner of Internet. Many people, organizations, universities and research agencies participate to manage the Internet.

1.1.1 Uses of Internet

Some uses of the Internet are as follows.

- To access information, news, research, and educational material
- To conduct business
- To access sources of entertainment such as online games, magazines etc.
- To shop for goods and services
- To meet and talk with people around the world in discussion groups or chat rooms
- To access other computers and exchange files
- To send messages and receive messages from other connected users

1.1.2 Working of Internet

The Internet is a collection of millions of computers. These computers are linked together on a computer network. The network is used by the computers to communicate with one another. A personal computer can be connected to Internet by using phone lines, DSL or cable modem. These devices communicate with the server of Internet service provider.

WWW.ITECVIEW.COM©

A user computer is usually connected with ISP using phone line. ISPs are connected to larger ISPs. Several main transmission media carry the heaviest amount of traffic on Internet. These carriers of network traffic are collectively called **Internet backbone.** Backbones are connected through fiber optic lines, submarine cables or satellite links. In this way, every computer on the Internet is connected to every other computer.

The computers connected to the Internet exchange data and information using servers and clients. A client computer accesses data, information and services from different servers. The server computers are also called host computers.

1.1.3 Impacts of Internet on Society

The impacts of the Internet on the society are as follows:

- **Globalization:** Internet has converted this world into a global village. People are connected with one another using the Internet. It has provided more opportunities of interaction. It also enables people to understand the views of one another.
- **Flow of Information:** Internet contains information on all types of topics. People can search information on any topic. Search engines are used to search information on Internet. Easy access to information has enabled people to be informed.
- **Better Understanding:** Internet is an easy way of communication among people. They can exchange their views. It has created better understanding. People can understand the views and thinking of one another. It helps in creating harmony and peace in the world.
- **Comfort in Life:** The use of Internet has provided many comforts for society. People can perform their duties easily. They can sell and buy goods online. They can run their business using Internet.
- **Knowledge:** Internet is a source of huge information. People can get information from websites. They can interact with different people, scholars and learned persons to get knowledge.

1.2 History of Internet

The work on Internet was started in 1960s during the Cold War of Russia and America. America wanted to communicate with its Armed Forces. A network of four computers was developed in the beginning for this purpose.

ARPANET

During Cold War, Russia launched **Sputnik** satellite and America developed a network known as **ARPANET.** It was developed for **Advance Research Project Agency (ARPA).** ARPA worked on a project to send information to armed forces at long distances. Different organizations and universities also involved in this research.

Paul Baran

In 1964, Paul Baran published a paper on Distributed Communication Networks. It was a theoretical idea for data transfer. It included:

- Decentralization of data storage
- Digital Packets
- Routes for data transfer

NCP Protocol

Protocol is a collection of rules. In 1970s, NCP (Network Control Protocol) was developed. It was the first standard protocol for data transfer.

DARPA

DARPA stands for Defense Advanced Research Projects Agency. It was a new name of ARPA. Now 37 computers were connected together. DARPA started work to share data not only on single network but also among different networks. Stanford Research Institute took part in this research.

TCP/IP

A team of Stanford Research Institute developed a protocol called TCP/IP. It stands for Transmission Control Protocol/Internet Protocol. It was designed to share and transfer data among different networks.

BITNET & CSNET

BITNET stands for Because It's Time Network and CSNET stands for Computer Science Network. These were established for non-military persons. These networks were used for academic and research purposes. In these networks, IBM computers and university computers were connected together.

World Wide Network

Soon the idea of Internet came out. Different universities and research institutes started work on a global network. In 1986, NSFNet was developed. It connected academic researched of different counties together.

Internet

Internet is now the largest network in the world. Millions of computers are connected together through different medium. People can share information, data. Any computer with a modem and telephone line can become a part of the Internet.

1.3 Connection to the Internet

Anyone can connect to the Internet by using the services of any Internet Service Provider. Following things are required to establish a connection to Internet:

- **Computer:** Any computer can be used for connection to Internet.
- **Modem:** Modem is an electronic device that is used to transfer data over Internet. It links the computer with the Internet using telephone line.
- **Dialup Software:** Dialup software is used to establish a connection between a personal computer and a server of an ISP.
- **ISP Connection:** ISP stands for Internet Service Provider. It provides the link between the computer and the Internet. A person has to get an account of an ISP for establishing a connection to Internet.
- **Web Browser:** Internet browser is software that is used to view web pages. Microsoft Internet Explorer and Netscape Communicator are popular web browsers.

More high-speed connections are also available to connect to the Internet such as DSL, ISDN and cable television Internet services. **DSL** stands for **digital subscriber line** and **ISDN** stands for **Integrated Services Digital Network**. These technologies provide high-speed Internet connections using telephone lines. **Cable modem** provides highspeed Internet connections through cable television network. These technologies are always connected to the Internet when the computer is running. The users do not have to dial a connection to connect to the Internet, which is necessary in case of **dial-up access**.

1.4 Addressing Scheme

Internet addresses are used to access different computers on the Internet. Each computer on the Internet has a unique address. This address is used to establish a connection with that computer. There are two types of addressing schemes on the Internet:

1.4.1 IP Addressing

IP address is a unique number that refers to a computer connected to the Internet. IP stands for **Internet Protocol**. This address is a 32-bit number. It consists of four numbers separated by period. Each number is between 0 and 255. These numbers are known as **octets**. An example of IP address is 127.80.144.30. Normally, first number indicates a network and the last number indicates a specific computer.

A server on the Internet has a static IP address. It does not change very often. When a computer connects to the Internet, it also gets an IP address by the ISP. This IP address is used only during the session and is not static.

1.4.2 DNS Addressing

IP address is difficult to remember for the users. All servers on the Internet also have human-readable names. It is known as **domain name**. A domain name consists of text instead of number. It is easier to remember a domain name than IP address.

For example, **www.hotmail.com** is a permanent and human-readable name. It is easier to remember than IP address. It actually has two parts:

- Host name
- Domain

The domain represents the institution that uses the address. The domain names are called top-level domains. Each domain name has a **top-level domain** that indicates the type of organization associated with domain name. The domain names outside United States also includes the country code. The country code of Pakistan is **pk** e.g. www.hec.gov.pk

Internet Corporation for Assigned Names and Numbers (ICANN) is a group that assigns and controls the top-level domains.

A method used to store domain names and corresponding IP addresses is called **domain name system (DNS)**. When the user specify a domain name, **DNS server** converts it into its corresponding IP address. DNS server is associated with an Internert access provider.

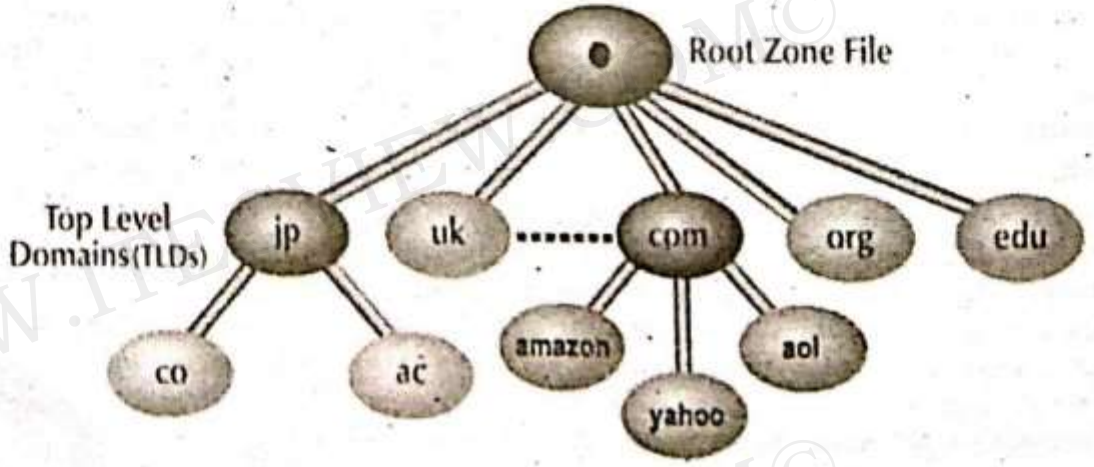


Figure: Top Level Domains

The top-level domains are as follows:

Domain	Type
com	Commercial organizations
edu	Educational institutions
gov	Government departments
mil	Military organizations
net	Network providers
org	Non-profit organizations
museum	Accredited museums
biz	Business organizations
info	Organizations or individuals providing general information
name	Individuals of families
pro	Professional like doctors, engineers
aero	Aviation community members
coop	Cooperative societies and unions

1.5 Services of Internet

Following are some important services of the Internet:

1.5.1 World Wide Web (WWW)

WWW stands for **World Wide Web**. It is also called **Web**. It provides the facility to publish information on the Internet. It is a collection of documents or web pages stored on computers permanently connected with Internet around the world.

A web page is a document that may include text, graphics, sound and video etc. These web pages are connected to one another using **hyperlinks**. Web pages are also known as **hypertext documents**. Anyone can view web pages through a web browser. A collection of related web pages is called **website**. Each website has a unique address. Websites are stored on a host computer on the Internet. These computers are known as **web server**. The process of launching a web page is called **publishing the page**.

Purpose and Uses of WWW

- **Advertisement:** Manufacturers use WWW to provide information of their products.
- **Shopping:** It is possible for user to buy an item using WWW. You can see products and prices, and even order them by using your credit card etc.
- **Flight Information:** It is used to provide the latest information on flights arrivals and departures.
- **Television Station:** WWW provides information about programs, the latest news and special events. Some television show different events live on their websites.
- **Celebrities, Sportsman and Film Actors:** Important celebrities can contact their fans, provide information about them and send their pictures.
- **Government Authorities:** Government authorities can provide information about services and facilities for the people.
- **News:** Most newspapers have their website. Latest news and interviews are provided on these websites.
- **Encyclopedia Publisher:** Online encyclopedias are available. You can find detailed information on any topic in the world.

- **Search Engines:** Search engine is a website that provides the facility to find required websites on a particular topic. You can find thousands of websites about any topic.
- **Medical:** Many sites in the world provide detailed information on diseases. You can also consult a panel of on-line doctors to get the advise about any medical problem.

1.5.2 Email

Email stands for **Electronic Mail**. It is the exchange of messages and files through Internet. Message can be in the form of graphics, sounds, video clips or simple text. It is a fast way of sending messages anywhere in the world. The sender and receiver may be sitting in the same building or anywhere in the world. One email message can be sent to multiple recipients. Email facility is provided by many Internet Service providers or specialized websites. **Yahoo, Hotmail** and **Gmail** are examples of popular email providing websites.

Working of Email

An email program is used to create, send and receive emails. It is also known as **email client**. When the user sends an email message, the computer connects to an email server and transmits a copy of the message to that server.

Email server is a host computer on the Internet that is used to send and receive emails. The email server receives the message and finds out the email server where that email is to be sent. It connects to that server and transmits another copy of the message.

Email Attachment

An email message can also contain attachments. **Attachment** is a process of sending connecting files with email message. Any type of file can be attached to email such as documents, spreadsheets, audio and video etc.

MIME stands for **Multipurpose Internet Mail Extension**. It is an Internet protocol that supports email attachments. Different websites provide different file size that can be attached with an email message.

Advantages and Uses of Email

Some advantages of email are as follows:

- It is very fast and timely.
- It is very cheap and inexpensive.
- People can send and receive message anywhere in the world.
- People can share your information among different people.
- It is possible to send and receive email through mobile phone.
- Email gives us the facility to send pictures, sounds and videos.

Disadvantages of Email

Some limitations of email are as follows:

- Email does not provide high privacy. An email passes from one computer system to another through different networks. The system administrator of a network or a hacker can read the contents of emails.
- Junk email is an unwanted email. It is also known as spam. A person may send such email that the receiver does not want to read. A large number of junk emails may fill the space of email account.
- An email message cannot express emotions properly. The receiver cannot view the facial expression or voice of the sender.

- Email is read when the user connects to the Internet and checks email account. An urgent message may be ignored or delayed.
- Email may not be delivered successfully due to communication errors.
- Most of the viruses are spread through email messages. Virus can spread if the user opens an email containing a virus.

Email Address

Every email account has unique address. An email address usually has two parts which are separated by the symbol @. These parts are as follows:

- User ID
- Identity of email service

Example

myemail@hotmail.com

In this example:

myemail myemail is called user ID. It is used to sign into email service.

@ It is "at" sign. It separates account ID from name of email service provider.

hotmail.com It indicates the email service provider.

1.5.3 Social Networking

Social networking is a way of building online communities. The people use social networking websites to interact with one another. **MySpace** is an example of social networking sites. Professionals also use social networking sites such as **LinkedIn**.

1.5.4 VoIP

VoIP stands for **Voice over Internet Protocol**. It is also called **Internet telephony**. It is used to speak to other users over the Internet. **VoIP** uses the Internet instead of telephone network to connect a user to other user. It requires high-speed Internet connection like DSL.

1.5.5 Mailing List

A **mailing list** is a collection of email names and addresses given a single name. An email message sent to a mailing list is received by every person on the list. Different people discuss different topics and distribute their ideas to other people by email. Softwares are available to administrate mailing lists. The most common program is **listserv**.

1.5.6 News Groups

A **newsgroup** is a discussion group on the Internet. People exchange information on a vast range of topics such as news, business, science and computer. A user sends message to the newsgroup to participate in discussion. Other users in the newsgroup read and reply to *the message*. A software is required to obtain articles from **news server**. A news server is a host computer that exchanges articles with other servers on the Internet. These servers use **Network News Transfer Protocol (NNTP)** to communicate. The users have to subscribe on a news group to view articles on a specific topic.

1.5.7 E-Commerce

E-Commerce stands for electronic commerce. E-Commerce means to carry out financial and business dealings using Internet. A person can deal with his customers throughout the world. People can buy and sell good on Internet. Even payments can be made using Credit Card numbers etc. The trend of E-Commerce is rapidly becoming popular.

1.5.7.1 Role of E-Commerce

The role of e-commerce in daily life is becoming very important. E-commerce can be used in the following ways:

1. Online Education

Online education is becoming very popular. Different types of interactive tutorials are available on the Internet. The students can browse these online books and tutorials or can download them after purchasing. Some websites provide online lectures for the students.

2. Electronic Banking

Many banks are now introducing electronic banking. Using your computer, you can connect to the bank's computer system via the Internet and control your daily financial dealing from home. It reduces the staff and building of banks. Many customers pay their bills from their bank accounts using this facility.

3. Electronic Shopping

It has become very easy for the people to shop from home using Internet. Different manufacturers present their products on the Internet. People can browse the website, place an order and even make a payment using credit card. It has made shopping very easy.

4. Job Search

Internet is used to search different types of jobs all over the world. Many websites are developed that provide information to the people about job vacancies. Job seekers can also apply for the job using Internet.

5. Conducting Auctions

Many websites provide the facility of auction. People participate in the auction to purchase a product. They can also pay the price using their credit cards etc. A popular website that provides this facility is eBay.

6. Collaboration with Partners

Businessmen can collaborate with their partners using Internet. They can discuss ideas, exchange views and make strategies in collaboration with business partners in the world.

7. Marketing and Advertising

E-commerce is playing an important part to market and advertise products all over the world. The use of popular websites can be an effective way of introducing a product to the customers.

8. Providing Customer Services

Businessmen can interact with their customers using the Internet. They can discuss different issues about their products. They can also deal with their complaints and provide different services to them.

9. Online Travel Reservations

Online travel reservation is a popular use of e-commerce. People can reserve seats in airline flights, hotels or car using the Internet.

10. Online Trading

Online trading is a process of conducting business using the Internet. The stockbrokers can do all trading activities electronically. They can submit and receive bids using computers. They can also interconnect with computer display screens where brokers match the buyers

with sellers. It reduces the cost as no paper or special building is required to conduct these activities.

1.5.7.2 Types of E-Commerce

Three types of E-Commerce are as follows:

1. Business-to-consumer

Business-to-consumer or B2C e-commerce consists of the sale of goods and services to the general public. In this type, customers or consumer can visit the website and purchase goods online. It is done by an electronic **storefront** that contains product details, graphics and shopping cart. **Shopping** cart is used to collect the purchases. Online shopping is normally performed using a secure Internet connection.

2. Consumer-to-consumer

Consumer-to-consumer or C2C e-commerce takes place between two consumers. For example, one consumer sells an item through online auction. The other consumer purchases the item by offering highest bid. Many websites provide the facility of online auction where C2C e-commerce occurs. A popular website providing this facility is eBay.

3. Business-to-business

Business-to-business or B2B e-commerce takes place between two businesses. One business provides services to other business. Online advertisement, recruiting, sales, marketing, technical support and training are some examples of B2B e-commerce. For example, some companies provide online purchasing, tracking and transaction facilities to other companies.

1.5.7.3 Advantages of e-commerce

Following are some important advantages of e-commerce:

- **Geographic Reach** – Organizations at distant places can interact with customers easily. Any person from any part of the world can be their customer.
- **Speed** – interaction between buyer and seller takes place almost instantaneously.
- **Productivity** – enables to search and obtain information, products, and services online. It increases the productivity of individuals and organizations.
- **Information Sharing** – digital information can be transmitted anywhere in the world easily. It is easier to do this than any traditional sharing method.
- **New Features** – e-commerce participants are constantly identifying new product features and services that they can offer online. For example, the organizations can deliver digital music online.
- **Lower Costs** – the cost of conducting business online is much lower as traditional physical resources are not needed.
- **Competitive Advantage** – companies using e-commerce are more competitive because they can offer their products and services to more customers well in time.

1.5.7.4 Functions of e-commerce Web sites

The four functions performed by B2C e-commerce Web sites are:

- **Catalog and content management** – It is the information of a commercial Web site. It must be up-to-date at all times and easy to access for the customer.
- **Shopping and checkout** – The site must allow the customer to identify multiple items they want to purchase as they shop then allow them to purchase all of them.

- **Back office processing** – It deals with final steps of sale such as processing credit card payments, updating inventories, and preparing to send items to customers.
- **Advertising** – The Web site must work with search engines, ads on other sites, online marketing, and email to advertise the products and services to online customers.

1.5.8 TELNET

TELNET is a program that is used to connect to a remote computer on Internet. A computer can act like a terminal directly linked to the remote computer. It allows a user to logon to a remote computer. The user can logon to the computer and use it as if he is connected to the computer directly.

1.5.9 FTP

File Transfer Protocol is used to transfer files over the Internet. Audio, video, graphics and data files can be uploaded or downloaded using this protocol. A process of transferring a file from remote computer to local computer is called **downloading**. The process of transferring a file from a local computer to remote computer is called **uploading**.

Different software are available to use this protocol for transferring files on the Internet. **CuteFTP** and **WS_FTP** are examples of FTP software.

1.5.10 Internet Relay Chat (IRC)

Internet Relay Chat is a real-time Internet-based chat service. **Chatting** is a process of exchanging views on the Internet by typing text messages in real-time. Many users join chat groups for exchanging views. Chat group is a group of users with similar interest. Chat groups are also called **channels**. IRC requires the use of IRC client program that displays a list of current IRC channels.

1.5.11 Instant Messaging (IM)

Instant messaging is a real-time Internet communication service. It uses text-based conversation similar to chat group. The users set up a list of contacts often called **buddy list**. The contact must be online to participate. The online members in the buddy list can exchange messages and files. IM software detects the presence of online members. Two examples of IM are **America Online's Instant Messenger** and **Yahoo Messenger**.

1.6 Basic Terminologies of Internet

Following are some important terminologies of Internet:

1. Website

A collection of related web pages that you can access electronically is called web site. It contains text, graphics, sound and video. There are different types of websites that provide different types of contents such as news, information, education and entertainment etc. A website must be stored on a web server to be accessible all over the world.

2. Home page

Most websites have a starting point, called a home page. It is similar to a book cover or table of contents for the site. It provides information about the site's purpose and content.

3. Hyper link

A hyper link is simply part of the text (or graphics) on a web page. When the user clicks on it, it automatically:

- Takes the user to a different part of the same page

- Takes the user to a different page within the web site
- Takes the user to a page in a different web site
- Enables the user to download file
- Lunches an application, video or sound.

4. Image Map

Image map is an image whose different parts are linked with different web pages.

5. Frame

Frames are used to divide a web page into different parts. Different web pages can be displayed in each part.

6. Table

Table is a set of rows and columns. Data can be displayed in tables in a web page.

7. HTTP

HTTP stands for **Hyper Text Transfer Protocol.** It is the foundation of data communication for the World Wide Web. It works as a request-response protocol in the client-server model. It requires a client program at one end and an HTTP server at the other end. For example, a web browser such as Internet Explorer acts as **client** and the application a hosting computer acts as a **server.** The client submits an HTTP **request** message to the server. The server returns a **response** message to the client. A response contains completion status information about the request and may contain any content requested by the client. The HTTP protocol is designed for communications between clients and servers.

8. HTML

It stands for Hyper Test Markup Language. HTML is the fundamental language to design web pages.

9. Webmaster

Webmaster is the person who develops and maintains a web site.

10. Domain Name

Domain name is a unique name given to a website. The name is linked to a web server where the information of website is stored. www.hec.gov.pk is an example of domain name.

11. URL

URL stands for Uniform Resource Locator. The URL is the web address for any given web document. Every web document has a unique URL. Each URL has several parts, which can be demonstrated using the address:

http://www.google.com/services/index.html

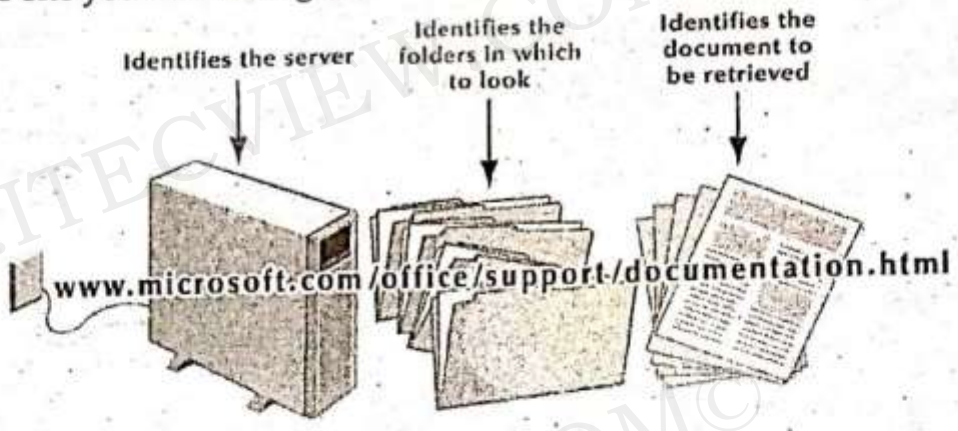
http:// This part of the address indicates that it is a Web page

WWW This indicates that the Web page you are looking at is part of the World Wide Web. Many Web sites do not use www but are still part of the Web.

google.com This part of the address is the domain name and indicates the unique address of a Web site. The domain name also often indicates the type of web site.

/services/ The "/" symbol indicates you have moved into a specific directory in the Web server. Directories are like the folders on your computer and help to organize Web pages in Web sites.

index.htm A word with ".htm" or ".html" following it indicates the name of the specific page in the Web site you are looking at.



13. ISP

ISP stands for **Internet Service Provider.** ISP provides the link between the individual user's computer and the Internet. The most common type of physical connection used for this service is a dialup modem that operates over normal telephone lines.

An ISP is a company, which provides you access to the Internet and provides facilities of WWW, Email and FTP etc. Different ISPs in Pakistan are Comsats, Brain Net and WOL.

14. SMTP

SMTP stands for **Simple Mail Transfer Protocol.** It is one of the most popular network services is electronic mail. It supports electronic mail on the Internet. It is a system for sending messages to other computers users based on email addresses. SMTP provides the facilities for mail exchange between users on same or different computers. It also supports:

- Sending a single message to one or more recipients
- Sending messages that include text, voice, video or graphics

Sending messages to users on networks outside the Internet.

15. DNS

DNS stands for **domain name system.** IP address is difficult to remember for the user. The domain names are alphabetic and are easier to remember.

The Domain Name Server resolves the name of the website typed in the browser with IP address of the server that sends the requested information from that website. For example, the domain name www.example.com might be translated to 198.105.232.4.

The DNS server is typically local in most major network environments. It must be updated periodically to include new domain names.

1.7 Advantages of Internet

Some important advantages of Internet are as follows:

1. Information Search

Internet contains information on all types of topics. People can search information on any topic. Search engines are used to search information on Internet.

2. Email

Email is an inexpensive and fast way of communication. It is used to send messages, pictures and files from one part of the world to another part.

3. E-Commerce

E-Commerce means to carry out financial and business dealings using Internet. A person can deal with his customers throughout the world.

4. Online Results

Internet provides the facility to display results of students on websites. Many colleges and universities provide online results. The students view the results from any part of world.

5. Online Shopping

People can buy and sell good on Internet. Even payments can be made using Credit Card numbers etc. The trend of E-Commerce is rapidly becoming popular.

6. Fast Communication

Internet is a way of fast communication. People can communicate with one another in less time from any part of the world.

7. Discussion Groups

People can communicate over the Internet using discussion groups. Different people discuss different topics and distribute their ideas to other people by email. Softwares like listserv are available to administrate discussion groups.

8. Job Search

Internet is used to search different types of jobs all over the world. Many websites are developed that provide information to the people about job vacancies. Job seekers can also apply for the job using Internet.

9. Entertainment

Internet provides a lot of entertainment to the people. The users can play online games, listen songs, watch movies and live matches etc. Many websites provides entertainment material. People can also download entertainment stuff.

10. Online Education

The process of getting education through Internet is called online education. Many websites provides tutorials and lectures on different subjects and topics. Some websites allow the users to download these tutorials. It is a fast and cheaper way of educating a large number of people.

11. Research

Internet provides many facilities for research work. It contains detailed information on different topics. The researchers can find the required information on Internet. They can also interact with other researchers in the world to share knowledge and get guidance from them.

12. Advertisement

Many commercial organizations use Internet to advertise their products. They provide information about their products to the people all over the world. Internet is a quick and effective way of advertisement.

13. Video Conferencing

People can use Internet for video conferencing. They can talk on different topics by watching to one another.

14. Government Authorities

Government authorities can provide information about services and facilities for the people.

15. News

Most newspapers have their website. Latest news and interviews are provided on these websites.

16. Encyclopedia Publisher

Online encyclopedias are available. You can find detailed information on any topic in the world.

17. Search Engines

Search engine is a website that provides you the facility to find the required websites on a particular topic. You can find thousands of websites about any topic.

18. Medical

Many sites in the world provide detailed information on diseases. You can also consult a panel of on-line doctors to get the advice about any medical problem.

1.8 Disadvantages of Internet

Some important disadvantages of Internet are as follows:

1. Hacking

Hacking is one of the most important disadvantages of Internet. The hackers access the data stored on computers across Internet. They can use this data illegally or even destroy it.

2. Immorality

Internet contains a large number of immoral websites. These websites contains such material that is against the moral values of our society. These websites are damaging the character of young people.

3. Security Problems

Internet has created many security problems. Important data can be hacked on Internet. Hackers also damage different websites and delete their contents. They also retrieve critical data of different organizations and governments.

4. Viruses

Internet is the most important source of spreading computer viruses. People spread viruses using Internet and emails. Many websites also contain different viruses that are copied to the computers when the users visit these websites.

5. Wastage of Time

Many people use Internet without any positive purpose. The young people waste their time in chatting. It affects their performance and makes them inefficient.

6. Cyber Crime

Internet is a source of many cyber crimes. People use Internet for negative activities. They hacks the credit card numbers of the people and use them for shopping. Some people use Internet to spread illegal and immoral material. Many governments are introducing laws to stop cyber crimes.

1.9 Benefits of Internet for Students

Some important benefits of Internet for students are as follows:

1. Online Education

Many colleges and universities in the world provide the facility to get education using Internet. The students can earn degrees and diplomas from home using the Internet.

2. Downloading Software

The students can download many educational software from the Internet. These software can help the students to improve their knowledge.

3. Online Results

Internet provides the facility to display results of students on websites. Many colleges and universities provide online results. The students view the results from any part of world.

4. Online Tutorials & Lectures

Many websites provides tutorials and lectures on different subjects and topics. Some websites allow the users to download these tutorials. The students can use these tutorials and lectures to learn different subjects.

5. Research

Internet provides many facilities for research work. It contains detailed information on different topics. The research students can find the required information on Internet. They can also interact with other researchers in the world to share knowledge and get guidance.

6. Online Encyclopedia

Online encyclopedias are available on the Internet. The students can find detailed information on any topic in the world.

7. Discussion Groups

The students can communicate with other students over the Internet using discussion groups. Different students discuss different educational topics and distribute their ideas to other students.

8. Information Search

Internet contains information on all types of topics. Search engines are used to search information on Internet. The students can search required information on any topic.

1.10 Benefits of Internet for Business Organization

Some important benefits of Internet for business organization are as follows:

1. Marketing and Advertising

A business organization can use the Internet to market and advertise products all over the world. The use of popular websites is an effective way of introducing a product to the customers.

2. Providing Customer Services

A business organization can interact with the customers using the Internet. The businessman can discuss different issues about their products. He can also deal with their complaints and provide different services to them.

3. Online Trading

Online trading is a process of conducting business using the Internet. A business organization can do all trading activities electronically. The products can be displayed on the website where the customer can buy these products. The customers can also pay by credit card. It is and easier and quicker way of trading.

4. Wider Customers Base

Organizations at distant places can interact with customers easily. Any person from any part of the world can be their customer. It provides a wider customers base for any business organization.

5. Competitive Advantage

Companies using the Internet in business are more competitive because they can offer their products and services to more customers well in time.

1.11 Web Browser

A **web browser** is software that is used to view web pages. It acts as an interface between the user and the Internet. It can understand HTML. It can display text and graphics. Browsers are also known as web clients or universal clients. Some popular web browsers are:

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari

1.11.1 Functions of Web Browser

Some important functions of web browsers are as follows:

- **Web Browsing:** It is the most important function of a web browser. The user can browse different websites using web browser and see the contents of websites. The web browsers provide different buttons to browser easily such as **Back, Forward, Stop** and **Refresh** etc.
- **Tabbed Browsing:** Tabbed browsing allows the user to open multiple pages in the same browser window. Each webpage is loaded in separate tab. The user can move between different pages more easily.
- **Search Box:** The web browsers provide a built-in search box. The user can designate the preferred default search engine for search.
- **Favorites:** A web browser provides the facility to store favorite website address in a list. The user can go to any web page in the list by clicking on it. The user does not need to remember the addresses of favorite websites.
- **Maintaining Browsing History:** A web browser maintains browsing history. It stores the visited links so that the user may visit these websites again easily.
- **Mailing:** A web browser also provides the facility to read or send e-mail. The user can also open up newsgroups etc.
- **Printing:** A web browser can print the web pages. The user can print the whole web page or any part of the web page by selecting.
- **AutoComplete:** A web browser stores the addresses of all websites that have been visited by the user. AutoComplete feature displays complete address of a website as the user starts typing that address in the address bar. The user does not need to type the complete address. It saves a lot of time.

1.11.2 Types of Browsers

Different types of web browsers are as follows:

1. Text-based Browsers

Text-based browser is used to see only textual information but not graphics. All text is of similar size. It can either be underlined or boldface. Keyboard is used to move in the web page. It does not support mouse. The most commonly used text-based browser is Lynx.

2. Graphical browsers

Graphical browser is used to see formatted text, graphics and animation. It can also play sound and video. There are many different graphical browsers. The most popular are Netscape and Internet Explorer.

1.12 Web Server

A **web server** is a computer that is used to store information for the users on the Internet. A web server is much faster and powerful than ordinary computer. It may be located anywhere in the world. Servers are connected to the Internet 24 hours a day.

The web sites are published on the WWW and you use your browser programs to read these web sites. In fact the data is stored on web servers. These web servers are connected to Internet. The users read data and information from these web servers.

1.13 Microsoft Internet Explorer

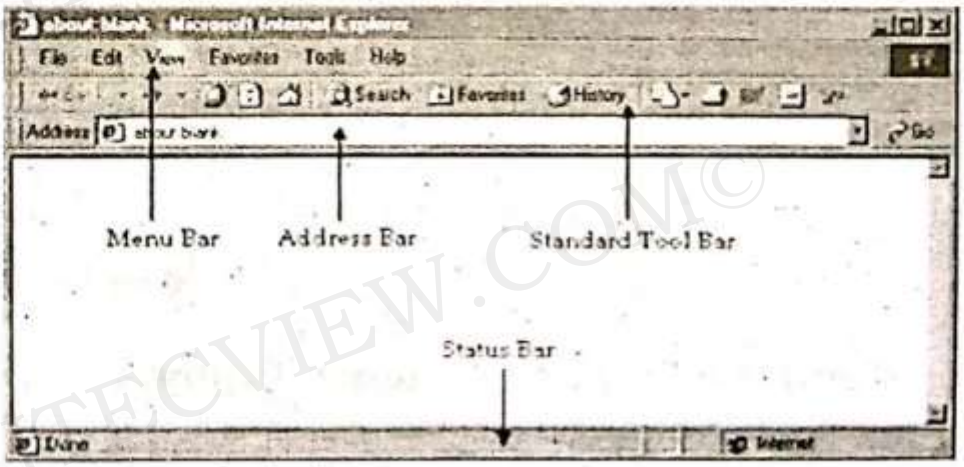
Internet Explorer is software that is used to view web pages. Microsoft Internet Explorer is a very popular web browser. It is a part of Microsoft Windows 98/2000/Me.

Starting Internet Explorer

Start > Programs > Internet Explorer OR use Internet Explorer icon on desktop or quick launch.

1.13.1 Main Window of Internet Explorer

The main window of Internet Explorer is as follows:



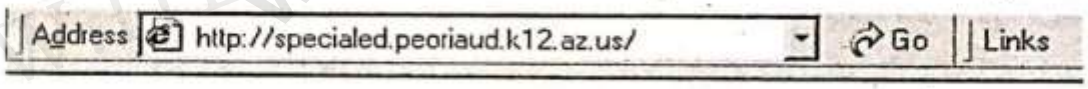
Internet Explorer Toolbar

The main toolbar has different buttons. Each has a different function and purpose in Internet Explorer:

- **Back Button:** This button is used to go back to the previous web page. It is disabled if the user is viewing the first web page.
- **Forward Button:** This button is used to go forward to the next web page. The user can go to the next page if Back button was clicked earlier. The button is disabled if the user has not moved back.
- **Stop Button:** The stop button stops any current operations by Internet Explorer. The page you are downloading will be cancelled.
- **Refresh Button:** This button is used to reload the current web page being viewed. It is useful if the page updates very frequently. The user can view the changes as soon as they are available.
- **Home Button:** This button is used to return to the default start-up page of Internet Explorer. The user can set any web page as default page.
- **Search Button:** It is used to move to the default search page for Internet Explorer.
- **Favorites Button:** This button is used to open **Favorites** menu. The **Favorites** menu is used to store the addresses of web page that are used frequently. The user can go to any web page in the list by clicking on it. The user does not need to remember the addresses of his favorite websites. The user can also add a favorite to the list. The **Add** button is used to add addresses in the list.
- **History Button:** This button is used to open the history folder. This folder contains all the links to pages that you have previously visited.
- **Mail Button:** This button is used to read or send e-mail. The user can also open newsgroups from this menu.
- **Print Button:** The print button is used to open Print dialog box.
- **Edit Button:** This button is used to open a web page editor. It will launch the editor and open the current web page in it for editing.

Address Bar

Address Bar displays the exact URL location of the page you are currently viewing. You can also type a Web address into this bar and then press the "Go" icon to go to that site.



Status Bar

The bottom of the page is the Status Bar. It displays the progress of the browser while it downloads files. It also shows the links when you move the mouse on a hyperlink.

1.13.2 AutoComplete Feature of Internet Explorer

Internet Explorer stores the addresses of all websites that have been visited by the user. These addresses are stored in **History** folder. The AutoComplete feature of Internet Explorer displays the compelte address of a website as the user starts typing that address in the address bar. The user does not need to type the complete address. It saves a lot of time.

1.13.3 Saving a Web Page

A web page can be saved on computer's hard drive or to a floppy disk. The following procedure is used to save a web page:

1. Saving Page to Hard Drive

The following procedure is used to save page on hard drive:

1. Open the page in your browser.
2. Click on the "File" menu.
3. Select "Save As" from the file menu.
4. A dialogue box will open. You will have to choose where you want the file to be saved, under what name, and in what format.
5. Click on the "Save" button in the dialogue box.

2. Saving Page to Floppy Disk (A:)

1. Follow the same steps as for saving to hard drive until opening the save dialog box.
2. Insert a diskette into the floppy disk drive.
3. Select **A:** drive in **Drive** box.
4. If there are already folders on disk, select the folder to save the page.
5. Click **Save**.

Once the page is saved, you can view without connecting to the Internet.

1.14 Web Directories

Web directory is used to search general information on a broad topic or to explore different areas of a broad topic. The directories are arranged by subject. They usually include links to the top level of the website rather than to pages within the website. The user searches by selecting a broad category and then narrows the search step by step. For example, a user may start the search with category of Social Sciences. He can move to History, By Region, Africa, Egypt, Ancient Egypt and then Pyramids. The results of different directories vary widely as websites includes in each category are selected by the people instead of machines.

Some examples of web directories are as follows:

- www.ipl.org
- www.looksmart.com
- www.webbrain.com

1.15 Search Engines

Search engine is a website that provides the facility to find the required websites on a particular topic. A user can search any topic on Internet using search engines. Internet contains a lot of information. A large number of websites about different topics are available. A user cannot remember all websites. He can search different websites using search engines.

Search engine contains the record of many websites. It searches the required websites and display them to the user. Search engines are very powerful tools for finding information.

1.15.1 Searching the Web

Different methods of searching the web are as follows:

1. Search using Boolean Logic

The Boolean operators are also used to search the required information from search engines. All search engines do not provide the facility of using Boolean operators.

The Boolean operators are as follows:

- **AND** operator is used to search on the basis of all words in the search string such as "British AND Novels". This search string will search for the websites that contain both words. The symbol + can also be used instead of AND.
- **OR** operator is used to search on the basis of any word in the search string such as British OR Novels. This search string will search the websites that contain any of the two words. The symbol – can also be used instead of OR.
- **NOT** operator is used to search on the basis of any word other than the search string such as "New York NOT City". This search string will search for the websites that contain the word "New York" but not "New York City".

2. Search by Phrase

Keywords are used to search the required information from search engine. Keyword is a word or set of words that is related to the topic being searched. The keywrod is typed in double quotation if it consists of two or more words in the form of phrase. For example, the following phrases can be used to search information about Pakistani universities:

"Pakistani Universities"

"Universities in Pakistan"

3. Search using Truncation

The truncation method is used to expand the seach to include a specified word root with different word endings. The first part of a keyword is given along with a symbol usually asterisk *. The following example will search Pakistan, Pakistani and Pakpattan etc:

Pak *

Most Popular Search Engines

Some popular search engines are as follows:

Google	http://www.google.com	**AltaVista**	http://www.altavista.com
Yahoo	http://www.yahoo.com	**Hotbot**	http://www.hotbot.com
Lycos	http://www.lycos.com	**Excite**	http://www.excite.com

1.16 Types of Websites

Different types of websites are as follows:

1. Portal

Portal is a website that provides different types of Internet services such as:

- Search engine
- Sports and weather
- Reference tools
- Maps
- Auctions
- News
- Free web publishing services
- Stock quotes
- Shopping malls
- Emails

Some portals contain information for general public and some provide information for specific group of people like laywers, doctors etc. A website for specific group of people with similar interests is called **Web community**. Some pupular portals are Yahoo, AltaVista, America Online, Excite, The Microsoft Network and Netscape. A type of portal is specially designed ofr Web-enabled devices. It is known as **wireless portal**. Wireless portals provide all facilities to mobile users.

2. News

News website provides different types of news related to current affairs, politics, sports, weather etc. They provide stories, articles and other material. Popular magazines and newspapers provide the news on websites. Some examples of news websites are:

- www.geo.tv
- www.jang.com.pk

3. Informational

An informational website provides information with facts and figures. Governments provide information about different departments and policies for general public. Some examples of informational websites are as follows:

- www.pakistan.gov.pk
- www.most.gov.pk
- www.punjab.gov.pk

4. Business/Marketing

A business/marketing webiste provides material to promote its products and services. Today, most of business companies have webistes. The customers can view the products and interact with the company. Many companies provide the facilities of online shopping using the Internet. Some examples of business/marketing websites are as follows:

- www.dell.com
- www.ibm.com

5. Educational

Educational webistes provide educational material related to teaching and learning. Many websites provide online training classes where people can learn different skills. Colleges and universities provide information about classes, admissions etc. on websites. Educational institutes also display results and course material on websites. Some examples of educational websites are as follows:

- www.pu.edu.pk
- www.uaf.edu.pk

6. Entertainment

Entertainment websites provide material for recreation and enjoyment. These webistes provide music, videos, sports, games and chats etc. Many websites also provides online shows, live sports etc. Some examples of entertainment websites are as follows:

- www.lollywood.com
- www.pakistanmusic.com

7. Advocacy

Advocacy websites provide material to convince people of a cause, opinion or idea. Some websites represents the viewpoint of a particular commnity or group. Some intend to create awareness in the people about a particular cause.

8. Personal

Personal websites contain material of an individual or a family. People publish personal websites to share their views with others, to seek job or just for fun.

1.17 Multimedia on Web

Multimedia is an application that uses text with grphics, animation, audio, video and virtual reality. The use of multimedia makes websites more attractive and interesting for the people. The websites with multimedia contents take more time to download. Multimedia consists of the following:

1. Graphics

Graphic or **graphical image** is the digital form of chart, photogrpah or drawing. Many websites use different colorful graphics, logos and charts to present information in an attractive way. The use of graphics increases the size of web page.

Acronym	Name
BMP	Bit map
GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
PCX	PC Paintbrush
PNG	Portable Network Graphics
TIFF	Tagged Image File Format

Table: Graphics formats used on the Web

Two commonly used graphic formats are JPEG and GIF. JPEG format compresses graphics and reduces its size. Compression reduces the size but also decreases the quality. GIF format also compresses images. This format is suitable for the images with a few distinct colors such as line drawings, borders etc. A new format PNG is an improve form of GIF format. BMP, PCX and TIFF formats are heavy in size and are not used on the Web normally. Many websites use thumbnails of images. **Thumbnail** is a small version of a larger graphics.

2. Animation

Animation is the display of motion by displaying a series of still images in a sequence. It makes websites more attractive and lively. Scrolling text is an example of animation. Animation easily captures the attention of the user. **Animated GIF** is a popular type of animation. It is actually a set of different images combined into a single GIF file.

3. Audio

Audio includes music, speech and other sounds. Audio is stored as individual sound files on the Web. Different formats of audio files are MP3, wav, WMA, MPEG, RealAudio and QuickTime. Audio files are also compressed to reduce size.

Some applications on the Web use streaming audio. **Streaming** is a process of transferring data in a continuous and even flow. With **streaming audio,** the user can listen to the music as it is downloading. MP3, QuickTime, Windows Media Format and RealAudio are accepted standards for transmitting streaming audio data on the Internet.

4. Video

Video is a set of full-motion images played at various speeds. Many videos also contain audio as well. Many websites provide live telecast of matches and shows. The users can download the video files from websites and watch them on their computers. Video files are found in different formats and are normally compressed to reduce their size. **Moving Pictures Experts Group (MPEG)** defines a popular video compression standard. The current standard is known as MPEG-4.

Streaming video allows the user to view video as it is downloading. Most commonly used streaming video standards are AVI (Audio Video Interleaved), QuickTime, Windows Media Format and RealVideo.

5. Virtual Reality

Virtual reality is a technique used to simulate a real or imaginary environment that appears in three-dimensional space. Virtual reality software creates indefinite space and depth known as **VR world**. It can display a building with furniture and equipments. The user can view the building on the computer by virtually moving the input device.

Virtual reality is commonly used in 3-D games. It is also used to create simulations of scientific structures, products and buildings etc.

6. Plug-Ins

Plug-in is a program that extends the capability of a web browser. Web browsers can display basic multimedia elements. Sometimes, they require plug-ins to display some files. Different plug-in programs are as follows:

Plug-In	Description
Acrobat Reader	Used to view Portable Document Format (PDF) files
Flash Player	Used to view graphics and animation, hear sound and music display Web pages across entire screen
QuickTime	Used to view animation, music, audio and video in a Web page
RealOne Player	Used to listen to live music and video
Shockwave Player	Used for 3-D graphics and streaming audio
Windows Media Player	Used to listen to live audio, play WMA and MP3 files, burn CDs and watch DVDs

1.18 Netiquette

Netiquette is an abbreviation of Internet etiquettes. It is the behavior expected from the users on the Internet. It includes different rules related to World Wide Web, email, message boards, chat rooms and instant messaging etc. Some netiquettes are as follows:

- Use short and concise messages with correct spelling and grammar.
- Use humorous word carefully as it may be misinterpreted.
- Use polite and sophisticated language.
- Do not send span mail to any one.
- Avoid the use of all capital letters.
- Use emoticons to expression emotions.
- Use abbreviations like AOA for Assalam-o-Alaikum
- Read frequently asked questions (FAQ) if available on a website or newsgroup.
- Never think that the available material is perfect.
- Do not read private email of any person.

1.19 Web Page Program Development

A collection of connected documents on the Internet is called **World Wide Web**. Each document is called a web page. It may consist of text, graphics, audio or video etc. The developer of web pages are called **web page authors**.

Different techniques used to develop web pages are as follows:

1. HTML

HTML stands for **Hyper Text Markup Language.** It is a formatting language used to develop web pages. Web pages can be viewed in a **web browser. Internet Explorer** and **Netscape Navigator** are two most popular web browsers.

HTML is not a programming language. It is used to define text, graphics, audio and video in web pages. It uses **tags or markups** to specify how the contents of web page will be displayed. HTML code can be written in any text editor like Notepad.

2. Scripts

Script is an interpreted program that runs on the **client** computer. It is written in a scripting language. An interpreted language used to write scripts is called scripting language.

3. Applet

Applet is a small Java program that can be embedded in HTML. It runs faster than script because it is compiled. Applet is different from Java applications. It cannot access certain resources on the local computer such as files and serial devices like modems, printers, etc. It is also prohibited from communicating with most other computers across a network. Normally, an applet can only connect to the computer from which the applet was sent.

4. Servlet

An applet that runs on a web server is called **servlet.** It responds to requests from clients. Servlets extend a Web server by generating content dynamically.

5. ActiveX Controls

ActiveX is a set of object-oriented technologies that allow components on a netwok to communicate with one another. **ActiveX control** runs on client computer instead of server.

Advantages of using Scripts, Applets, Servlets & ActiveX Controls

There are many advantages of using scripts, applets, servlets and ActiveX controls. They provide the following facilities:

- **Multimedia Effects:** They add special multimedia effects to web pages such as animated graphics, scrolling text, calendars and advertisements etc.
- **Interactive Capabilities:** They provide interactive capabilities to web pages.
- **Counters:** They are used to create counter for websites. A counter is used to track the number of visitors to a website.
- **Image Map:** A graphic that is used as a hyperlink is called image map. Different parts of the same image may link to different web pages.
- **Processing Forms:** Processing forms are used to collect data from visitors. The information entered in the forms is processed by scripts or applets.
- **CGI Programs:** CGI stands for **Common Gateway Interface.** It is a standard that defines how web server communicates with outside sources. A program that manages this communication is known as **CGI program** or **CGI script.**

1.19.1 Roles in Web Site Development Team

- **Server Administrators:** They take care of the technical issues and are responsible for running and maintaining the Web server. They also generate statistical reports on the users of a website.
- **HTML Coders:** They write HTML code for website pages and troubleshoot the site.

- **Designers:** They are the graphic artists responsible for the appearance of the website.
- **Writers and Information Designers:** Writers prepare content for online display such as hypertext information and navigation paths. They work closely with designers to create page templates.
- **Software Programmers:** They write the programs for interactive website using different tools such as **Common Gateway Interface (CGI)** scripts, **Java scripts**. They also develop back end applications to interface with databases.
- **Database Administrators:** They maintain the databases and ensure data accessibility and security.
- **Marketers:** They generate contents and work to make the website more popular.

1.20 Scripting Languages

A type of language used to write scripts is called **scripting language**. Scripting languages are used with interpreters. It means that a program developed with a scripting language must be run in the environment that contains the interpreter of scripting language. It cannot be executed as a stand-alone application. Scripting language is easier to learn.

Different scripting languages are as follows:

1. Java

Java is a popular object-oriented programming language. It was developed by **Sun Microsystem**. It is used to write stand-alone applications, applets and servlets. Java source code is compiled into **bytecode** instread of **object code**. Bytecode cannot be executed by operating system. It is executed by Java interpreter.

The code used to create Java application is called **JavaBeans** or **Beans**. It is platform indepedent and can run on different types of computers or operating systems.

2. JavaScript

JavaScript is an interpreted language that is used to add dynamic contents in Web pages. It is used to include messages, scrolling text, animations, menus, pop-up windows etc. JavaScript is directly embedded in HTML.

JavaScript was developed by Sun Microsystems and Netscape Communicator. It is an open language. It can be used without purchasing. Netscape supports JavaScript and Internet Explorer supports JScript. **JScript** is a subset of JavaScript.

3. VBScript

VBScript is an interpreted language that is used to add dynamic contents in Web pages. It is used to include messages, scrolling text, animations, menus, pop-up windows etc. VBScript is directly embedded in HTML.

4. PERL

PERL stands for **Practical Extraction and Report Language**. It was developed by Larry Wall at NASA's Jet Propulsion Laboratory. It is a powerful language for writing scripts.

5. DHTML

DHTML stands for **Dynamic Hyper Text Markup Language**. It is used to add dynamic contents and interactivity in web pages. It is used to apply colors and add animations etc. It uses **documents object model (DOM), style sheets** and **scripting languages**.

- **Document Object Model:** It defines the contents of web page as objects. It considers fonts, graphics and tables as objects.

- **Style Sheet:** Style sheet contains the characteristics of a document. Cascading style sheet contains the formatting details of each object in the document. Different style sheets can be used for applying different formatting to different objects.
- **Scripting Language:** Scripting language is used to manipulate the objects in the web page. A script can move, display, or change the appearance of an object as different events occur.

6. XML

XML stands for **Extensible Markup Language.** It allows developers to create customized tags. It is used to send complete record from server to the client that reduces network traffic. It can also be used to create a link that points to multiple websites. It uses **Extensible Stylesheet Language (XSL)** as its style sheet specification.

7. XTHML

XHTML stands for **extensible Hypertext Markup Language.** It combines the formatting strengths of HTML 4.0 and the data structure and extensibility strengths of XML. It uses the tags and attributes of HTML along with the syntax of XML. The first version of XHTML 1.0 was released in 2000. The **World Wide Web Consortium (W3C)** sets the standards for XHTML and other internet languages.

8. WML

WML stands for **Wireless Markup Language.** It is used to design web pages for microbrowsers. It is mostly used for handheld computers, cellular telephones and pagers. It uses **Wireless Application Protocol (WAP)** as a standard. WAP defines how wireless devices communicate with the Web.

9. The .NET Platform

.NET Platform is created by Microsoft. It is used to develop applications with **XML Web services.** These services are used to share data between Web applications created in any programming language and any operating system. It provides easier way of communication.

.NET applications run under any web browser and any type of .NET-compliant smart client. A computer or a device that can accesses XML Web services is called **.NET-compliant smart client.** Different .NET-compliant smart clients are personal computers, notebooks, handheld computers, web-enabled telephones, Tablet PC and Xbox™.

Tablet PC is a mobile computer that uses Windows XP operating system. It supports pen input and provides voice recognition facility. **Xbox™** is video game console. It provides the facility to update games regularly. It also allows interacting with other games via broadband Internet connection.

Visual Studio .NET (VS.NET) is a set of tools to develop applications for .NET platform. It provides editors for cascading style sheets, XML and HTML. It supports more than 25 programming languages. **Visual Basic .NET (VB.NET), Visual C++®.NET** and **Visual C#™.NET** are three popular languages.

10. Web Page Authoring Software

Web page authoring software is used to develop web pages with graphical images, video, audio, animation etc. It is also called **HTML editor.** It also generates HTML tags as the user designs web pages. HTML tags can be customized to modify the look of web page. Popular web page authoring software are Macromedia Dreamweaver, Microsoft FrontPage, Adobe GoLive and Macromedia Flash.

1.21 Web Hosting

Web hosting is a service that provides space on the Internet for storing web pages. Different organizations provide the facility of web hosting. Some websites also provide this facility free of cost. A website must be hosted on a Web server to be accessible from all over the world.

1.21.1 Web Hosting Services

A good web site hosting company will provide at least the following services:

- Technical support
- FTP access to update web site
- A number of email accounts
- Online control panel for managing web site
- A database management system, such as MySQL or MS SQL.
- Specialized server applications such as ColdFusion, SQL Server, or PHP if required.

1.21.2 Types of Web Hosting

Different types of web hosting are as follows:

1. Shared Hosting

Shared hosting is a type of hosting in which a web site is hosted on a server with many other web sites. However, the web site is configured as a separate web site on the server with its own domain name etc. The benefit of shared hosting is that it is cost effective.

2. Dedicated Servers

A **dedicated server** hosts one or more web sites of only one user. It gives more control over the web site. It can also ensures that other web sites do not impact on the web site. However, the dedicated servers are much more expensive than shared hosting. It should only be used if the website receives very heavy traffic or the user requires extra security etc. The user can log in to a dedicated server like a personal computer.

3. Virtual Dedicated Servers

Virtual dedicated servers are a low-cost alternative to dedicated servers. The web host can put many virtual servers on each machine to reduce costs. It appears to be a dedicated server when the user logs in to a virtual server.

4. Managed Hosting

A **managed hosting** means that the user has a dedicated server and the web site hosting provider manages the server for the user. Typically, the hosting provider is responsible for the hardware, operating system and virus protection etc. However, the user has total control over the machine and the web site content.

1.21.3 Web hosting technologies

Different web hosting technologies are as follows:

- **Windows Hosting:** Windows hosting is the hosting of web services running on Windows operating system. It should be used if the user wants to use ASP (Active Server Pages) as server scripting or a database like Microsoft Access or Microsoft SQL Server. Windows hosting is also the best to develop web site using Microsoft Front Page.

- **Unix Hosting:** Unix hosting is the hosting of web services running on Unix operating system. Unix was the first web server operating system. It is known for reliability and stability. It is also less expensive than Windows.
- **Linux Hosting:** Linux hosting is the hosting of web services running on the Linux operating system.

1.22 Web Publishing

The process of developing and maintaining web pages is known as **web publishing**. A website must be stored on a web server to be accessible all over the world. Many tools are available to publish web pages on the Internet.

1.23 Cookie

A **cookie** is used to maintain information about a user state. Cookies can be used to store user preferences and user information that can be used across pages in the website. A cookie is also known as **web cookie, browser cookie** and **HTTP cookie**.

A cookie is stored as **text strings** on the computer. A Web server sends a cookie and the browser stores it. The browser returns the cookie to the server when the page is referenced again. The most common use of a cookie is to store a user ID. A cookie may contain the string such as ID=96352398.

Example

The website *www.amazon.com* uses cookies. The user fills a form with name and address to order a book. Amazon assigns an ID, stores user's information with that ID in its database on the server. It also sends the ID to user's browser as a cookie. The browser stores ID on the hard disk. The browser sends the ID back to the server when the user visits Amazon again. The server customizes the web page according to the user ID.

1.23.1 Types of Cookies

Cookies can be temporarily and persistent.

- **Temporary cookie:** Temporary cookie is also called **session cookie**. It is stored temporarily in browser's memory and is deleted when the session ends or the browser is closed.
- **Permanent cookie:** Permanent cookie is also called **persistent cookie**. It is stored permanently on computer's hard drive. If it is deleted, it will be recreated when the user visits the website again.

1.23.2 Uses of Cookies

Cookies are used for different purposes such as follows:

- Most websites use them for personalization of user preferences. For example, a news website may allow the user to customize the page to display the news of his interests. The website stores user preference in cookies.
- Some websites also use cookies to store user's passwords. The users do not need to enter the password again and again.
- Some website use cookies to track visits of users and web pages visited by the users.
- Online shopping sites generally use session cookies to keep the track of items in a user's shopping cart.
- Some website may use cookies to target advertisement. These sites store a user's interests and browsing habits in the cookie and display related ads on the pages.

1.23.3 Browser Setting for Cookies

The browser can be set to accept cookies as follows:

- The user can set the browser to accept cookies automatically.
- It can be set to prompt the user to accept a cookie.
- It can be set to disable the use of cookies altogether. However, the user may not be able to use many of e-commerce web sites if cookies are disabled.

1.23.4 Privacy Concerns about Cookies

Webmasters obtain information about users' IP addresses, browser type, last pages visited using cookies. It allows Web server to recognize a user who has already visited the website from the same computer. It allows the compilation of user profiles and large amounts of data related to the interests of the user.

Advertisers and webmasters are currently using cookies to develop detailed profiles of users and their browsing habits. Each click on a particular type of page in a website is added to the profile. The information is primarily used for website design and placement of banner advertisements. However, this information can be sold for other commercial purpose that creates a concern to personal privacy.

1.24 TCP/IP

TCP/IP stands for **transmission control protocol/Internet protocol**. It is a technology that is used to manage the transmission of data by breaking it into packets. It is commonly used in Internet transmissions. When a computer send data over the Internet, the data is divided into small pieces called **packets**. Each packet contains data and the destination, sender and the sequence information used to reassemble the data at the destination. These packets travel via devices called **routers**. This process is known as **packet switching**.

1.25 Intranet and Extranet

Intranet

Intranet is an internal network of an organization that uses Internet technologies. It is normally used to provide the information of the organization to the employees. It is a small Internet within an organization. It uses TCP/IP technologies and has a Web server. It also supports multimedia Webpages that are viewed in a web browser. Some intranet may also provide access to the Internet. Some organizations also use extranet. It allows the customers to access a part of intranet.

Extranet

Extranet is a collection of two or more intranets. Different organizations can use extranet to share the information of their intranets. An organization can apply security measures to provide limited access to the employees of other organization using the extranet.

Difference between Intranet and Extranet

Intranet	Extranet
1. Intranet contains the information of one organization.	1. Extranet may have the information of two or more organizations.
2. Intranet is used to provide information to the employees.	2. Extranet is used to share information with other organizations.

1.26 Web 2.0

Web 2.0 is a set of techniques that provide an upgraded presentation and usefulness for World Wide Web. It provides more opportunities for individuals to collaborate, interact with one another and create new content using applications like blogs, wikis and podcasts.

Blogs and Vlogs

Blog stands for **Web log**. It is a Web site that contains a writer's reflections and comments. Many free sites provide the facility to set up a blog such as www.blogsport.com. **Vlog** stands for **video log**. It is a set of personal videos. Many vlogs are posted on **YouTube**.

Wiki

A **wiki** is an online information source that can be edited, deleted or modified by anyone. A wiki may contains errors and it should not be used as a primary research source. The most popular wiki is **Wikipedia**.

Podcasts and vodcasts

A **podcast** is a combination of the words **iPod** and **broadcast**. It is a program in digital format available for download and replay. Most podcasts are available for free. **iTunes** is a popular source. A user can create podcast with software and a microphone. **Vodcast** stands for **video podcast**. It refers to online delivery of video clip on demand.

Some popular video-sharing Web sites are as follows:

- YouTube
- iTunes
- iTunes Higher Education

Social networking

Social networking is a group of individuals creating a community. It is used to share pictures and common interests. Some examples of social networking sites are as follows:

- LinkedIn
- Ning
- Digg
- RSS Feeds

Tagging

A **tag** is a user-generated word or phrase that helps in organizing Web content. Web developers refer to these tags as metatags. The tags help in locating a Web page or blog when people search.

Collaboration Tools

Collaboration tools assist people work online as team. **Google Docs** allows Web-based collaboration on Word documents, spreadsheets and presentations

1.27 Web 3.0

Web 3.0 is the next generation of the Web. It is believed to make search tasks faster and easier by acting as a personal assistant. The browser may remember the interest of the user while browsing. The user can be less specific in the searches.

WWW.ITECVIEW.COM©

Short Questions

Q.1. What is Internet?

Internet is a huge collection of computers all over the world that are all connected to one another. It is a global network of computers. The computers are connected through telecommunications links.

Q.2. Is the Internet a single network?

Internet is not a single network. It consists of tens of thousands of networks around the world.

Q.3. What are hosts on the Internet?

The computers connected to the Internet are called hosts.

Q.4. Which communication links are used in Internet?

The communication links used in Internet are Phone lines, Fiber optics lines, Satellites and wireless connections.

Q.5. Who is the owner of Internet?

No government or organization is the owner of Internet. Many people, organizations, universities and research agencies participate to run Internet.

Q.6. What do you know about ARPANET?

During Cold War America developed a network named ARPANET. It was developed for Advance Research Project Agency. It was used to send information to armed forces at long distances.

Q.7. What is DARPA?

DARPA stands for Defense Advanced Research Projects Agency. DARPA worked to share data not only on single network but also among different networks.

Q.8. Why are TCP/IP protocols important to the functioning of the Internet?

TCP/IP stands for Transmission Control Protocol/Internet Protocol. This protocol is used to share and transfer data among different networks. This is the most commonly used protocol over the Internet. TCP/IP protocols are the "language of the Internet" and are what makes the Internet work

Q.9. Discuss the main functions of the Internet.

There are three main functions of the Internet:

Communication: Internet can be used to contact and exchange information with friends and organizations anywhere in the world.

Retrieve: People can access a broad range of data and information from websites.

Shop, Buy and Sell: People can buy and sell goods and services on the Internet

Q.10. Which protocol is the heart of the Internet?

The TCP/IP protocol is the heart of the Internet.

Q.11. How many addressing schemes are used on the Internet.

Two types of addressing schemes on Internet are IP addressing and DNS Addressing.

Q.12. What is the purpose of IP addressing?

IP address is a unique number that refers to a computer connected to the Internet. IP stands for Internet Protocol. This address is a 32-bit number. It consists of four numbers separated by period. Each number is between 0 and 255. These numbers are known as octets. An example of IP address is 127.80.144.30

Q.13. Describe the composition of an IP address.

IP address is a string of four numbers. Each string is separated by periods or dots.

Q.14. Write the advantage of using a domain name to address a node than IP address.

A domain name is easier to remember and more readable by a human user.

Q.15. Explain the difference between an IP address and a DNS address.

IP address is a numerical address based on four integers separated by periods. DNS address is a symbolic name that it easier to remember and pass along to others. IP addresses are fine for computers like Web servers and routers, but people find it easier to remember names instead of numbers.

Q.16. What does a domain name server do?

It translates DNS addresses into IP addresses.

Q.17. How is an Internet address determined? Who assigns Internet addresses?

An Internet address, or domain name, is obtained ultimately from ICANN. The top level (.com, .edu, etc.) can be chosen, although .edu must be validated as an educational institution. The only requirement is that the name is unique. The domain name is leased on a yearly basis for a small fee.

Q.18. List three top-level domain names.

The top-level domains are .edu, .com, .gov,.net, .org.

Q.19. What are included in an Internet address?

An Internet address has four parts:

1. The name of the host computer.
2. The name of the institution or organization
3. The domain name.

Q.20. What is an application service provider? What benefits does it provide?

Application service providers are companies that develop, install and operate an information technology application on the Internet for the user. It charges fee for this. The application on the Internet can be made available to anyone having access to the Internet. ASPs offer the benefit of taking full responsibility for maintaining the service, ensuring that it is always available for use.

Q.21. List different ways to connect to the Internet.

Normally, dialup access is used to connect to the Internet at home or office. More high-speed connections are also available to connect to the Internet such as DSL, ISDN and cable television Internet services. These technologies provide high-speed Internet connections using telephone lines. Cable modem provides highspeed Internet connections through cable television network.

Q.22. What is ISP connection?

ISP stands for Internet Service Provider. It provides the link between the computer and the Internet. A person has to get an account of an ISP for establishing a connection to Internet.

Q.23. Write some facilities provided by Internet.

WWW: It is also called Web. It provides the facility to publish information on Internet. It is a collection of documents stored on computers permanently connected with Internet around the world.

Email: Email stands for Electronic Mail. Email is the exchange of messages and files through Internet. It is a fast way of sending messages anywhere in the world in a very short time.

Email Discussion Groups: People can communicate over the Internet using email facility. Softwares are available to administrate email discussion groups such as the listserv.

NewsGroup: Newsgroup is a global electronic bulletin board system. People exchange information on a vast range of topics such as news, recreation, business, science, and computer.

Ecommerce: E-Commerce means to carry out financial and business dealings using Internet. People can buy and sell good on Internet. Even payments can be made using Credit Card numbers etc

Q.24. What is difference between The Internet and Web?

The Internet is the physical connection of millions of networks. The Web uses the Internet for its existence. The Web consists of hypertext embedded on Web pages that are hosted on Web sites

Q.25. What are some uses of WWW?

WWW can be used in manufacturing, shopping, flight information, television stations, Govt. authorities and medial field.

Q.26. How web pages are created?

Web pages are created in hypertext using special languages. The most popular hypertext language is called HTML. It stands for Hyper Text Markup Language. It is used to create web pages.

Q.27. When is HTML needed for the WWW? Who uses HTML?

HTML is used to make Web Pages. HTML is a small set of key commands to tell a browser how to display the text and pictures in a Web Page. It specifies the position, size and color of text, graphics, sound and video. It also identifies what pages to display when links are clicked.

Q.28. Write some popular languages to create web pages.

Some popular languages to create web pages are XHTML, XML, WML and WAP.

Q.29. Name some advantages of Email.

Email is free, easily accessible, convenient and quick.

Q.30. Write four problems associated with e-mail.

E-mail can be a threat to privacy. It can be faked. It cannot communicate emotions properly. It may be delayed due to communication problems.

Q.31. Which types of files can be attached to email messages?

The types of files that can be attached to email messages are Word documents, pictures and other multimedia files etc.

Q.32. Define email address.

Every email account has unique address. An email address usually has two parts i.e. User ID and Identity of email service.

Q.33. Give one example of email address.

pakistan@yahoo.com

Q.34. Explain the main difference between an IMAP mail service and a POP mail service. Which is more powerful?

IMAP mail service stores all emails on the mail server. POP mail service moves all emails to local host for long-term storage. IMAP is more powerful as it gives more options to manage email folders remotely or locally. It is also better for people who need to work from multiple locations. It does not use much memory on local host.

Q.35. How are chat sessions and instant messages different?

Chat sessions are live, interactive discussions with people currently on the Internet. Anyone can join the chat. Chat sessions are open to all. All participants in the chat can see the responses of others. Instant messaging is more private. The interaction is normally only between two people at a time.

Q.36. What is an FTP site and why would someone visit such a site on the Internet?

An FTP site represents a server on which someone has placed files for users to download. Access to the files is across the Internet. FTP is the method or program used to download the files. This can be done using a browser, typing in ftp:// rather than http:// before the rest of the URL.

Q.37. What is Web Site?

A collection of related web pages is called website. Each website has a unique address. It contains text, graphics, sound, and video.

Q.38. Define Home Page of a Web site.

Most websites have a starting page called a home page. It is similar to a book cover or table of contents for the site. It provides information about the site's purpose and content.

Q.39. Write a short note on ISP.

ISP stands for Internet Service Provider. It is a company that provides Internet connections. ISP also provides the facilities of Email. A connection from ISP is necessary to connect to Internet.

Q.40. What benefits does streaming offer Internet users?

Streaming allows Internet users to receive video or voice over the Internet more quickly than normal video. As soon as some amount of video signal has arrived, it begins to run. The user can start viewing it as it continues to download in the background. It also allows users with slow connections to watch video that would normally have required a faster connection.

Q.41. What is web browsing or web surfing? List two names of Web Browser.

Web browsing or web surfing is a process of searching information on the World Wide Web. A software used to search and view web pages is known as web browser. The two names of browsers are Internet Explorer and Netscape.

Q.42. Distinguish between HTTP and HTML.

Hypertext Transfer Protocol (HTTP) is the standard used for the transfer of requests and responses. Hypertext Markup Language (HTML) is a standard used to create webpages.

Q.43. State the use of Web Browser.

A browser is a software that is used to view web pages. It acts as an interface between the user and the Internet. It can display text and graphics. Browsers are also known as **web clients** or **universal clients.**

Q.44. List names of some popular web browsers.

Some popular web browsers are Internet Explorer, Netscape, Mozilla Firefox,Opera & Google Chrome.

Q.45. Explain how the browser's caching ability improves download time.

Browsers store recently downloaded web pages in the cache. When the user access a page, the browser checks the cache first to see whether the content is stored locally. If so, the browser loads the content from the hard drive, rather than downloading it again from the Internet.

Q.46. What are two types of web browsers.

There are two types of web browsers.

Text-Based Browser: Text-based browser is used to see only textual information but not graphics. Keyboard is used to move in the web page. It does not support mouse. The most commonly used text-based browser is Lynx.

Graphical Browsers: Graphical browser is used to see formatted text, graphics and animation. It can also play sound and video. There are many different graphical browsers. The most popular are Netscape and Internet Explorer.

Q.47. What are the functions of portals.

Today's portals offer four functions:

1. Search capability
2. Access to specialized functions
3. Personalization of content
4. Communities of interest

Q.48. List different types of e-commerce.

Business-to-consumer: It consists of the sale of goods and services to the general public. The customers or consumer can visit the website and purchase goods online.

Consumer-to-consumer: It takes place between two consumers. For example, one consumer sells an item through online auction. The other consumer purchases the item by offering highest bid.

Business-to-business: It takes place between two businesses. One business provides services to other business. Online advertisement, recruiting, sales, marketing, technical support and training are some examples of B2B e-commerce.

Q.49. List the functions of B2C sites.

B2C sites offer four common functions:

1. Catalog and content management
2. Shopping and checkout
3. Advertising.

Q.50. What are the forms of advertising commonly found in B2C e-commerce?

Four forms of advertising are common in B2C e-commerce are Search engine links, Banner ads, Interactive marketing and E-mail.

Q.51. What is domain name?

Domain name is the unique name given to a web site. This name is linked to a web server where the information of that web site is stored. **www.urdupoint.com** is the example of a domain name.

Q.52. Distinguish between a vertical portal and a horizontal portal.

Vertical portals specialize in providing information related to a particular industry such as automobiles, healthcare, or investments. Vertical portals are narrow in scope. Unlike vertical portals, horizontal portals provide services and links to Web sites of interest to a wide variety of users.

Q.53. Describe the use of Web Server.

Web server is a computer that is used to store information for the users on the Internet. It may be located anywhere in the world. Servers are connected to the Internet 24 hours a day.

Q.54. What is the difference Web Browsers and Servers?

Web browsers display Web document and enable users to link to other Web pages. Web servers respond to the requests of browsers. They find and send requested resources back to the browser.

Q.55. What is the difference between client and server software? Give an example of each.

Client software runs on your computer. Client applications are Web browser, email, newsreader, FTP, telnet or file compressors.

Server software runs on a remote computer. Server applications are Web (HTTP) server, Usenet (NNTP) server or email (POP, SMTP, or IMAP) server.

Q.56. Which part of the following URL is a domain name? What are the other parts?

URL: $\underbrace{\text{http://}}_{A}\underbrace{\text{www.pu.edu.pk/}}_{B}\underbrace{\text{index.html}}_{C}$

Part A specifies hypertext transfer protocol. Part B is a domain name. Part C is a file.

Q.57. Name three components of a URL.

The three components of a URL are protocol, address of the host computer and path to the resource (file).

Q.58. Distinguish between HTTP and FTP.

HTTP is to establish a connection with a Web server and transmit HTML pages or any other files to Web browser. FTP is used to download and upload files on the Internet.

Q.59. How does the WWW work?

Web information is stored in documents called web pages. Web pages are files stored on computers called web servers. Computers reading the web pages are called web clients. Web clients view the pages with a program called a web browser such as Internet Explore.

Q.60. How does a browser fetch a Web Page?

A browser fetches a page from a web server by a request. A request is a standard HTTP request containing a page address. An address may look like this: http://www.urdupoint.com/default.htm.

Q.61. How is a web page accessed?

There are two main ways to access a page. The user can enter URL for the page and press Enter. The user can also click a hyperlink that opens the desired page.

Q.62. Write the protocol and the domain name of the following URL. Also state the top level domain.

http://www.student.net/learning/english.html.

The protocol is HTTP (HyperText Transfer Protocol). The domain name is student.net. The top level domain is net.

Q.63. Differentiate between URL and Website.

The URL is the web address for any given web document. Every web document has a unique URL. URL is written in the web browser to visit the web page. Website is a collection of web pages that are stored on computers around the world.

Q.64. How does a URL enable a user to find and retrieve information?

URL is a document's address on World Wide Web. Every Web page has its own unique URL. When a user types any URL, special servers on the Internet translate the URL to a numeric value. This value is the actual address of the webpage. This numerical address points to the server on which the Web Page resides, so the page can be obtained for viewing by the user.

Q.65. What do email addresses and URLs have in common?

URLs include a host name (the computer on which the Internet document is stored). E-mail addresses also have host names. For e-mail addresses, the host name is the name of the computer that provides e-mail services to that individual.

Q.66. Explain the importance of Search Engine.

Search engine is a website that provides the facility to find the required websites on a particular topic. A person can search any topic on Internet using search engines.

Q.67. List any five websites providing free email facility.

Some websites that provide free email facility are www.hotmail.com, www.yahoo.com, www.pakwatan.com, www.msn.com and www.go.com.

Q.68. List five search engines.

Some important search engines are www.google.com, www.altavista.com, www.yahoo.com, www.lycos.com and www.excite.com.

Q.69. What is Client-Server Model?

The exchange of data on Internet is based on client-server model. The user's computer is called client. The computer providing information is called **server**.

Q.70. How does web casting differ from video conferencing?

Webcasting uses the Internet to link people from anywhere in the world into a video conference. They can view the conference using a computer or handheld device. They can interact in the discussion through attached digital Web cameras or microphones.

Q.71. What is FTP?

FTP stands for File Transfer Protocol. It is used to transfer files over the Internet. Audio, video, graphics and data files can be uploaded or downloaded using this protocol.

Q.72. Explain the difference between FTP and Telnet.

FTP allows users to move files between Internet hosts. Telnet allows users to log in to accounts on remote hosts and conduct computing sessions (e.g. run OS commands and applications) on those hosts remotely.

Q.73. What is web hosting?

Web hosting is a facility for providing space on Internet for storing web pages. Web hosting is provided by different organization commercially.

Q.74. Define host computer.

A larger computer used to store web pages on the Internet is known as **host computer**.

Q.75. List the tools for retrieval on the Internet.

The tools for retrieval on the Internet include:

- FTP
- Portal
- Web directory
- Web casting
- Search engines
- Streaming.

Q.76. What role do plug-ins play in using WWW? With what software are they used?

Plug-in is small software programs that extend the capabilities of Internet browser. It gives added features to the browser. It is used to play sounds, music and video from within the browser.

Q.77. What is uploading and downloading.

The processing of copying data from your computer to Internet is known as uploading. The process of copying data from Internet to your computer is known as downloading.

Q.78. What is web publishing?

The process of developing and maintaining web pages is called web publishing. It does not require programming skills. Many tools are available to publish websites on the Internet.

Q.79. What is Nettiquettes?

Netiquette is an abbreviation of Internet etiquettes. It is the behavior expected from the users on the Internet. It includes different rules related to World Wide Web, email, message boards, chat rooms and instant messaging etc.

Q.80. What is mail server?

A computer that hold electronic mail messages for clients on a Internet. It manages the sending and receiving of emails. It is a very powerful computer that remains connected to the Internet all the time. It may be located at any place in the world.

Q.81. What chat server?

A dedicated server that manages text-based chat among different people all over the world is called **chat server**. It is a very powerful computer that remains connected to the Internet all the time. It may be located at any place in the world.

Q.82. What is difference between web hosting service and web server?

Web hosting is a service that provides space on the Internet for stroing web pages. Different organizations provide the facility of web hosting. Some websites also provide this facility free of cost. Web server is a computer that stores the web documents to be viewed by Internet visitors. It connected to the Internet for all the time.

Q.83. What is Gopher?

A system allowing users to search for files via menus or directory structures. Uses plain English names and is text based only

Q.84. Why are root servers important to the successful functioning of the Internet? How many root servers comprise the Internet?

Root servers are specially designated computers distributed around the world that contain the IP addresses of all of the top level domain registries for both global registries (e.g. .com and .edu) and country registries (e.g. .uk, .jp). There are 13 of these servers. They are managed by ICANN.

Q.85. What is the difference between a search engine and a directory?

Search engine is a program that takes terms provided by a user and searches the Internet for Web Pages that "match" the terms. It returns the links to the pages. A directory is a Web Page that contains lists of Web Sites and their URL's, categorized by subject. To locate a Web Site through a directory, you have to enter a keyword, and the Web Page then displays a list of the relevant categories

Q.86. Can intranets and extranets use the Internet for transmission?

Yes, both intranets and extranets may use the Internet for transmission needs

Q.87. Explain the difference between intranet and extranet.

Intranet contains the information of one organization. Intranet is used to provide information to the employees. Extranet may have the information of two or more organizations. Extranet is used to share information with other organizations.

Q.88. What's the difference between a mailing list and a Usenet newsgroup?

Mailing list messages are delivered to the inbox. Usenet newsgroup messages reside on a news server and the user must view messages through a newsreader or Web interface as they will not be delivered

Q.89. What is the definition of the term "listserv"?

A listserv is a mailing list program that accepts subscription requests and posts messages to the list members.

Q.90. Define the terms static Web page and dynamic Web page.

Static Web page is a page that is stored on a server. Dynamic Web page is created by a server-side program and is not a file in itself.

Q.91. What is spam and how can you reduce the amount of spam you receive?

Spam is junk email. The user can use spam-blocking software to reduce the amount of spam. The ISP may also use spam-blocking software on the mail server.

Q.92. What is an email client?

Email client is an application package that allows the user to send, receive and manage email such as Microsoft Outlook, Eudora and Pegasus. Some browsers such as Netscape and Mozilla can be considered email clients because they have a separate email feature to connect to POP and IMAP email.

Q.93. What is the significance of Domain Name System in the evolution of the Internet?

The user can type a domain name as URL for a Web page or email address instead of using IP addresses. IP addresses are difficult to remember and user can use domain easily.

Q.94. What you need to publish a web site?

A user id, password and host address for Web server
The directory path to account on Web server
The URL for homepage
Construction kits such as Miscrosoft Frontpage or Macromedia Dreamworks

Q.95. Are cookies harmful?

No. They are just short text strings. They can often make browsing better by allowing a server to recall any customized information set by the user.

Q.96. Can a company read my personal information from hard disk with a cookie?

No. Only the cookie that is sent in the first place is returned to the server. It is not modified or manipulated in any way.

Multiple Choice

1. **Collection of millions of computer interlinked to each other is called:**
   a. Interlink b. Internet c. Collection d. Group
2. **What is the domain name in the following address? name@server.com**
   a. Com b. Server.com c. Name d. server
3. **Which of the following is an Internet Protocol?**
   a. EFT b. MAN c. TCP/IP d. EDI
4. **TCP/IP stands for:**
   a. Transmission Control Protocol/Internet Protocol
   b. Transfer Collector protocol/International Protocol
   c. Transitional Covered Protocol/Intranet Protocol
   d. Transferred Collected Protocol/Internet Protocol
5. **The USA Defense Department initiated the Internet in:**
   a. 1969 b. 1976 c. 1978 d. 1980
6. **ARPA stands for:**
   a. American Rational Program Association
   b. Advanced Residential Programming Association
   c. Advance Research Project Agency
   d. Arizona's Russian Program of Association
7. **DARPA is the abbreviation of:**
   a. Defensive Association of Russian Policy with America
   b. Dared Association of Republic Partition
   c. Defense Advanced Research Projects Agency
   d. Defensive Advanced Regional Political Agency
8. **Who owns the Internet?**
   a. The U.S. government b. Pakistan Telecommunications
   c. The United Nations (U.N.) d. None
9. **Computers in internet are connected through:?**
   a. Fiber-optics cables b. Telephone lines c. Satellite c. All of these
10. **Software used to connect a PC and the server of ISP is called:**
    a. Communication software b. Dial up software c. Modem software d. ISP's link
11. **What is the protocol in the URL "http://www.Microsoft.com"?:**
    a. www b. http c. Microsoft d. com
12. **ISP stands for:**
    a. International Service Provider b. Internet Service Provider
    c. Interlinked Services Provision d. Intranet's Service Party
13. **Two popular types of e-commerce are banking and:**
    a. Shopping b. E-Mailing c. Chatting d. Messaging
14. **Software used to access the Internet is called:**
    a. Browser b. Packaged c. Spreadsheet d. HTTP

WWW.ITECVIEW.COM©

15. **A collection of documents or web pages stored on computes around the world is called:**
a. Internet b. Web c. LAN d. FTP
16. **Web pages are written in which type of language?**
a. Hypertext b. Multimedia c. Hyperlinks d. Assembly
17. **Web pages are connected to one another using:**
a. Hyperlinks b. HTTP c. Interlink d. Multimedia
18. **A collection of related web pages is called:**
a. Web-link b. Web site c. Internet d. Linking
19. **URL stands for:**
a. Universal Research Limit b. United Russian Language
c. Universal Resource Locator d. Uniform Resource Locator
20. **A Web browser looks for the Uniform Resource Locator, which is the Web site's**
a.. code b. File extension c. Address d. Protocol
21. **E-mail stands for:**
a. Electric-mail b. Electronic mail c. Elective-mail d. Elaborated-mail
22. **The format of an email address is:**
a. name#server.com. b. name&server.com.
c. name@server.com d. name$server.com
23. **Symbol that separates parts on an e-mail address is:**
a. $ b. # c. @ d. ^
24. **To carry out financial & business dealings using Internet is called:**
a. E-Commerce b. E-Customer c. E-Marketing d. E-Financing
25. **Which of the following is not an example of e-commerce?**
a. Online banking b. Online Shopping c. Online chatting d. Online Educations
26. **A program used to connect to a remote computer on Internet is called:**
a. Intranet b. Tele-net c. Telnet d. RealNet
27. **FTP stands for:**
a. File Transfer Protocol b. First Tele Program c. Find Tele Path d. Failed Transmit Pole
28. **Which of the following is an example of FTP?**
a. Personal b. Web server b. SQL d. Cute-FTP
29. **In the Web address, http://www.prenhall.com, http is the ____.**
a. URL b. domain name c. protocol d. extension
30. **Which is the other name given to browsers?**
a. Universal client's b. Linkers c. Web Clients d. Servers
31. **A computer used to store information for users on the Internet is called:**
a. Web Server b. Web Client c. Web application d. domain name
32. **The most widely protocol that is used for sending & receiving e-mail is:**
a. HTTP b. TCP/IP c. SMTP d. NCP
33. **An Internet Service Provider gives you access to:**
a. E-mail. b. FTP c. Chat lines. d. All of the above
34. **The three-character extension that designates a URL as a commercial site is:**
a. gov b. com c. org d. mil
35. **In a URL, a resource is a:**
a. Browser b. Server c. Domain Name d. Computer File
36. **FTP Facility is used to:**
a. Transfer Protocol b. Transfer File c. Connect to the Internet d. Self-check computer
37. **Paknet, Comsat and Cyber Net are examples of:**
a. Mobile Companies b. ISP c. Hardware Sellers d. Software developers
38. **This is not part of a complete URL.**
a. Browser b. Path c. Resource name d. Protocol

39. **Services on the Internet include**
a. World wide web b. FTP c. e-mail d. all of these

40. **Electronic commerce (e-commerce) is the process of:**
a. Sharing of business information.
b. Maintaining business relationships.
c. Conducting business transactions through the use of telecommunications networks.
d. All of the above are correct

41. **This is not one of the protocols used to access information.**
a. FTP b. HTTP c. HTML d. Gopher

42. **Which of the following is required when typing a URL?**
a. Inserting spaces between the protocol and server
b. Using periods to separate the URL components
c. Typing http://
d. Inserting slash marks between the path and resource name

43. **Copying data from Internet to computer is called:**
a. Uploading b. Downloading c. Transferring d. None

44. **Transferring information from computer to Internet is called:**
a. Downloading b. Down Seizing c. Uploading d. Pasting

45. **Which browser shows the formatted text in different fonts & colors?**
a. Text-based b. Interactive c. default d. Graphic-based

46. **This is not a feature found in the two most popular browser programs.**
a. Program icon b. Navigation buttons c. Address toolbar d. Menu bar

47. **A search engine is:**
a. Software to searches for engines.
b. A company that does searches for you.
c. Web site that sells products and services
d. Website that looks through databases for matching criteria

48. **When you conduct a search, the computer sends out a:**
a. Request b. Snake. c. Spider d. Lizard.

49. **This is not one of the leading search engines.**
a. Google b. Yahoo c. Excite d. Netscape Navigator

50. **Which of the following can only shows the text and not the graphics?**
a. IE b. Netscape Navigator c. Lynx d. Netscape Communicator

51. **Shortcut to refresh a web page is**
a. F2 b. F7 c. F1 d. F5

52. **Page that is being downloaded can be cancelled by clicking:**
a. Cancel button b. Stop Button c. Remover d. Re-loader

53. **Progress of browser while visiting a page is shown through:**
a. Address bar b. Status bar c. Menu bar d. Standard toolbar

54. **Option for saving any web page exists in:**
a. File menu b. Edit menu c. Tools menu d. Favorites menu

55. **When you send another file with your e-mail message, it is called:**
a. Enclosure b. Supplement c. Attachment d. Heavy mail

56. **The computer that manages sending and receiving of emails is called:**
a. Mail Server b. Web Browser c. HTTP d. Outlook

57. **A discussion group to exchange information such as news, business, science and computer is called:**
a. Tele-system b. Teleprocessing c. Newsgroup d. Chatting

58. **Which button contains the list of previously visited web pages?**
a. History b. Refresh c. Address d. Favorite

59. **Recently visited Web sites are stored in:**
a. Cookies folder b. History folder
c. Temporary Internet Files folder d. Temp folder

60. **In 1986, the National Science Foundation provided its _____ of high-speed transmission circuitry to carry the bulk of Internet traffic**
a. Hu b. LA c. Backbone d. World Wide Web

61. **Tim Berners-Lee is best known as:**
a. The creator of the Internet. B. The founder of Netscape.
c. The CEO of CERN. D. The creator of the Web.

62. **A message sent over the Internet is composed of uniformly sized:**
a. Packets. b. Browsers. C. Links. D. Protocols.

63. **Packeting for Internet transmission is handled through:**
a. Hyperlinks. B. FTP. C. TCP. d. IP.

64. **The protocol responsible for routing packets to the correct address is:**
a. HTTP. B. FTP. C. TCP. D. IP

65. **Which of the following is a top-level domain?**
a. Http b. com c. HTML d. URL

66. **A plug-in is:**
a. A computer program that plugs into an expansion card or port.
b. A special Web site that allows users to gain additional services.
c. A software component that enhances the browser's functionality
d. A small program embedded in a Web page that runs when the browser loads the page.

67. **The term referring to software used by search engines to follow links across the Web and set up indexes to pages found is:**
a. Applet. B. Spider c. browser. D. Indexer.

68. **Internet environment in which users read and post messages on a particular topic is called:**
a. Usenet. b. FTP. C. Telnet. d. e-mail.

69. **The Internet-based protocol for uploading and downloading files on the Internet is:**
a. Usenet. B. FTP c. Telnet. D. e-mail.

70. **You will use FTP to:**
a. Send email attachments to other users over the Internet.
b. Upload or download files over the Internet.
c. Remotely log into a host computer via the Internet.
d. Post and read newsgroup messages via the Internet.

71. **You will use Usenet to:**
a. Send email attachments to other users over the Internet.
b. Upload or download files over the Internet.
c. Remotely log into a host computer via the Internet.
d. Post and read newsgroup messages via the Internet.

72. **You will use Telnet to:**
a. Send email attachments to other users over the Internet.
b. Upload or download files over the Internet.
c. Remotely log into a host computer via the Internet.
d. Post and read newsgroup messages via the Internet.

73. **The category of e-commerce that home users are most likely to engage in is**
a. FTP. b. B2C c. SSL. d. B2B.

74. **The category of e-commerce that takes place between a construction company and its materials supplier is called:**
a. bricks-and-clicks. b. B2C. c. intranets. d. B2B.

75. The term portal refers to:
a. a special form of banner ad that links to the Web site for the company being advertised.
b. a personalized Web page that provides access to Web sites of interest and presents customized advertisements to the user.
c. a dedicated Internet connection between a company and its suppliers.
d. a secure firewall, which is like a doorway that needs a special key in order to open.

76. When supplier companies wish to connect their customer companies to their own private and isolated network using Web technology, they employ a(n):
a. LAN. b. Intranet. c. VPN. d. Extranet.

77. The TCP/IP standards belong to _____.
a. Microsoft b. Sun Microsystems c. Xerox d. None

78. An IP address is a(n) _____.
a. String of four numbers separated by periods
b. String of names separated by dots to specify the exact Internet location
c. Open standard
d. None

79. Which of the following is NOT a top-level domain?
a. .edu b. .jp c. .gov d. .url

80. Which a new top-level domain name created by the Internet Ad Hoc Committee?
a. .air b. .alt c. .me d. .name

81. Which of the following is a restricted domain name?
a. .edu b. .org c. .gov d. .net

82. Computers are connected to the Internet through:
a. Direct connections.
b. Dial-up connections through modems.
c. Broadband connections through high-speed alternatives to modems.
d. All of the above

83. Another name for a T1 line is _____.
a. E1 b. XrL4 c. G3 d. X10

84. The URL contains all of the following EXCEPT the _____.
a. Protocol that must be used to access information b. domain name
c. Name of the resource d. user name

85. Real-time audio or video broadcasts do not include:
a. MP3 music b. TV broadcasts c. Radio broadcasts d. None

86. Webcasts transmit :
a. Video clips b. Streaming audio c. Streaming video d. Both b and c

87. To play some media files, a(n) _____ may be needed.
a. Add-to b. plug-in c. helper d. Both b and c
e. Both b and d are correct

88. The small files that are stored on visitors' hard disks are called _____.
a. Tokens b. Frames c. Cookies d. Packets

Answers

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

49. d	50. c	51. d	52. b	53. b	54. a	55. c	56. a
57. c	58. a	59. b	60. c	61. d	62. a	63. c	64. d
65. b	66. c	67. b	68. a	69. b	70. b	71. d	72. c
73. b	74. b	75. b	76. d	77. d	78. a	79. d	80. d
81. a	82. d	83. a	84. d	85. a	86. d	87. e	88. d

Fill in the Blanks

1. DARPA is the abbreviation of _____.
2. SMTP stands for _____.
3. A special language used for wireless remote phones is called _____.
4. ISP stands for _____.
5. _____ is a large group of millions of computers all connected to one another in world.
6. Many companies use the Internet to _____their products.
7. To establish connection to the Internet, a computer, a web browser, _____, _____ and _____ are required.
8. ____ is used to establish a connection between personal computer and a server of ISP.
9. Most of sites are part of _____.
10. Computers used to find information are called _____.
11. Just because of _____video-conferencing has been made possible.
12. _____ is used to gain information about news, product, topic or anything on Internet.
13. FTP stands for _____.
14. A web server is much faster & _____than _____computer.
15. _____ must be all time connected to internet.
16. Web pages are connected to one another using _____.
17. _____ is a group of web pages following same theme & connected through hyperlinks.
18. org stands for _____ in www.pakweb.org.pk.
19. _____ protocol is used on net for sending files from one place to another.
20. To save any audio, video or data file from Internet is called _____.
21. In www.pageresource.com, page resource is the _____.
22. In any URL, presence of words like .pk, .uk, .au indicates _____.
23. In an e-mail @ is called _____.
24. @ symbol in an email address separates _____ from the name of _____.
25. Search engines basically have underlying _____through which information is obtained.
26. It is always a good idea to use different _____to broaden your findings on Internet.
27. E-Commerce stands for _____.

Answers

1. Defense Advanced Research Projects Agency		2. Simple Mail Transfer protocol
3. WAP	4. Internet Service Provider	
5. Internet	6. Sell	7. ISP, Dialup software, Modem
8. Dialup Software	9. WWW	10. Servers

WWW.ITECVIEW.COM©

11. Internet	12. Search Engines	13. File Transfer Protocol
14. Powerful, Ordinary	15. Web Servers	16. Hyperlinks
17. Web Site	18. Organization	19. FTP
20. Downloading	21. Domain Name	22. Country
23. At	24. Email ID, Email Providing Site	
25. Database	26. Search Engines	27. Electronic Commerce

True / False

1. Internet is a global network of computers
2. PTCL is the owner of Internet
3. Dialup software is used to establish a connection between a personal computer and a server of an ISP.
4. The first Web browsers were text-only browsers
5. A complete URL has three parts.
6. A collection of related web pages is called browser
7. Web pages are created in hypertext language.
8. HTML documents will have the .html or .htm extension
9. The URL is the web address for any given web document.
10. Email stands for Electronic Mail.
11. Every email account has unique address
12. myemail@hotmail.com is invalid email address
13. The symbol @ separates the account ID from the name of the e-mail service provider.
14. FTP is specially designed for professionals
15. ISP is a company that provides Internet connections
16. Comsats and Cybernet are examples of Browser.
17. TELNET is a program that is used to connect to a remote computer on Internet.
18. A browser is a software that is used to view web pages
19. The most commonly used text-based browser is Internet Explorer.
20. Graphical browser can play sound and video
21. The exchange of data on Internet is based on client-server mode
22. FTP means to carry out financial and business dealings using Internet
23. Search engine is software that provides the facility to find the required websites on a particular topic.
24. Google and Yahoo are examples of web browser.

Answers

1. T	2. F	3. T	4. T	5. F	6. F
7. T	8. T	9. T	10. T	11. T	12. F
13. T	14. T	15. T	16. F	17. T	18. T
19. F	20. T	21. T	22. T	23. F	24. F

Part - 2

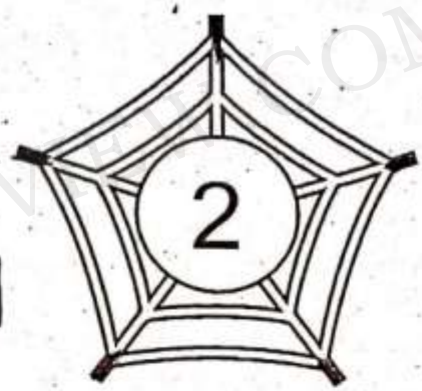
Hyper Text Markup Language

Includes:

Chapter 2	Introduction to HTML
Chapter 3	Lists, Images and Tables
Chapter 4	Hyperlinks and Image Maps
Chapter 5	Frames and Forms

WWW.ITECVIEW.COM©

HyPER TEXT Mark UP LANGUAGE



Introduction to HTML

2.1 HTML

HTML stands for **Hyper Text Markup Language**. It was developed in early 1990s. It is a formatting language used to develop web pages. Hypertext means that HTML marks certain words in the document as **links** to other parts of the same document or other document. When the user clicks on a link, browser stars displaying the document attached to that link.

HTML is not a programming language. It is used to define text, graphics, audio and video in web pages. It uses **tags** or **markups** to specify how the contents of web page will be displayed. HTML code can be written in any text editor like Notepad.

HTML was developed a few years ago as a subset of SGML. SGML stands for **Structured Generalized Mark-up Language**. It also describes formatting and hypertext links, and different components of a document. It is basically an application-oriented document format. SGML is not ideal for transmission across the Internet to different types of computers and browser applications.

HTML is simpler than SGML. It was designed for use over the Internet in the form of Web pages. HTML is more suited for this purpose. It is used to design and create pages for different types of users of the Web.

2.1.1 Tags and Attributes

HTML commands are known as **tags**. The HTML tags are surrounded by the two characters < and >. These characters are called **angle brackets**. HTML tags normally come in pairs like **<head>** and **</head>**. The first tag in a pair is called the **start tag** and the second tag is called the **end tag**.

An **element** is a command that tell a browser to do something such as <H1>. It is a fundamental component of an HTML document. Different tags indicate different elements i.e. HEAD, TABLE etc. The text between the start and end tags is called **element content**. HTML tags are not case sensitive. It means that **<head>** is same as **<HEAD>**.

Some elements may also include some attributes. An **attribute** is the additional information about the element that is included inside the start tag. Some attributes are compulsory and some are optional.

2.1.2 Benefits of HTML

Some advantages of HTML are as follows:

- **Simplicity:** HTML is a simple but powerful language. Any person can use HTML easily to create web pages.
- **Platform Independent:** HTML is platform independent. It works on any platform. It means that HTML documents created for one type of computer system can be used on different types of computer systems.
- **Easy Navigation:** HTML web pages are linked together using hyperlinks. It provides easy navigation between different documents.
- **Attractive and Easier to Read:** HTML allows the use of graphics and animation. It delivers richer content and arranges the content creatively.

2.1.3 Limitations of HTML

Some limitations of HTML are as follows:

- **No Programming Capabilities:** HTML provides no programming capabilities. It only provides formatted text, pictures and sounds etc.
- **Static:** HTML can only be used to create static web pages.
- **Fixed Tag Set:** HTML does not provide the facility to create customized tags. It only provides a predefined set of tags. It is not possible to extend HTML.
- **Not Reusable:** HTML code is used for specific pages and cannot be reused. The user cannot display the same HTML document in a different manner without creating different documents each time.

2.2 Web Page Creation using HTML

The basic requirements to create web pages using HTML are as follows:

- **Text Editor:** It is used to write HTML commands. Any simple text editor like Notepad or WordPad can be used for this purpose.
- **Web Browser:** It is used to test the web pages. Internet Explorer and Netscape Navigator are two popular web browsers.

2.2.1 Basic Structure of HTML Document

The basic structure of an HTML document is as follows:

```
<HTML>

    <HEAD>
    }  ← Head Section
    </HEAD>

    <BODY>
    }  ← Actual Document
    </BODY>

</HTML>
```

In every HTML document, **<HTML>** and **</HTML>** tags indicate the beginning and end of the document. These tags inform the browser that the document is an HTML file. All other HTML tags must be written within these tags.

The HTML document can be divided into two sections:

- Head Section
- Body Section

1. Head Section

<HEAD> and </HEAD> tags identify the heading or title of document. The tags **<TITLE>** and **</TITLE>** can appear only in this section. The text between the title tags appears as title in the browser's window. Its length should not be more than 60 characters.

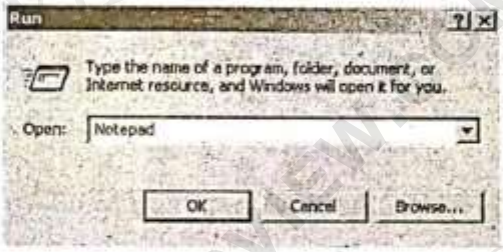
2. Body Section

The body of an HTML document contains the text that is displayed on a web page. The body section is enclosed within **<BODY>** and **</BODY>** tags. It gives a description of the document layout and structure.

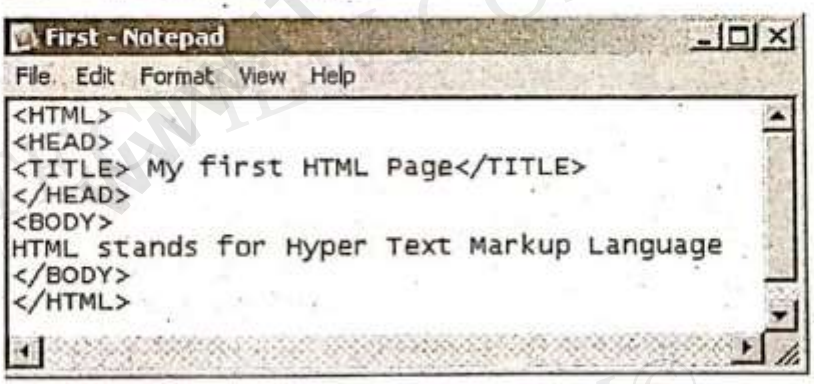
Practice 2.1

Write HTML code that displays a simple message on the web page.

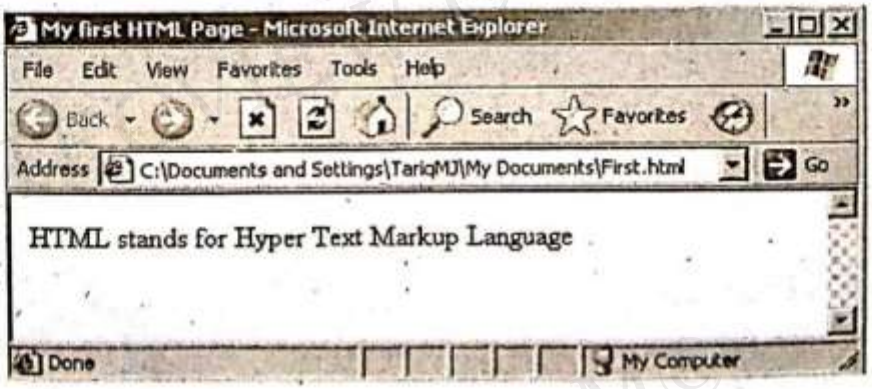
1. Click **Start** button and select **Run** option. A dialog box will open.



2. Type "Notepad" in **Open** textbox and click **OK**. Notepad will open.
3. Type the following code in Notepad.



4. Save the file as "First.html" on hard disk.
5. Double click the file. The web page will appear in the browser as follows:



Attributes

Some important attributes of <BODY> tag are as follows:

Attribute	Description
BGCOLOR	It specifies the background color of the web page.
BACKGROUND	It specifies the background picture of the web page.
TEXT	It specifies the color of the text displayed on the web page.

Example

```
<BODY BGCOLOR="red" TEXT="yellow">
```

In the above example, the background color of the web page is red and text color is yellow. The color name or its hexadecimal code can be given as the value of the attribute.

```
<BODY BACKGROUND="MyPic.gif" TEXT="green">
```

In the above example, "MyPic.gif" will appear in the background of the web page. It should be stored in the same directory in which the web page is stored. If the picture is stored in some other directory, the complete path of the picture should be given in the attribute.

2.3 Headings

Headings are very important tags in the BODY of HTML document. It is used to display different types of heaings. The opening tag for a heading is <H*n*> and the closing tag is </H*n*> where *n* is the size of the heading. Its value is from 1 to 6. The value 1 represents the largest and the value 6 represents the smallest heading size.

Attributes

Some important attributes of heading tag are as follows:

Attribute	Description
ALIGN	It specifies the alignment of the heading. The possible values are **center, left** and **right**. The default value is **left**.

Example

```
<H1 ALIGN="center">Hello Heading</H1>
```

The above line will display the heading in the center of the page.

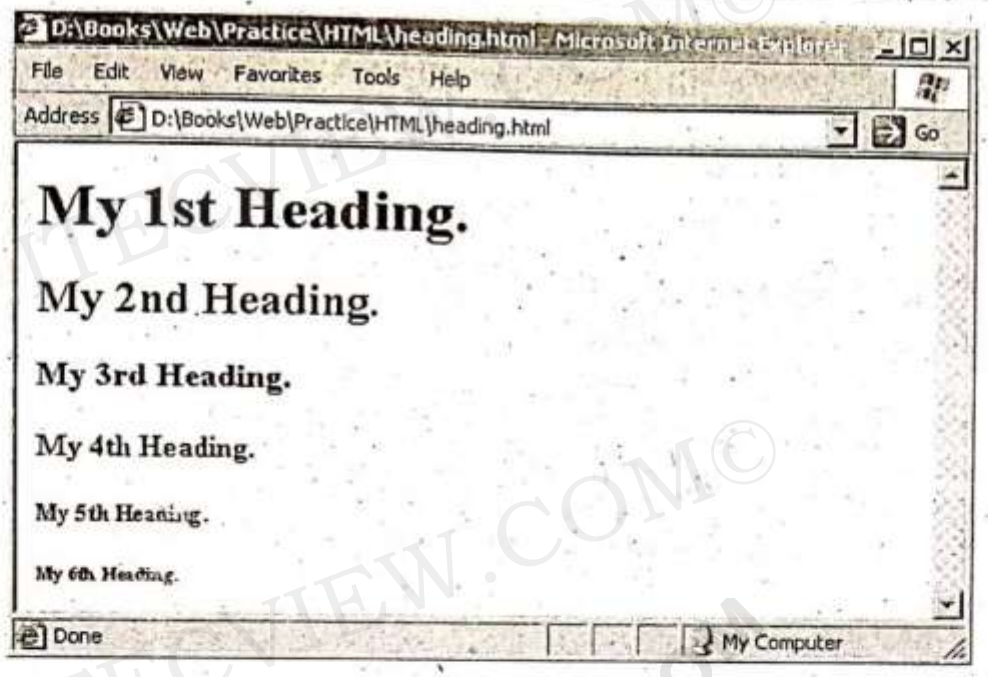
Practice 2.2

Write HTML code that displays six types of heading in a web page.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H1> My 1st Heading. </H1>
<H2> My 2nd Heading. </H2>
<H3> My 3rd Heading. </H3>
<H4> My 4th Heading. </H4>
<H5> My 5th Heading. </H5>
<H6> My 6th Heading. </H6>
</BODY>
</HTML>
```

2. Save the file as "heading.html".
3. Double click the file. The web page will appear in the browser as follows:



2.4 Paragraphs

Paragraphs are used to add text to a document in such a way that it will automatically adjust the end of line to suit the window size of the browser. Each line of text will stretch the entire length of the window. Paragraphs are defined with the **<P>** tag. The use of ending tag **</P>** is optional.

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

HTML automatically adds an extra blank line before and after a paragraph.

Attributes

Some important attributes of paragraph tag are as follows:

Attribute	Description
ALIGN	It specifies the alignment of paragraph. The possible values are **center**, **left** and **right**. The default value is **left**.

Example

```
<P ALIGN="right">I am a paragraph.</P>
```

The above example will align the paragraph to the right side of the page.

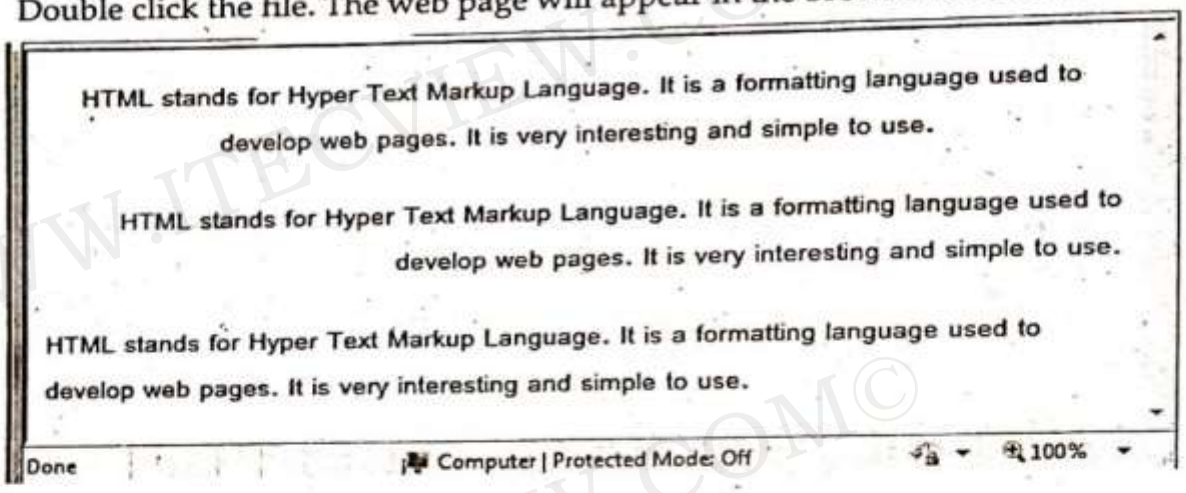
Practice 2.3

Write HTML code that displays three paragraphs with different paragraph alignments.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<P ALIGN="center">HTML stands for Hyper Text Markup Language. It is a formatting language used to develop web pages. It is very interesting and simple to use. </P>
<P ALIGN="right">HTML stands for Hyper Text Markup Language. It is a formatting language used to develop web pages. It is very interesting and simple to use. </P>
<P ALIGN="left">HTML stands for Hyper Text Markup Language. It is a formatting language used to develop web pages. It is very interesting and simple to use. </P>
</BODY>
</HTML>
```

2. Save the file as "para.html".
3. Double click the file. The web page will appear in the browser as follows:



2.5 Line Breaks

Line breaks are used to decide where the text will break on a line or continue to the end of the window. It can be used to move the control to the next line. The **
** tag is used to end a line without inserting a new paragraph. The **
** has no closing tag.

Example

The following text will be displayed in single line.

```
Sentence One.
Sentence Two.
```

The following text will appear in two different lines.

```
Sentence One. <BR>
Sentence Two. <BR>
```

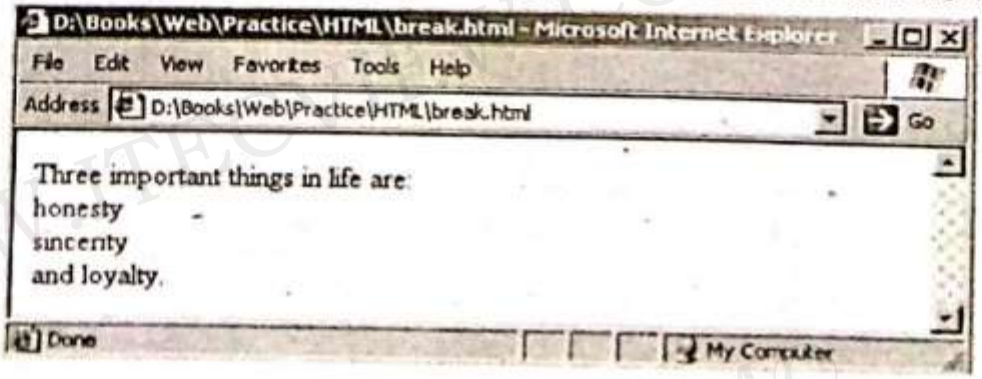
Practice 2.4

Write HTML code that displays text in four lines using <BR> tag.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
    Three important things in life are: <BR>honesty<BR>sincerity<BR>and loyalty.
</BODY>
</HTML>
```

2. Save the file as "break.html".
3. Double click the file. The web page will appear in the browser as follows:



2.6 Horizontal Line

Horizontal lines are used to separate different areas of a web page. It is used to display a horizontal line in the page. The tag for horizontal line is **<HR>**. This tag has no closing tag.

Attributes

Some important attributes of HR tag are as follows:

Attribute	Description
ALIGN	It specifies the alignment of the line. Possible values are left, right and center. Line appears in center by default.
SIZE	It specifies the size of the line. The default size is 2 pixels.
WIDTH	It specifies the width of line. Size can be given in pixels or percentage. The width of line is according to the width of webpage by default.
NOSHADE	It turns off the shading of the line.
COLOR	It specifies the color of the line.

Example

```
<HR WIDTH=50 ALIGN="center" SIZE=4>
<HR WIDTH=50% SIZE=7 NOSHADE>
```

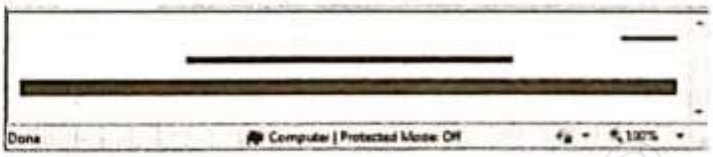
Practice 2.5

Write HTML code that displays three horizontal lines with different attributes.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<HR WIDTH=50 ALIGN="right" SIZE=4 COLOR="red">
<HR WIDTH=50% ALIGN="center" SIZE=7 COLOR="brown">
<HR WIDTH=100% SIZE=15 COLOR="blue">
</BODY>
</HTML>
```

2. Save the file as "line.html".
3. Double click the file. The web page will appear in the browser as follows:



2.7 Text Formatting

Different tags to format text are as follows:

Bold

<B> tag is used to display text in bold face style. It is closed by **</B>** tag.

Example

```
<B> Formatting makes things cool. </B>
```

Italic

<I> tag is used to display text in italic style. It is closed by **</I>** tag.

Example

```
<I> Italic text looks stylish.</I>
```

WWW.ITECVIEW.COM©

Underline

<U> tag is used to display text as underlined. It is closed by </U> tag.

Example

```
<U> Underlined text looks prominent.</U>
```

Superscript

^{tag is used to display text as superscript. It is closed by} tag.

Example

```
N<SUP>2</SUP>
```

The above example will display N^2

Subscript

_{tag is used to display text as subscript. It is closed by} tag.

Example

```
N<SUB>2</SUB>
```

The above example will display N_2

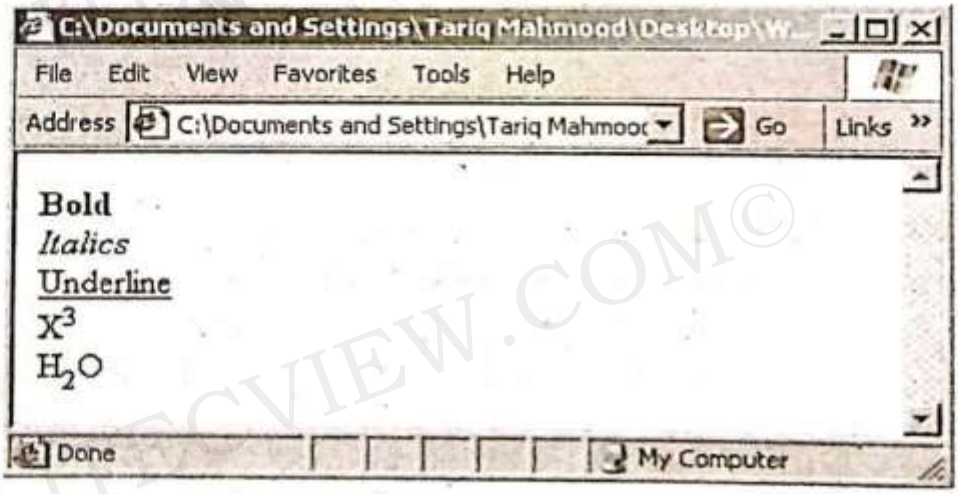
Practice 2.6

Write HTML code that displays text in different formatting.

1. Open Notepad and type the following code:

```
<HTML>
<B>Bold</B><BR>
<I>Italics</I><BR>
<U>Underline</U><BR>
X<SUP>3</SUP><BR>
H<SUB>2</SUB>O<BR>
</HTML>
```

2. Save the file as "Text.html".
3. Double click the file. The web page will appear in the browser as follows:



2.7.1 Fonts

<FONT> tag is used to specify the characteristics of font. Different characteristics include the typeface, size and color. It is closed with </FONT> tag.

Attributes

Some important attributes of FONT tag are as follows:

Attribute	Description
FACE	It specifies the font of the text. The possible values are "Arial", "Arial Black", "Courier" and "MS sans Serif" etc.
SIZE	It specifies the font size of the text. It can be from 1 to 7. The default size is 3. The size can also be given as relative to the default size. The size 1 displays 8pt size, size 2 displays 10pt and so on.
COLOR	It specifies the color of the text. It can be given as color name or hexadecimal value of the color.

The following tables shows the color name and its hexadecimal value:

Color	Hexadecimal Value	Color	Hexadecimal Value
Red	#FF0000	Gray	#808080
Green	#00FF00	Dark Red	#800000
Blue	#0000FF	Dark Green	#008000
White	#FFFFFF	Yellow	#FFFF00
Black	#000000		

Example

```
<FONT SIZE=+2 FACE="Verdana">Verdana</FONT><BR>
<FONT SIZE=+2 FACE="Arial">Arial</FONT><BR>
<FONT SIZE=+2 FACE="Helvetica">Helvetica</FONT><BR>
<FONT SIZE=+2 FACE="Impact">Impact</FONT><BR>
<FONT SIZE=+2 FACE="Comic Sans MS">Comic Sans MS</FONT><BR>
```

Practice 2.7

Write HTML code that displays text of different sizes and colors.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<FONT FACE="Arial" SIZE="+1">One Size Larger</FONT> <BR>
<FONT SIZE="-1" COLOR="red">Small & Colorful</FONT>
</BODY>
</HTML>
```

2. Save the file as "Font.html".
3. Double click the file. The web page will appear in the browser as follows:

2.7.2 <BIG> and <SMALL> Tags

<BIG> and <SMALL> tags change the relative size of text with respect to surrounding text. <BIG> tag makes the text a little bigger and <SMALL> tag makes the text a little smaller.

However, different browsers display the effect of these tags differently. In Internet Explorer, text formatted with <BIG> tag is always one size larger than the surrounding text. The tag has no effect if the size of surrounding text is 7. The text formatted with <SMALL> is one size smaller than surrounding text. The tag has no effect if size of surrounding text is 1.

Example

The following line will display the text one size larger than normal size:

```
<BIG>Hello World</BIG>
```

The following line will display the text one size smaller than normal size:

```
<SMALL> Hello World </SMALL>
```

2.7.3 Preformatted Text

This tag is used to display the text as the user has typed. HTML ignores extra line breaks and the text apears in single line. The text is placed between **<PRE>** and **</PRE>** tags.

Example

The following text will appear in single line. HTML ignores extra line breaks so the text appears in single line.

```
Say this city has ten million souls                    Can you believe it?
No body can believe it
```

The following text will appear in the browser as it appears below:

```
<PRE>
Say this city has ten million souls                    Can you believe it?
No body can believe it
</PRE>
```

2.8 Spaces

If an HTML document has many spaces, additional spaces are removed automatically. If the user inserts five spaces, four of them will be removed. Additional spaces can be inserted in the text by using character entity ** ** that stands for **non breaking space.**

Example

```
<B> Hello    World<B>
```

The above example will display **Hello World**

```
<B> Hello          World</B>
```

The above example will display **Hello World**

Common Character Entities

Some common character entities are as follows:

Entity Name	Description	Output
	space	
<	less than	<
>	greater than	>
&	ampersand	&
"	quotation mark	"
¢	cent	¢
£	pound	£

¥	yen	¥
§	section	§
©	copyright	©
®	registered trademark	®
×	multiplication	×
÷	division	÷

2.9 <DIV> Tag

<DIV> tag is used to define a section in an HTML document. It can be used to group large section of HTML elements together and format them with cascading style sheet. The closing tag **</DIV>** is used to indicate the end of section.

Attributes

Some important attributes of <DIV> tag are as follows:

Attribute	Description
ALIGN	It specifies the alignment of the DIV. The possible values are center, left and right. The default value is left.
STYLE	It is used to include inline cascading style sheet

Example

```
<DIV STYLE="background-color:blue;text-align:right">
        <P>Hello World!</P>
</DIV>
```

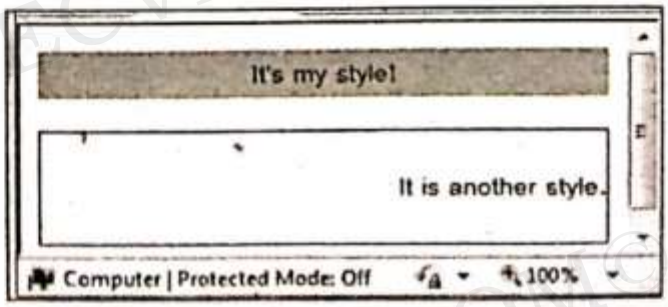
Practice 2.8

Write HTML code that displays two sections of text with different formatting with inline CSS using <DIV> tag.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<DIV STYLE="background-color:orange;text-align:center">
        <P>It's my style!</P>
</DIV>
<DIV STYLE="border:1px solid black;text-align:right ">
        <P>It is another style.</P>
</DIV>
</BODY>
</HTML>
```

2. Save the file as "Div.html".
3. Double click the file. The web page will appear in the browser as follows:



2.10 <SPAN> Tag

<SPAN> tag is used to provide additional structure to HTML document. It is used to group and apply styles to inline elements. It can be used to give different characteristics to specific parts of other elements. The HTML span element begins with <SPAN> tag and ends with </SPAN> tag. A style can be applied to the content placed inside the SPAN tag using STYLE attribute.

Example

```
<P>This is a paragraph
        <SPAN STYLE="color:#FF0000;">This is a paragraph</SPAN>
This is a paragraph</P>
<P><SPAN STYLE="color:#6600ff;">This is another paragraph</SPAN></P>
```

Practice 2.9

Write HTML code that displays a paragraph using <SPAN> tag. It uses style attribute to change the characteristics of part a paragraph element.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<P STYLE="text-align: 'center'; font: '16pt courier'; color: 'blue'">
The color of the sky is blue, but if you look at the <SPAN STYLE="color: 'green'">trees, they
are green</SPAN>. </P>
</BODY>
</HTML>
```

2. Save the file as "Span.html".
3. Double click the file. The web page will appear in the browser.

2.11 <MARQUEE> Tag

HTML provides a special feature to display scrolling and animated text in the Web page. This tag is supported by Internet Explorer, Firefox and Netscape 7 and above. Netscape has a blink tag to display blinking text but Internet Explorer does not support blink tag.

Marquee tag can be used to scroll text or image in different directions. The text placed inside the marquee tag scrolls horizontally across the screen or the web page. The scrolling text is normally used to give messages or display some special announcements to the visitors.

Attributes

Attribute	Description
BEHAVIOR	It indicates how the contents will scroll. The possible values are **scroll, slide** and **alternate**. The **scroll** value is the default value. It indicates that the content scrolls off the edge of marquee area and then reappears on the other side. The **slide** value is almost the same. It also indicates that when the content reaches the left edge, it stops without scrolling off. The **alternate** value makes the content bounce back and forth. All contents remain visible all the time.
BGCOLOR	It is used to set the background color of the marquee. Its value can be given as a combination of red, green and blue or name of the color. The hexadecimal value of the color can also be used.

DIRECTION	It indicates the direction to which the marquee will scroll. The possible values are up, down, left and right. The default value is left. The left value indicates that the marquee starts at the right and moves to left across the page. The right value indicates that the marquee starts at the left and moves to right across the page
HEIGHT	It defines the height of marquee. It is given in pixels or percentage.
HSPACE	It is used to set the horizontal space to the left and right of the marquee. Its value is given in pixels.
LOOP	It specifies the number of times the marquee will scroll. The default value is INFINITE.
SCROLLDELAY	It is used together with SCROLLAMOUNT to set the speed of the scrolling. Marquee moves the content by displaying the content and then delays for some period of time specified in SCROLLDELAY. Its value is given in milliseconds. The default delay is 85.
SCROLLAMOUNT	It is used together with SCROLLDELAY to set the speed of the scrolling. Marquee moves the content by displaying the content and delays for some period of time. It then displays the content in a new position specified in SCROLLAMOUNT that sets the size in pixels of each jump. A higher value for SCROLLAMOUNT makes the marquee scroll faster. The default value is 6.
WIDTH	It defines the width of marquee. It is given in pixels.
VSPACE	It is used to set the vertical space at the top and bottom of the marquee. It is given in pixels.

Example

```
<MARQUEE> Marquee</MARQUEE>
<MARQUEE LOOP=2>Hello</MARQUEE>
```

Practice 2.10

Write HTML code that displays three marquees with different attributes.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<FONT SIZE="8" FACE="courier" COLOR=blue>
<MARQUEE BEHAVIOR=SLIDE HEIGHT=25 WIDTH=300 BGCOLOR=red>Hello World!
</MARQUEE></FONT>
<FONT SIZE="4" FACE="courier" COLOR=green>
<MARQUEE WIDTH=100% BEHAVIOR=SCROLL HSPACE=15 VSPACE=25
BGCOLOR=yellow>Hello World! </MARQUEE></FONT>
<FONT SIZE="4" FACE="courier" COLOR=blue>
<MARQUEE DIRECTION=UP WIDTH=100% BEHAVIOR=SCROLL BGCOLOR=yellow
LOOP=3>Hello World! </MARQUEE></FONT>
</BODY>
</HTML>
```

2. Save the file as "Marquee.html".
3. Double click the file. The web page will appear in the browser.

2.12 HTML Comments

<COMMENT> tag is used to comment HTML code. It is recommended to use <!--....--> to comment the tags. This tag is compatible to all browsers. The comments can be inserted in any section of an HTML document. The comments are ignored by the browsers. They are primarily used to explain the code in an HTML document etc.

Example

```
<P>Following are comments </P>
<COMMENT>Comments line </COMMENT>
<!—Another comments line -->
```

2.13 XML

XML stands for **Extensible Markup Language**. It means that the user can define his own set of tags. It makes XML more flexible than HTML. XML tags represent the logical structure of data and can be interpreted and used in different ways by different applications. XML inherits the features of SGML and combines the features of HTML with it. The major framework is taken from SGML.

XML is a Meta language to describe other languages. It can be used to create a markup language for a specific purpose. XML resembles and complements HTML. XML describes data, such as student name, marks and address. HTML defines the tags that describe the display of data such as with bulleted list or table. XML allows the developers to define an unlimited set of tags thus bringing great flexibility to the users.

2.14 XHTML

XHTML stands for **extensible Hypertext Markup Language**. It combines the formatting strengths of HTML 4.0 and the data structure and extensibility strengths of XML. It uses the tags and attributes of HTML along with the syntax of XML. The first version of XHTML 1.0 was released in 2000. The **World Wide Web Consortium (W3C)** sets standards for XHTML and other internet languages. XHTML was created for two main reasons:

1. To create a more strict standard for making web pages and reduce incompatibilities between browsers
2. To create a standard that can be used on different devices without changes

2.14.1 Changes in XHTML from HTML

There are several main changes in XHTML from HTML:

- All tags must be in lower case
- All documents must have a doctype
- All documents must be properly formed
- All tags must be closed
- All attributes must be added properly
- The name attribute has changed
- Attributes cannot be shortened
- All tags must be properly nested

2.14.2 Flavors of XHTML 1.0

Three flavors of XHTML 1.0 are as follows:

1. XHTML Transitional
2. XHTML Frameset
3. XHTML Strict

1. XHTML Transitional

The most commonly used version of XHTML is 1.0. This version mostly resembles with HTML 4.0.1. It is the best choice when documents need to use HTMLs presentational elements or when pages need to be developed without using style sheets. This version is used to convert existing HTML pages to XHTML. It also does not provide support for frames.

2. XHTML Frameset

XHTML Frameset version should be used when the documents need to use the frame elements to divide the browser into multiple windows. It also supports the element set of XHTML Transitional.

3. XHTML Strict

XHTML Strict most closely represents the future of XHTML. The element set for XHTML Strict contains a subset of the elements from XHTML Transitional. However, it does not include support for strictly presentational elements and elements that will not be included in future versions of XHTML. In the future, XHTML documents will separate presentation from content and use style sheets to define presentation formatting such as font types, colors and styles. XHTML Strict should be used with **Cascading Style Sheets (CSS)**.

2.14.3 Validating XHTML Documents

Valid documents must be formed properly and should be according to the rules of a DTD. **DTD** stands for **Document Type Definition**. Document Type Definition is used to identify the type of markup language used in a web page.

1. XHTML Transitional

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

2. XHTML Frameset

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

3. XHTML Strict

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

Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
        <TITLE> Computer Science </TITLE>
</HEAD>
<BODY>
<B>Subject: </B> Introduction to IT <BR />
<B>Teacher Name: </B> Muhammad Naeem <BR />
<B>Class Timing: </B>  Monday- Wednesday, 2:00pm–5:00pm <BR />
<P /></BODY>
</HTML>
```

2.14.4 Differences between XHTML and HTML

The difference between XHTML and HTML is as follows:

- XHTML documents contain the XML and DOCTYPE declarations at the top of the document. XML declaration is optional but DOCTYPE declaration is required. The DOCTYPE declaration was optional in HTML. XHTML requires that all WebPages contain <HTML>, <HEAD>, <BODY> elements.
- XHTML documents must be well formed. However, HTML does not strictly require well formed documents.
- XHTML requires closing tag for every element. HTML does not require closing tags strictly.
- XHTML requires that all attributes should be enclosed in quotation marks. HTML does not require quotation marks. However, they are required around attribute values that consist of only letters, numbers and the characters dash, period, underscore and colon.
- XHTML elements and attributes are case sensitive but HTML elements and attributes are not.

Exercise Questions

1. Write HTML code that displays an essay of two paragraphs. The heading of essay should appear italic in the center. The first paragraph should be aligned left in red color. The second paragraph should be aligned right in green color. The heading and the paragraphs should be separated by a horizontal line.
2. Write HTML code that displays the following text in three separate lines:

- H_2SO_4
- A^N
- X^3Y^2

The text should appear bold, center-aligned and in blue color.

WWW.ITECVIEW.COM©

Lists, Images and Tables

3.1 Lists

HTML can display different items in the form of lists. Lists are used to present text in more readable form. Different types of lists are as follows:

- Ordered Lists
- Unordered Lists
- Definition Lists

3.1.1 Ordered Lists

An **ordered list** is a list of items in which each item is marked with a number or letter. Ordered list is also known as **numbered list**. <OL> tag is used to create an ordered list. It stands for **ordered list**. <LI> tag is used with each item in the list. It stands for **list item**. It has no closing tag. </OL> tag is used to close an ordered list.

Attributes

Some important attributes of ordered list are as follows:

Attribute	Description
TYPE	It specifies the number scheme for the list. The default number scheme is 1, 2, 3... Possible schemes are: 1 for 1, 2, 3... A for A, B, C... a for a, b, c... I for I, II, III... i for i, ii, iii...
START	It specifies the starting number of the list.
VALUE	It changes the numbering sequence in the middle of the list. It is used in <LI> tag.

Example

```
<OL TYPE="A">
  <LI> Red
  <LI> Green
  <LI> Blue
</OL>
```

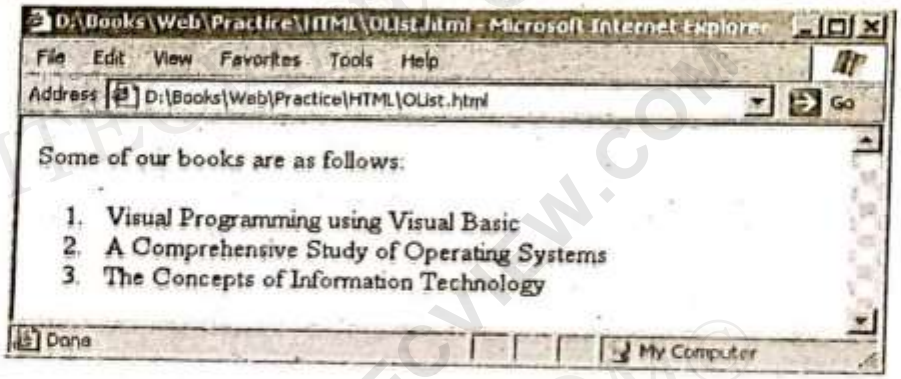
Practice 3.1

Write HTML code that displays the names of three books in ordered list.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
Some of our books are as follows:
<OL>
 <LI>Visual Programming using Visual Basic
 <LI>A Comprehensive Study of Operating Systems
 <LI>The Concepts of Information Technology
</OL>
</BODY>
</HTML>
```

2. Save the file with "OList.html".
3. Double click the file. The web page will appear in the browser as follows:



Practice 3.2

Write HTML code that displays the following output:

My favorite fruit are as follows:
1. Mango
2. Apple
3. Banana

My favorite cities are as follows:
A. Lahore
B. Karachi
C. Peshawar

My favorite colors are as follows:
i. Orange
ii. Red
iii. Green

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
My favorite fruit are as follows:
<OL>
<LI>Mango
<LI>Apple
```

Output:

```
<LI>Banana
</OL>
My favorite cities are as follows:
<OL TYPE="A">
<LI>Lahore
<LI>Karachi
<LI>Peshawar
</OL>
My favorite colors are as follows:
<OL TYPE="i">
<LI>Orange
<LI>Red
<LI>Green
</OL>
</BODY>
</HTML>
```

2. Save the file with "OList2.html".
3. Double click the file. The web page will appear in the browser.

3.1.2 Unordered List

An **unordered list** is a list of item in which each item is marked with a symbol. Unordered list is also known as unnumered list. **<UL>** tag is used to create an unordered list. It stands for **unordered list**. **<LI>** tag is used with each item in the list. **</UL>** tag is used to close an unordered list.

Attributes

Some important attributes of unordered list are as follows:

Attribute	Description
TYPE	It specifies the type of symbol to appear with each item of list. Possible values are FILLROUND, SQUARE, DISC, CIRCLE. The default value is CIRCLE.

Example

```
<UL>
<LI>Item No. 1
<LI>Item No. 2
<LI>Item No. 3
</UL>
```

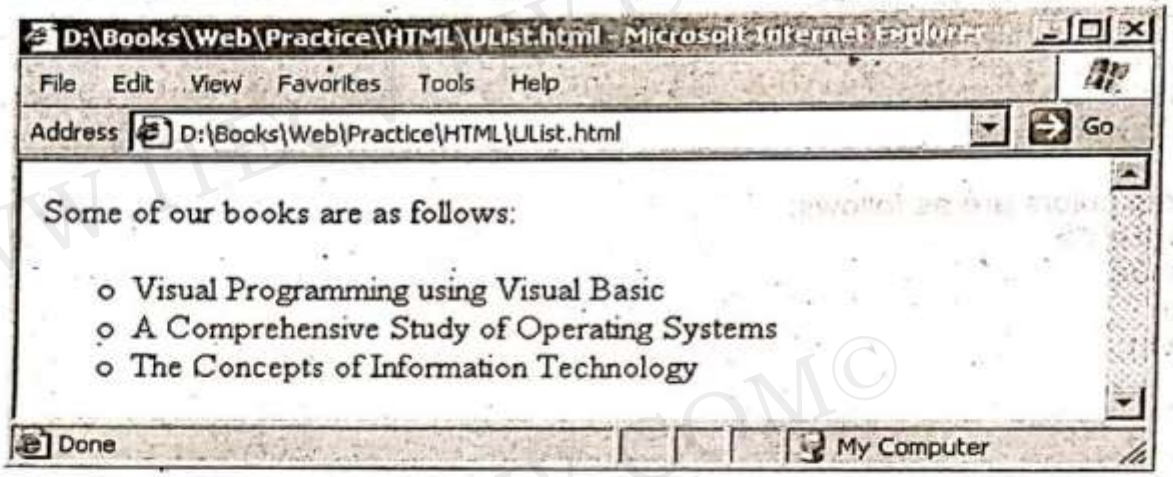
Practice 3.3

Write HTML code that displays the names of three books in unordered list.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
Some of our books are as follows:
<UL TYPE="circle">
<LI>Visual Programming using Visual Basic
<LI>A Comprehensive Study of Operating Systems
<LI>The Concepts of Information Technology
</UL>
</BODY>
</HTML>
```

2. Save the file with "UList.html".
3. Double click the file. The web page will appear in the browser as follows:



3.1.3 Definition Lists

A **definition list** is not a list of items. It consists of terms and description of terms. **<DL>** tag is used to create a definition list. It stands for **definition list**. **<DT>** tag is used with each term in the definition list. It stands for **definition term**. **<DD>** tag is used with each description of the term. It stands for **definition description**.

Example

```
<DL>
<DT>HTML</DT>
<DD>Hyper Text Markup Language</DD>
<DT>XML</DT>
<DD>Extensible Markup Language</DD>
</DL>
```

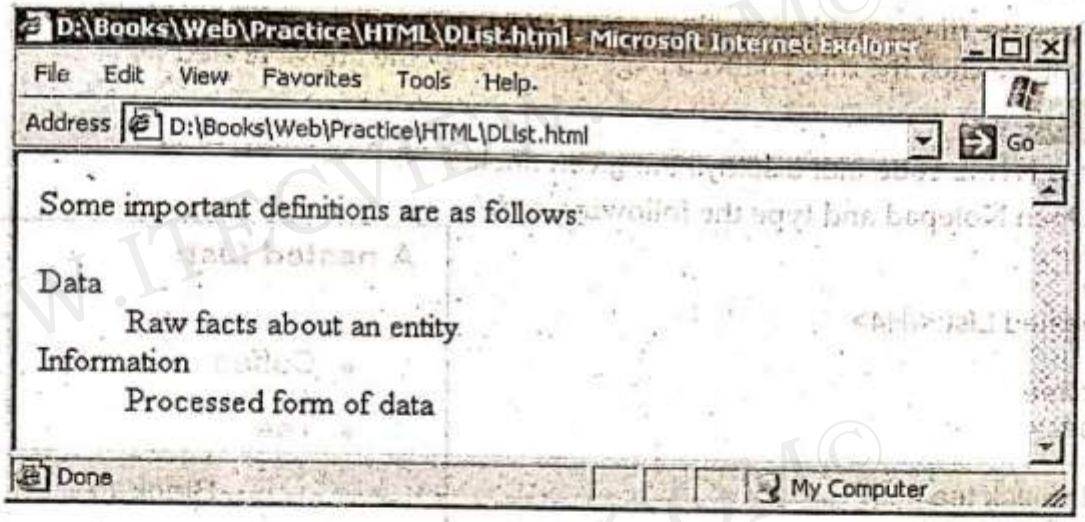
Practice 3.4

Write HTML code that displays two definition lists for the terms data and information.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
Some important definitions are as follows:
<DL>
<DT>Data</DT>
<DD>Raw facts about an entity</DD>
<DT>Information</DT>
<DD>Processed form of data</DD>
</DL>
</BODY>
</HTML>
```

2. Save the file with "DList.html".
3. Double click the file. The web page will appear in the browser as follows:



3.1.4 Nested Lists

A list within a list is known as **nested lists**. HTML can display items as nested lists. A nested list can have different types of lists. For example, an ordered list can be nested within an unordered list.

Example

```
<UL TYPE="SQUARE">
<LI>Item No. 1
<LI>Item No. 2
<OL TYPE="i" START="3">
<LI>Sub Item No. 1
</OL>
</LI>
<LI>Item No. 3
</UL>
```

Practice 3.5

Write HTML code that displays the given nested list:

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>My Favorites:</H3>
<OL>
  <LI>Books
    <UL TYPE="square">
      <LI>Holly Quran
      <LI>Alif Laila
    </UL>
  <LI>Fruits
    <UL TYPE="square">
      <LI>Mango
      <LI>Apple
    </UL>
</OL>
</BODY>
</HTML>
```

My Favorites:

1. Books
   - Holly Quran
   - Alif Laila
2. Fruits
   - Mango
   - Apple

2. Save the file as "NList.html".
3. Double click the file. The web page will display the above nested list.

Practice 3.6

Write HTML code that displays the given nested list:

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>A nested List:</H4>
<UL>
  <LI>Coffee
  <LI>Tea
    <UL>
      <LI>Black tea
      <LI>Green tea
    </UL>
  </LI>
  <LI>Milk
</UL>
</BODY>
</HTML>
```

A nested List:

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk

2. Save the file as "NList1.html".
3. Double click the file. The web page will display the above nested list.

Practice 3.7

Create a webpage containing four ordered lists: Burger, Ice cream, soft drink and frozen yogurt. Each ordered list should contain a nested, unordered list of favorite flavors. Provide a minimum of three flavors in each unordered list.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
        <TITLE>Nested Lists</TITLE>
</HEAD>
<BODY>
<H3>MY favorite burger, ice cream, soft drink and frozen
yogurt.</H3>
<OL>
<LI>Burger
<UL>
        <LI>Chicken Burger
        <LI>Beef Burger
        <LI>Fish Burger
</UL>
<LI>Ice Cream
<UL>
        <LI>Vanilla
        <LI>Chocolate
        <LI>Strawberry
</UL>
<LI>Soft Drink
<UL>
        <LI>Pepsi
```

Output:

1. Burger
   - Chicken Burger
   - Beef Burger
   - Fish Burger
2. Ice Cream
   - Vanilla
   - Chocolate
   - Strawberry
3. Soft Drink
   - Pepsi
   - Coke
   - Dew
   - 7up
4. Frozen Yogurt
   - Vanilla
   - Chocolate
   - Strawberry

WWW.ITECVIEW.COM©

```
        <LI>Coke
        <LI>Dew
        <LI>7up
</UL>
<LI>Frozen Yogurt
<UL>
        <LI>Vanilla
        <LI>Chocolate
        <LI>Strawberry
</UL>
</OL>
</BODY>
</HTML>
```

2. Save the file as "NList2.html".
3. Double click the file. The web page will display the above nested list.

Practice 3.8

Write HTML code that displays the following nested list:

College Organizational Structure

- Dean's Office
  - Dean
  - Associate Dean
  - Assistant Dean
  - Staff
    1. Science Staff
    2. Arts Staff
    3. Commerce Staff
- Department
  A. Chairman
  B. Assoc. Chairman
  C. Faculty
     - Science
     - Arts
     - Commerce
  D. Graduate Students
  E. Undergraduate Students

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
        <TITLE>ORGANIZATIONAL STRUCTRE</TITLE>
</HEAD>
```

WWW.ITECVIEW.COM©

```
<BODY>
<H2 ALIGN="left"><B>College Organizational Structure</B></H2><BR>
<UL>
        <LI>Dean's Office</LI>
                <UL TYPE = "square">
                  <LI>Dean</LI>
                  <LI>Associate Dean
                  <LI>Assistant Dean
                  <LI>Staff
                        <OL>
                           <LI>Science Staff
                           <LI>Arts Staff
                           <LI>Commerce Staff
                        </OL>
                </UL>
        <LI>Department
                <OL TYPE="A">
                  <LI>Chairman
                  <LI>Assoc. Chairman
                  <LI>Faculty
                  <UL>
                           <LI>Science
                           <LI>Arts
                           <LI>Commerce
                        </UL>
                  <LI>Graduate Students
                  <LI>Undergraduate Students
                </OL>
</UL>
</BODY>
</HTML>
```

2. Save the file as "NList3.html".
3. Double click the file. The web page will display the above nested list.

Practice 3.9

Write HTML code that displays the given nested list:

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>My Computer</H3>
<UL>
 <LI>Input Devices
 <OL>
  <LI>Keyboard
  <LI>Mouse
 </OL>
 <LI>Output Devices
 <OL>
  <LI>Monitor
        <UL>
          <LI>CRT Monitors
          <LI>LCD Monitors
```

My Computer

- Input Devices
  1. Keyboard
  2. Mouse
- Output Devices
  1. Monitor
     - CRT Monitors
     - LCD Monitors
  2. Printer
     - Laser Printers
     - Inkjet Printers
- Processor
  1. Control Unit
  2. ALU
     - Arithmatic Unit
     - Logic Unit

WWW.ITECVIEW.COM©

```
        </UL>
  <LI>Printer
        <UL>
          <LI>Laser Printers
          <LI>Inkjet Printers
        </UL>
</OL>
<LI>Processor
<OL>
  <LI>Control Unit
  <LI>ALU
        <UL>
          <LI>Arithmatic Unit
          <LI>Logic Unit
        </UL>
 </OL>
</UL>
</BODY>
</HTML>
```

2. Save the file as "NList4.html".
3. Double click the file. The web page will display the above nested list.

3.2 Images in HTML

Different image formats supported by HTML are as follows:

1. Graphic Interchange Format (GIF)

Graphic Interchange Format uses a compression technique that preserves all color information for the image. It uses a maximum of 256 colors. GIF format is best for displaying images like logos, icons and buttons. A GIF file with fewer number of colors has smaller file size and allows greater compression. However, the use of GIF format is limited because of color depth limitation. It should not be used for photographs and complex graphics.

GIF images are in two different versions. These are **GIF 87** and **GIF 89a** format. GIF 89a is newer and has the following features:

- **Interlacing:** Interlacing is a capability that enables the browser to display the image as it loads. The image gradually gets clearer until it is loaded completely. Interlaced images are slightly larger in size than non-interlaced images.
- **Transparency:** GIF 89a format can be used to set a single color to be transparent. It will allow the background color or image to show through it.
- **Animated GIFs:** GIF 89a images can also be animated using special software. Animated GIF images are actually a number of GIF images saved in a single file.

2. Joint Photographic Expert Group (JPG, JPEG)

Joint Photographic Expert Group is a good format for images. It can contain millions of colors. It does not provide the option of transparency or interlacing. It allows to specify the degree of compression to create a balance between image quality and file size. It uses a compression technique that discards some colors that may cause a loss of quality of the original image. The higher compression reduces the file size but provides lower image quality. A new addition to JPEG format is Progressive JPEG. Pro-JPEGs provide superior compression and give a wider range of quality settings. Pro-JPEGs also support interlacing.

3. Limited Support or Non-Supported Image Formats

The following image formats are supported by some computers but not all. These are are not recommended to be included in HTML pages. These images can be converted to supported formats by using graphics editor such as **Corel Photo Paint or Adobe Photoshop.**

- **Portable Network Graphics (PNG):** Portable Network Graphics is good for combinations of text and graphics within one image. It supports transparency and interlacing but not animation. It was specially designed for use on the web. It was intended to place GIF format. However, it is not popular because of lack of browser support. PNG is only supported by Internet Explorer.
- **BMP:** BMP stands for **bitmap picture**. It is most commonly used in Windows environment. It provides high quality images. BMP images are not compressed and produce large file size. That is why it is not used on the Internet.
- **TIFF:** TIFF stands for **Tagged Image File Format**. It is widely supported format to store bitmap images on computer. It is commonly used in desktop publishing, 3D applications and medical imaging applications.
- **PCX:** It was originally developed by ZSOFT for PC Paintbrush program. PCX is a graphics file format for graphics programs running on PCs.

3.2.1 Using Images in HTML

A web page may contain different types of graphics. Graphics make websites more attractive and interesting. HTML can be used to add static or animated graphics in websites. The image put into a webpage are called **inline images**. The most popular types of images used in web pages are **.gif** and **.jpg**. **<IMG>** tag is used to insert images in web pages. This tag has no ending tag.

Attributes

Some important attributes of <IMG> tag are as follows:

Attribute	Description
SRC	It specifies the name and path of image to be displayed. Only name of the image is specified if image is stored in the same directory in which web page is stored. Otherwise, complete path of image is specified.
ALIGN	It specifies the alignment of image. Possible alignments are as follows: left . The image is aligned to the left of the screen right The image is aligned to the right of the screen center The image is aligned to the center of the screen It also specifies the text alignment with respect to the image. Possible alignments are as follows: top The text is written at the top next to the image bottom The text is written at the bottom next to the image middle The text is written at the middle next to the image
BORDER	It specifies the size of border around the image. The possible values are 1, 2, 3 and so on. The default value is 0.
WIDTH	It specifies the width of the image in pixels.
HEIGHT	It specifies the height of the image in pixels.
HSPACE	It specifies the amount of space to left and right side of image in pixels.
VSPACE	It specifies the amount of space to top and bottom of the image in pixels.
ALT	It specifies the text to be displayed if the browser cannot display image.

Examples

```
<IMG SRC="Pic.jpg" ALIGN="center" BORDER=2>
```

The above example displays "Pic.jpg" image in the center of the page with border of size 2 around it.

```
<IMG SRC="Pic.jpg" ALIGN="right" ALT="Image not found">
```

The above example displays "Pic.jpg" image at right side of the page. If the browser cannot display the image, the text "Image not found" will be displayed.

Practice 3.10

Write HTML code that displays an image aligned right. The code should display a message if there is any problem in dispalying the image.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Images in HTML</H3>
Images are very important for a website. Following is a small image:<P>
<IMG SRC="Pic.jpg" ALIGN="left" ALT="Image not found">
<BR><BR>Oops... Check it now <P>
<IMG SRC="Pic.gif" WIDTH=100 HEIGHT=100 ALIGN="right" BORDER=1 ALT="Error">
</BODY>
</HTML>
```

2. Save the file as "image.html".
3. Double click the file. The web page will appear in the browser as follows:

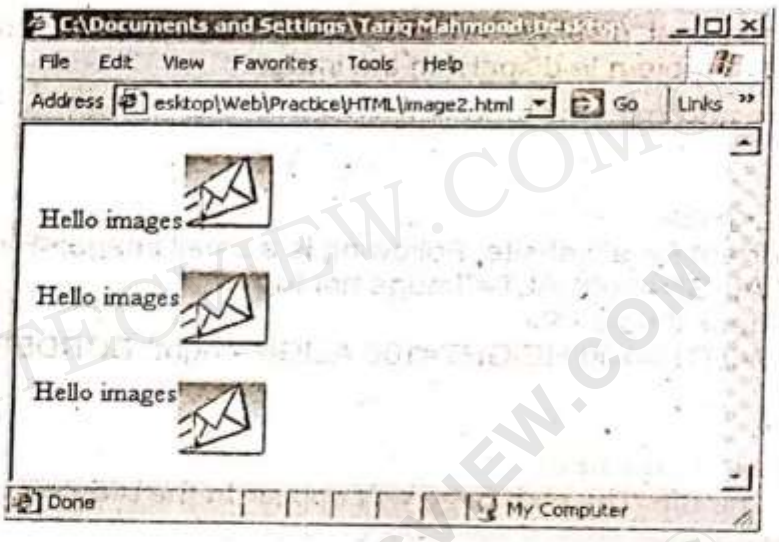
Practice 3.11

Write HTML code that displays text with image. The code should display the text in all possible alignments with the image.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<P>Hello images<IMG SRC="mail.jpg" ALIGN="bottom" WIDTH="48" HEIGHT="48"></P>
<P>Hello images<IMG SRC="mail.jpg" ALIGN="middle" WIDTH="48" HEIGHT="48"></P>
<P>Hello images<IMG SRC="mail.jpg" ALIGN="top" WIDTH="48" HEIGHT="48"></P>
</BODY>
</HTML>
```

2. Save the file as "image2.html".
3. Double click the file. The web page will appear in the browser as follows:



3.3 Tables

Tables are a very important tag in the HTML language. A table consists of rows and columns. The data in a table is more readable. There are three main elements in HTML for building tables. These are <TABLE>, the table row <TR> and table cell <TD>. **<TABLE>** tag is used to define a table. **<TR>** tag is used to define a row in a table and **<TD>** tag is used to define a data cell. **<TH>** tag is used to define the header row of the table.

Attributes

Some important attributes of <TABLE> tag are as follows:

Attribute	Description
ALIGN	It specifies the horizontal alignment of the table. Possible alignments are left, right and center.
VALIGN	It specifies the vertical alignment of the table. Possible alignments are top, bottom and middle.
WIDTH	It specifies the width of the table. It can be given as number of pixels or as percentage relative to the screen.
BORDER	It specifies the thickness of the border of the table.
CELLPADDING	It specifies the distance between data in the cell and cell boundaries.
CELLSPACING	It specifies the spacing between adjacent cells of the table.
COLSPAN	It specifies the number of columns for a cell. It is used inside **<TH>** or <TD> tags. A cell may consist of many columns.
ROWSPAN	It specifies the number of rows for a cell. It is used inside <TD> or <TD> tags. A cell may consist of many rows.

Example

```
<TABLE ALIGN="left" BGCOLOR="red" BORDER=1 CELLPADDING=5 CELLSPACING=2
WIDTH="90%">
```

3.3.1 The <TR> Tag

The <TR> tag is used to create a row in the table. TR stands for **table row.**

Attributes

Some important attributes of <TR> tag are as follows:

Attribute	Description
ALIGN	It specifies the horizontal alignment of row. Possible alignments are left, right and center.
BGCOLOR	It specifies the color of the background for the row.
VALIGN	It specifies the vertical aligns the content of the row. The possible values are top, middle or bottom.

Example

```
<TR ALIGN="right" BGCOLOR="red" VALIGN=top>
```

3.3.2 The <TD> Tag

The <TD> tag is used to create cells in the row of a table. TD stands for **table data.**

Attributes

Some important attributes of <TD> tag are as follows:

Attribute	Description
ALIGN	It specifies the horizontal alignment of cell. Possible alignments are left, right and center.
BACKGROUND	It specifies the background image inside of the table cell
BGCOLOR	It specifies the color of the background for the cell
COLSPAN	It stretches the length of the cells to 2 or more cell lengths.
HEIGHT	It specifies the heights of the cell in pixels
NOWRAP	It locks the content so that it all stays on the same line.
ROWSPAN	It stretches the height of the cells to 2 or more cells
VALIGN	It specifies the vertical alignment of the content of the cell. The possible values are top, middle or bottom.
WIDTH	It specifies the width of the actual cell in pixels or percentage

Example

```
<TD ALIGN="left" BGCOLOR="red" COLSPAN=3 HEIGHT=200 VALIGN="top" WIDTH=300>
```

Practice 3.12

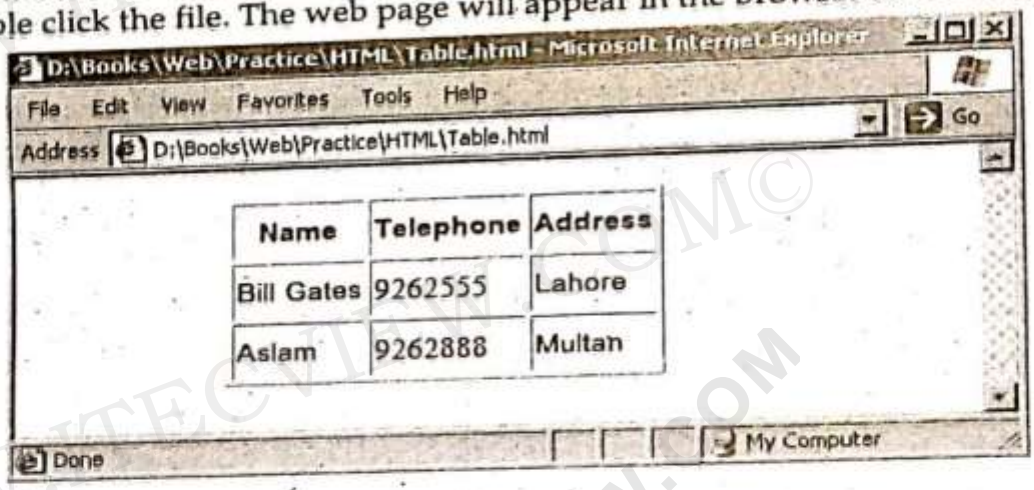
Write HTML code to generate a table with three columns Name, Telephone and Address. The table should display two records.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<TABLE BORDER="1">
```

```
<TR><TH>Name</TH><TH>Telephone</TH><TH>Address</TH></TR>
<TR><TD>Bill Gates</TD><TD>9262555</TD><TD>Lahore</TD></TR>
<TR><TD>Aslam</TD><TD>9262888</TD><TD>Multan</TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table.html".
3. Double click the file. The web page will appear in the browser as follows:



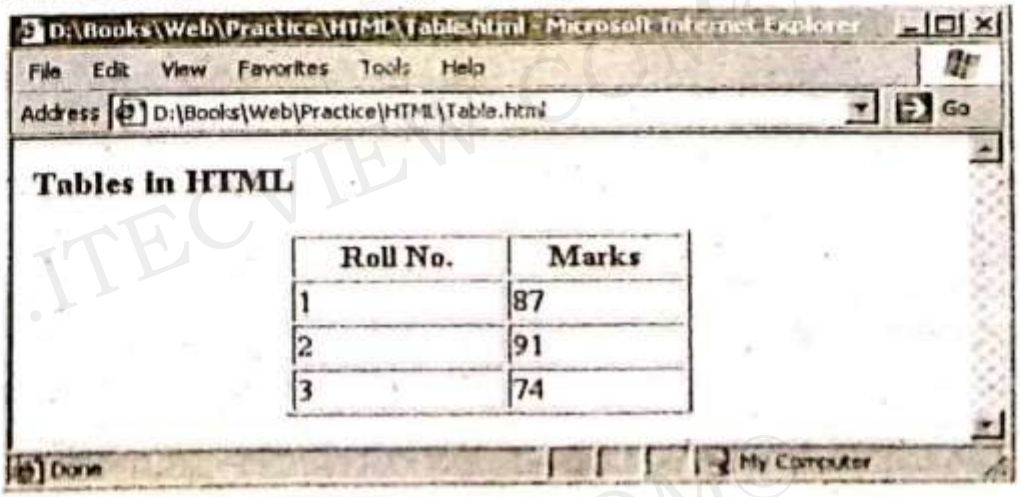
Practice 3.13

Write HTML code that displays the Roll No. and marks of three students in a table.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Tables in HTML</H3>
<TABLE BORDER=1 WIDTH=200 ALIGN="center">
  <TR> <TH>Roll No.</TH><TH>Marks</TH> </TR>
  <TR> <TD>1</TD><TD>87</TH> </TR>
  <TR> <TD>2</TD><TD>91</TD> </TR>
  <TR> <TD>3</TD><TD>74</TD> </TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table1.html".
3. Double click the file. The web page will appear in the browser as follows:



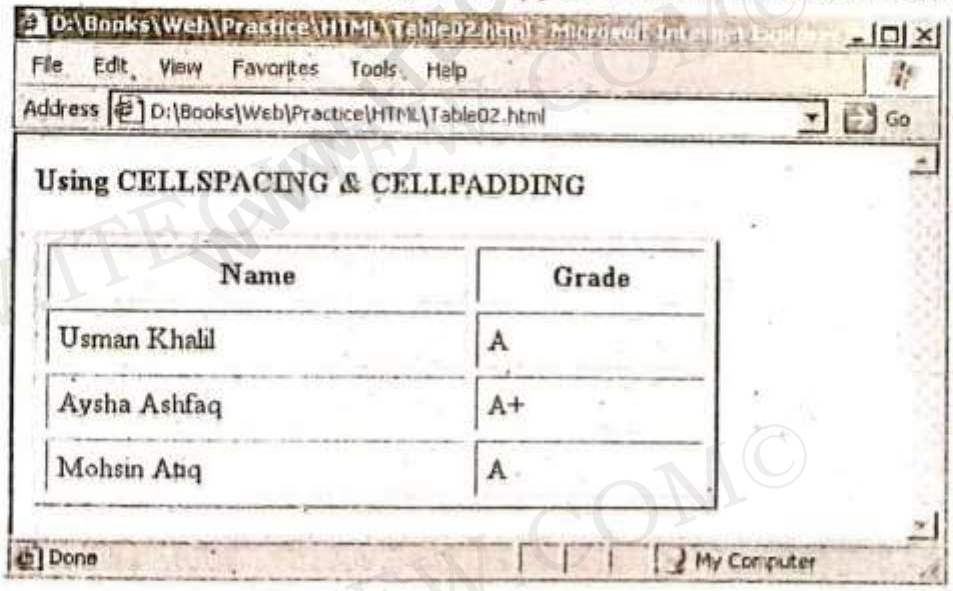
Practice 3.14

Write HTML code that displays the names and grades of three students in a table.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Using CELLSPACING & CELLPADDING</H4>
<TABLE BORDER=2 WIDTH=80% CELLPADDING=5 CELLSPACING=5>
 <TR>
      <TH>Name</TH><TH>Grade</TH>
 </TR>
 <TR>
      <TD>Usman Khalil</TD><TD>A</TH>
 </TR>
 <TR>
      <TD>Aysha Ashfaq</TD><TD>A+</TD>
 </TR>
 <TR>
      <TD>Mohsin Atiq</TD><TD>A</TD>
 </TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table3.html".
3. Double click the file. The web page will appear in the browser as follows:



Practice 3.15

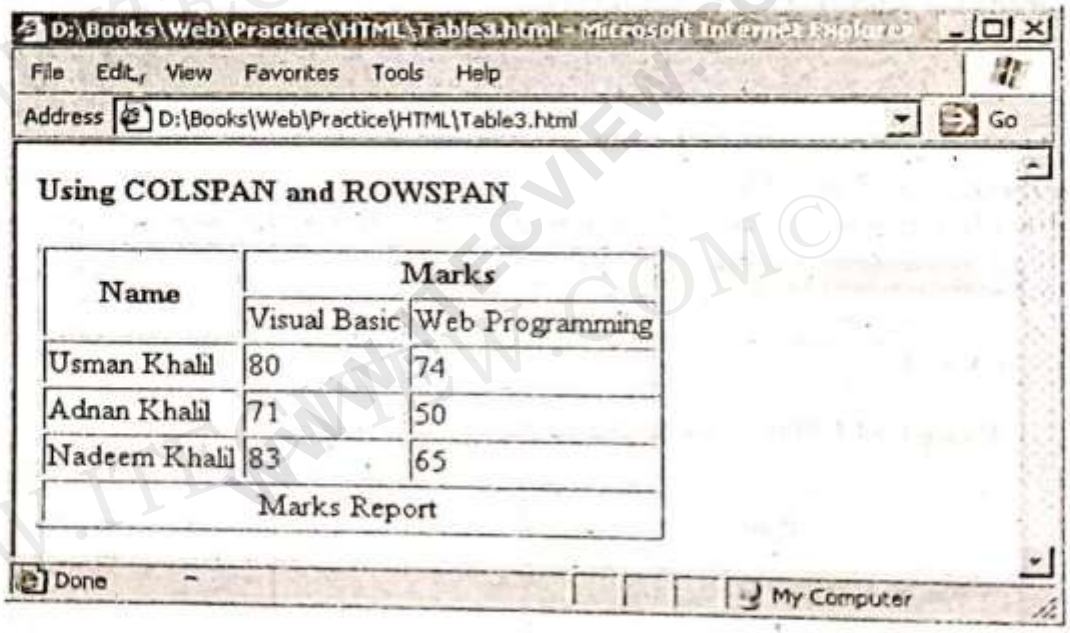
Write HTML code that displays the following table:

Name	Marks	
	Visual Basic	Web Programming
Usman Khalil	80	74
Adnan Khalil	71	50
Nadeem Khalil	83	64
Marks Report		

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Using COLSPAN and ROWSPAN</H4>
<TABLE BORDER=1>
 <TR>
      <TH ROWSPAN=2>Name</TH><TH COLSPAN=2>Marks</TH>
 </TR>
 <TR><TD>Visual Basic</TD><TD>Web Programming</TD></TR>
 <TR><TD>Usman Khalil</TD><TD>80</TH><TD>74</TD></TR>
 <TR><TD>Adnan Khalil</TD><TD>71</TD><TD>50</TD></TR>
 <TR><TD>Nadeem Khalil</TD><TD>83</TD><TD>65</TD></TR>
 <TR><TD COLSPAN=3 ALIGN="center">Marks Report</TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table4.html".
3. Double click the file. The web page will appear in the browser as follows:



Practice 3.16

Write an HTML document that shows the names of fruits in one column and colors in second column of table. Use attributes to format width, border and cell spacing for the table.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<TABLE BORDER="1" CELLSPACING="2" WIDTH="25%">
<TR><TH>Fruit</TH><TH>Color</TH></TR>
<TR><TD>Mango</TD><TD>Yellow</TD></TR>
<TR><TD>Orange</TD><TD>Orange</TD></TR>
<TR><TD>Apple</TD><TD>Red</TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table5.html".
3. Double click the file. The web page will appear in the browser as follows:

Fruit	Color
Mango	Yellow
Orange	Orange
Apple	Red

Practice 3.17

Write HTML code that displays the following table:

Player Name	Team	Runs	100s	Highest
Sachin Tendulkar	India	13909	38	186
Inzamam-ul-Haq	Pakistan	11141	10	137
Sanath Jayasuriya	Sri Lanka	10219	18	189
Sourav Ganguly	India	10123	22	183
Mohd Azharuddin	India	9378	7	153

Top 5 Run-Makers in One Day Internationals

The background color of heading row should be yellow and the background color of the coloumn of players' names should be red.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=2 WIDTH="98%">
<CAPTION ALIGN="bottom">Top 5 Run-Makers in One Day Internationals</CAPTION>
<TR BGCOLOR="yellow">
        <TH ALIGN=left WIDTH=25%>Player Name</TH>
        <TH ALIGN=left WIDTH=15%>Team</TH>
        <TH WIDTH=11% ALIGN=right>Runs</TH>
        <TH WIDTH=9% ALIGN=right>100s</TH>
        <TH WIDTH=9% ALIGN=right>Highest</TH>
</TR>
<TR>
        <TD BGCOLOR="red">Sachin Tendulkar</TD>
        <TD>India</td>
        <TD ALIGN=right><B>13909</B></TD>
        <TD ALIGN=right>38</TD>
        <TD ALIGN=right>186</TD>
</TR>
<TR>
        <TD BGCOLOR="red">Inzamam-ul-Haq</TD>
        <TD>Pakistan</TD>
        <TD ALIGN=right><B>11141</B></TD>
        <TD ALIGN=right>10</TD>
        <TD ALIGN=right>137</TD>
</TR>
```

```
<TR>
        <TD BGCOLOR="red">Sanath Jayasuriya</TD>
        <TD>Sri Lanka</td>
        <TD ALIGN=right><B>10219</B></TD>
        <TD ALIGN=right>18</TD>
        <TD ALIGN=right>189</TD>
</TR>
<TR>
        <TD BGCOLOR="red">Sourav Ganguly</TD>
        <TD>India</TD>
        <TD ALIGN=right><B>10123</B></TD>
        <TD ALIGN=right>22</TD>
        <TD ALIGN=right>183</TD>
</TR>
<TR>
        <TD BGCOLOR="red">Mohd Azharuddin</TD>
        <TD>India</TD>
        <TD ALIGN=right><B>9378</B></TD>
        <TD ALIGN=right>7</TD>
        <TD ALIGN=right>153</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "ColorTable.html".
3. Double click the file. The above table will appear in the browser.

Practice 3.18

Write HTML code that creates a table with two list in the left cell and a picture in the right cell.

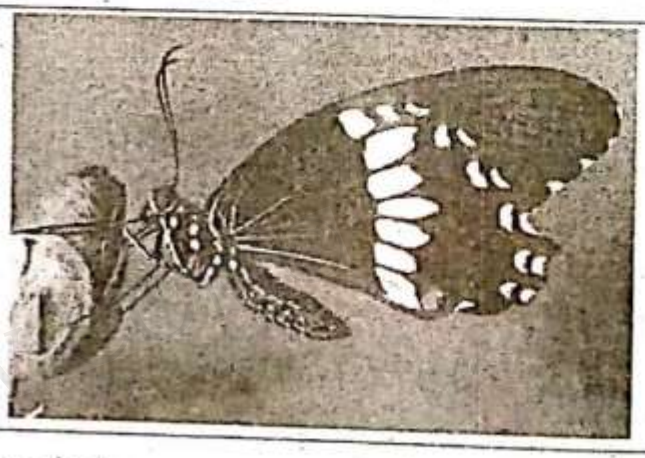
Images and Table

My favorite sports:

- Cricket
- Football
- Hockey

My favorite flowers:

1. Rose
2. Jasmine



1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
        <TITLE>Tables</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="CENTER">
```

```
<H1>Images and Tables</H1>
<DIV ALIGN="center">
<TABLE CELLPADDING="2" CELLSPACING="2" BORDER="1" WIDTH="100%">
<TR>
    <TD VALIGN="top">My favorite sports: <BR>
    <UL>
        <LI>Cricket
        <LI>Football
        <LI>Hockey
    </UL>
My favorite flowers: <BR>
    <OL>
        <LI>Rose
        <LI>Jasmine
    </OL>
    </TD>
    <TD VALIGN="top" ALIGN="center">
<IMG SRC="butterfly.jpg" WIDTH="196" HEIGHT="236"><BR></TD>
   </TR>
</TABLE>
 <BR>
</DIV>
</BODY>
</HTML>
```

2. Save the file as "Table6.html".
3. Double click the file. The web page will appear in the browser.

Note: The picture "butterfly.jpg" must be saved in the same folder in which the page "Table.html" is saved. Otherwise, the complete path of the picture should be given.

Practice 3.19

Write HTML code that creates a pair of tables side by side within another table. The table should appear as shown in the following figure. Use the table attributes of cellspacing, cellpadding and border.

One	Two
Three	Four

1	2
3	4

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Nested Tables</TITLE>
</HEAD>
<BODY>
<TABLE CELLSPACING="4" CELLPADDING="6" BORDER="4" ALIGN="center">
<TR>
        <TD ALIGN="center">
            <TABLE CELLSPACING="4" CELLPADDING="6" BORDER="4" ALIGN="center">
                <TR><TD>One</TD><TD>Two</TD></TR>
                <TR><TD>Three</TD><TD>Four</TD></TR>
```

```
        </TABLE>
      </TD>
      <TD ALIGN="center">
        <TABLE CELLSPACING="4" CELLPADDING="6" BORDER="4" ALIGN="center">
      <TR><TD>1</TD><TD>2</TD></TR>
      <TR><TD>3</TD><TD>4</TD></TR>
        </TABLE>
      </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "NestedTable.html".
3. Double click the file. The web page will appear in the browser.

Practice 3.20

Write HTML code that creates three image links in a borderless table. Construct the table so that there is just a little space between the images.

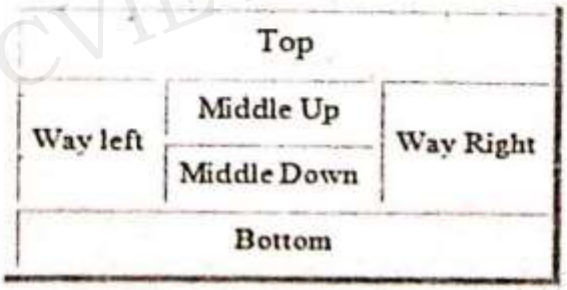
1. Open Notepad and type the following code:

```
<HTML>
<HEAD><TITLE>Image Links</TITLE></HEAD>
<BODY>
<TABLE CELLSPACING="3" CELLPADDING="0" BORDER="0" ALIGN="center">
<TR><TD><A HREF="home.html"><IMG SRC="home.gif" WIDTH="96" HEIGHT="27"
BORDER="0" ALT="Home"></A></TD></TR>
<TR><TD><A HREF="mailto:me@work"><IMG SRC="email.gif" WIDTH="96" HEIGHT="27"
BORDER="0" ALT="Email"></A></TD></TR>
<TR><TD><A HREF="links.html"><IMG SRC="links.gif" WIDTH="96" HEIGHT="27"
BORDER="0" ALT="Links"></A></TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "ImageTable.html".
3. Double click the file. The web page will appear in the browser.

Practice 3.21

Write HTML code that creates puzzle looking table as follows:



1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
        <TITLE>Difficult Table</TITLE>
</HEAD>
<BODY>
<TABLE CELLSPACING="4" CELLPADDING="6" BORDER="4" ALIGN="center">
<TR><TD COLSPAN="3" ALIGN="center"><BIG>Top</BIG></TD></TR>
<TR><TD ROWSPAN="2" ALIGN="center">Way Left</TD>
<TD ALIGN="center">Middle Up</TD>
<TD ROWSPAN="2" ALIGN="center">Way Right</TD>
</TR>
<TR><TD ALIGN="center">Middle Down</TD></TR>
<TR><TD COLSPAN="3"></TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table7.html".
3. Double click the file. The web page will appear in the browser.

Practice 3.22

Write HTML code that displays the following table using different color shadings and other table attributes.

Result of Students

Name	Physics	Biology	Chemistry	English	Urdu
Usman Khalil	89	77	82	94	80
Abdullah	91	83	76	81	68
Faiq	90	85	73	89	70
Mohsin	93	91	87	83	82

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Result of Students</TITLE>
</HEAD>
<BODY>
<TABLE CELLSPACING="0" CELLPADDING="2" BORDER="0" ALIGN="center">
<TR BGCOLOR="#BBDDBB">
   <TD ALIGN="center" COLSPAN="6">
   <FONT SIZE="5"><I><B> Result of Students</B></I></FONT></TD>
</TR>
<TR>
   <TD><FONT SIZE="2"><B>Name</B></FONT></TD>
   <TD ALIGN="center"><FONT SIZE="2"><B>Physics</B></FONT></TD>
   <TD ALIGN="center"><FONT SIZE="2"><B>Biology</B></FONT></TD>
   <TD ALIGN="center"><FONT SIZE="2"><B>Chemistry</B></FONT></TD>
   <TD ALIGN="center"><FONT SIZE="2"><B>English</B></FONT></TD>
   <TD ALIGN="center"><FONT SIZE="2"><B>Urdu</B></FONT></TD>
</TR>
```

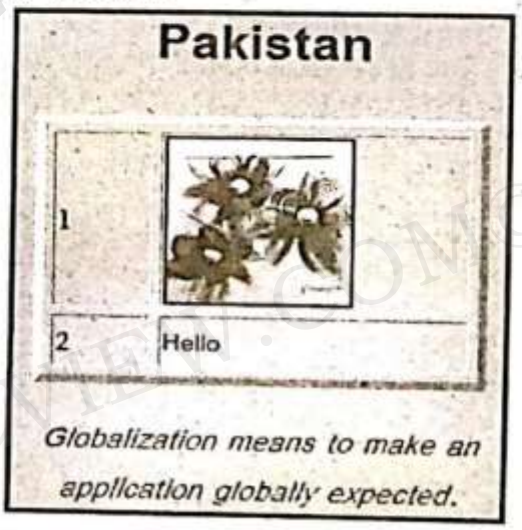
```
<TR BGCOLOR="#DDFFDD">
    <TD>Usman Khalil</TD>                <TD ALIGN="center">89</TD>
    <TD ALIGN="center">77</TD>           <TD ALIGN="center">82</TD>
    <TD ALIGN="center">94</TD>           <TD ALIGN="center">80</TD>
</TR>
<TR>
    <TD>Abdullah</TD>                    <TD ALIGN="center">91</TD>
    <TD ALIGN="center">83</TD>           <TD ALIGN="center">76</TD>
    <TD ALIGN="center">81</TD>           <TD ALIGN="center">68</TD>
</TR>
<TR BGCOLOR="#DDFFDD">
    <TD>Faiq</TD>                        <TD ALIGN="center">90</TD>
    <TD ALIGN="center">85</TD>           <TD ALIGN="center">73</TD>
    <TD ALIGN="center">89</TD>           <TD ALIGN="center">70</TD>
</TR>
<TR>
    <TD>Mohsin</TD>                      <TD ALIGN="center">93</TD>
    <TD ALIGN="center">91</TD>           <TD ALIGN="center">87</TD>
    <TD ALIGN="center">83</TD>           <TD ALIGN="center">82</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table8.html".
3. Double click the file. The web page will appear in the browser.

Practice 3.23

Write HTML code that displays the following output. Consider output for:

(i) Proper line breaks and alignments. Font color for paragraph=green.
(ii) Background color of the page must be pink.
(iii) Image dimension (100x100). Image "water.jpg" is saved on D: drive. This image is working as hyperlink for hotmail.



1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Image Link in Table</TITLE>
</HEAD>
```

```
<BODY BGCOLOR="pink">
<H1 ALIGN="center">Pakistan</H1>
<TABLE BORDER="5" CELLSPACING="5" WIDTH="20%" ALIGN="center">
<TR>
        <TD WIDTH="50">1</TD>
        <TD><A HREF="http://www.hotmail.com" TARGET="_blank">
        <IMG SRC="D:\Flower.jpg" WIDTH=100 HEIGHT=100></A></TD>
</TR>
<TR>
        <TD>2</TD>
        <TD>Hello</TD>
</TR>
</TABLE>
<FONT COLOR="green" SIZE="4">
<P ALIGN="center"><I>Globalization means to make an <BR> application globally
expected.</I></P>
</FONT>
</BODY>
</HTML>
```

2. Save the file as "Table9.html".
3. Double click the file. The web page will appear in the browser.

3.3.3 The <CAPTION> Tag

<CAPTION> tag is used to specify the caption of table. Caption is always displayed in the center with respect to the table. It also appears within the width of the table. The ALIGN attribute of this tag specifies the location of the caption. The possible alignments are **top** and **bottom**. The caption appears at the top of table by default. It is used within <TABLE> tag.

Example

```
<CAPTION ALIGN="top">
It is my caption.
</CAPTION>
```

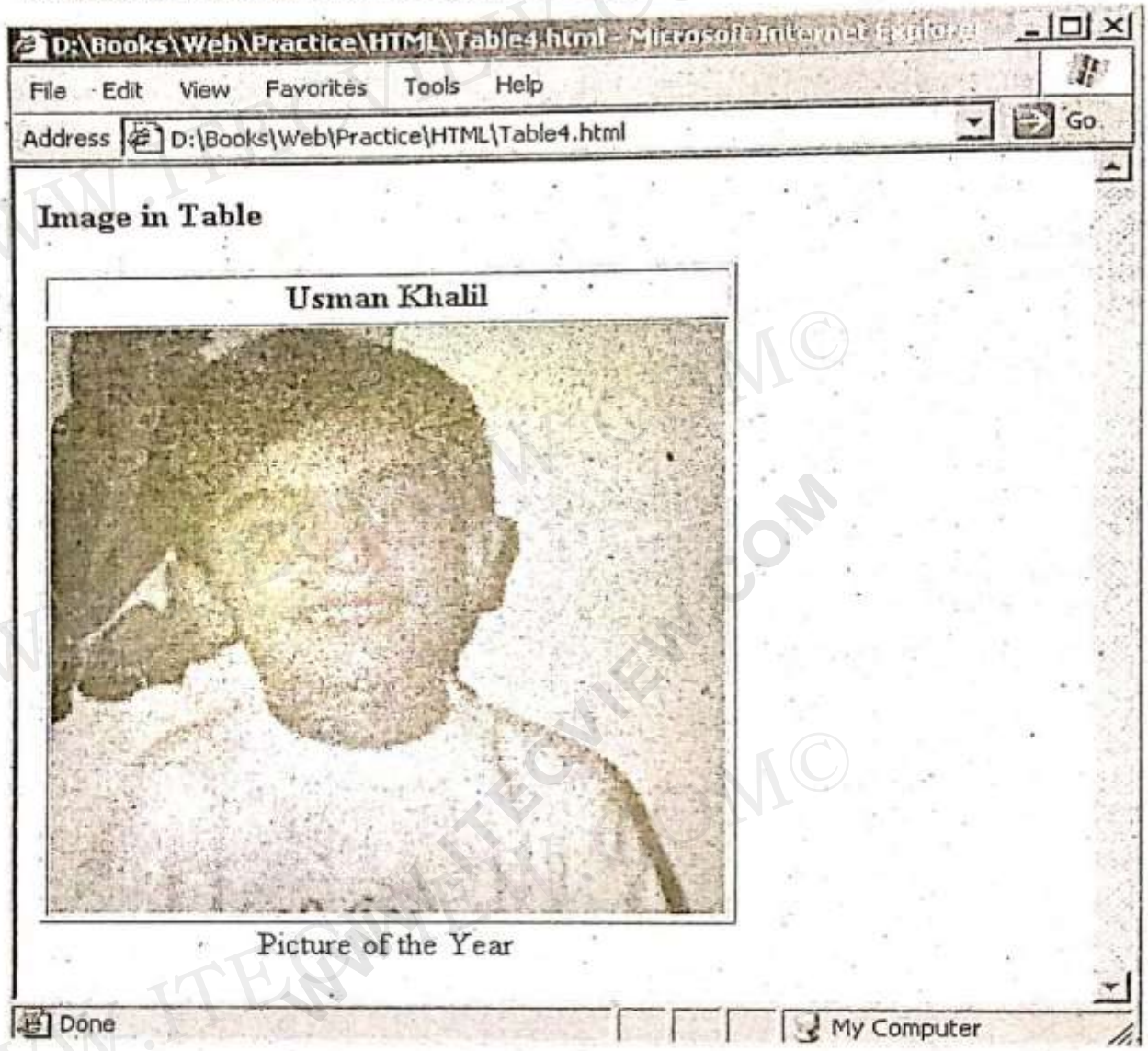
Practice 3.24

Write HTML code that displays an image in a table. The caption of table should appear at the bottom.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Image in Table</H4>
<TABLE BORDER=1 WIDTH=300>
  <CAPTION ALIGN="bottom">Picture of the Year</CAPTION>
  <TR>
        <TH>Picture of Usman Khalil</TH>
  </TR>
  <TR>
        <TD><IMG SRC="Usman.jpg" WIDTH=350 HEIGHT=300></TD>
  </TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table10.html".
3. Double click the file. The web page will appear in the browser as follows:



Exercise Questions

1. Write HTML code that uses lowercase Roman numbers, starts at a value of 9 and contains four elements.
2. Write HTML code that creates a nested list. The outer list should display the names of three faculties of university. Each faculty should contain the names of departments in that faculty as unordered list.
3. Create a tic-tac-toe game in the following form:

X	X	X
0	0	X
0		0

4. Write HTML code that display the following score in a table:

Batsman	Score
Salman Butt	59
Kamran Akmal	71
Younis Khan	65
Muhammad Yousuf	83
Inzamal-ul-Haq	120
Total	298

5. Write HTML code that displays the following table:

1	2	3	4	5	6
1		2		3	
1			2		
1					

6. Write HTML code that displays the following time table:

Class	Days	Time
Introduction to IT	Mon - Wed	09:00AM - 10:00AM
Programming Basics	Mon - Wed	10:00AM - 11:00AM
Operating Systems	Mon - Wed	11:00AM - 12:00PM
Digital Electronics	Thu - Sat	09:00AM - 10:00AM
Numerical Analysis	Thu - Sat	10:00AM - 11:00AM
Note: *Lab will be conducted daily from 12:00PM to 02:00PM*		

Class Schedule for Semester 1

WWW.ITECVIEW.COM©

Hyperlinks and Image Maps

4.1 Hyperlinks

Each website on World Wide Web contains many pages that are linked with one another. The links are used to move from one location to another on the Internet. A link may consist of simple text as well as an image. The link is also known as **hyperlink**. A hyperlink consists of two parts. These are link label and link destination. **Link label** is the visible element that is clicked by the user. It can be simple text or image. **Link destination** is the location that will be visited when the user clicks the link label.

Hyperlinks can be very easily identified by following properties:

- They are normally underlined.
- Color of hyperlinks is normally different from normal text.
- When mouse pointer moves to a hyper link, pointer changes to a pointing hand.

4.1.1 Types of Hyperlinks

Hyperlinks can be of three kinds:

- **Page-Jumps:** These are the hyperlinks that are linked with other locations within the same page. The location to which this hyperlink is linked is called **Bookmark**. These bookmarks help the user to easily navigate the site without too much scrolling.
- **Internal Links:** These are the hyperlinks that are linked with other web pages within the same website.
- **External Links:** These are the hyperlinks that are linked with other websites or other web pages of other websites.
- **Email Links:** These are the hyperlinks that are linked with emails. They start email application when the user clicks on them.

4.2 Creating Hyperlinks in HTML

<A> (anchor) tag is used to create a hyperlink in a web page. **HREF** attribute of **<A>** tag specifies location to which the link is connected. An anchor can be linked to the following:

- Another web page
- Another part of the same web page

WWW.ITECVIEW.COM©

- A file etc.

Be default, hypertext appears in blue color with underlined style. The default style can be changed by specifying the following attributes of <BODY> tag of the web page:

Attribute	Description
LINK	It specifies the color of hyperlinks. It can be specified by using color name or its hexadecimal code.
ALINK	It specifies the color of the hyperlinks that are activated. It can be specified by using color name or its hexadecimal code.
VLINK	It specifies the color of the hyperlinks that have been visited. It can be specified by using color name of its hexadecimal code.

4.2.1 Links to another Web Page

A hyperlink can refer to a web page in the same website. This type of hyperlink is frequently used in almost all websites. The visitors view different web page of a website by clicking on different hyperlinks that refer to different pages.

Example

```
<A HREF="contact.html">Contact Us</A>
```

In the above example, the text **Contact Us** is a hyperlink that refers to **contact.html** file. The file should be in the current directory. If the file is stored in another directory, complete path must be given as follows:

```
<A HREF="images/contact.html">Contact Us</A>
```

The "TARGET" Attribute

The **target** attribute of <A> tag is used to specify the window to open the linked page. By default, the page opens in the current window.

```
<A HREF="images/contact.html" TARGET="_blank">Contact Us</A>
```

The above example will open the **contact.html** page in a new window.

Practice 4.1

Write HTML code that connects three web pages using hyperlinks.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Welcome to Online Book Shop</H4>
Online book shop provides you the facility to purchase the latest published books on a variety
of topics. You can purchase the books by using your credit cards.
Please check the following links for further information: <P>
Click here to <A HREF="contact.html" TARGET="_blank">contact us</A> <P>
Click here to view <A HREF="books.html">list of books</A> <P>
</BODY>
</HTML>
```

2. Save the file as "Main.html".
3. Open a new file in Notepad and type the following code:

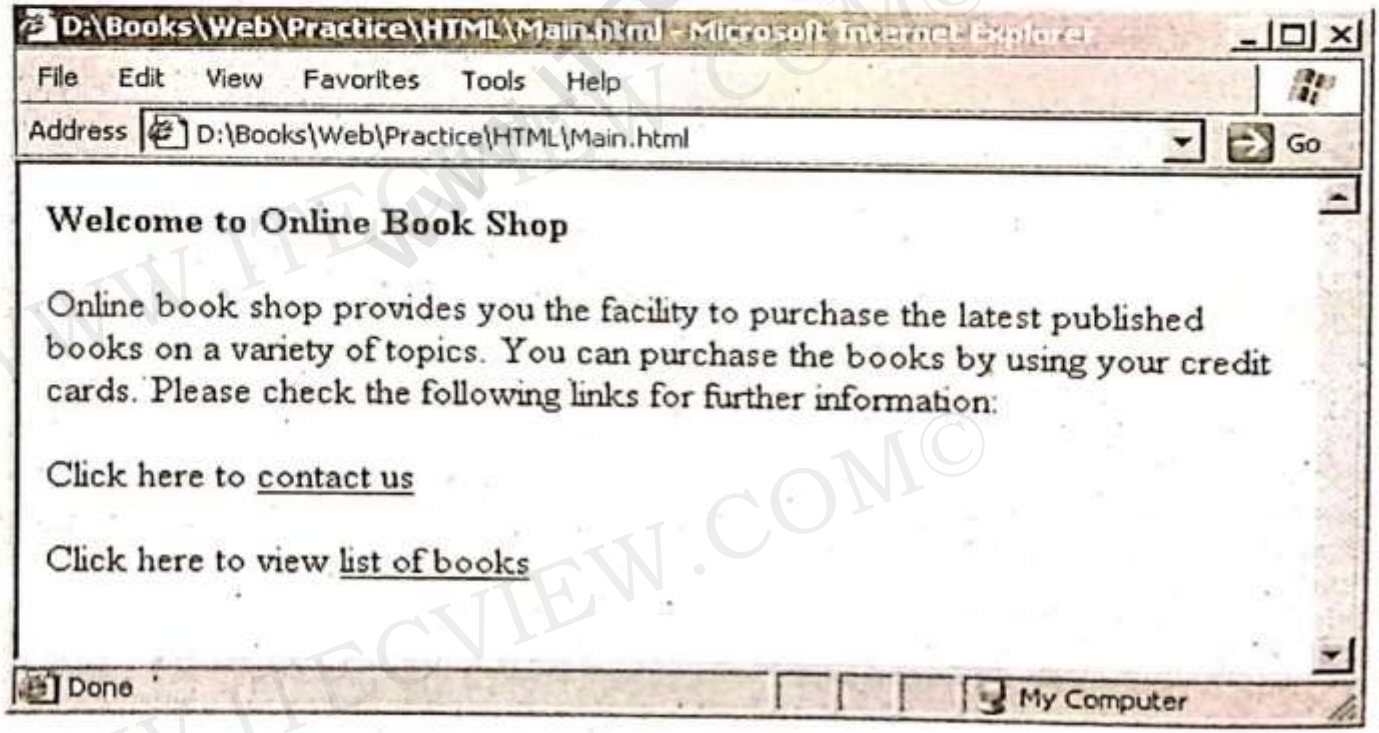
```
<HTML>
<BODY>
<H4>Contact Us</H4>
```

```
Our contact details are as follows: <P>
Web: www.itseries.com.pk <BR>
Email: itseries@msn.com.pk <BR>
Mob: 0300-6627905, 0321-6627905
<P>Click here to go to <A HREF="Main.html">main page</A> <P>
</BODY>
</HTML>
```

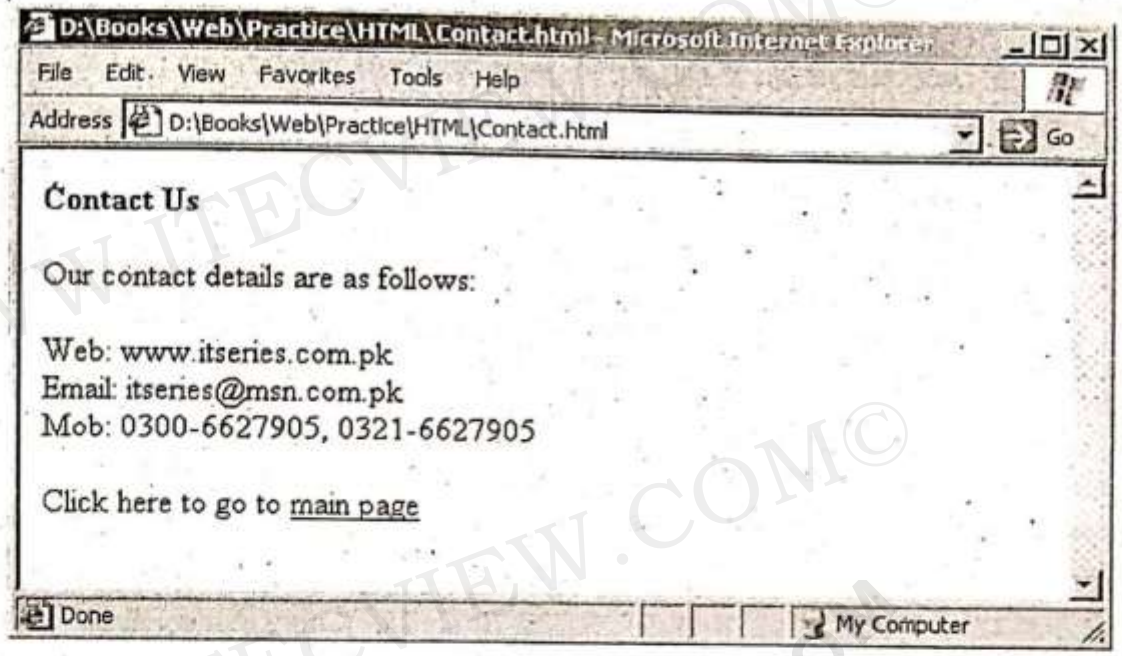
4. Save the file as "Contact.html".
5. Open another file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Books</H4>
Some of our books are as follows: <P>

1. Visual Programming using Visual Basic <BR>
2. A Comprehensive Study of Operating Systems <BR>
3. The Concepts of Information Technology <BR>
4. A Fundamental Study of Database Management Systems <BR>
5. A Practical Approach to Web Programming

<P>Click here to go to <A HREF="Main.html">main page</A> <P>
</BODY>
</HTML>
```

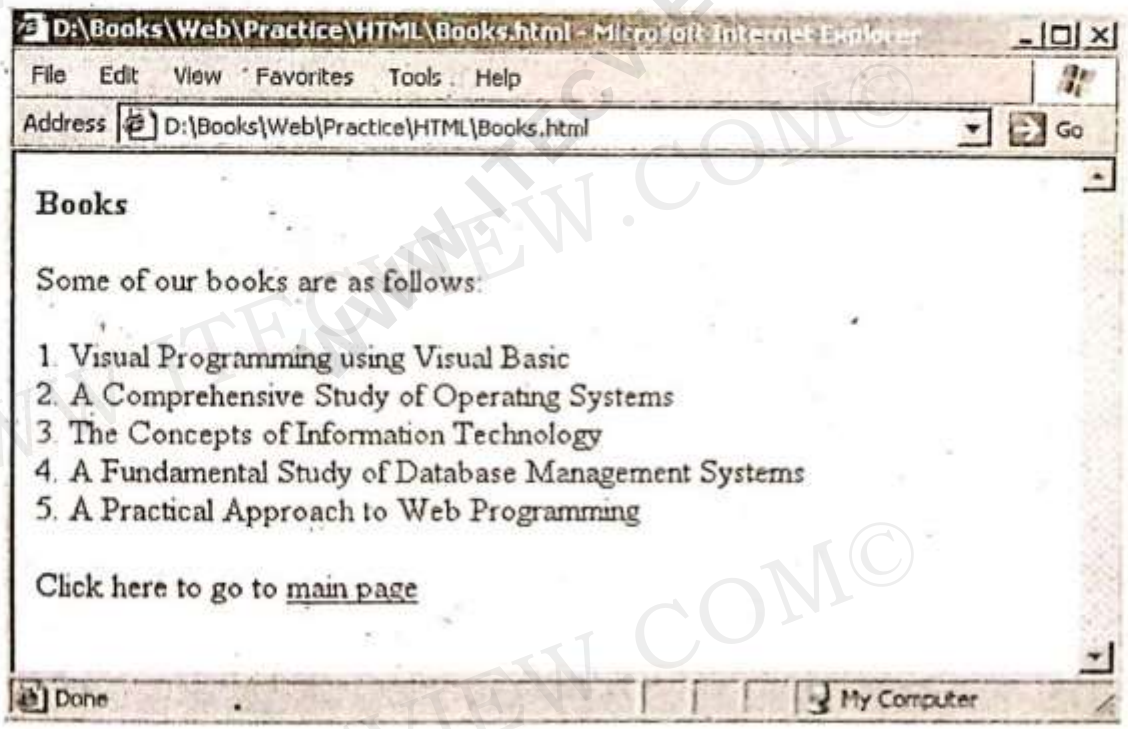
6. Save the file as "Books.html".
7. Double click "Main.html". The web page will appear in the browser as follows:



8. Click on **contact us** hyperlink. "Contact.html" will appear in the browser.



9. Click on main page hyperlink to move to "Main.html" again.
10. On "Main.html", click on list of books hyperlink. "Books.html" will appear in the browser as follows:



4.2.2 Links to the Page of other Website

A hyperlink can refer to a web page of another website. The visitor can move to that page by clicking that hyperlink.

Example

```
<A HREF="http://www.itseries.com.pk/Books.asp">Click to view list of books</A>
```

http:// must be used if the link refers to a web page of another website.

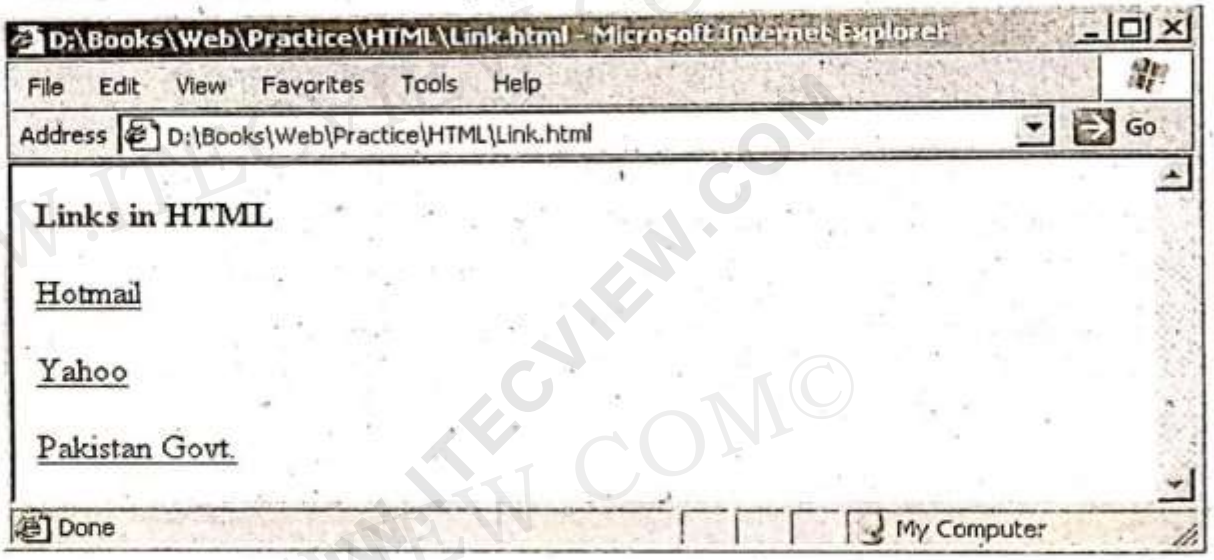
Practice 4.2

Write HTML code that displays three hyperlinks to three websites.

1. Open Notepad and type the following code:

```
<HTML>
<BODY LINK="red" ALINK="green" VLINK="orange">
<H4>Links in HTML</H4>
<A HREF="http://www.hotmail.com">Hotmail</A> <P>
<A HREF="http://www.yahoo.com">Yahoo</A> <P>
<A HREF="http://www.pakistan.gov.pk">Pakistan Govt.</A>
</BODY>
</HTML>
```

2. Save the file as "Link.html".
3. Double click the file. The web page will appear in the browser as follows:



4.2.3 Links to other Parts of the Same Page

A hyperlink can refer to another part of the same web page. This is done in two steps:

- Identify the location that will be connected with the hyperlink. **NAME** attribute of **<A>** tag is used for this purpose.
- Create the hyperlink

Example

```
<A NAME="location"></A>
<A HREF="#location">Jump</A>
```

Practice 4.3

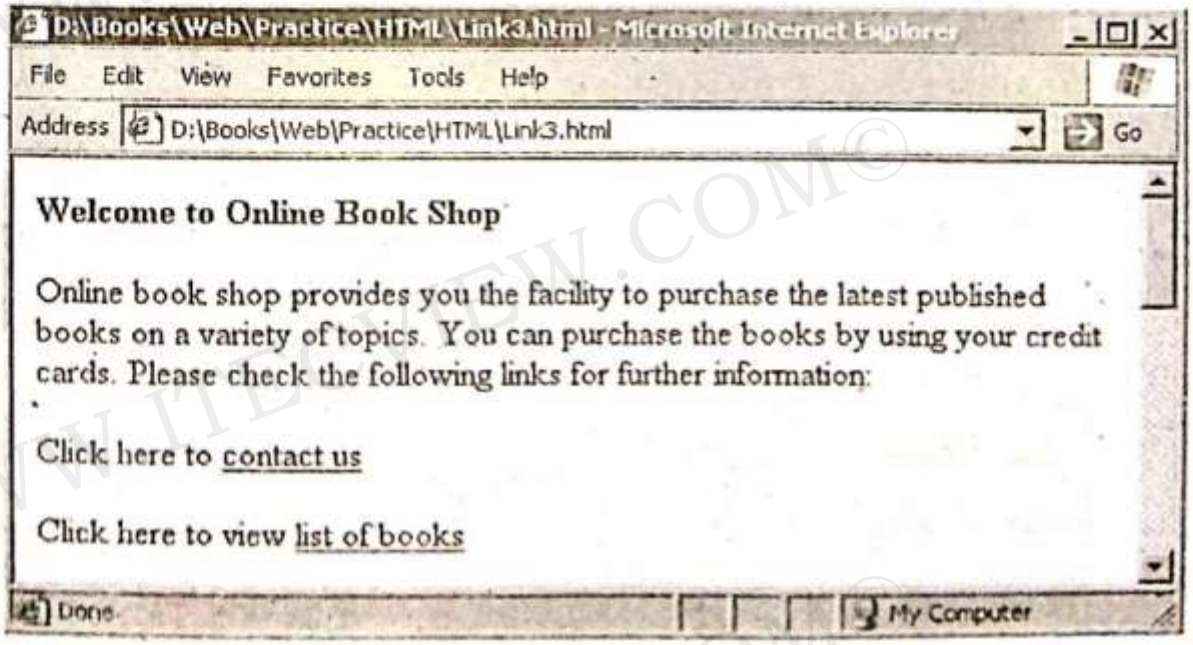
Write HTML code that connects different parts of the same page using hyperlinks to jump from one part to another.

1. Open Notepad and type the following code:

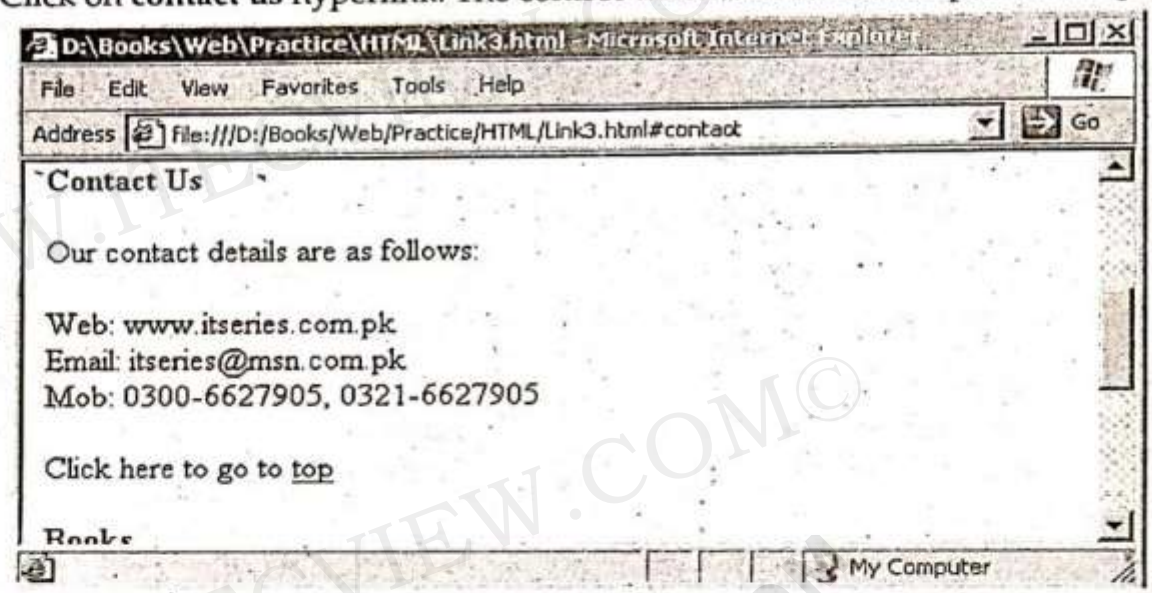
```
<HTML>
<BODY>
<A NAME="top"></A>
<H4>Welcome to Online Book Shop</H4>
```

```
Online book shop provides you the facility to purchase the latest published books on a variety
of topics. You can purchase the books by using your credit cards.
Please check the following links for further information: <P>

Click here to <A HREF="#contact">contact us</A> <P>
Click here to view <A HREF="#books">list of books</A> <P>

<A NAME="contact"></A>
<H4>Contact Us</H4>

Our contact details are as follows: <P>

Web: www.itseries.com.pk <BR>
Email: itseries@msn.com.pk <BR>
Mob: 0300-6627905, 0321-6627905

<P>Click here to go to <A HREF="#top">top</A> <P>

<A NAME="books"></A>
<H4>Books</H4>
Some of our books are as follows: <P>
1. Visual Programming using Visual Basic <BR>
2. A Comprehensive Study of Operating Systems <BR>
3. The Concepts of Information Technology <BR>
4. A Fundamental Study of Database Management Systems <BR>
5. A Practical Approach to Web Programming

<P>Click here to go to <A HREF="#top">top</A> <P>

</BODY>
</HTML>
```

2. Save the file as "Link3.html".
3. Double click the file. The web page will appear in the browser as follows:



4. Click on **contact us** hyperlink. The control will move to contact part of the page.



Note that the address bar displays **Link3.html#contact**. It indicates that the browser is displaying a part of **Link3.html** named **contact**.

5. Click on **top** hyperlink. It will move to the top of the page. You can again move to books part by clicking on **list of books** hyperlink.

4.2.4 Email Hyperlinks

An email hyperlink is a type of hyperlink that is linked to send email to the specified email address. As the user clicks on the hyperlink, the default email editor opens. The **mailto** attribute is used in **<A>** tag to create email hyperlink.

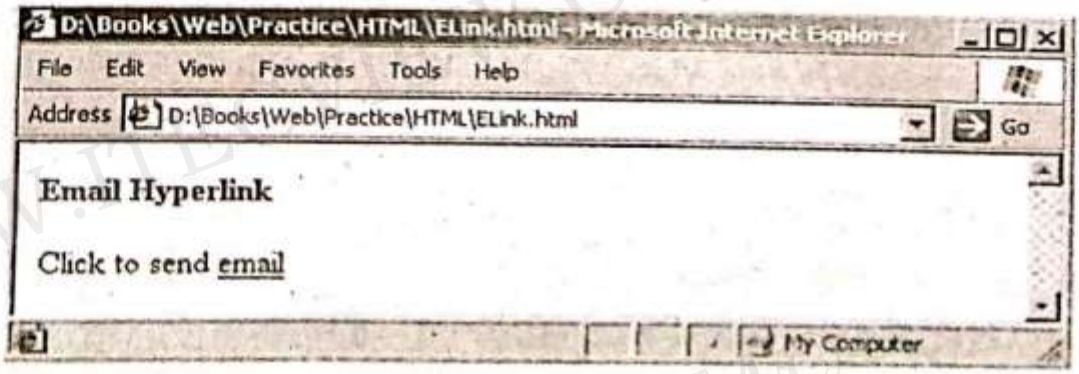
Practice 4.4

Write HTML code that displays an email hyperlink.

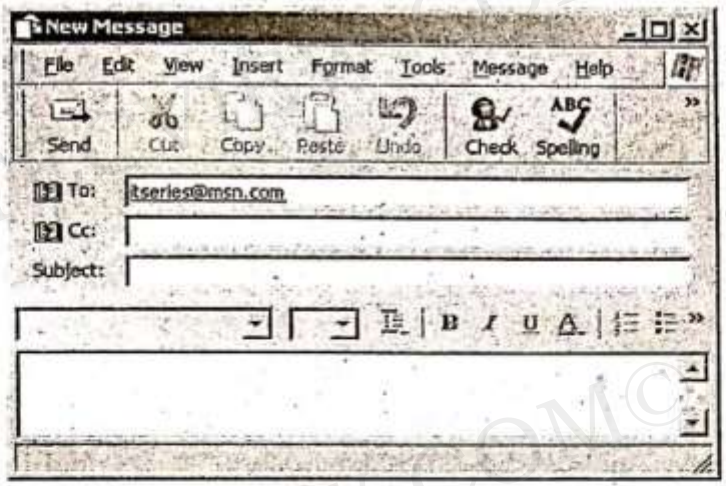
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Email Hyperlink</H4>
Click to send <A HREF="mailto:itseries@msn.com">email</A>
</BODY>
</HTML>
```

2. Save the file as "ELink.html".
3. Double click the file. The web page will appear in the browser as follows:



4. Click on **email** hyperlink. The default email editor will open as follows:



Note that the email address specified in mailto attribute automatically appears in **To** textbox of the email editor.

4.2.5 Images as Hyperlinks

Images can also be used as hyperlinks. The method of using images as hyperlinks is similar to text hyperlinks.

Example

```
<A HREF="http://www.itseries.com.pk"><IMG SRC="itmono.jpg"></A>
```

In the above example, the image **itmono.jpg** will become a hyperlink.

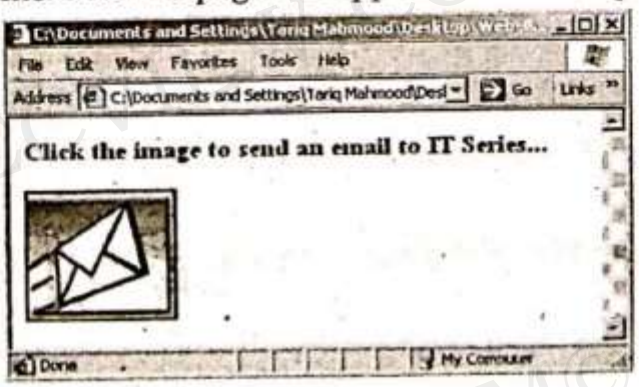
Practice 4.5

Write HTML code that displays an image as email hyperlink.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
 <H3>Click the image to send an email to IT Series...</H3>
 <P>
 <A HREF="mailto:info@itseries.com.pk">
  <IMG SRC="mail.jpg" ALT = "Send email to IT Series">
 </A>
</BODY>
</HTML>
```

2. Save the file as "ImgLink.html".
3. Double click the file. The web page will appear in the browser as follows:



Practice 4.6

Write HTML code that displays your CV.

1. Open Notepad and type the following code:

```
<HTML><BODY>
<H2>Usman Khalil</H2>
<H5>    Madina Town, Faisalabad, Pakistan.<BR> Ph: 0300-6627905 <BR>
        Email: <A HREF= "mailto:usman@itseries.com.pk">usman@itseries.com.pk</A>
</H5> <HR>
<H3>Objective</H3>
<UL>
 <LI>To work as Web Programmer
</UL>
<H3>Education</H3>
<UL>
 <LI>M.Sc. <I>Computer Science</I><BR> University of East London,UK
</UL>
<H3>Skills</H3>
<OL>
 <LI><B>Languages</B>
  <UL>
        <LI>HTML / XHTML
        <LI>C / C++
        <LI>Cascading Style Sheets
  </UL>
 <LI><B>Web Technologies</B>
  <UL>
        <LI>ASP
        <LI>JSP
  </UL>
 <LI><B>Databases</B>
  <UL>
        <LI>Oracle
        <LI>SQL Server
        <LI>MS Access
  </UL>
</OL>
<H3>Experience</H3>
<TABLE BORDER="1">
<TR>
 <TH>Organization</TH> <TH>Position</TH> <TH>Duration</TH>
</TR>
<TR>
 <TD>Tec2Day</TD> <TD>Programmer</TD> <TD>January 2003 to April 2004</TD>
</TR>
<TR>
 <TD>SoftVision</TD> <TD>Web Master</TD> <TD>May 2004 to March 2005</TD>
</TR>
<TR>
 <TD>New Creative Soft</TD> <TD>Web Programmer</TD> <TD>April 2005 to date</TD>
</TR>
</TABLE>
```

WWW.ITECVIEW.COM©

```
<H3>Interests & Activities</H3>
<UL>
        <LI>Cricket
        <LI>Literature
        <LI>Music
</UL><HR>
</BODY>
</HTML>
```

2. Save the file as "CV.html".
3. Double click the file. The web page will appear in the browser as follows:

4.2.6 Hyperlinks in Lists

The hyperlinks can also be displayed in ordered or unordered lists. The hyperlink tag is nested in the list tag to display hyperlinks in lists.

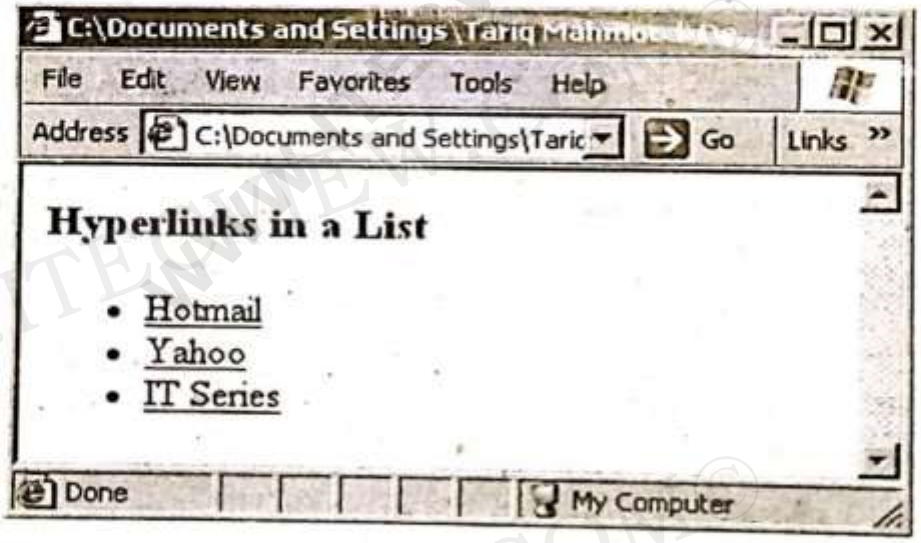
Practice 4.7

Write HTML code that displays three hyperlinks in an unordered list.

1. Open Notepad and type the following code:

```
<HTML>
  <BODY>
    <H3>Hyperlinks in a List</H3>
    <UL>
      <LI>
          <A HREF="http://www.hotmail.com">Hotmail</A>
      <LI>
          <A HREF="http://www.yahoo.com">Yahoo</A>
      <LI>
          <A HREF="http://www.itseries.com.pk">IT Series</A>
    </UL>
  </BODY>
</HTML>
```

2. Save the file as "ListLink.html".
3. Double click the file. The web page will appear in the browser as follows:



4.2.7 Hyperlinks in Tables

The hyperlinks can also be displayed in tables. The hyperlink tag is nested in the table tag to display hyperlinks in tables.

Practice 4.8

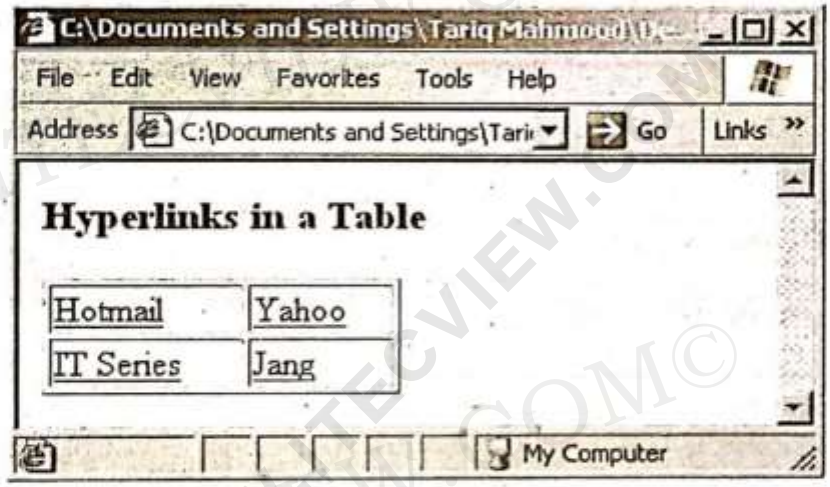
Write HTML code that displays four hyperlinks in two columns and two rows of a table.

1. Open Notepad and type the following code:

```
<HTML>
  <BODY>
    <H3>Hyperlinks in a Table</H3>
```

```
    <TABLE BORDER="1" WIDTH="50%">
     <TR>
          <TD><A HREF="http://www.hotmail.com">Hotmail</A></TD>
          <TD><A HREF="http://www.yahoo.com">Yahoo</A></TD>
         </TR>
     <TR>
          <TD><A HREF="http://www.itseries.com.pk">IT Series</A></TD>
          <TD><A HREF="http://www.jang.com.pk">Jang</A></TD>
         </TR>
    </TABLE>
  </BODY>
</HTML>
```

2. Save the file as "TableLink.html".
3. Double click the file. The web page will appear in the browser as follows:



4.3 Image Maps

An **image map** is a clickable image that is divided into different regions or areas. These regions are also called **hotspots**. Each region of image map is linked to a different URL. It loads different web pages when the user clicks its different parts.

For example, the following image contains three shapes. Each shape of the image can be used as a separate hyperlink.

The "USEMAP" Attribute

Image map is associated with an image by adding the **usemap** attribute to **<IMG>** tag.

```
<IMG SRC="shapes.jpg" WIDTH="375" HEIGHT="102" BORDER="0" USEMAP="#shapes">
```

In the above line, **USEMAP="#shapes"** associates the image map with the image.

The <MAP> Tag

The **<MAP>** tag defines the map for creating an image map. It tells the browser where the hot spots exist in the image. It also links the hot spots with different locations.

```
<MAP NAME="shapes">
<AREA SHAPE="circle" COORDS="58,50,40" HREF="circle.html" TITLE="Circle">
<AREA SHAPE="rect" COORDS="136,11,227,89" HREF="rect.html" TITLE="Rectangle">
<AREA SHAPE="tri" COORDS="309,13,358,89,257,89" HREF="tri.html" TITLE="Triangle">
<AREA SHAPE="default" NOHREF>
</MAP>
```

The above example defines three "hot spot" areas i.e. circle, rectangle and triangle. Each area is linked to a different page. The name of the above map is **shapes.** It is used in the **USEMAP** attribute of the image. The above map defines the following areas:

- **circle:** a circular shape
- **rect:** a rectangular shape
- **tri:** an arbitrary polygon, with 3 or more points
- **default:** represents the remaining area of the image not defined by any **area** tags.

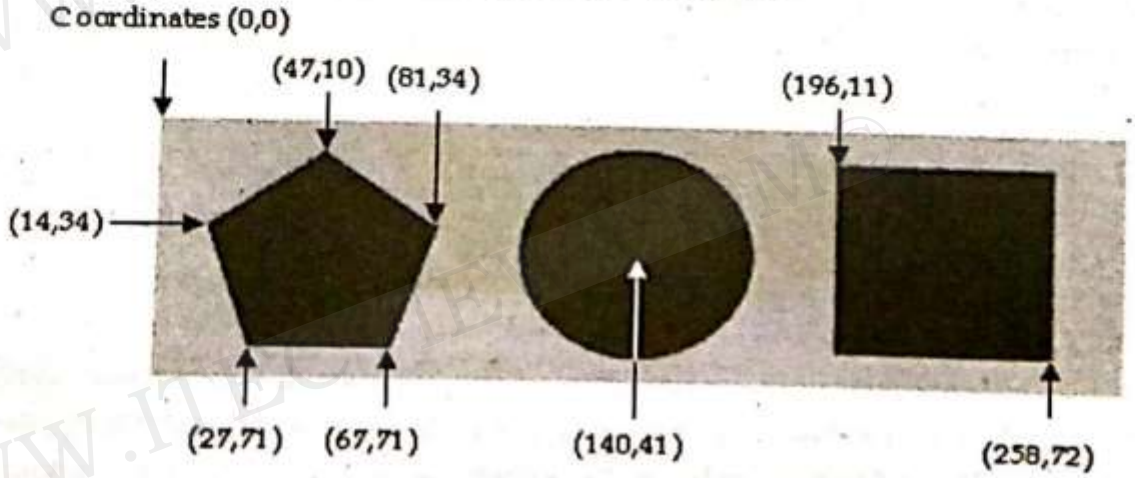
COORDS attribute specifies the coordinates that define the corners of the shape. The coordinates depend on the shape specified in the **shape** attribute:

Shape	Coordinates
rect	COORDS="x1,y1,x2,y2" x1, y1 defines the top left corner of rectangle and x2, y2 define the bottom right corners of the rectangle.
circle	COORDS="x,y,r" x, y define the centre of the circle and r defines the radius of the circle.
poly	COORDS="x1,y1,x2,y2,x3,y3,..." (The corners of the polygon)

Note that all coordinate values are relative to the **top left** corner of the image. The top left corner always has coordinates **(0,0)**. The **default** shape type does not require coordinates. If **NOHREF** attribute is used, the hot spot will not link to anything. The **TITLE** attribute specifies the area a title. The title appears as tool tip when mouse is rolled over the hot spot.

Practice 4.9

Write HTML code that dispalys an image map whose different parts refers to different web pages. The practice uses the following image "Test.jpg":



1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<IMG SRC="Test.jpg" border="0" USEMAP="#TestMap">
```

```
<MAP NAME="TestMap">
<AREA SHAPE="poly"      COORDS="47,10,14,34,27,71,67,71,81,34"      HREF="poly.html"
TITLE="Polygon">
<AREA SHAPE="circle" COORDS="140,41,34" HREF="circle.html" TITLE="Circle">
<AREA SHAPE="rect" COORDS="196,11,258,72" HREF="rect.html" TITLE="Rectangle">
<AREA SHAPE="default" NOHREF>
</MAP>
</BODY>
</HTML>
```

2. Save the file as "ImageMap.html".
3. Double click the file. The web page will appear in the browser as follows:

In the above example, each shape in the image refers to different web page.

Exercise Questions

1. Write HTML code that displays three hyperlinks to different websites. The websites should open in a new window when the user clicks on the hyperlinks.
2. Write HTML code that creates an unordered list containing hyperlinks to five Web pages.
3. Write HTML code that displays three images as hyperlinks. Two image hyperlinks should refer to any two websites and the third image should be an email hyperlink.

WWW.ITECVIEW.COM©

Frames and Forms

5.1 Frames

A frame is a container window that can display a web page. A web page may consist of many frames. Each frame in the browser window can display different web pages. Many web pages can be displayed at the same time by using frames. Frame is used to break the web page into different sections. These sections work independently but appear simultaneously in the browser. Frames were first introduced with Netscape Navigator 2.0. They are accepted widely by most of the browsers. Frames can be powerful in designing the web pages if they are used properly. The frames usually are used for site navigation as follows:

1. The title frame appears at the top of the webpage.
2. The navigational frame appears on the left side of the webpage.
3. The contents frame appears on the remaining part of the webpage.

5.1.1 Advantages of Frames

The advantages of frames are as follows:

1. The frames allow to keep one part of page static and change the other parts of page.
2. The frames can be used to display multiple pages in the same browser.
3. Frames can be used to reduce the load on server because all pages are not reloaded when a new page is visited.
4. Frames can provide a navigational tool if they are used properly.

5.1.2 Disadvantages of Frames

The disadvantages of frames are as follows:

1. Search engines do not index the web pages with frames.
2. It is difficult to print the content of all frames as compared to a normal web page.
3. Most browsers do not allow to bookmark an individual frame page.

<FRAMESET> Tag

Frameset is a container or placeholder for multiple frames in same browser window. <FRAMESET> tag is used to define the style and number of frames in a page. The ROWS and COLS attributes of frameset are used to specify the frameset characteristics. A frameset can be specified as COLS or ROWS but not both.

Attributes

Some important attributes of <FRAMESET> tag are as follows:

Attribute	Description
ROWS	It divides the screen into multiple rows. Its value can be given as: • Number of pixels • Percentage of the screen • * symbol to indicate all remaining space
COLS	It divides the screen into multiple columns. Its value can be given as: • Number of pixels • Percentage of the screen • * symbol to indicate all remaining space
BORDER	It specifies the width of the border in pixels drawn around the frames.
FRAMESPACING	It specifies the amount of space between the frames in pixels.

Examples

```
<FRAMESET ROWS="40%,40%,20%>
</FRAMESET>
```

The above example divides the screen in three horizontal parts. The first two parts consist of 40% each and the last part consists of 20% of the screen.

```
<FRAMESET COLS="40%,30%,*>
</FRAMESET>
```

The above example divides the screen in three vertical parts. The first part consists of 40% of screen, second part consists of 30% and remaining area is allocated to the third part.

<FRAME> Tag

<FRAME> tag is used to specify the web page that will be displayed in that frame. **<FRAME>** tag is used within **<FRAMESET>** tag.

Attributes

Some important attributes of <FRAME> tag are as follows:

Attribute	Description
SRC	It specifies the page to be displayed in the frame.
MARGINHEIGHT	It specifies the amount of space to be left at the top and left side of frame in pixels
MARGINWIDTH	It specifies the amount of space to be left along the sides of frame in pixels
NAME	It specifies the name of the frame
NORESIZE	It makes the frame un-resizable
SCROLLING	It specifies the appearance of horizontal and vertical scrollbars of frame. The possible values are **yes, no** and **auto.**
BORDER	It specifies the width of the frame border in pixels.

Example

```
<FRAMESET ROWS="33%,33%,*>
<FRAME SRC="top.html" NAME="Top">
<FRAME SRC="center.html" NAME="Center">
```

```
<FRAME SRC="bottom.html" NAME="Bottom">
</FRAMESET>
```

The above example divides the screen into three horizontal frames. The first frame named **Top** contains **top.html**, the second frame named **Center** contains **center.html** and third frame named **Bottom** contains **bottom.html**.

Target Frame

The **target** attribute in **<A>** tag is used to specify the target frame in which the page will be loaded. This attributes is frequently used with frames. It can be used to open different web pages in the same frame while other frames remain static. The possible values of **target** attribute are as follows:

Value	Description
_parent	The page is loaded into the frameset containing the current frame.
_top	The page will replace the topmost frameset and load into the whole browser window.
_blank	The page is loaded into a new browser window.
_self	The page is loaded into the same window in which the link exists. It is the default target.

<NOFRAMES> Tag

<NOFRAMES> tag is used to specify an alternate web page if the browser does not support frames. It is also very useful to specify a webpage in <NOFRAME> tag for the search engines to index the site.

Practice 5.1

Write HTML code that displays two horizontal frames of 20% and 80%. The top frame should display three hyperlinks. The pages referred by the hyperlinks must appear in the bottom frame.

1. Open Notepad and type the following code:

```
<FRAMESET ROWS="20%,80%">
<FRAME SRC="Top.html" NAME="Top">
<FRAME SRC="Home.html" NAME="Home">
</FRAMESET>
```

2. Save the file as "Frame.html".
3. Open a new file in Notepad and type the following code:

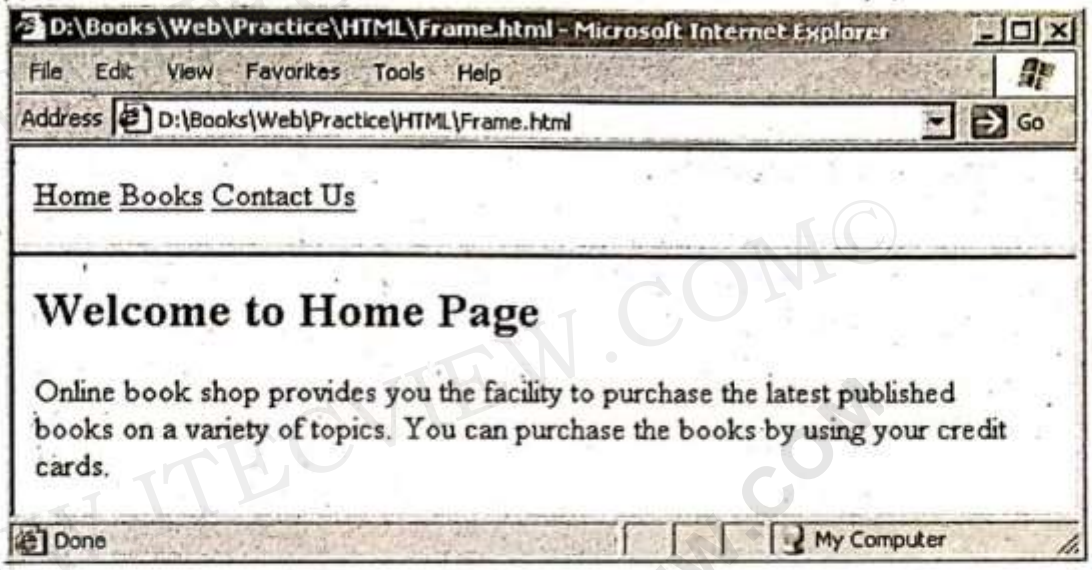
```
<HTML>
<BODY>
<H2>Welcome to Home Page</H2>
<H3>Online book shop provides you the facility to purchase the latest published books on a variety of topics. You can purchase the books by using your credit cards.</H3>
</BODY>
</HTML>
```

4. Save the file as "Home.html".
5. Open a new file in Notepad and type the following code:

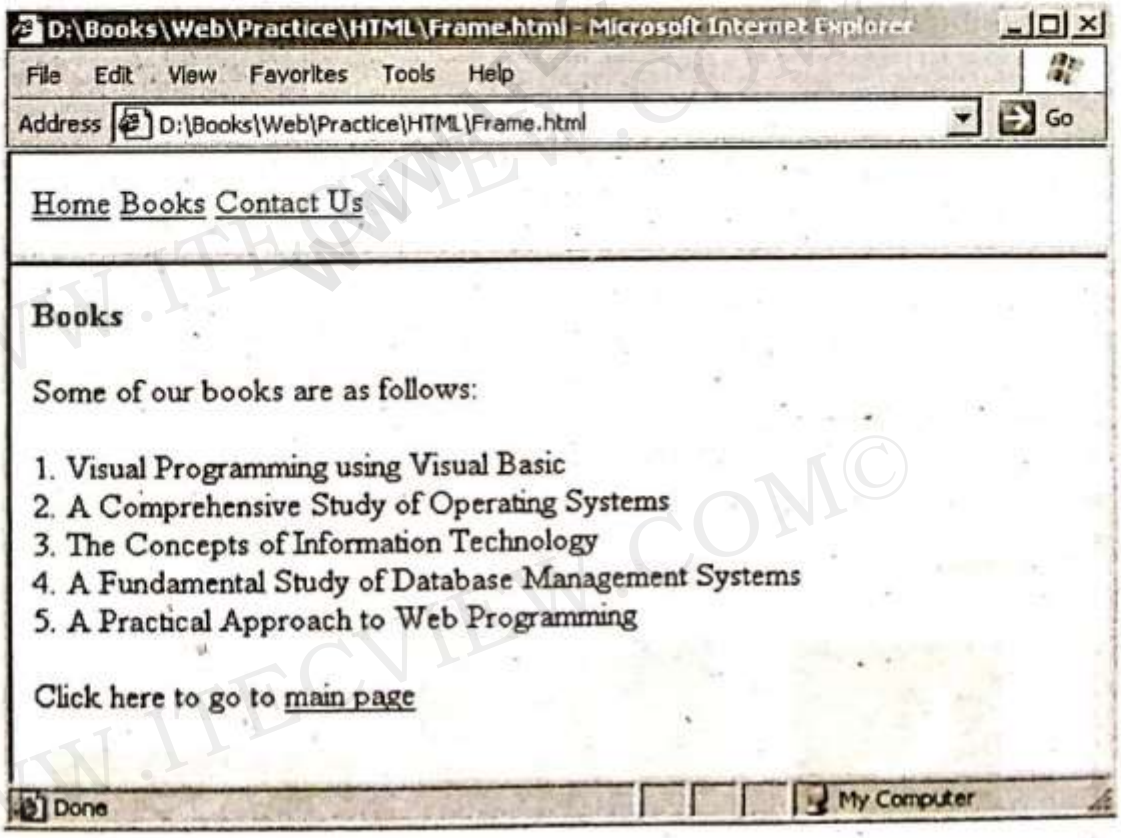
```
<HTML>
<BODY>
<A HREF="Home.html" TARGET="Home">Home</A>
<A HREF="Books.html" TARGET="Home">Books</A>
```

```
<A HREF="Contact.html" TARGET="Home">Contact Us</A>
</BODY>
</HTML>
```

6. Save the file as "Top.html".
7. Double click Frame.html file. The web page will appear as follows:



8. Click on **Books** hyperlink in the top frame. The page will change as follows:



Note that the contents of top frame remain static. The **Books.html** page appears in the **Home** frame as specified in the **target** attribute of the **<A>** tag in **Top.html**.

Practice 5.2

Write HTML code that divides the display screen in three frames and display three flags (Pak.jpg, Bangladesh.jpg and Iran.jpg) in each frame. Assume these images are placed in C: drive.

1. Open Notepad and type the following code:

```
<FRAMESET COLS="33%,33%,*">
<FRAME SRC="Pak.html" NAME="Pak">
<FRAME SRC="Bangladesh.html" NAME="Bangladesh">
<FRAME SRC="Iran.html" NAME="Iran">
</FRAMESET>
```

2. Save the file as "Flags.html".
3. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Pakistan </H4>
<IMG SRC="Pak.jpg">
</BODY>
</HTML>
```

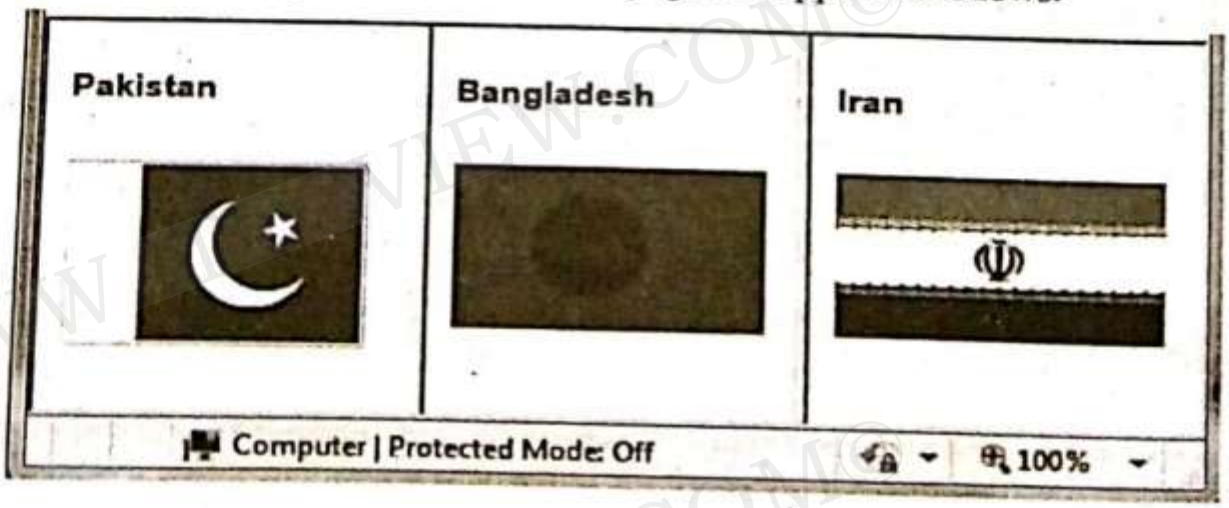
4. Save the file as "Pak.html".
5. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Bangladesh </H4>
<IMG SRC="Bangladesh.jpg">
</BODY>
</HTML>
```

6. Save the file as "Bangladesh.html".
7. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Iran </H4>
<IMG SRC="Iran.jpg">
</BODY>
</HTML>
```

8. Save the file as "Iran.html".
9. Double click "Flags.html" file. The web page will appear as follows:



Practice 5.3

Write HTML code that creates a frame with a window of contents on the left side of the window and have each entry in the table of contents. Use internal link to scroll down the document to the appropriate subsection.

1. Open Notepad and type the following code:

```
<FRAMESET COLS="20%,*">
<FRAME SRC="TOC.html" NAME="TOC">
<FRAME SRC="Main.html" NAME="Main">
</FRAMESET>
```

2. Save the file as "Frame2.html".
3. Open a new file in Notepad and type the following code:

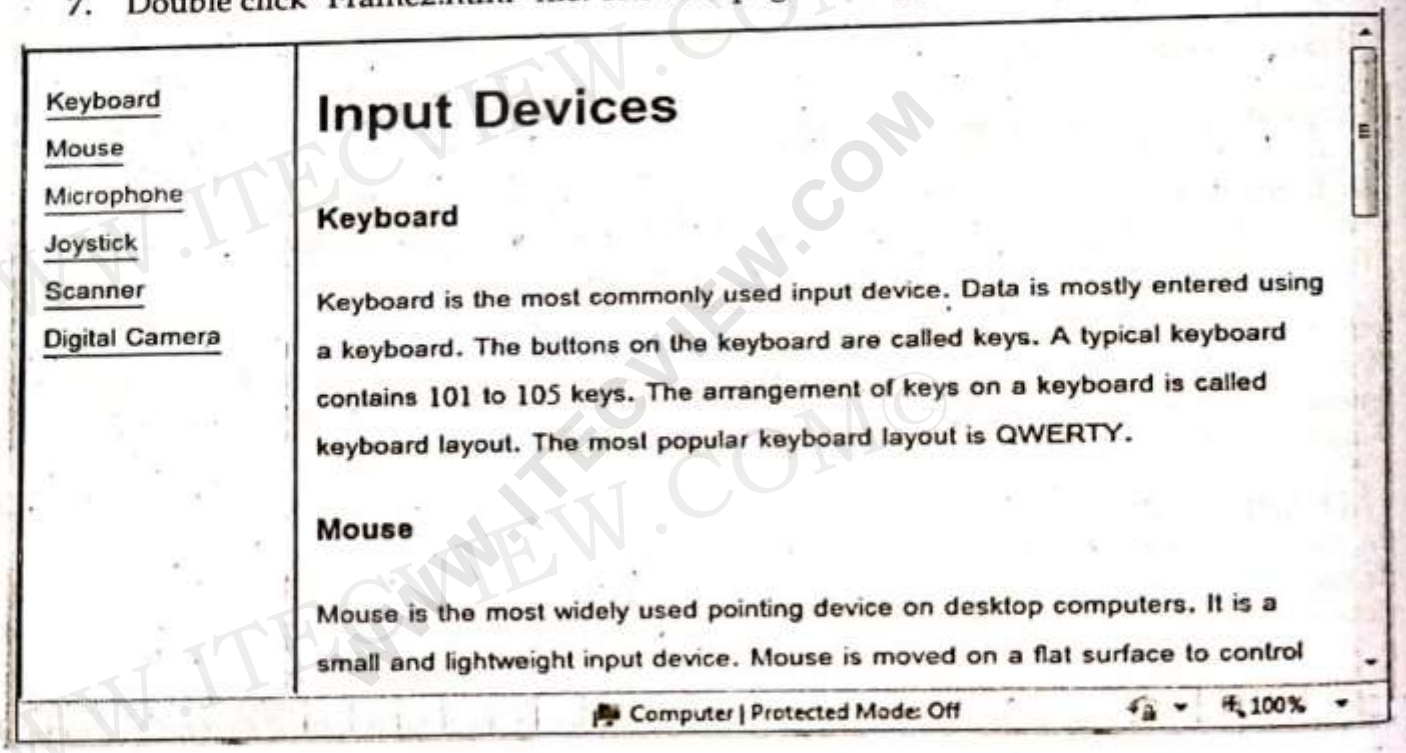
```
<HTML>
<BODY>

<H1>Input Devices</H3>

<A NAME="kb"><H3>Keyboard </H3></A>

Keyboard is the most commonly used input device. Data is mostly entered using a keyboard.
The buttons on the keyboard are called keys. A typical keyboard contains 101 to 105 keys.
The most popular keyboard layout is QWERTY.

<A NAME="ms"><H3>Mouse </H3>
Mouse is a small and lightweight input device. Mouse is moved on a flat surface to control the
movement of the cursor or pointer on the screen. A mouse is very easy to use. It is mostly
used in graphic applications.

<A NAME="mp"><H3>Microphone </H3>2:29 PM 13/08/2010
A microphone is an input device. It is used to digitally record audio data like such as human
voice. It enables a user to input text and issue commands orally. The software in the
computer converts the sound waves into digital form.

<A NAME="js"><H3>Joystick </H3>
A joystick consists of a base and a stick. The stick can be moved in any direction to move an
object around the computer screen. A joystick can perform a similar function to a mouse or
trackball. The most common use of a joystick is for playing computer games.

<A NAME="sc"><H3>Scanner</H3>
Digital scanner is an input device. It is used to read the pictures and other printed material
placed on its glass. It reads the picture, translates it in digital form and stores it in the
computer. Digital scanner uses laser technology to read images.

<A NAME="ca"><H3>Digital Camera </H3>
Digital camera is an input device. It stores images digitally in its memory rather than recording
on film. The pictures taken with a digital camera can be transferred to a computer system.
The number of digital photos depends on the amount of memory in camera.

</BODY>
</HTML>
```

4. Save the file as "Main.html".

5. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>

<A HREF="Main.html#kb" TARGET="Main">Keyboard</A><BR>
<A HREF="Main.html#ms" TARGET="Main">Mouse</A><BR>
<A HREF="Main.html#mp" TARGET="Main">Microphone</A><BR>
<A HREF="Main.html#js" TARGET="Main">Joystick</A><BR>
<A HREF="Main.html#sc" TARGET="Main">Scanner</A><BR>
<A HREF="Main.html#ca" TARGET="Main">Digital Camera</A><BR>

</BODY>
</HTML>
```

6. Save the file as "TOC.html".
7. Double click "Frame2.html" file. The web page will appear as follows:



8. Click any link in the left window. The related part will appear.

5.1.3 Nested Frames

A frame within a frame is called **nested frame**. Nested frames are used to divide the page into multiple levels.

Example

```
<FRAMESET ROWS="20%,80%">
<FRAME SRC="Top.html" NAME="Top">
<FRAMESET COLS="20%,80%>
<FRAME SRC="Left.html" NAME="Left">
<FRAME SRC="Main.html" NAME="Main">
</FRAMESET>
</FRAMESET>
```

The above example creates the following sketch:

Displays Top.html	
Displays Left.html	Displays Main.html

Practice 5.4

Write HTML code that creates nested frames. The top frame should display the name of organization. The left frame should display a list of hyperlinks. The right frame should display the pages refered by the hyperlinks in the left frame.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Books</H4>
Some of our books are as follows: <P>
1. Visual Programming using Visual Basic <BR>
2. A Comprehensive Study of Operating Systems <BR>
3. The Concepts of Information Technology <BR>
4. A Fundamental Study of Database Management Systems <BR>
5. A Practical Approach to Web Programming
</BODY>
</HTML>
```

2. Save the file as "Books.html".
3. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Contact Us</H4>
Our contact details are as follows: <P>
Web: www.itseries.com.pk <BR>
Email: itseries@msn.com.pk <BR>
Mob: 0300-6627905, 0321-6627905
</BODY>
</HTML>
```

4. Save the file as "Contact.html".
5. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Welcome to Online Book Shop</H4>
Online book shop provides you the facility to purchase the latest published books on a variety
of topics. You can purchase the books by using your credit cards.
</BODY>
</HTML>
```

6. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<A HREF="Main.html" TARGET="Main">Main</A><BR>
<A HREF="Books.html" TARGET="Main">Books</A><BR>
```

```
<A HREF="Contact.html" TARGET="Main">Contact Us</A>
</BODY>
</HTML>
```

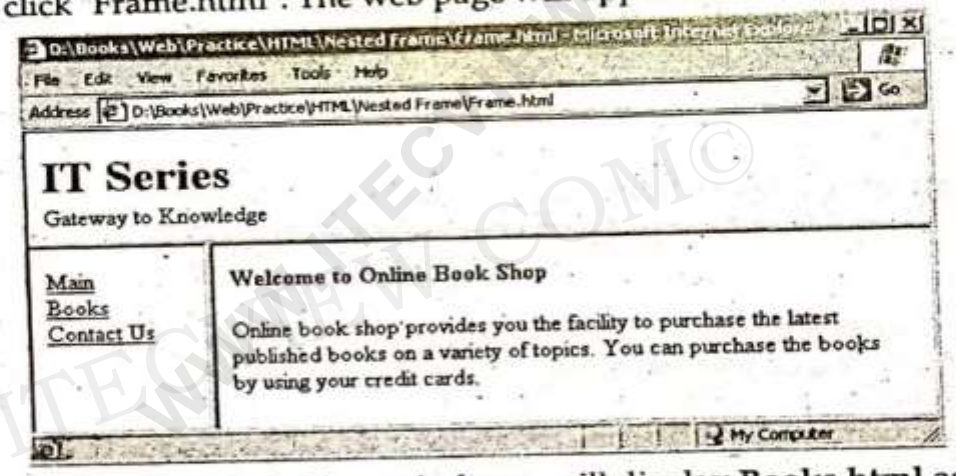
7. Save the file as "Menu.html".
8. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H1>IT Series</H1>Gateway to Knowledge
</BODY>
</HTML>
```

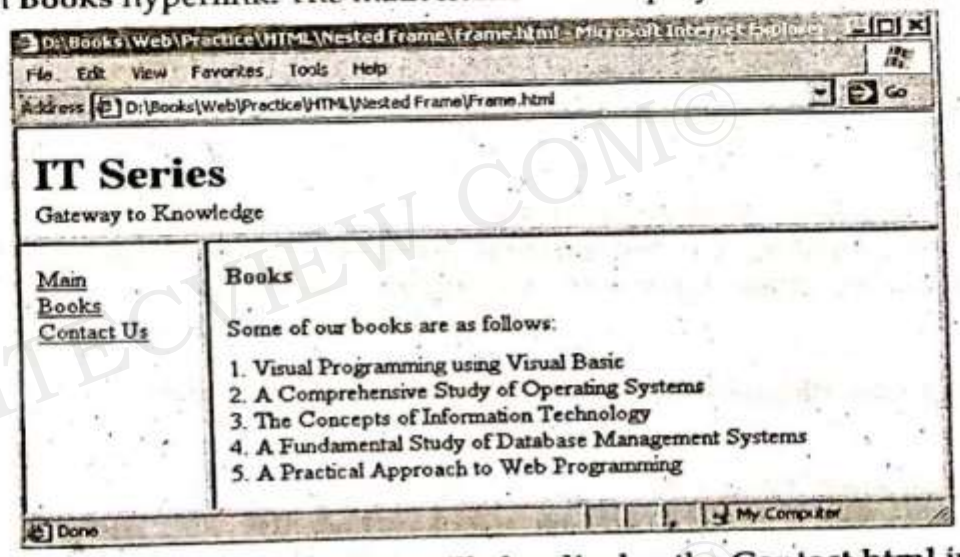
9. Save the file as "IT.html".
10. Open a new file in Notepad and type the following code:

```
<FRAMESET ROWS="20%,80%">
<FRAME SRC="Top.html" NAME="Top">
<FRAMESET COLS="20%,80%">
<FRAME SRC="Menu.html" NAME="Left">
<FRAME SRC="Main.html" NAME="Main">
</FRAMESET>
</FRAMESET>
```

11. Save the file as "Frame.html".
12. Double click "Frame.html". The web page will appear in the browser as follows:



13. Click on **Books** hyperlink. The main frame will display **Books.html** as follows:



Similarly, the **Contact Us** hyperlink will also display the **Contact.html** in **Main** frame.

5.2 Forms

Forms are used to create interactive websites. The visitors can send feedback, emails, comments and suggestions by using these forms. Form is an easier way for the user to communicate with the webmaster or the owner of the website.

5.2.1 Uses of Forms

Some uses of forms are as follows:

1. Educational Sites

Forms are used to collect names, addresses, emails and other information by the educational websites.

2. Online Orders

Many websites use forms to get online orders from the visitors. The user can send the order by entering information in the forms.

3. Feedback

Many websites use forms to get feedback from the users. The feedback of the visitors is an important source for improving the services or products of a company.

4. Interface for Chatting

Many users use forms for chatting with other people on the Internet. They have to fill different forms to participate in the chat. A form consists of different **form elements**. Each type of element is used for different purpose. Some important form elements are as follows:

- Textbox
- Checkbox
- Password field
- List Box
- Button
- Radio Button
- Text area

<FORM> Tag

<FORM> tag is used to create a form in a web page. All form elements are inserted within <FORM> and </FORM> tags. <FORM> tag indicates the start of form and </FORM> indicates the end of form. A web page may contain many forms.

Attributes

Some important attributes of <FORM> tag are as follows:

Attribute	Description
NAME	It specifies the name of the form
METHOD	It specifies how the form data is sent to the server. The possible values are as follows: **GET:** It sends the data as part of URL. The data included with URL can be seen by other people. It is also remembered by the browser. It is the default value. **POST:** It sends the data encoded in HTTP data stream. It is mostly used to send a large amount of data. The data is not displayed in URL. Therefore, it is more secure.
ACTION	It specifies where the form data will be submitted. It is normally sent to a script on the server or submitted as email.

Example

```
<FORM NAME="Form1" METHOD="post" ACTION="mailto:itseries@msn.com"> </FORM>
```

5.3 Form Elements

Most of the form elements are inserted in a form by using <INPUT> tag. The **TYPE** attribute of <INPUT> tag indicates the type of form elements to be inserted in the form. The general syntax of using this tag is as follows:

```
<INPUT TYPE="form_element">
```

In the above example, **form_element** is the type of element to be inserted in the form.

5.3.1 Text Field

Text fields are used to get input from the user in the form of letters, numbers etc.

Attributes

Some important attributes of text field are as follows:

Attribute	Description
NAME	It specifies the name of the text field
VALUE	It refers to the contents of the text field
DEFAULTVALUE	It indicates the default value of the text field
MAXLENGTH	It specifies the maximum number of characters that can be entered in the text field

Example

```
<INPUT TYPE="text" NAME="firstname">
<INPUT TYPE="text" NAME="lastname">
```

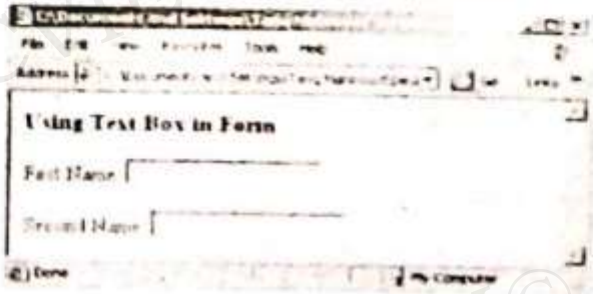
Practice 5.6

Write HTML code that displays two text boxes to input first and second name

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<H3>Using Text Box in Form</H3>
<FORM NAME="form1">
First Name: <INPUT TYPE="text" NAME="fname"><P>
Second Name: <INPUT TYPE="text" NAME="sname"><P>
</FORM>
</BODY>
</HTML>
```

2. Save the file as 'Textbox.html'
3. Double click the file. The web page will appear as follows:



5.3.2 Password Field

Password fields are used to input passwords in the form. Each character inserted in this field appears as *. The attributes of password field are same as text field.

Example

```
<INPUT TYPE="password" NAME="Password">
```

Practice 5.7

Write HTML code that displays a password field to input password.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Password Field in Form</H3>
<FORM NAME="form1">
Enter password: <INPUT TYPE="password" NAME="password">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Password.html".
3. Double click the file. The web page will appear as follows:

5.3.3 Radio Button

Radio buttons are used when the user has to choose one option from multiple choices.

Attributes

Some important attributes of radio button are as follows:

Attribute	Description
NAME	It specifies the name of the radio button
VALUE	It refers to the value attached with the radio button
CHECKED	It specifies that the radio button is selected by default

Example

```
<INPUT TYPE="radio" NAME="gender" VALUE="Male"> Male
<INPUT TYPE="radio" NAME="gender" VALUE="Female"> Female
```

All radio buttons with same name become a group. The user can select one radio button from the group. If the names are different, the user can select all radio buttons at the same time.

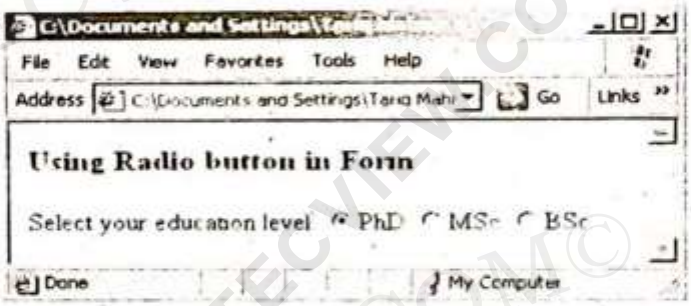
Practice 5.8

Write HTML code that displays three education levels of PhD, MSc and BSc using radio buttons

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Radio button in Form</H3>
<FORM NAME="form1">
Select your education level:
<INPUT TYPE="radio" NAME="edu" VALUE="PhD" CHECKED>PhD
<INPUT TYPE="radio" NAME="edu" VALUE="MSc">MSc
<INPUT TYPE="radio" NAME="edu" VALUE="BSc">BSc
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Radio.html".
3. Double click the file. The web page will appear as follows:



5.3.4 Checkbox Field

Checkboxes are used to provide many options to the user. The user can choose one or multiple options from available choices. The attributes of checkbox are same as radio button.

Example

```
<INPUT TYPE="checkbox" NAME="travel"> Traveling
<INPUT TYPE="checkbox" NAME="sports"> Sports.
<INPUT TYPE="checkbox" NAME="chatting"> Chatting
```

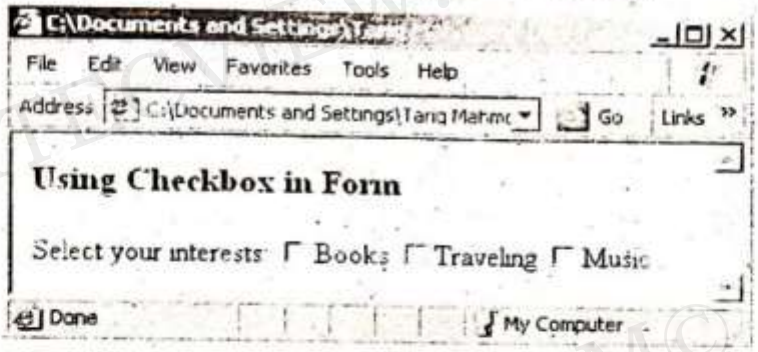
Practice 5.9

Write HTML code that displays three interests for user to choose using checkboxes.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Checkbox in Form</H3>
<FORM NAME="form1">
Select your interests
<INPUT TYPE="checkbox" NAME="books">Books
<INPUT TYPE="checkbox" NAME="travel">Traveling
<INPUT TYPE="checkbox" NAME="music">Music
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Checkbox.html".
3. Double click the file. The web page will appear as follows:



5.3.5 Select Field

Select field is used to provide predefined list of items to the user. The user can choose one or multiple options from the list depending upon the type of list. **<SELECT>** tag is used to create a list. **<OPTION>** tag is used to create an item in the list.

Attributes

Some important attributes of <SELECT> tag are as follows:

Attribute	Description
NAME	It specifies the name of the radio button
MULTIPLE	It specifies that multiple option can be selected at the same time by clicking CTRL key
SELECTED	It specifies that the current option is selected

Example

```
<SELECT NAME="City">
<OPTION VALUE="Islamabad" SELECTED>Islamabad
<OPTION VALUE="Lahore">Lahore
<OPTION VALUE="Karachi">Karachi
<OPTION VALUE="Peshawar">Peshawar
<OPTION VALUE="Quetta">Quetta
</SELECT>
```

Practice 5.10

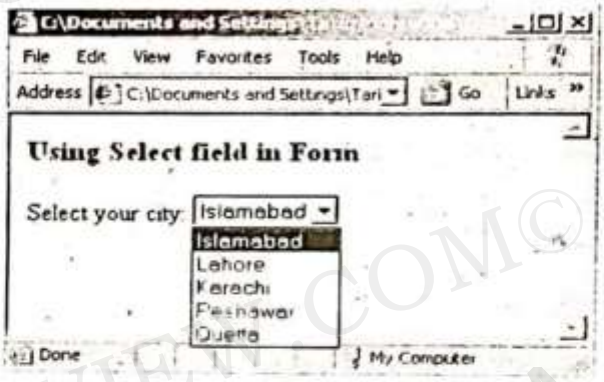
Write HTML code that displays names of five cities in a Select field.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Checkbox in Form</H3>
<FORM NAME="form1">
Select your city:
<SELECT NAME="City">
        <OPTION VALUE="Islamabad" SELECTED>Islamabad
        <OPTION VALUE="Lahore">Lahore
        <OPTION VALUE="Karachi">Karachi
        <OPTION VALUE="Peshawar">Peshawar
        <OPTION VALUE="Quetta">Quetta
</SELECT>
```

```
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Select.html".
3. Double click the file. The web page will appear as follows:



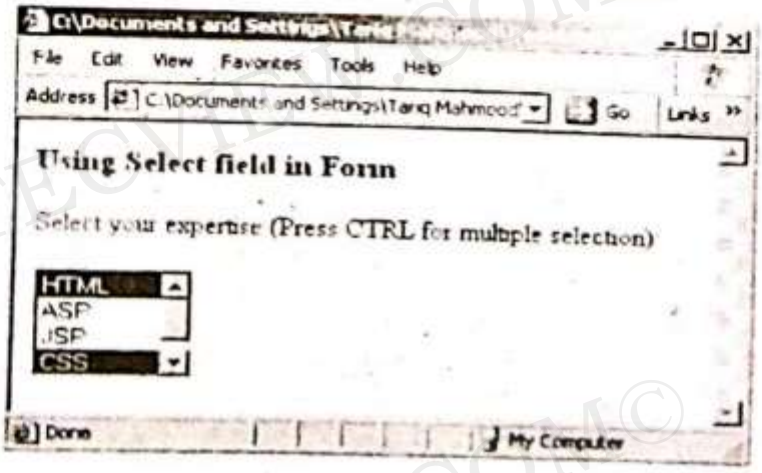
Practice 5.11

Write HTML code that displays names of different expertise in a Select field. The user should be able to select multiple item from the list.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Select field in Form</H3>
<FORM NAME="form1">
Select your expertise (Press CTRL for multiple selection):<P>
<SELECT NAME="Expertise" MULTIPLE>
<OPTION VALUE="HTML" SELECTED>HTML
<OPTION VALUE="ASP">ASP
<OPTION VALUE="JSP">JSP
<OPTION VALUE="CSS">CSS
<OPTION VALUE="JavaScript">JavaScript
</SELECT>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Select2.html".
3. Double click the file. The web page will appear as follows:



5.3.6 Text Area Field

Text area is similar to text box but it may consist of multiple rows. It is used to get lengthy input from the user like comments and suggestions etc. <TEXTAREA> tag is used to create text area in a form.

Attributes

Some important attributes of text area are as follows:

Attribute	Description
NAME	It specifies the name of the text area
VALUE	It specifies the contents of the text area
DEFAULTVALUE	It specifies the default value of the text area
ROWS	It specifies the number of rows in the text area
COLS	It specifies the number of columns in the text area

Example

```
<TEXTAREA COLS=25 ROWS=20>
The text for text area is written here
</TEXTAREA>
```

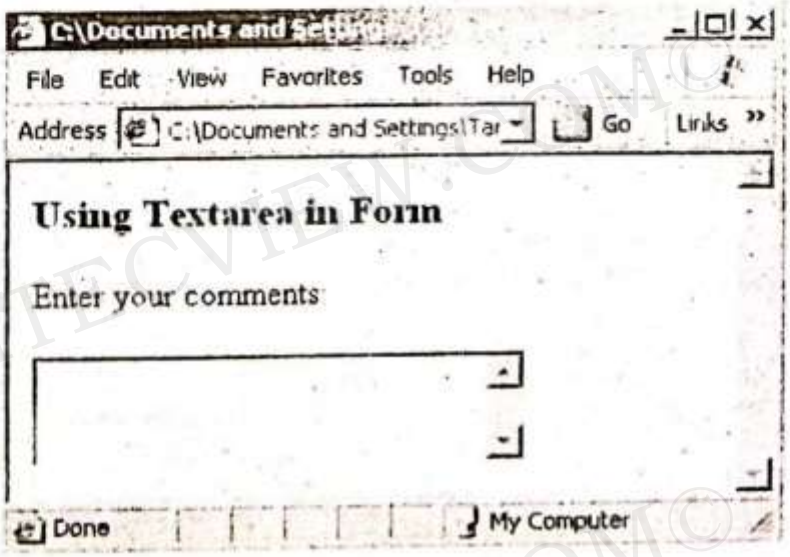
Practice 5.12

Write HTML code that displays a text area of 25 columns and 3 rows for comments.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Textarea in Form</H3>
<FORM NAME ="form1">
Enter your comments <P>
<TEXTAREA COLS=25 ROWS=3>
</TEXTAREA>
</BODY>
</HTML>
```

2. Save the file as "Textarea.html".
3. Double click the file. The web page will appear as follows:



5.3.7 Button

Button is used in forms to process the form data. <BUTTON> tag is used to create a button in the form.

Attributes

Some important attributes of this tag are as follows:

Attribute	Description
NAME	It specifies the name of the button
VALUE	It specifies the caption of the button

Example

```
<INPUT TYPE="button" VALUE="Hello Buttons">
```

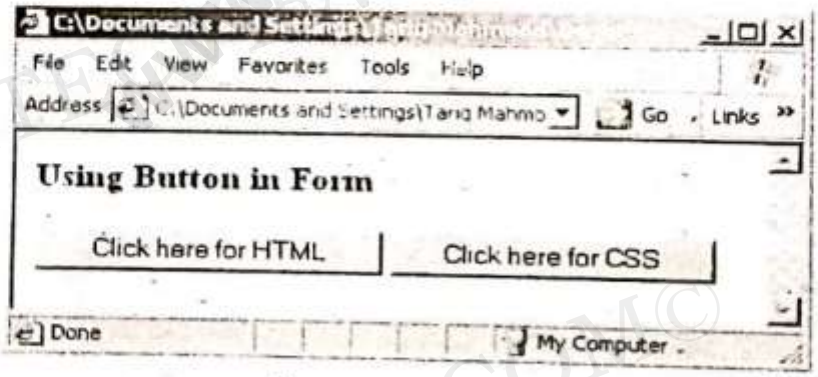
Practice 5.13

Write HTML code that displays two buttons.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using Button in Form</H3>
<FORM NAME="form1">
<INPUT TYPE="button" VALUE="Click here for HTML">
<INPUT TYPE="button" VALUE="Click here for CSS">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "button.html".
3. Double click the file. The web page will appear as follows:



5.3.8 File Upload Field

File upload field is used to get files from the user in a form. The visitors can select a file from the disk using this field. <FILE> tag is used to create file upload field.

Attributes

Some important attributes of this tag are as follows:

Attribute	Description
NAME	It specifies the name of the button
VALUE	It specifies the file name and path of the selected file

Example

```
<INPUT TYPE="FILE" NAME="F">
```

Practice 5.14

Write HTML code that displays File Upload button

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Using File Upload Button in Form</H3>
<FORM NAME="form1">
<INPUT TYPE="file">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "File.html".
3. Double click the file. The web page will appear as follows:

4. Click **Browse** button. The **Choose File** dialog box will open for selecting a file.

5.3.9 Submit Button

Submit button is a predefined button. It is used to process the form data. When the user clicks this button, it processes the form data according to the value specified in **ACTION** attribute of the form. The attributes of this tag are similar to the attributes of **<BUTTON>** tag.

Example

```
<INPUT TYPE="submit">
```

5.3.10 Reset Button

Reset button is a predefined button. It is used to reset the form. When the user clicks this button, it clears all data entered by the user in different form elements. The attributes of this tag are similar to the attributes of **<BUTTON>** tag.

Example

```
<INPUT TYPE="reset">
```

Practice 5.15

Write HTML code to design a feeback form that accepts detailed data from the user. The data includes first name, second name, gender, area, interests and photo.

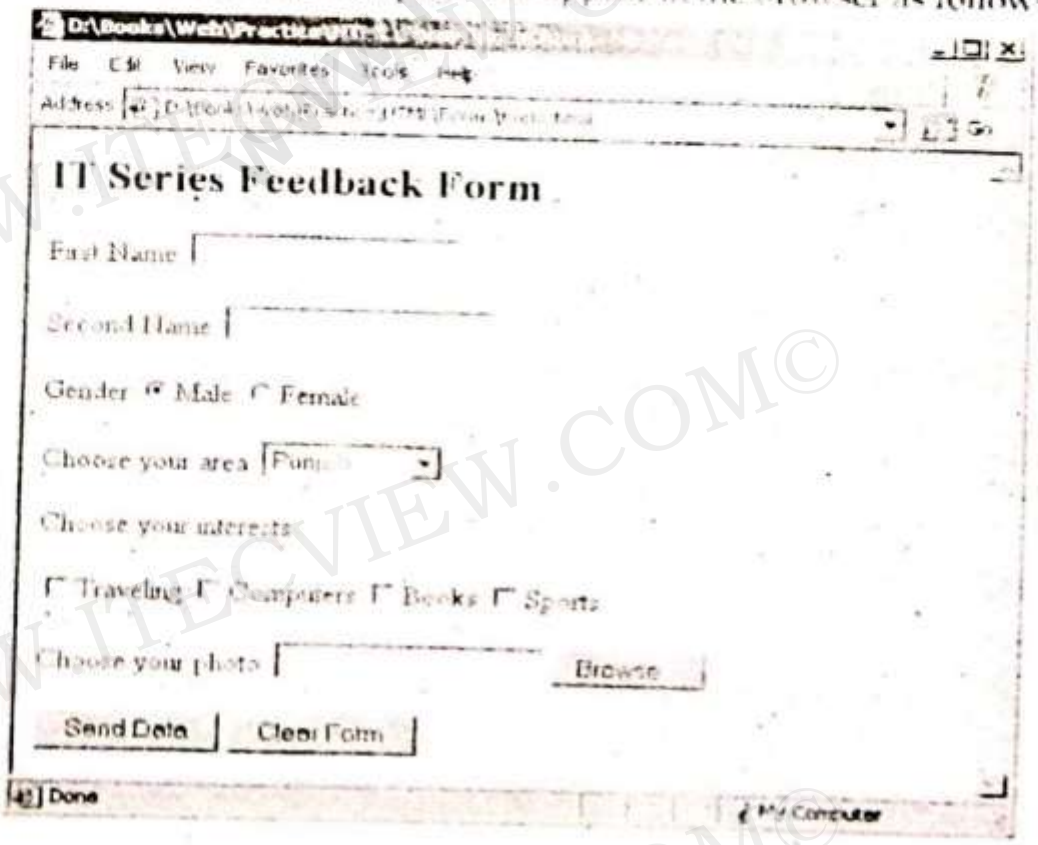
1. Open Notepad and type the following code:

```
<HTML>
```

WWW.ITECVIEW.COM©

```
<BODY>
<H2>IT Series Feedback Form</H2>
<FORM NAME="form1" METHOD="get" ACTION="mailto:itseries@msn.com">
First Name: <INPUT TYPE="text" NAME="fname"><P>
Second Name: <INPUT TYPE="text" NAME="sname"><P>
Gender:<INPUT TYPE="radio" NAME="gender" VALUE="male" CHECKED>Male
<INPUT TYPE="radio" NAME="gender" VALUE="female">Female <P>
Choose your area:
<SELECT NAME="area">
<OPTION VALUE="Punjab">Punjab
<OPTION VALUE="Sindh">Sindh
<OPTION VALUE="NWFP">NWFP
<OPTION VALUE="Balochistan">Balochistan
<OPTION VALUE="AJK">AJK
</SELECT>
<P>Choose your interests: <P>
<INPUT TYPE="checkbox" NAME="traveling" VALUE="travel">Traveling
<INPUT TYPE="checkbox" NAME="computers" VALUE="computers">Computers
<INPUT TYPE="checkbox" NAME="books" VALUE="books">Books
<INPUT TYPE="checkbox" NAME="sports" VALUE="sports">Sports
<P>Choose your photo:
<INPUT TYPE="File" NAME="Photo"><P>
<INPUT TYPE="submit" VALUE="Send Data" NAME="submit">
<INPUT TYPE="reset" VALUE="Clear Form" NAME="reset">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form.html"
3. Double click the file. The web page will appear in the browser as follows:

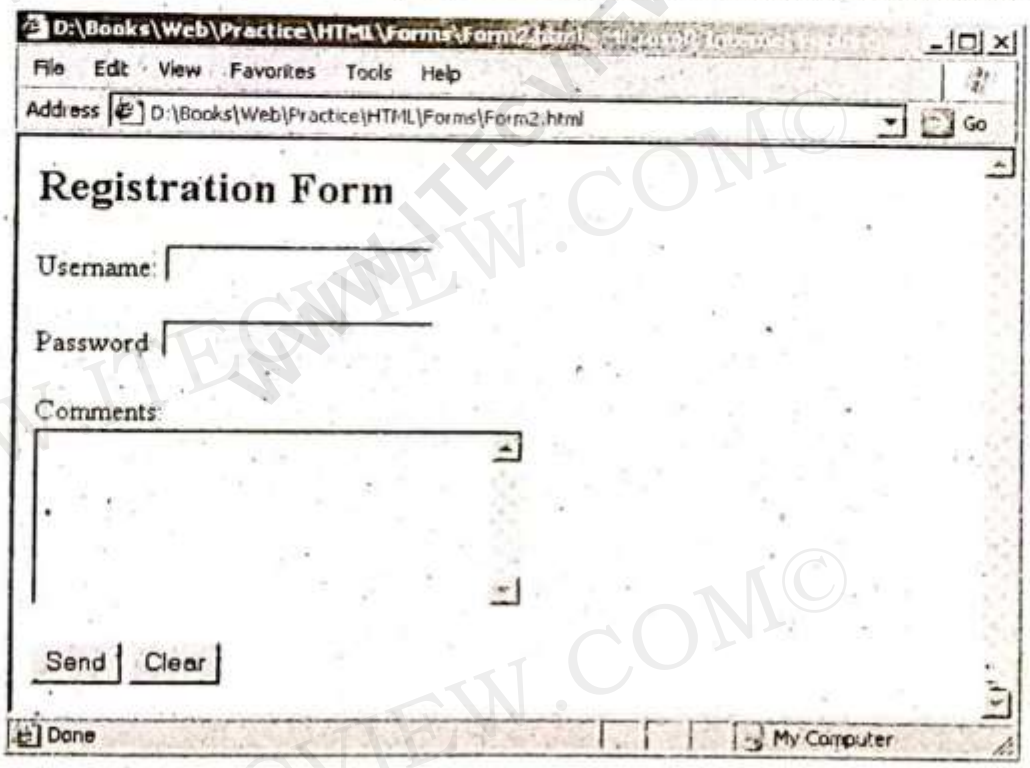


Practice 5.16

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<HTML>
<BODY>
<H2>Registration Form</H2>
<FORM NAME="form1" METHOD="get" ACTION="mailto:itseries@msn.com">
Username: <INPUT TYPE="text" NAME="name"><P>
Password: <INPUT TYPE="password" NAME="pw" MAXLENGTH=8><P>
Comments:<BR>
<TEXTAREA COLS=30 ROWS=10>
</TEXTAREA>
<P><INPUT TYPE="submit" VALUE="Send" NAME="submit">
<INPUT TYPE="reset" VALUE="Clear" NAME="reset">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form2.html".
3. Double click the file. The web page will appear in the browser as follows:



Practice 5.17

Write HTML code to create a complete form for a pizza shop. It should input name, phone, payment method, pizza size, ingredients, and soft drinks.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
    <TITLE>Pizza Form</TITLE>
```

```
</HEAD>
<BODY>
<H1 ALIGN = "center">Tasty Pizza</H1>
<FORM METHOD = "post" ACTION = "mailto:itseries@msn.com">
<DT> Enter your name:
<DD> <INPUT TYPE = "text" NAME = "name" SIZE = "30" MAXLENGTH = "50">
<DT> Enter your phone:
<DD> <INPUT TYPE = "text" NAME = "num" SIZE = "8" MAXLENGTH = "15">
<DT> Enter payment method:
<DD> <INPUT TYPE = "radio" NAME = "pay" VALUE = "1"> Cash
<DD> <INPUT TYPE = "radio" NAME = "pay" VALUE = "2"> Check
<DT> Enter pizza size:
<DD> <INPUT TYPE = "radio" NAME = "size" VALUE = "1"> Small
<DD> <INPUT TYPE = "radio" NAME = "size" VALUE = "2"> Medium
<DD> <INPUT TYPE = "radio" NAME = "size" VALUE = "3"> Grand
</DL>
<B>Select ingredients:</B>
<UL>
<LI><INPUT TYPE = "checkbox" NAME = "g1" VALUE = "1"> Extra Cheese
<LI><INPUT TYPE = "checkbox" NAME = "g2" VALUE = "2"> Mushrooms
<LI><INPUT TYPE = "checkbox" NAME = "g3" VALUE = "3"> Peppers
<LI><INPUT TYPE = "checkbox" NAME = "g4" VALUE = "4"> Sausage
<LI><INPUT TYPE = "checkbox" NAME = "g5" VALUE = "5"> Pepperoni
<LI><INPUT TYPE = "checkbox" NAME = "g6" VALUE = "6"> Olives
</UL>
Select soft drink: <BR>
<SELECT NAME = "drink">
<OPTION SELECTED> Coke
<OPTION> Pepsi
<OPTION> 7up
<OPTION> Dew
</SELECT>
<BR><BR>
<INPUT TYPE = "submit" VALUE = "SUBMIT">
<INPUT TYPE = "reset" VALUE = "CLEAR">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form3.html".
3. Double click the file. The web page will appear in the browser.

Practice 5.18

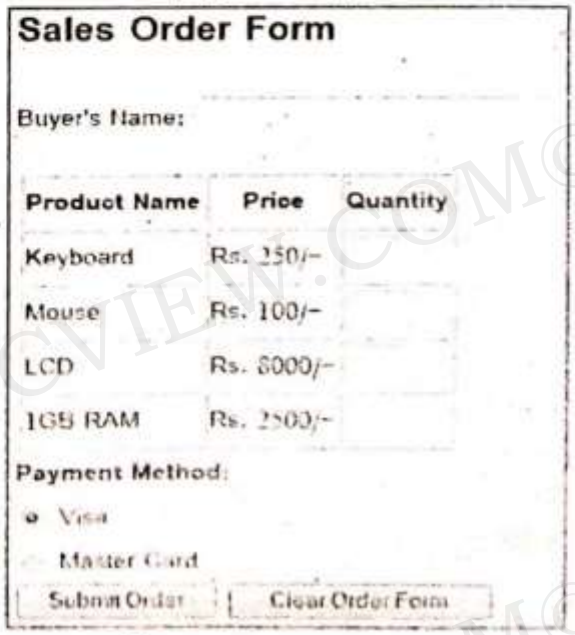
Write HTML code to create a sales order form for computer accessories shop.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE> Computer Sales </TITLE>
</HEAD>
<BODY>
<H2 ALIGN="left"> Sales Order Form </H2>
<FORM METHOD = "post" ACTION = "mailto:itseries@msn.com">
```

```
<P>Buyer's Name: <INPUT TYPE = "text" NAME = "name" SIZE = "30"></P>
<TABLE BORDER = "border">
  <TR> <TH> Product Name </TH> <TH> Price </TH> <TH> Quantity </TH>   </TR>
  <TR>
    <TD> Keyboard </TD>
    <TD> Rs. 250/-.</TD>
    <TD> <INPUT TYPE = "text" NAME = "keyboard" SIZE ="2"> </TD>
  </TR>
  <TR>
    <TD> Mouse </TD>
    <TD> Rs. 100/- </TD>
    <TD> <INPUT TYPE = "text" NAME = "mouse" SIZE = "2"> </TD>
  </TR>
  <TR>
    <TD> LCD </TD>
    <TD> Rs. 8000/- </td>
    <TD> <INPUT TYPE = "text" NAME = "lcd" SIZE = "2"> </TD>
  </TR>
  <TR>
    <TD> 1GB RAM </TD>
    <TD> Rs. 2500/- </TD>
    <TD> <INPUT TYPE = "text" NAME = "ram" SIZE = "2"> </TD>
  </TR>
</TABLE>
<B> Payment Method: </B> <BR>
<INPUT TYPE = "radio"  NAME = "payment"  VALUE = "visa" CHECKED> Visa <BR>
<INPUT TYPE = "radio"  NAME = "payment"  VALUE = "mc"> Master Card <BR>
<INPUT TYPE = "submit"  VALUE = "Submit Order">
<INPUT TYPE = "reset"  VALUE = "Clear Order Form">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form4.html".
3. Double click the file. The web page will appear in the browser as follows:



Exercise Questions

1. Write HTML code that displays the following form:

A Guest Book Form

First Name: Last Name:

Email: Year of Graduation:

College: Major:

Comments

Would you like to share your comments? Yes No

Send Entry Clear Form

2. Write HTML code that displays the following form in table:

Please fill this survey form

First Name:		MI:		Last Name:	
City:		State:		Zip Code:	
Choose a sport:					
Basketball: ☑ Football: ☐ Hockey: ☐					
Choose a year:					
Freshman: Sophomore: Junior: Senior:					
Choose your favorite fast food:					
Chips Pizza Nachos					
Send It	Clear It				

WWW.ITECVIEW.COM©

3. Write HTML code that displays the following form in table:

What type of bike do you want to purchase?

☐ Street bike ☐ Mountain bike ☐ dirt bike

What accessories do you need?

☐ Gloves ☑ Helmet ☐ Camel water bag

Where do you go biking?

Parks

Shipping information

First Name Last Name Address

City State Zip code

Special delivery instructions

Payment information

Credit card number Expires 12/10

Send Order Start Over

WWW.ITECVIEW.COM©

Short Questions

Q.1. Define Markup Language

A structured language that lets you identify common sections of a document such as headings, paragraphs, and lists

Q.2. What is HTML?

HTML stands for Hyper Text Markup Language. It is a formatting language used to develop web pages. It uses tags or markups to specify how the contents of web page will be displayed. HTML code can be written in any text editor like Notepad.

Q.3. HTML is a subset of which markup language?

HTML is a subset of Standard Generalized Markup Language (SGML)

Q.4. Write three characteristics of HTML that make it ideal for the World Wide Web.

HTML is an easy and open language. It is a non-proprietary language. It can be used on different platforms.

Q.5. What is XML?

XML stands for Extensible Markup Language. It allows developers to create customized tags. It makes XML more flexible than HTML. XML tags represent the logical structure of data and can be interpreted and used in different ways by different applications. XML inherits the features of SGML and combines the features of HTML with it.

Q.6. What feature distinguishes XML from HTML?

HTML has a fixed set of elements. XML lets you create your own elements. XML describes data, such as student name, marks and address. HTML defines the tags that describe the display of data such as with bulleted list or table.

Q.7. Explain how different browsers affect the display of a Web page.

The logic for interpreting HTML tags is different in every browser. It results in different interpretations of appearance of HTML files.

Q.8. Differentiate between XHTML and HTML.

XHTML documents contain DOCTYPE declarations at the top of document. DOCTYPE was optional in HTML. XHTML documents must be well formed. However, HTML does not strictly require well formed documents. XHTML elements and attributes are case sensitive but HTML elements and attributes are not. XHTML requires closing tag for every element. HTML does not require closing tags.

Q.9. What is XTHML?

XHTML stands for extensible Hypertext Markup Language. It combines the formatting strengths of HTML 4.0 and the data structure and extensibility strengths of XML. It uses the tags and attributes of HTML along with the syntax of XML. The first version of XHTML 1.0 was released in 2000. The World Wide Web Consortium (W3C) sets the standards for XHTML and other internet languages.

Q.10. Define Hypermedia and Meta language.

The linking of different types of media such as sounds, video clips, images, etc. on the World Wide Web is called hypermedia. A language used to describe other languages is called meta language.

Q.11. What is the work of World Wide Web Consortium (W3C)?

W3C organization was founded in 1994 at Massachusetts Institute of Technology. It develops the standards for Internet languages such as HTML, XML and CSS.

Q.12. Define tag and attribute.

Tag is the basic command of HTML. HTML tags are surrounded by the two characters < and >. HTML tags normally come in pairs like <head> and </head>. An attribute is additional information about element that is included inside start tag. Some attributes are compulsory and some are optional.

Q.13. List some benefits of HTML.

- **Simplicity:** HTML is a simple but powerful language.

- **Platform Independent:** HTML is platform independent. It works on any platform.
- **Easy Navigation:** HTML provides easy navigation between different documents.
- **Attractive and Easier to Read:** HTML allows the use of graphics and animation.

Q.14. List some limitations of HTML.

Some limitations of HTML are as follows:

- **No Programming Capabilities:** HTML provides no programming capabilities. It only provides formatted text, pictures and sounds etc.
- **Static:** HTML can only be used to create static web pages.

Q.15. List basic requirements to create web pages using HTML.

The basic requirements to create web pages using HTML are as follows:

- **Text Editor:** It is used to write HTML commands. Any simple text editor like Notepad or WordPad can be used for this purpose.
- **Web Browser:** It is used to test the web pages. Internet Explorer and Netscape Navigator are two popular web browsers.

Q.16. Which are two main sections of an HTML page?

The two main sections of an HTML page are as follows:

- **Head Section:** <HEAD> and </HEAD> tags identify the heading or title of document. The tags <TITLE> and </TITLE> can appear only in this section.
- **Body Section:** The body of an HTML document contains the text that is displayed on a web page. The body section is enclosed within <BODY> and </BODY> tags.

Q.17. List some important attributes of BODY tag.

Some important attributes of <BODY> tag are as follows:

- **BGCOLOR:** Specifies the background color of the web page.
- **BACKGROUND:** Specifies the background picture of the web page.
- **TEXT:** Specifies the color of the text displayed on the web page.

Q.18. What will be the effect of using the following tags?

a. <b>

bold

b. <h2>

-heading style 2

c. <i>

-italics

d. <p>

-paragraph

e. <font size="+3">

It increases the font size by 3 sizes. This is 3 sizes larger than the "normal" size.

f. <font size="2">

It is the absolute size of 2. The "normal" is size 3, therefore this will be 1 size smaller than normal

g. <ol>

-ordered list (numeric)

Q.19. What is the difference between <BR> tag and <P> tag?

 is a line break and subsequent text will move to the line below. <P> is the paragraph tag and text will move two lines below. There will be a blank line between text before the <P> tag and text after <P> tag

Q.20. Name two elements designed for use with style sheets.

<DIV> and <SPAN>

Q.21. Which types of lists can be created in HTML?

Different types of lists are Ordered Lists, Unordered Lists and Definition Lists.

Q.22. What is an ordered list?

An ordered list is a list of items in which each item is marked with a number. Ordered list is also known as numbered list.

Q.23. What is the use of OL tag?

<OL> tag is used to create an ordered list. It stands for **ordered list**. **<LI>** tag is used with each item in the list. It stands for **list item**. It has no closing tag. **</OL>** tag is used to close an ordered list.

Q.24. What is an unordered list?

An unordered list is a list of item in which each item is marked with a symbol. Unordered list is also known as unnumered list.

Q.25. What is the use of UL tag?

<UL> tag is used to create an unordered list. It stands for **unordered list**. **<LI>** tag is used with each item in the list. **</UL>** tag is used to close an unordered list.

Q.26. What is a definition list?

A definition list is not a list of items. It consists of terms and description of terms.

Q.27. What is the use of DL tag?

<DL> tag is used to create a definition list. It stands for **definition list**. **<DT>** tag is used with each term in the definition list.

Q.28. Which types of images are supported by HTML?

Different image formats supported by HTML are as follows:

- **Graphic Interchange Format (GIF):** Graphic Interchange Format uses a maximum of 256 colors. GIF format is best for displaying images like logos, icons and buttons.
- **Joint Photographic Expert Group (JPG, JPEG):** It is a good format for images. It can contain millions of colors. It does not provide transparency or interlacing.
- **Portable Network Graphics (PNG):** It is good for combinations of text and graphics within one image. It supports transparency and interlacing but not animation. PNG is only supported by Internet Explorer.
- **BMP:** BMP stands for **bitmap picture**. It is most commonly used in Windows environment. It provides high quality images. BMP images are not compressed and produce large file size. That is why it is not used on the Internet.
- **TIFF:** TIFF stands for **Tagged Image File Format**. It is widely supported format to store bitmap images on computer. It is commonly used in desktop publishing, 3D applications and medical imaging applications.
- **PCX:** It was originally developed by ZSOFT for PC Paintbrush program. PCX is a graphics file format for graphics programs running on PCs.

Q.29. What do know about interlacing?

Interlacing is a capability that enables the browser to display the image as it loads. The image gradually gets clearer until it is loaded completely. Interlaced images are slightly larger in size than non-interlaced images.

Q.30. What are the three image file formats that can be used on the Web?

The three image file formats that can be used on the Web are GIF, JPG and PNG.

Q.31. Which file formats support 24-bit color?

JPG and PNG file formats support 24-bit color.

Q.32. What is lossless file compression?

Lossless file compression is the creation of smaller file size without discarding color information

Q.33. Which file formats support transparency?

GIF and PNG file formats support transparency.

Q.34. Explain "lossy" image compression.

Lossy image compression is the creation a smaller file size by discarding color information

Q.35. Why is PNG not a more popular image format?

PNG is not a more popular image format due to lack of browser support.

Q.36. Which property specifies the location of the image to be used?

The SRC property specifies the location of the image to be used in the web page.

Q.37. What are the most common uses of tables and why?

Tables are used to organize a Web page. The user can break the table into very small cells and combine cells when necessary. It allows flexibility in a Web page's appearance. In addition, tables are very useful to display data in a 2-D format such as sales people/sales or people/occupations, etc.

Q.38. Why did tables become so popular?

Initially developers had very few ways to align text on a Web page. Tables are popularly used to align forms etc. In addition, tables can be used to control images better.

Q.39. What are the three basic table elements?

The three basic table elements are <TABLE> <TR> <TD>.

Q.40. Define Table, TR ,TD AND TH tag.

<TABLE> tag is used to define a table. <TR> tag is used to define a row in a table and <TD> tag is used to define a data cell. <TH> tag is used to define the header row of the table.

Q.41. What is difference between CELLPADING and CELLSPACING attribute of table.

CELLPADDING is used to insert spacing within the table cells on all four sides. Its value is given in pixels. CELLSPACING is used to insert spacing between the table cells on all four sides. Its value is also given in pixels.

Q.42. What is the use of COLSPAN and ROWSPAN attribute of table.

COLSPAN attribute allows cells to span multiple columns of a table. The column cells always span to the right. ROWSPAN attribute allows cells to span multiple rows of a table. The rows always span down.

Q.43. In reference to <IMG> tag, describe the effect of alignment attributes, left and right.

Left alignment will position the image on the left margin and text will wrap to the right of the image, starting at the top of image and continuing to wrap. Right alignment will position the image on right margin and the text will wrap to left of image, starting at the top of image and continuing to wrap.

Q.44. What is the attribute and value for removing the hypertext border for an image?

BORDER=0 in the <IMG> element is used for removing the hypertext border for an image.

Q.45. What is hyperlink? What are important properties of a hyperlink?

A link that connects one web page with another is called a hyperlink. The links are used to move from one location to another on the Internet. Some important properties of hyperlinks are as follows:

- They are normally underlined.
- Color of hyperlinks is normally different from normal text.
- When mouse pointer moves to a hyper link, pointer changes to a pointing hand.

Q.46. List different types of hyperlinks.

- **Page-Jumps:** These hyperlinks are linked with other locations within the same page. These are used to easily navigate the site without too much scrolling.
- **Internal Links:** These hyperlinks are linked with other web pages within same website.
- **External Links:** These hyperlinks are linked with other websites or other web pages of other websites.
- **Email Links:** These hyper links are linked with emails. They start email application when the user clicks on them.

Q.47. What is the difference between a local hypertext link and a remote hypertext link?

Local link points to a file on the same server. Remote link points to a file on a different server

Q.48. What is difference between a visited link and an active link?

A visited link is a link in the history list of the browser. A link is active when the user is hovering over the link and depressing the mouse button

Q.49. What is the benefit of using hyperlinks? Discuss a possible result of using too many hyperlinks within a document.

Hyperlinks are designed to provide users with an easy method of accessing additional information. Developers must be careful not to add too many links to a page since it may make the page confusing to users and result in fewer people visiting a Web page.

Q.50. Write some of the benefits of providing text links in addition to graphical links?

Answers will vary: With a text link you have the ability to include more information about what is in the related link. This can help the user avoid unnecessary links and improve their visit to a site.

Q.51. What is the purpose of TARGET attribute of a hyperlink?

The TARGET attribute of a hyperlink is used to specify the window to open the linked page.

Q.52. Define <A> tag and HREF.

All hyperlinks are created with the anchor tag such as <A>. Its closing tag is </A>. The HREF attribute of <A> is used to specify the link destination such as <A HREF="destination.html">

Q.53. What is the difference between absolute path and relative path.

A path that points to computer's root directory is called absolute path. A path that tells browser where a file is located relative to the document viewed by browser currently is called relative path.

Q.54. What is a frame?

A frame is a container window that can display a web page. Frame is used to break the web page into different sections. These sections work independently but appear simultaneously in the browser.

Q.55. What are some benefits of using frames?

The frames allow the user to keep one part of page static and change other parts. The frames can be used to display multiple pages in the same browser. Frames can be used to reduce the load on server because all pages are not reloaded when a new page is visited.

Q.56. What are some of the problems associated with using frames?

Search engines do not index web pages with frames. It is difficult to print contents of all frames as compared to a normal web page. Most browsers do not allow to bookmark an individual frame page.

Q.57. Explain the difference between frameset and frame and how they are used together.

Frameset is a container or placeholder for multiple frames in same browser window. A frameset tells the browser how to lay out a page. The frames are individual HTML files that are used for each section of the frames. Framesets are very important for the use of frames.

Q.58. What is the default setting for scrollbars in a frame?

The default setting for scrollbars in a frame is AUTO.

Q.59. What attribute prevents the user from adjusting the size of the frames?

The NORESIZE attribute prevents the user from adjusting the size of the frames.

Q.60. What element is the placeholder for the structure of a frames document?

<FRAMESET> element is the placeholder for the structure of a frames document.

Q.61. What are the two attributes you use to structure the look of a frameset?

COLS and ROWS attributes are used to structure the look of a frameset.

Q.62. How does the order of <FRAME> element within <FRAMESET> affect the display of frameset?

The frames fill top to bottom in row framesets. The frames file left to right in columns framesets.

Q.63. What is the purpose of <NOFRAMES> element?

The purpose of <NOFRAMES> element is to provide alternate content for browsers that cannot display frames.

Q.64. What is the benefit of NORESIZE attribute?

It allows HTML author to restrict the user's ability to resize frames.

Q.65. What are the two tasks you must perform to add targeting to a frameset?

The frames should be assigned names using NAME attribute in FRAME element. Target links to display their content in the named frame.

Q.66. List the four special target names.

The four special target names are _self, _blank, _parent and _top.

Q.67. Are target names case sensitive?

Yes. The target names are case sensitive.

Q.68. Which special target name is the default behavior for links in a frameset?

_self is the default behavior for links in a frameset.

Q.69. How can you control the display of content within a frameset?

The tables can be used to keep the content aligned within the frameset.

Q.70. What additional browser elements can affect the display of frameset?

Scrollbars, frame margins and frame borders can affect the display of frameset.

Q.71. What are forms?

Forms are used to create interactive websites. The visitors can send feedback, emails, comments and suggestions by using these forms. Form is an easier way for the user to communicate with the webmaster or the owner of the website.

Q.72. List some uses of forms.

- **Educational Sites:** Forms are used to get names, addresses and emails etc. by educational sites.
- **Online Orders:** Many websites use forms to get online orders from the visitors. The user can send the order by entering information in the forms.
- **Feedback:** Many websites use forms to get feedback from the users. The feedback of the visitors is an important source for improving the services or products of a company.

Q.73. List some important elements of form.

Some important form elements are, Textbox, List Box, Radio Button, Checkbox, Button, Text area and Password field.

Q.74. How does METHOD attribute of form work?

The METHOD attribute specifies how the form data is sent to the server. The possible values are "get" and "post". The value "get" sends the data as part of URL. It is the default value. The value "post" sends data encoded in HTTP data stream.

Q.75. Why action="post" is better than action="get" for submitting sensitive information?

When you use GET, all information is included in the URL, which can be seen by other people and is remembered by the browser. With the POST method, the computer keeps no record of the information and it is not displayed in the URL. Therefore, it is more secure.

Q.76. How can you prevent slow download times when using images on Web pages?

There are several ways to download images faster on a Web page. One way to reduce download time is to set the height ="" and width="" attributes. One way that might seem to reduce download time is to use lowsrc attribute. The smaller version of the image can make the image appear to download faster, but it usually does not.

Q.77. What are some of the benefits of using forms?

Forms are a mechanism to gather information from a user. The information can be something that the user inputs into a form.

Q.78. Why would you want to collect form data in a database?

It is useful to collect data in a database wherever you would like to store or keep or study the data entered by users. Once the data is in database, developers can generate valuable reports to help the users modify a Web site.

Q.79. Discuss different form layouts and how tables might be useful in creating forms.

Forms are useful to gather data from users. Tables can be useful to create structure for Website. By combining them, developer has added additional formatting to the form layout and data collection.

Q.80. When would you use a text box rather than a text area?

Text box is used when the input is relatively short such as name, address or city etc. Text area is used when the input is relatively lengthy such as feedback, comments or suggestions etc.

Multiple Choice

1. **What does HTML stand for?**
   a. Home Tool Markup Language
   b. Hyper Text Markup Language
   c. Hyperlinks and Text Markup Language
   d. None
2. **HTML syntax is:**
   a. Case-sensitive
   b. Space-sensitive
   c. Both a and b
   d. None
3. **An example of a Web editor is:**
   a. MS Excel
   b. MS Access
   c. MS FrontPage
   d. MS Internet Explorer
4. **HTML is a _____ language.**
   a. Programming
   b. Processing
   c. Markup
   d. Visual
5. **A Web markup language is used to:**
   a. Execute Java applets embedded in a Web page.
   b. Describe how a Web browser should display text
   c. Control connections to a database stored on a Web server.
   d. Create simple programs to add animation or other dynamic content to a Webpage.
6. **Web pages developed with HTML:**
   a. Can be displayed in a variety of browsers
   b. Can be displayed only in a Microsoft browser, such as Internet Explorer
   c. Can be viewed only in an e-mail message
   d. Cannot contain hyperlinks
7. **The main part of the Notepad window, where text is displayed as you type:**
   a The title bar
   b. The menu bar
   c. The text area
   d. The scroll bar
8. **The title of a Web page appears:**
   a. Between the < title> and < /title> tags in the HTML code
   b. On the favorites list if the Web page is saved as a favorite
   c. On the title bar of the Web browser when the Web page is active
   d. All of the above
9. **HTML tags typically appear:**
   a. At the beginning of Web page
   b. In pairs
   c. At the end of a Web page
   d. None
10. **HTML tags are distinguished from other text in a Web page by using:**
    a. < >
    b. { }
    c. [ ]
    d. ( )
11. **A Web page contains the information that is displayed in the browser window is:**
    a. TITLE
    b. BODY
    c. BACKGROUND
    d. HEADING
12. **Choose the correct HTML tag for the largest heading:**
    a. <heading>
    b.<head>
    c. <h6>
    d. <h1>
13. **A Web page title should:**
    a. Be concise yet descriptive
    b. Identify the subject or purpose of the page
    c. Both a and b
    d: Neither a nor b
14. **What is the correct HTML tag for inserting a line break?**
    a. <BREAK>
    b. <LB>
    c.
    d. <OL>
15. **What is the correct HTML for adding a background color?**
    a. <body color="yellow">
    b. <body bgcolor="yellow">
    c. <Background>yellow</background>
    d. None
16. **Choose the correct HTML tag to make a text bold:**
    a. <B>
    b. <BOLD>
    c. <BLD>
    d. <BB>

WWW.THEUNIREVIEW.COM©

17. The HTML tags < sub> and < /sub> cause text to be displayed as:
a. Substitute text b. Superscript text c. Subscript text d. All

18. The HTML tags < sup> and < /sup> cause text to be displayed as:
a. Substitute text b. Superscript text c. Subscript text d. All

19. Which of the following is not an attribute of the <HR> tag?
a. NOSHADE b. SHADE c. ALIGN d. WIDTH

20. Which of the following is not a valid list tag in HTML?
a. <DL> b. <ML> c. <UL> d. <OL>

21. Which of the following is not a valid attribute for either <OL> tag or <LI> tag?
a. BEGIN b. START c. TYPE d. VALUE

22. If an image that is used has a link as a border, the border:
a. is always black b. is invisible
c. Displays default link colors d. Changes color when mouse moves over it

23. The attribute "href", used with the < a> tag:
a. Stands for "hypertext reference" b. Uses the name of a linked page or file
c. Uses the URL of a linked page or file d. All

24. The default value for ALIGN attribute is:
a. Left b. Center c. Right d. Not specified

25. What is the default display for a text link?
a. Black and bold b. Blue and underlined c. Purple d. Green

26. Each of the following types of files my be used on Web pages except:
a. GIF b. JPEG c. PNG d. PIC

27. An image should be used:
a. To convey content b. To visually organize a page
c. To provide a hyperlink d. All

28. Which attribute of <IMG> tag defines the horizontal space that separates the image from the text?
a. HSPACE b. SRC c. VSPACE d. WIDTH

29. Which attribute of <IMG> tag can have values bottom, middle, top, left and right?
a. Align b. Alt c. border d. height

30. Which attribute of <IMG> tag defines the URL of the image to be loaded?
a. hspace b. src c. vspace d. width

31. Text contained between the paired tags <u></u> will be:
a. Bold b. Underlined c. Italicized d. Upper case

32. The Web pages have file extensions:
a. .htm or .html b. .htm or .doc c. .htm or .pdf d. .http or .html

33. Choose the correct HTML tag to make a text italic:
a. <ii> b. <italics> c. <i> d. <I>

34. What is the correct HTML for making a hyperlink?
a. <a>http://www.itseries.com.pk</a>
b. <a name="http://www.itseries.com.pk">itseries.com.pk</a>
c. <a url="http://www.itseries.com.pk">itseries.com.pk</a>
d. <a href="http://www.itseries.com.pk">IT Series</a>

35. How can you make an e-mail link?
a. <mail>xxx@yyy</mail> b. <a href="mailto:xxx@yyy">
c. <a href="xxx@yyy"> d. <mail href="xxx@yyy">

36. Which of the following is not information that is typically contained in a header?
a. Title of a Web page
b. Last update information
c. Table of contents
d. Logo

37. The intersection of a row and a column is.
a. Row b. Column c. Cell d. Border

38. The attribute of the < table> tag defines the number of columns in the table is:
a. cellspacing b. cellpadding c. cols d. width

39. In an HTML table, you can set attributes for:
a. An entire row of information
b. A single cell
c. One or more cells within a row
d. All of the above

40. Which of the following HTML tags indicate the start and end of a table?
a. < table> < /table> b. < tr> < /tr> c. < th> < /th> d. < td> < /td>

41. All other table tags are inserted between the:
a. < table> < /table> b. < tr> < /tr> c. < th> < /th> d. < td> < /td>

42. To position a caption below a table, use the _____ attribute of the < caption> tag.
a. align= "top" b. align= "above" c. align= "bottom" d. align= "below"

43. The attribute of the < table> tag defines the width of the table border in pixels is.
a. Align b. bgcolor c. borde d. cellspacing

44. The line that encloses the perimeter of the HTML table.
a. Row b. column c. cell d. border

45. The attribute of the < tr> tag sets the vertical alignment of the row is.
a. vrow b. valign c. halign d. align

46. A table should be used:
a. If it will help organize information in such a way that it is easier to read
b. If a Web page needs to display a structured, organized list of information
c. If a Web page includes text and images to be positioned in a very specific way
d. All of the above

47. Which of the following is not valid syntax for the <HR> tag.
a. <HR> b. <HR WIDTH="75"> c. <HR WIDTH="75\%"> d. <HR SIZE="5\%">

48. Which of the following is an attribute of the <TABLE> tag?
a. CELLSPACER b. CELLSPACING c. CELLSPACES d. CELLS

49. Which of the following is not an attribute of the <TD> tag?
a. ROWSPAN b. SPACING c. COLSPAN d. ALIGN

50. Which of the following is not an attribute of the <TABLE> tag?
a. BORDER b. WIDTH c. CELLPADDING d. COLSPAN

51. Which of the following is not a valid value for VALIGN attribute of <TD> tag?
a. Center b. Top c. TALL d. BOTTOM

52. Which of the following statements is true?
a. Tables can be nested
b. Tables cannot be used to produce newspaper layout
c. Images cannot be included inside tables
d. The <BR> attribute is not useful for making a column

53. Which of the following statements is false?
a. HTML editors are never helpful in producing tables
b. Tables are useful for helping to layout a form
c. Images can be used as hyperlinks
d. Hyperlinks can be included inside a table

54. Which code fragment has a bug?

a. <TD>Table data.</TD>
b. <TD>Table data
c. <TH Hello>There</TH>
d. <TR ALIGN = "left">Hey</TR>

55. Which of the following will not produce a table with a single row of text?

a. <table><tr>text</tr></table>
b. <table><tr><td>text</td><td>text</td></tr></table>
c. <table><tbody><tr>text</tr></tbody></table>
d. <table><tr>text</tr><tr>text</tr></table>

56. Which code fragment does not have a bug?

a. <PAR>Testing, one, two, three.</PAR>
b. <PARAGRAPH>Testing, one, two, three.</PARAGRAPH>
c. <P>Testing, one, two, three.</P>
d. <PH>Testing, one, two, three.</PH>

57. Which code fragment has a bug?

a. <TABLE BORDER = "5">
b.<TABLE BORDER = "\#FFFFFF">
c. <TABLE CELLSPACING = "9">
d. <TABLE CELLPADDING = "9">

58. Frames can be used to:

a. Allow a Web site visitor to view more than one Web page at a time
b. Create a navigation menu, replacing menu lists and menu bars
c. Display Web page elements that need to remain on screen as the screen changes
d. All

59. The Web pages or images displayed in frames

a. Cannot have links embedded in them
b. Must be composed of text only
c. Must be composed of graphics only
d. Can interact and link to other

60. The information in frame definition file defines which ____ appear in each frame.

a. Attributes
b. Coordinates
c. Web pages
d. All

61. The tag is used to define each frame; it does not need an end tag.

a. <defineframe>
b. <frame>
c. <startframe>
d. <framehere>

62. The attribute of <FRAME> tag turns frame borders on or off is:

a. < frameborder >
b. < bordercolor >
c. < marginwidth >
d. < marginheight >

63. The attribute of <FRAME> tag adjusts the margin on left and right of a frame is:

a. < frameborder >
b. < bordercolor >
c. < marginwidth >
d. < marginheight >

64. Which of the following is legal HTML syntax?

a. <FRAMESET COLS= "50%, 50%">
b. <FRAMESET ROW= "50%, 50%">
c. <FRAMESET COLUMNS= "50, 50">
d. <FRAME COLS= "50%, 50%">

65. Which piece of code could be used to divide screen into four parts of equal area?

a. <FRAMESET COLS= "50%, 50%" ROWS = "40%, 60%">
b. <FRAMESET COLS= "60%, 40%" ROWS = "40%, 60%">
c. <FRAMESET COLS= "1*, 1*, 1*, 1*">
d. <FRAMESET ROWS = "1*, 2*, 2*, 1*">

66. Which of the following is not an attribute of the FRAME tag?

a. SCROLLING
b. AUTO
c. MARGINWIDTH
d. FRAMEBORDER

67. Which of the following is valid HTML syntax?

a. <A HREF = "d.html" TARGET = "crow" D/A>
b. <A TARGET = "d.html" TARGET= "crow">D/A>
c. <A HREF = "d.html" TARGET = "crow">D/A>
d. <A TARGET = "d.html" TARGET = "crow >D/A>

68. Which of the following statements is true?
a. _blank value of TARGET attribute opens a new unnamed browser window
b. _top has the same meaning as _blank
c. _self has the same meaning as _blank
d. _parent has the same meaning as _blank

69. The scrolling attribute of the < frame > tag:
a. Locks the borders of a frame to prevent resizing
b. Defines the name of a frame that is used as a target
c. Indicates whether a scroll bar is present
d. Indicates the Web page or other file to be displayed in the frame

70. An image map is:
a. A special type of inline image
b. An image in which one or more areas are defined as hotspots
c. Used frequently for Web site navigation
d. All

71. A hotspot in an image map can link to:
a. Another part of the same Web page
b. A part of different Web page on same website
c. A Web page on a different Web site
d. All

72. Which of the following is not a valid attribute of the INPUT tag?
a. TEXT b. NAME c. SIZE d. MAXLENGTH

73. Which of the following is not a valid value for TYPE attribute of INPUT tag?
a. Checkbox b. RADIOBUTTON c. SUBMIT d. RESET

74. Which of the following is syntactically correct HTML code?
a. <INPUT TYPE= "submit" VALUES = "resetsubmit">
b. <INPUT TEXT = "submit" VALUE = "reset">
c. <INPUT TYPE = "submit" VALUE = "reset">
d. <INPUT TYPE = "reset" TEXT = "clear">

75. To provide a default-checked box, you can use which attribute of the INPUT tag?
a. Checkbox b. Checked c. Mark d. None

76. Which of the following pre-select a specific option?
a. <OPTION SELECT>
b. <OPTION MARK>
c. <OPTION SELECETED>
d. <OPTION CHECKED>

77. Which of the following is not an attribute of SELECT tag?
a. NAME b. MULTIPLE c. SIZE d. SELECTED

78. Which of the following is not an attribute of TEXTAREA tag?
a. WRAP b. MINLENGTH c. COLS d. NAME

79. Which of the following tags would not be present in every HTML document?
a. HTML b. BODY c. HEAD d. ALIGN

80. Which of the following statements is false?
a. FF in hexadecimal is 255 in decimal
b: 12 in hexadecimal is 18 in decimal
c. 1010 in binary is A in hexadecimal
d. 762 in hexadecimal is 1880 in decimal

81. Which of the following statements is false?
a. In the RGB color model, FFFFFF represents the color white
b. In the RGB color model, FFFF00 represents the color yellow
c. In the RGB color model, 000000 represents the color black
d. In the RGB color model, FF0000 represents the color rose

82. Which of the following is not a value of ALIGN attribute of HEADING tag?
a. LEFT b. MIDDILE c. RIGHT d. CENTER

83. HREF:
a. is a tag of A tag
b. is a value of A tag
c. is an attribute of A tag
d. stands for Hyperlink Repeat Electronic Filing

84. Which of the following is not an attribute of the IMG tag?
a. SRC b. ALTER c. HEIGHT d. WIDTH

85. What is the correct HTML for making a checkbox?
a. <Check>
b. <input type="check">
c. <checkbox>
d. <input type="checkbox">

86. Each list item in an unordered list is delineated by:
a. LI b. DISC c. UL d. HREF

87. The second level in nested unordered lists by default are preceded by:
a. Discs b. Circles c. Squares d. Numbers

88. Ordered lists by default are preceded by:
a. Discs b. Circles c. Squares d. Numbers

89. Form elements are mainly used to:
a. Display information b. Collects information c. Display images d. Format text

90. Which is not a preset value for the target attribute?
a. "_self" b. "_top" c. "_new" d. "_blank"

91. All of the following are file extensions for Web images EXCEPT:
a. .htm b .jpg c. .gif d. .png

92. To have an "invisible" table set the following property:
a. frame =0 b. border =0 c. cellspacing=0 d. cellpadding=0

93. The HTML code to add a background image is:
a. <body image = ...
b. <bodyimage= ...
c. <body background = ...
d. <bodybackground = ...

94. Clicking a hyperlink causes what to happen?
a. Another Web page displays
b. The same Web page re-loads itself
c. The browser opens
d. The computer connects to the Internet

95. Hyperlinks can be which of the following?
a. Text b. pictures c. neither of the above d. both a and b

96. In HTML, the tag for creating numbered lists is:
a. <ol> </ol>. b. <nl> </nl>. c. <ul> </ul>. d.<#> </#>.

97. In HTML, the tag for creating a bullet list is:
a. <ol> </ol>. b. <nl> </nl>. c. <ul> </ul>. d.<bl> </bl>.

98. All HTML documents begin with this tag:
a. <html>. b. <head>. c. <page>. d. <begin>.

99. A text editor allows the user to:
a. Write plain text. b. Write HTML c. write JavaScript. d. All

100. HTML break tags are used for separating:
a. Sections. b. Paragraphs. c. Lines. d. All.

101. Headings:
a. Are all at the same level of importance
b. Are only used to make text bold
c. are in six different levels for readability
d. None

102. To create a paragraph on a Web page, use the following tag/s:
a. <h1></h1> b.
 c. d. None

103. The font tag offers three formatting attributes. They are:
a. Align, type, and color
b. Color, face, and size
c. Color, size, and height
d. None

104. Horizontal rules:

a. Separate content on a Web page
b. Are part of standard but some attributes are deprecated
c. Display as a gray line across a Web page
d. None

105. Linking within a Web page is:

a. Convenient for navigation in long document
b. Not beneficial
c. not available in HTML
d. None

106. How is an email link different than any other link?

a. An email link uses the href attribute.
b. An email link uses the mailto: command.
c. To create an email link, use the <email></email> tags instead of the <a></a> tags.
d. None

107. Radio buttons are used for:

a. Single-option choices
b. Multiple choices
c. To collect text
d. All

108. Check boxes allow the user to:

a. Make multiple choices at a given time
b. make only once choice at a given time
c. Choose nothing from the list of options
d. All

Answers

1.	b	2.	d	3.	c	4.	c	5.	b	6.	a
7.	c	8.	d	9.	B	10.	a	11.	b	12.	d
13.	c	14.	c	15.	b	16.	a	17.	c	18.	b
19.	b	20.	b	21.	a	22.	c	23.	d	24.	a
25.	B	26.	d	27.	d	28.	a	29.	a	30.	b
31.	b	32.	a	33.	d	34.	d	35.	b	36.	c
37.	c	38.	c	39.	d	40.	a	41.	a	42.	c
43.	c	44.	d	45.	b	46.	d	47.	d	48.	b
49.	b	50.	d	51.	c	52.	a	53.	a	54.	c
55.	d	56.	c	57.	b	58.	d	59.	d	60.	c
61.	b	62.	a	63.	c	64.	a	65.	c	66.	b
67.	c	68.	a	69.	c	70.	d	71.	d	72.	a
73.	b	74.	c	75.	b	76.	c	77.	c	78.	b
79.	d	80.	d	81.	d	82.	b	83.	c	84.	b
85.	d	86.	a	87.	b	88.	d	89.	b	90.	c
91.	a	92.	b	93.	c	94.	a	95.	d	96.	a
97.	c	98.	a	99.	d	100.	d	101.	c	102.	c
103.	b	104.	c	105.	a	106.	b	107.	a	108.	a

Fill in the Blanks

1. A program used to create Web pages by interacting directly its interface is called a _____.
2. The language that all Web pages are written is _____.
3. HTML stands for _____.
4. HTML is a _____ language.
5. _____ control the structure and design of text and graphics on a Web page.
6. HTML tags are distinguished from other text in a Web page by using the symbol ___

7. Notepad automatically saves files as .txt, but Web pages should be saved as __ or __.
8. The HTML tags, <title> </title>, tell the browser what to display in the _____
9. The HTML tags, <a> </a>, are called _____ tags.
10. The code #0000FF is an example of a(n) _____ in HTML of a Web page.
11. The HTML tag, <hr>, used by itself indicates to the browser to insert a _____.
12. Links in a Web page frequently appear as _____ text underline
13. In HTML view, you can modify HTML tags and _____.
14. The last three characters of a file name are called the _____. File extension
15. To place an image on a Web page we should use the _____ tag.
16. _____ Tag is used to define a table.
17. _____ tag is used to define a row in a table.
18. _____ tag is used to define a column in row.
19. _____ tag is used to define the header row of the table.
20. _____tag is used to define the style and number of frames in a page.
21. A frame within a frame is called_____
22. _____ tag is used to create a from in a web page.
23. _____ attribute of <INPUT> tag indicates the type of form elements to be inserted in the form.
24. The _____ tag is used to end a line without inserting a new paragraph.
25. Paragraphs are defined with the_____ tag.
26. _____ tag is used to create an ordered list.
27. SGML stands for_____

Answers

1. Web Editor	2. HMTL	3. Hypertext Markup Language
4. markup	5. Markup tags	6. < >
7. .htm or .html	8. title bar	9. anchor
10. Color	11. Horizontal rule	12. Blue
13. Text	14. File extension	15. <img>
16. <TABLE>	17. <TR>	18. <TD>
19. <TH>	20. <FRAMESET>	21. Nested frame
22. <FORM>	23. type	24.
25. <p>	26. <OL>	27. Structured Generalized Markup language

True / False

1. A **</P>** tag must be used to terminate the **<P>** tag
2. HTML comments can extend for several lines of text?
3. The <H7> tag produces a heading with the largest font size.
4. The <BR> tag does not contain any attributes..
5. The <HR> tag contains WIDTH and HEIGHT attributes
6. The <DOCTYPE> tag tells a browser which language was use to create the page.
7. The <P>, <DIV>, and <HR> tags use the same ALIGN attribute.
8. An HTML document is divided into a head and a body .
9. The ** ** character entity can be used to create spaces in text.
10. The <SUP> tag is a physical style tag.
11. Browser safe colors use the following RGB values: 00 22 44 66 and 88.
12. Both Netscape & IE recognize FACE, SIZE, and COLOR attributes of <FONT> tag.

13. The default bullet style for an unordered list is CIRCLE.
14. In nested unordered lists, the browser changes bullet style and indentation of each nested list.
15. In nested ordered lists, the browser changes bullet style and the indentation of each nested list.
16. We can create ordered lists that are ordered by decimals, upper or lower case letters, and upper or lower case Roman numerals.
17. <LI> tag contains VALUE and TYPE attributes when used with ordered or unordered lists.
18. To create a definition list, we use <DEF>...</DEF> tags.
19. To define a term in a definition list, we use <DT>...</DT> tags.
20. Internet Explorer and Netscape Navigator equally forgive in treatment of missing table tables.
21. By default, Netscape Navigator and Internet Explorer underline all links.
22. JPEG images support animation and interlacing.
23. PNG is a imaging format that supports animation, transparency, and interlacing.
24. An image that serves as a hot link has a default border of one pixel.
25. The frame set page includes <HEAD> and <BODY> tags.
26. Asterisk * in <FRAMESET> tag indicates that row or column should consume all extra space.
27. To remove the default border that appears between frames we include NOBORDER attribute in <FRAMESET> tag.
28. The allowed values for SHAPE attribute of <AREA> tag include polygon and rect.
29. We use <MAP> ...</MAP> tags to define the areas of an image map.
30. To change the color of all unvisited links to green, we write: <BODY LINK = "green">

Answers

1.	F	2.	T	3.	F	4.	F	5.	F	6.	T
7.	T	8.	T	9.	T	10.	T	11.	F	12.	T
13.	F	14.	T	15.	F	16.	T	17.	F	18.	F
19.	T	20.	F	21.	T	22.	F	23.	T	24.	T
25.	F	26.	T	27.	F	28.	T	29.	T	30.	T

WWW.ITECVIEW.COM©

Part - 3

Cascading Style Sheets

Includes:

Chapter 6 Cascading Style Sheets

WWW.ITECVIEW.COM©

Cascading Style Sheets

6.1 DHTML

DHTML stands for **Dynamic Hyper Text Markup Language**. It is simply a term that is used to describe the new technologies. DHTML is not a standard defined by the **World Wide Web Consortium (W3C)**. It is used to add dynamic contents and interactivity in web pages. It is used to apply colors and add animations etc. It uses **documents object model (DOM), style sheets** and **scripting languages**.

- **Document Object Model:** Document Object Model defines the contents of web page as objects. It considers fonts, graphics and tables as objects.
- **Style Sheet:** Style sheet contains the characteristics of a document. Cascading style sheet contains the formatting details of each object in the document. Different style sheets can be used for applying different formatting to different objects.
- **Scripting Language:** Scripting language is used to manipulate the objects in the web page. A script can move, display, or change the appearance of an object as different events occur.

6.1.1 Benefits of DHTML

The benefits of using DHTML technology are as following:

1. Dynamic Style & Content

Dynamic HTML provides the facility of creating dynamic styles. Different styles can be generated dynamically in response to the user action. The contents of a web page can also be changed dynamically.

2. Less Complexity

Common scripting languages such as JavaScript and VBScript can be used to create interactive web pages. Complex and full-featured Web applications can be developed with DHTML without writing complex applets.

3. Reusability

DHTML code can easily be reused with multiple web pages. It reduces the total cost of web development and also saves a lot of time.

4. Support of Multiple Browsers

A single set of Web pages can be developed for all browsers that support the document object model for DHTML. Most popular browsers follow the standard features such as CSS. It ensures that the web pages are displayed similarly in many types of browsers.

5. Use of Third-Party Tools

DHTML supports a wide variety of Microsoft and third-party Web development and authoring tools like **Microsoft FrontPage, Microsoft Visual InterDev, SoftQuad HoTMetaL PRO 6.0** and **Pictorius iNet Developer 2.0.**

6. Quicker Downloads

DHTML files are much smaller than files for similar effects created with multimedia programs like Shockwave and Flash.

7. No Plug-Ins Required

DHTML uses tools already recognized by most browsers. The visitors do not have to download and install any special software to view the contents of the website. It increases the usability level of a website.

8. Increasing Popularity

The W3C is working on a final standards document for the **DOM**. Netscape and Internet Explorer have promised to follow to the standards. It will ease the cross-browser DHTML design for developers.

6.1.2 Disadvantages of DHTML

Some disadvantages of DHTML are as follows:

1. Incomplete Support

Netscape and Explorer support CSS and DOM but apply it to HTML document elements using different tags. Many DHTML effects require a **browser detection script** and two completely different sets of code to create the same effect in different browsers. For example, **WebTV** browser provides very limited support for JavaScript.

2. Expensive Tools

Several companies provide editors to write DHTML code. Dreamweaver and Fusion are two most popular tools. These tools can create effects quickly but the code is often large and hard to edit. The use of different tools also increases the cost of web development.

6.2 Cascading Style Sheets

Cascading Style Sheets were introduced in late 1996. Style sheet is a collection of formatting styles that can be applied to a web page. The designer can specify the design of a web page in a style sheet and then use it in the web page. The same style sheet can be used in multiple web page or even many websites. It provides the facility of creating websites with consistent designs and outlook.

Cascading Style Sheets are used to modify the properties of existing HTML tags. All Web browsers are based on a built-in style sheet. It instructs the browser how to display different elements in a web page. For example, when the browser sees the <P> tag, it knows to skip a line and start a new section. This information is stored in the built-in style sheet. When a new style sheet is defined, the built-in style sheet is overridden. So, the browser starts following the instructions of new style sheet instead of built-in style sheet.

Style sheets provide increased flexibility and the capability to add several attributes that are not available with normal HTML. Style Sheets are used to define styles for web page in many ways. Styles can be specified as follows:

- Inside an individual HTML element
- Inside the <head> element of an HTML page
- In an external CSS file

6.2.1 Cascading Order

Generally, all styles "cascade" into a new Style Sheet by the following rules in which the number 4 has the highest priority:

1. Browser default
2. External Style Sheet
3. Internal Style Sheet (inside the <head> tag)
4. Inline Style (inside HTML element)

An inline style inside an HTML element has the highest priority. It overrides every style declared inside **<HEAD>** tag, in an external style sheet and in a browser default style.

6.2.2 Benefits of Style Sheet

Some benefits of style sheets are as follows:

1. Override the Browser

Each browser has its own way of displaying web pages. In past, developers could not control how to display the page in a browser. But style sheet can be used to override the browser's style and set a new style for the web page.

2. Page Layout

Style sheets can be used to display fonts, change colors etc. without changing the structure of web page. A designer can separate visual design requirements from the logical structure of the web page. A style sheet is used to define the appearance of the elements on a web page.

3. Style Reusability

A style sheet can be used in many web documents by linking all web pages to a style sheet. It ensures that all web pages have a uniform appearance when they are displayed.

4. One time effort

A style sheet can be linked to several documents. If the style sheet is modified, the changes will be reflected in all the pages that are linked to it.

6.2.3 Parts of CSS

A style sheet consists of the following three parts:

- Selector
- Property
- Value

```
selector {property: value}
```

The **selector** is normally the HTML element or tag to be defined. The **property** is the attribute of the tag to be changed and **value** is the value of that **property**. The property and value are separated by colon and surrounded by curly braces.

Examples

```
H2 {color: green}
```

The above example changes the default color of **<H2>** tag. A style sheet can modify two or more attributes as follows:

```
H2 {color: green; font-family: Arial}
```

It is also possible to modify two or more tags at the same time as a group as follows:

```
H1, H2, H3
{
  color: green
}
```

The above example modifies the color of **<H1>**, **<H2>** and **<H3>** tags and sets to green.

6.3 Types of Style Sheets

Different types of style sheets are as follows:

6.3.1 External Style Sheets

An external style sheet is defined in a separate file stored with **.css** extension. It is very useful when the style is applied to many pages. An external style sheet is used to specify the style of an entire Web site in one file. The external style sheet file is then linked to the web page using the **<link>** tag. The **<link>** tag is written inside the **head** section.

Example

```
<HEAD>
<LINK REL="stylesheet" TYPE="text/css" HREF="mystyle.css">
</HEAD>
```

In the above example, the browser will read the style definitions from the file **mystyle.css** and format the document according to it. An external style sheet can be written in any text editor. The file should not contain any html tags.

```
H1 {color: red}
H2 {color: green}
H3 {color: blue}
```

Practice 6.1

Create an external CSS file that defines red color for H1, green color for H2 and blue color for H3. Use this file in a web page.

1. Open Notepad and type the following code:

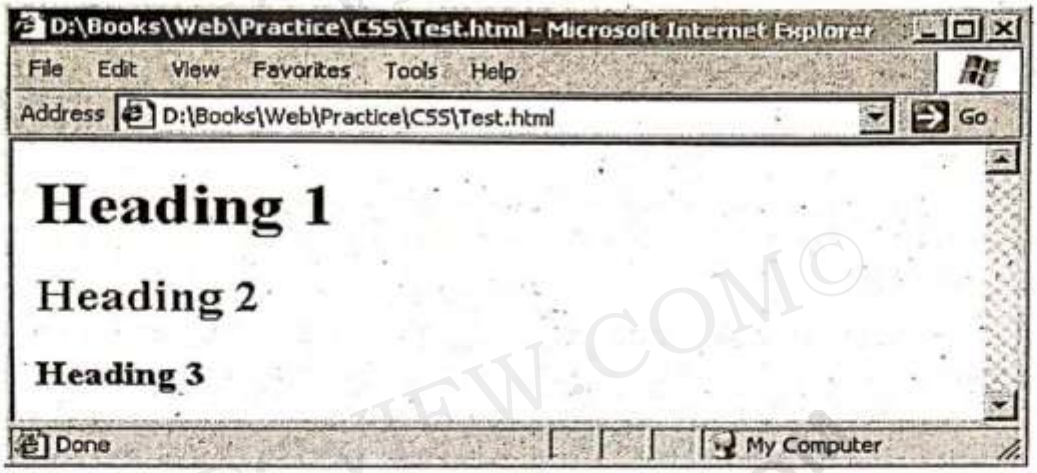
```
H1 {color: red}
H2 {color: green}
H3 {color: blue}
```

2. Save the file as "Test.css".
3. Open a new file in Notepad and type the following code:

```
<HTML>
<HEAD>
<LINK REL="stylesheet" TYPE="text/css" HREF="Test.css">
</HEAD>
<BODY>
<H1>Heading 1</H1>
<H2>Heading 2</H2>
<H3>Heading 3</H3>
```

```
</BODY>
</HTML>
```

4. Save the file as "Test.html".
5. Double click the file. The web page will appear as follows:



6.3.2 Internal Style Sheets

An internal style sheet is inserted in the head section of a web page. It only affects the web page in which it is inserted. It is inserted in a web page by using the **<STYLE>** tag.

Example

```
<HEAD>
<STYLE TYPE="text/css">
H4 {color: orange}
H5 {color: magenta}
</STYLE>
</HEAD>
```

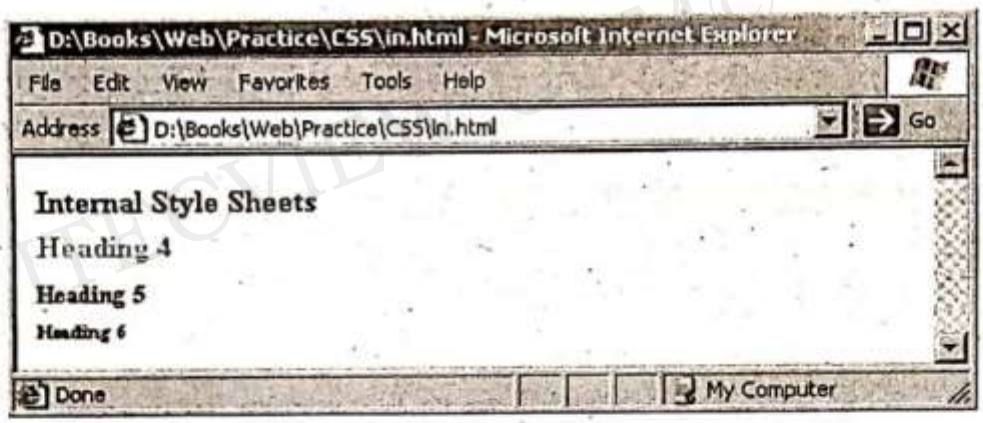
Practice 6.2

Write an internal style sheet that defines orange color for H4, magenta color for H5 and maroon color for H6.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
H4 {color: orange}
H5 {color: magenta}
H6 {color: maroon}
</STYLE>
</HEAD>
<BODY>
<H3>Internal Style Sheets</H3>
<H4>Heading 4</H4>
<H5>Heading 5</H5>
<H6>Heading 6</H6>
</BODY>
</HTML>
```

2. Save the file as "in.html".
3. Double click the file. The web page will appear in the browser.



6.3.3 Inline Styles

Inline style is applied to an individual tag. It modifies the attributes of a tag in the current occurrence of that tag. If the tag is used again, the default style of the tag will be used. The attributes are given in the opening tag.

Example

```
<P STYLE="color: red">It's my life</P>
```

The above example displays the contents of the paragraph in red color. But this attribute is valid for current occurrence only. If the **<P>** tag is used again in the document, it will display the contents in the default color.

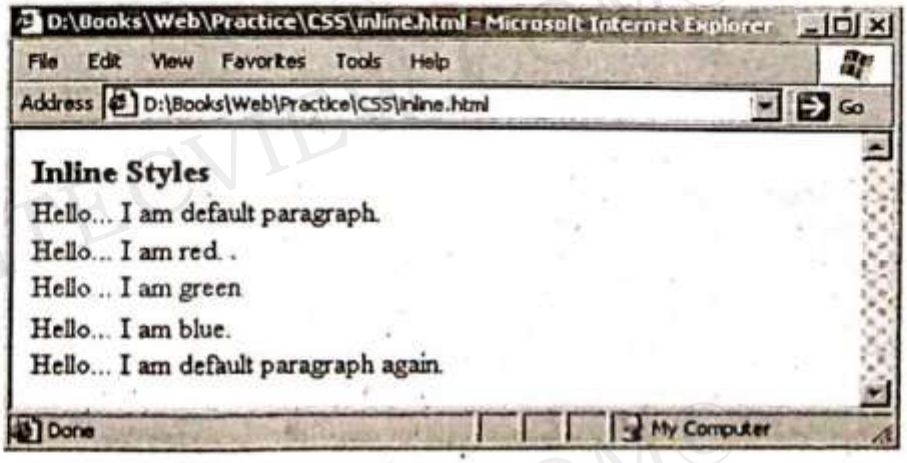
Practice 6.3

Write inline style sheet that defines different colors for different paragraphs.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H3>Inline Styles</H3>
<P>Hello... I am default paragraph.</P>
<P STYLE="color: red">Hello... I am red.</P>
<P STYLE="color: green">Hello... I am green.</P>
<P STYLE="color: blue">Hello... I am blue.</P>
<P>Hello... I am default paragraph again.</P>
</BODY>
</HTML>
```

2. Save the file as "inline.html".
3. Double click the file. The web page will appear in the browser as follows:



6.4 CSS and Text

HTML provides limited facilities for text formatting. CSS provides complete control on text formatting, presentation and style.

Text Color

Text color is changed by specifying any color to **color** property. The color value can be specified as hexadecimal code like **#00DDFF** or a named color value like **red** as follows:

```
H1 {color: red;}
H1 {color: #00DDFF;}
```

Typeface

Typeface is used to specify the font-family of the text as follows:

```
H2 {font-family: Verdana, Arial, Helvetica, sans-serif; }
```

The browser will keep going down the list of specified faces until it finds one that is installed on the system. If no font in the list is installed on the system, a font similar to the specified face will be used.

Text Size

CSS can be used to specify the text size. HTML can display text of size 7 at maximum. In CSS, **font-size** property is used to specify the size of the text. The size can be specified by using different units that are as follows:

- Units: points, and pixels
- Pre-defined keywords
- Percentages

Points

```
P {font-size: 50pt; }
```

The above line will create text like big text.

Pixels

```
P {font-size: 43px; }
```

More Units

There are a few other units of text-measurement that are less widely used:

- **in** is inches
- **cm** is centimetres
- **mm** is millimetres

Keywords

There are seven keywords that work similar to the **<font size>** values. The possible values are below:

```
B {font-size: large; }
```

- xx-small
- x-small
- small
- medium
- large
- x-large
- xx-large

WWW.ITECVIEW.COM©

Percentages

Percentage values are relative to the size of the parent value.

```
BODY {font-size: 20pt; }
P {font-size: 200%; }
```

In the above example, the font size of <BODY> is 20pt and font size of <P> is 200%. Any paragraph inside <BODY> will be sized 40pt.

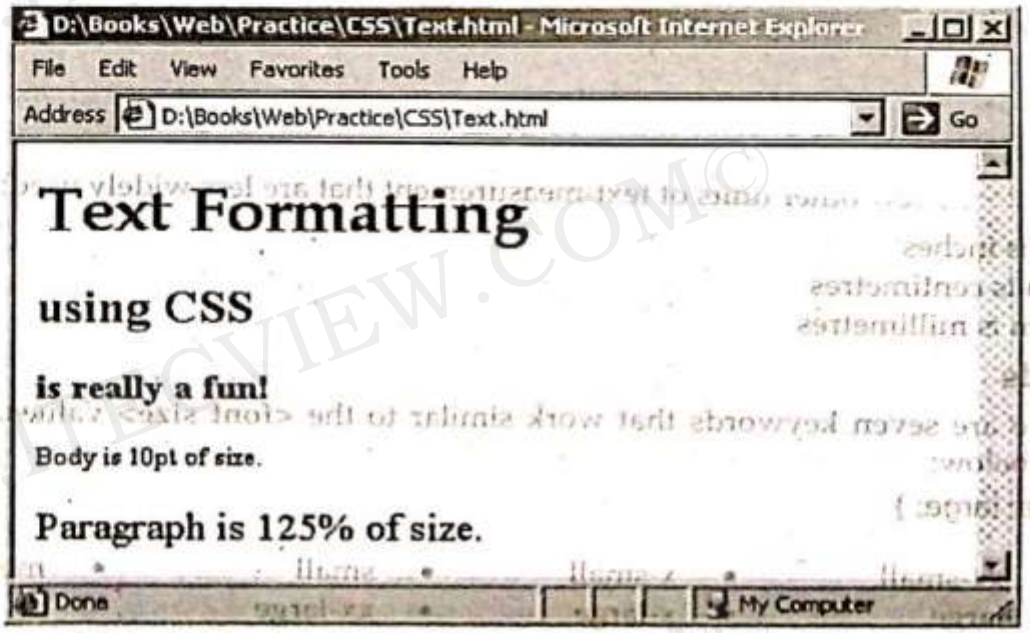
Practice 6.4

Create a web page that uses CSS code to define red color and Book Antiqua font for H1, green color for H2, blue color for H3, 10pt font size for body and 150% font size for paragraphs.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
H1 {color: red; font-family: Book Antiqua;}
H2 {color: green}
H3 {color: blue}
BODY {font-size: 10pt;}
P {font-size: 150%;}
</STYLE>
</HEAD>
<BODY>
<H1>Text Formatting</H1>
<H2>using CSS</H2>
<H3>is really a fun!</H3>
Body is 10pt of size.
<P>Paragraph is 125% of size.</P>
</BODY>
</HTML>
```

2. Save the file as "Text.html".
3. Double click the file. The web page will appear in the browser as follows:



6.4.1 Special Effects

Some special effects in CSS are as follows:

Italics

CSS command for italics is **font-style**. The possible values are **italic** or **normal**.

```
P {font-style: italic; }
```

Boldness

The property that controls the boldness of text is **font-weight**.

```
TD {font-weight: bold; }
```

Capitalization

CSS provides two properties to specify the capitalization of text. **font-variant** is used to set all characters in small versions of capital letters:

```
P {font-variant: small-caps; }
```

IT WILL DISPLAY TEXT LIKE THIS. The second property is **text-transform**:

```
P {text-transform: uppercase; }
```

The possible values are:

- **uppercase:** makes all the letters capitals;
- **lowercase:** makes all the letters small;
- **capitalize:** capitalize the first letter in every word;
- **none:** override any inherited transformation.

Text Decoration

The last formatting property is **text-decoration**. It is used to remove the underlines and other effects of the text. It is normally used to remove the underline from hyperlinks.

```
a:link {text-decoration: none; }
```

The possible values are as follows:

- **none:** removes any extra formatting and makes the text normal;
- **underline:** adds an underline;
- **overline:** adds a line over the text;
- **line-through:** creates a ~~strike-through~~ effect;
- **blink:** makes the text flash on and off. It is not supported by Internet Explorer.

Text Alignment

Text alignment is specified by using the property **text-align**. It only works on block-level elements i.e. the elements that add line breaks around themselves like **H1, P** etc.

```
P {text-align: center; }
```

The possible alignments are **left, right, center** and **justify**.

6.4.2 Text Layout

Text layout is used to control the exact spacing between letters, words and lines.

```
P {letter-spacing: 3px; }
```

The above example will add a space of 3 pixels after every letter.

```
P {word-spacing: 5px; }
```

The above example will add a space of 5 pixels after every world.

```
P { line-height: 3pt; }
```

The above example will insert a line spacing of 3 pixels.

Indents

```
P {text-indent: 10px; }
```

This command indents the first line in every paragraph.

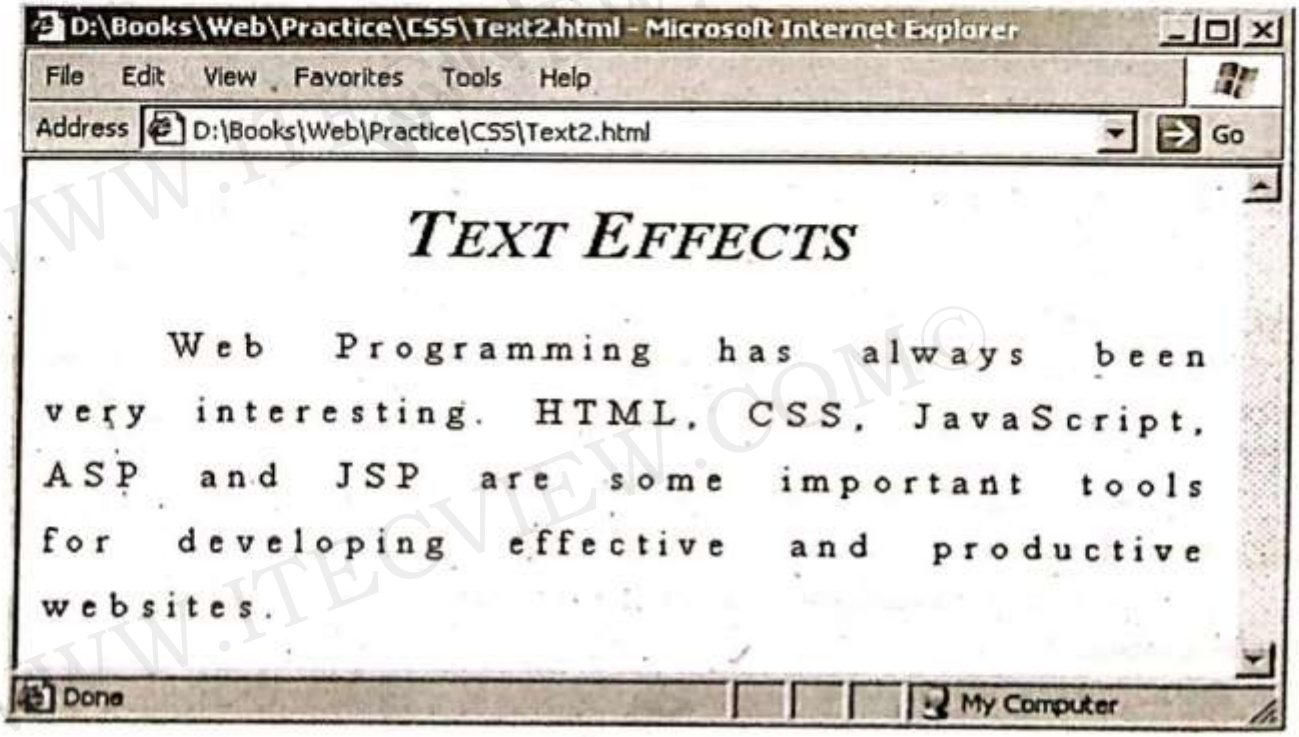
Practice 6.5

Create a web page that uses CSS code to define small-caps font, center alignment and italic style for H1 and 5px letter spacing, 5px word spacing, 30px line height, 50px text indent and justify text alignment for paragraphs.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
H1 {font-variant: small-caps; text-align: center; font-style: italic;}
P {letter-spacing: 5px; word-spacing: 5px; line-height: 30px; text-indent: 50px; text-align: justify;}
</STYLE>
</HEAD>
<BODY>
<H1>Text Effects</H1>
<P>Web Programming has always been very interesting. HTML, CSS, JavaScript, ASP and JSP are some important tools for developing effective and productive websites.</P>
</BODY>
</HTML>
```

2. Save the file as "Text2.html".
3. Double click the file. The web page will appear in the browser as follows:



6.5 Lists

CSS can be used to control the lists and their styles. Different properties related to lists are as follows:

List Style

The property to define the style of the list is **list-style**. It consists of different properties such as **list-style-type, list-style-position** and **list-style-image.**

list-style-type specifies the type of bullet displayed with each item in the list. The possible values are **disc, square, circle, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha** and **none.**

```
UL {list-style-type: square;}
```

List Position

list-style-position specifies the position of the bullet. The possible values are **inside** and **outside.** The **inside** property Indents the marker and the text. The **outside** property keeps the marker to the left of the text

```
UL {list-style-position: outside;}
```

List Image

CSS can be used to display an image with each item in the list as follows:

```
UL {list-style-image: url(redbullet.gif); }
```

If the image is unavailable, the browser will display the default dot.

Practice 6.6

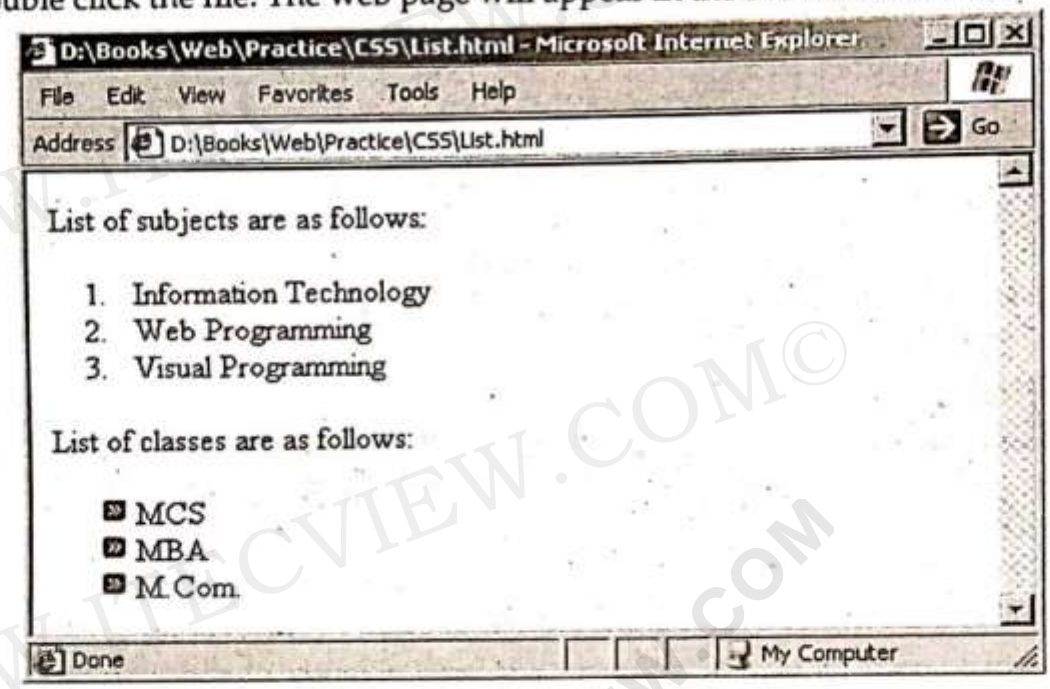
Create a web page that uses CSS code to define ordered list of lowercase roman number and outside position. It should also define unordered list with "Pic.gif" image and outside position.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
OL {list-style-type: low-roman; list-style-position: outside;}
UL {list-style-image: url(Pic.gif); list-style-position: outside;}
</STYLE>
</HEAD>
<BODY>
List of subjects are as follows:
<OL>
<LI>Information Technology
<LI>Web Programming
<LI>Visual Programming
</OL>
List of classes are as follows:
<UL>
<LI>MCS
<LI>MBA
<LI>M.Com.
</UL>
</BODY>
```

```
</HTML>
```

2. Save the file as "List.html".
3. Double click the file. The web page will appear in the browser as follows:



6.6 CSS and Hyperlinks

The following four style sheet entities are used to control hyperlinks in CSS:

```
a: link { }
a: visited { }
a: active { }
a: hover { }
```

a:link is used to specify the attributes of a hyperlink.

```
a:link {color: blue; text-decoration: none; }
```

a:visited is used to specify the attributes of the hyperlinks that have been visited.

```
a:visited {color: red; text-decoration: none; }
```

a:active is used to specify the attributes of the active hyperlinks.

```
a:active {color: green; text-decoration: none; }
```

a:hover is used to specify attributes of the hyperlinks on which the user is moving the mouse.

```
a:hover {color: magenta; text-decoration: none; }
```

Practice 6.7

Create a web page that uses CSS code that defines red color with no decoration for links, puple color with no decoration for visited links, blue color for active links and orage color with underline decoration for hover links.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
a:link {color: red; text-decoration: none; }
a:visited {color: purple; text-decoration: none; }
```

```
a:active {color: blue; }
a:hover {color: orange; text-decoration: underline; }
</STYLE >
</HEAD>
<BODY>
<A HREF="http://www.itseries.com.pk">IT Series</a>
</BODY>
</HTML>
```

2. Save the file as "Link.html".
3. Double click the file. The web page will appear as follows:

6.7 CSS and Backgrounds

CSS can be used to apply background colors and background images to different parts of a web page.

6.7.1 Background Colors

In HTML, background colors and background image can be applied to the whole page or to a table. CSS provides the facility to apply background color or image to even a single word. The property used for this purpose is **background-color**.

```
P {background-color: green; }
B {background-color: red;}
```

6.7,2 Background Images

CSS also provides the facility to add an image as a background to any part of the page. The property for this purpose is **background-image**.

```
BODY {background-image: url("Pic.gif"); }
```

6.7.3 Background Control

The style of displaying the background image can easily controlled in CSS. Different options for controlling background image are as follows:

Background Repetition

The repetition of image in the background can be controlled by using **background-repeat** property.

```
P {background-image: url("Pic.gif"); background-repeat: no-repeat; }
```

The above example will display the image once in the top-left of the paragraph. The possible values are as follows:

- **repeat-x:** repeats the image horizontally for one row
- **repeat-y:** repeats the image vertically for one column.

Background Attachment

Normally, the background image moves when the user scrolls up or down the page. **background-attachment** property is used to specify the background attachment.

```
BODY {background-image: url("Pic.gif"); background-attachment: fixed; }
```

The other value for this property is **scroll** that restores the default option.

Background Position

You can also specify the position of the background image relative to the element behind which it is applied. The possible values are **top, center, bottom, left** and **right.**

```
P {background-image: url("Pic.gif"); background-position: right bottom; }
```

You can also specify horizontal and vertical starting points as follows:

```
P {background-position: 60px 15px; }
```

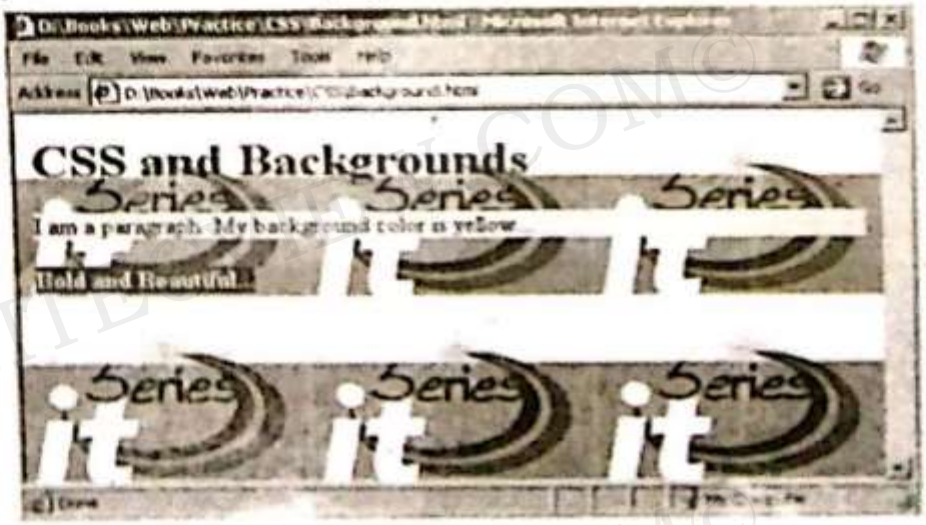
Practice 6.8

Create a web page that uses CSS code to define "itmono.jpg" as background image for body, yellow color for paragrah background, green background color and white text color for bold text.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<style type="text/css">
BODY {background-image: url("itmono.jpg"); }
P {background-color: yellow; }
B {background-color: green; color: white;}
</style >
</HEAD>
<BODY>
<H1>CSS and Backgrounds</H1>
<P>I am a paragraph. My background color is yellow...</P>
<B>Bold and Beautiful...</B>
</BODY>
</HTML>
```

2. Save the file as "Background.html".
3. Double click the file. The web page will appear in the browser as follows:



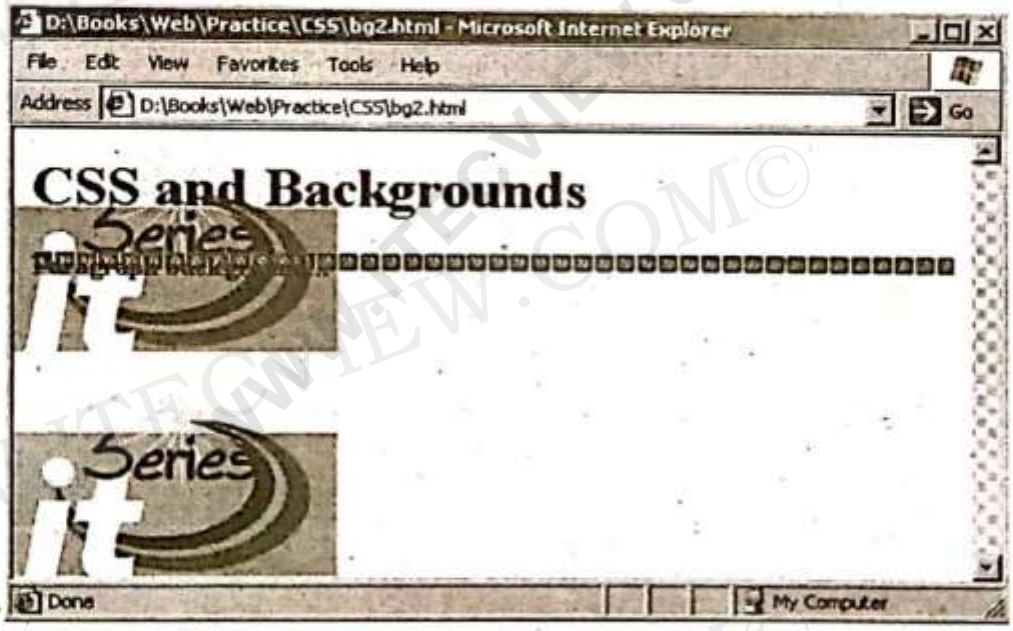
Practice 6.9

Create a web page that uses CSS code that defines "itmono.jpg" as background image of body. The image should repeat vertically once. It defines "Pic.gif" as background image of paragraph. The image should repeat horizontally once.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<style type="text/css">
BODY {background-image: url("itmono.jpg"); background-repeat: repeat-y}
P {color: blue; background-image: url("Pic.gif"); background-repeat: repeat-x }
</style >
</HEAD>
<BODY>
<H1>CSS and Backgrounds</H1>
<B><P>Paragraph background...</P></B>
</BODY>
</HTML>
```

2. Save the file as "bg2.html".
3. Double click the file. The web page will appear in the browser as follows:



In the above example, the image behind the BODY appears vertically for one column and the image behind the paragraph appears horizontally for one row.

6.8 CSS and Borders

There important properties that are used to change the borders are as follows:

- **border-width**
- **border-color**
- **border-style**

All three properties can be used together by using **border** attribute.

```
<P STYLE="border: 3px coral solid;">Hello World</P>
```

Hello World

In the above example, all three values are separated with space. 3px indicates the width, coral indicates the color and solid indicates the style of the border.

6.8.1 Parts of Border

There are four parts to each border:

- border-top
- border-bottom
- border-left
- border-right

The above four properties can be used to display each side of the border differently.

Border Width

Border width can be specified in pixels or one of the standard values **thin, medium** and **thick**.

```
border-width: thin;
```

Border Color

The **border-color** property is used to specify the color of the border. It can be specified as color name or its hexadecimal value.

```
border-color: red;
```

Border Style

There are eight possible styles for border. These are **solid, dashed, dotted, double, groove, ridge, inset** and **outset**.

```
border-style: dashed;
```

The **width, color** and **style** properties can also be applied to the individual parts of the border. In this case, the individual part must be specified in the statement as follows:

```
border-top-style: solid;
border-top-color: green;
```

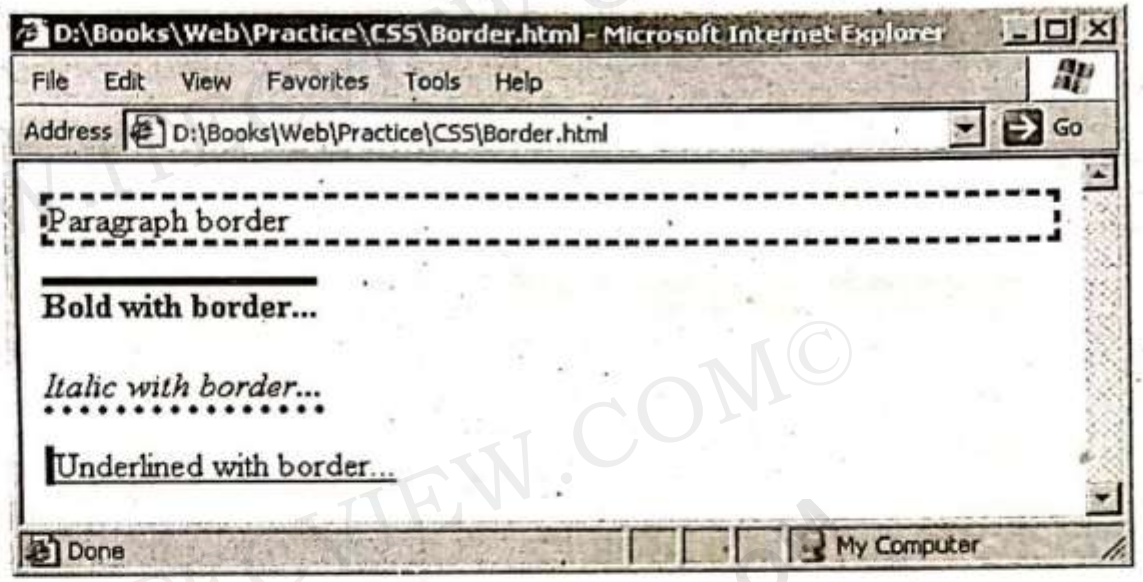
Practice 6.10

Create a web page that uses CSS code to define dashed green border for paragraphs, solid red top border for bold text, dotted blue bottom border for italic text and solid red left border for underlined text.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
P {border: 3px dashed green;}
B {border-top-style: solid; border-top-color: red;}
I {border-bottom-style: dotted; border-bottom-color: blue;}
U {border-left-style: solid; border-left-color: red;}
</STYLE >
</HEAD>
<BODY>
<P>Paragraph border</P>
<B>Bold with border...</B><BR><BR>
<I>Italic with border...</I><BR><BR>
<U>Underlined with border...</U>
</BODY>
</HTML>
```

2. Save the file as "Border.html".
3. Double click the file. The web page will appear as follows:



6.9 CSS and Tables

CSS can be used to customize the text, borders and padding of the tables. The color, border and background of the tables can be controlled as discussed earlier.

Padding

The Padding properties specify the space between the cell border and the content of a table. The top, right, bottom, and left padding can be changed independently using separate properties. A shorthand padding property is also created to control multiple sides at once. The following properties are used to specify the padding values:

- padding-top
- padding-bottom
- padding-right
- padding-left

```
TD {padding: 10px; }
```

The above example inserts padding of 10 pixels in all sides of the cell.

```
TD {padding: 40px 30px 20px 10px; }
```

In the above example, padding values for all sides are given. The sequence of sides is **top, right, bottom** and **left.**

Practice 6.11

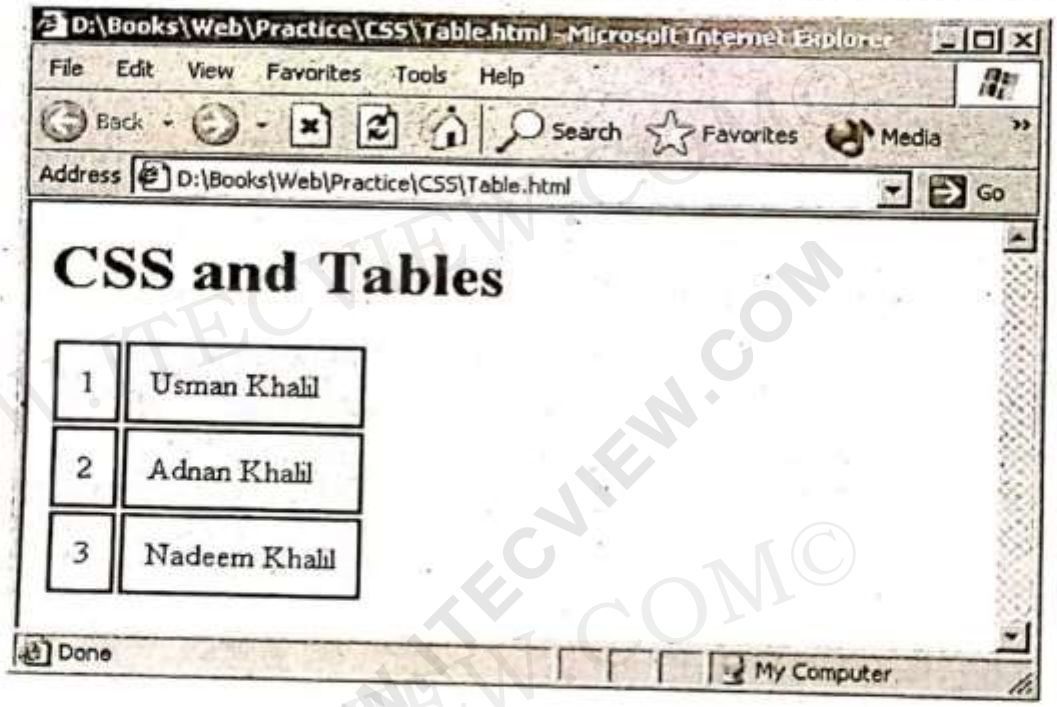
Create a web page that uses CSS code to define green text color, yellow background color, 2px solid red border and 10px padding for <TD> tag.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
TD {color: green; background-color: yellow; border: 2px solid red; padding:10px;}
</STYLE >
</HEAD>
<BODY>
```

```
<H1>CSS and Tables</H1>
<TABLE>
        <TR><TD>1</TD><TD>Usman Khalil</TD></TR>
        <TR><TD>2</TD><TD>Adnan Khalil</TD></TR>
        <TR><TD>3</TD><TD>Nadeem Khalil</TD></TR>
</TABLE>
</BODY>
</HTML>
```

2. Save the file as "Table.html".
3. Double click the file. The web page will appear in the browser as follows:



6.10 CSS and Scrollbars

CSS can customize the color of different parts of scrollbar. A scrollbar consists of different parts that are **base, arrow** and **track**.

Different properties that define the colors of scrollbar are as follows:

- scrollbar-base-color
- scrollbar-arrow-color
- scrollbar-track-color
- scrollbar-shadow-color
- scrollbar-lightshadow-color
- scrollbar-darkshadow-color
- scrollbar-highlight-color
- scrollbar-3dlight-color

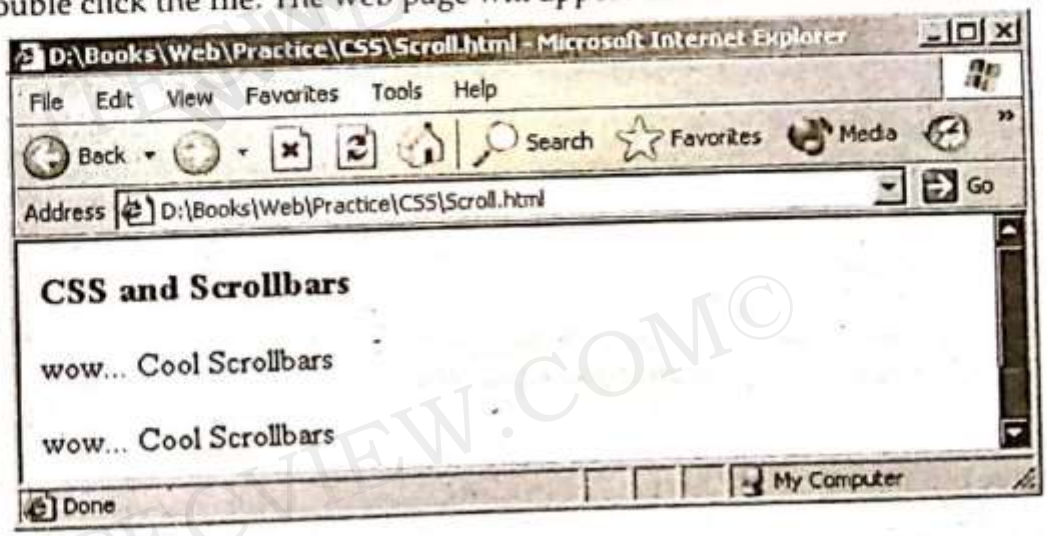
Practice 6.12

Create a web page that uses CSS code to define red base color, white arrow color, green track color, black shadow color, black light shadow color, gray dar shadow color, white highlight color and black 3D light color for the scrollbar.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
BODY {
scrollbar-base-color: red;
scrollbar-arrow-color: white;
scrollbar-track-color: green;
scrollbar-shadow-color: black;
scrollbar-lightshadow-color: black;
scrollbar-darkshadow-color: gray;
scrollbar-highlight-color: white;
scrollbar-3dlight-color: black;
}
</STYLE>
</HEAD>
<BODY>
<H3>CSS and Scrollbars</H3>
<P>wow... Cool Scrollbars</P>
<P>wow... Cool Scrollbars</P>
<P>wow... Cool Scrollbars</P>
</BODY>
</HTML>
```

2. Save the file as "Scroll.html".
3. Double click the file. The web page will appear in the browser as follows:



6.11 CSS and Cursors

CSS can be used to change the shape of the cursor. By default, cursor changes to different shapes as it is rolled over different objects. The default appearance of the cursor can be customized by using **cursor** attribute as follows:

```
style="cursor: hand; "
a {cursor: help; }
```

The possible values of cursor attribute are as follows:

Cursor property	Description
cursor: auto	The default cursor
cursor: crosshair	Gun-style cross
cursor: default	No change
cursor: hand	The normal link-hand
cursor: wait	The hourglass (non-animated, sadly)
cursor: text	The text-selecting
cursor: help	An arrow with a question-mark
cursor: move	Crosshair with arrows on the ends
cursor: n-resize	An arrow pointing north
cursor: ne-resize	An arrow pointing north-east
cursor: nw-resize	An arrow pointing north-west
cursor: e-resize	An arrow pointing east
cursor: w-resize	An arrow pointing west
cursor: s-resize	An arrow pointing south
cursor: se-resize	An arrow pointing south-east
cursor: sw-resize	An arrow pointing south-west

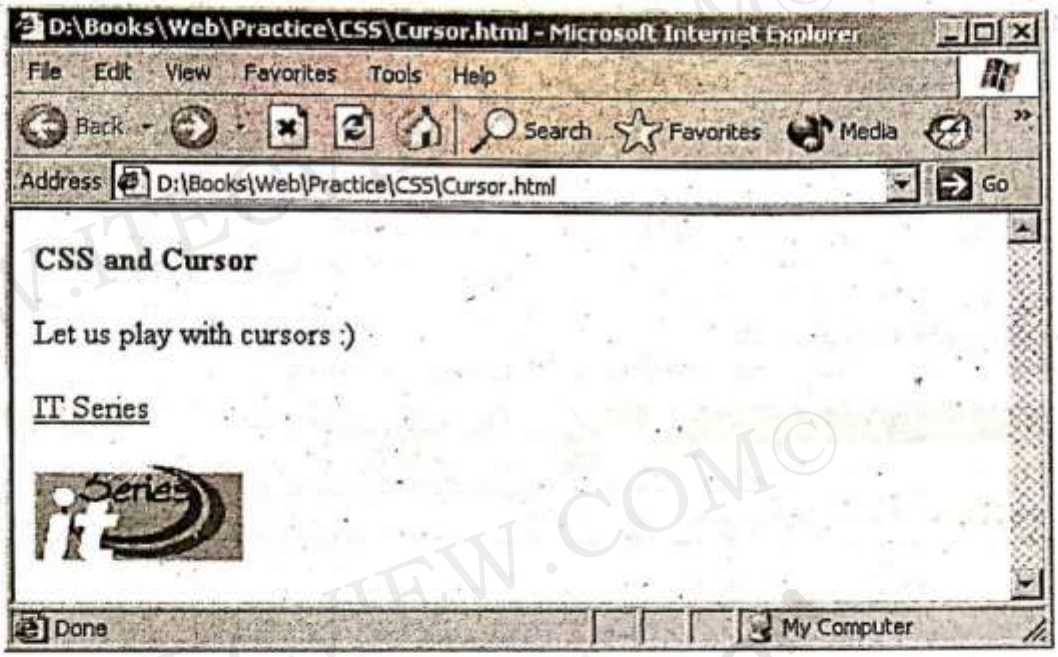
Practice 6.13

Create a web page that uses CSS code to define *wait* cursor for <A> tag, *help* cursor for <P> tag, *hand* cursor for <B> tag and *crosshair* cursor for <IMG> tag.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
A {cursor: wait;}
P {cursor: help;}
B {cursor: hand;}
IMG {cursor: crosshair;}
</STYLE>
</HEAD>
<BODY>
<B>CSS and Cursor</B>
<P>Let us play with cursors :)<P>
<A HREF="http://www.itseries.com.pk">IT Series</A>
<BR><IMG SRC="itmono.jpg" width=100 height=70>
</BODY>
</HTML>
```

2. Save the file as "Cursor.html".
3. Double click the file. The web page will appear in the browser.



6.12 CSS Class

CSS provides the facility to define two or more styles for the same tag. For example, an unordered list can be displayed with two different images using CSS as follows:

```
UL.One {list-style-image: url(Pic.gif); }
UL.Two {list-style-image: url(arrowright.gif); }
```

The above two lines define two different styles for **<UL>** tag. The style to be used in a particular **<UL>** tag is specified using class attribute as follows:

```
<UL class="One">
<UL class="Two">
```

The first **<UL>** tag will use class **One** and the second **<UL>** tag will use class **Two**. Both unordered list will be displayed with different images.

Practice 6.14

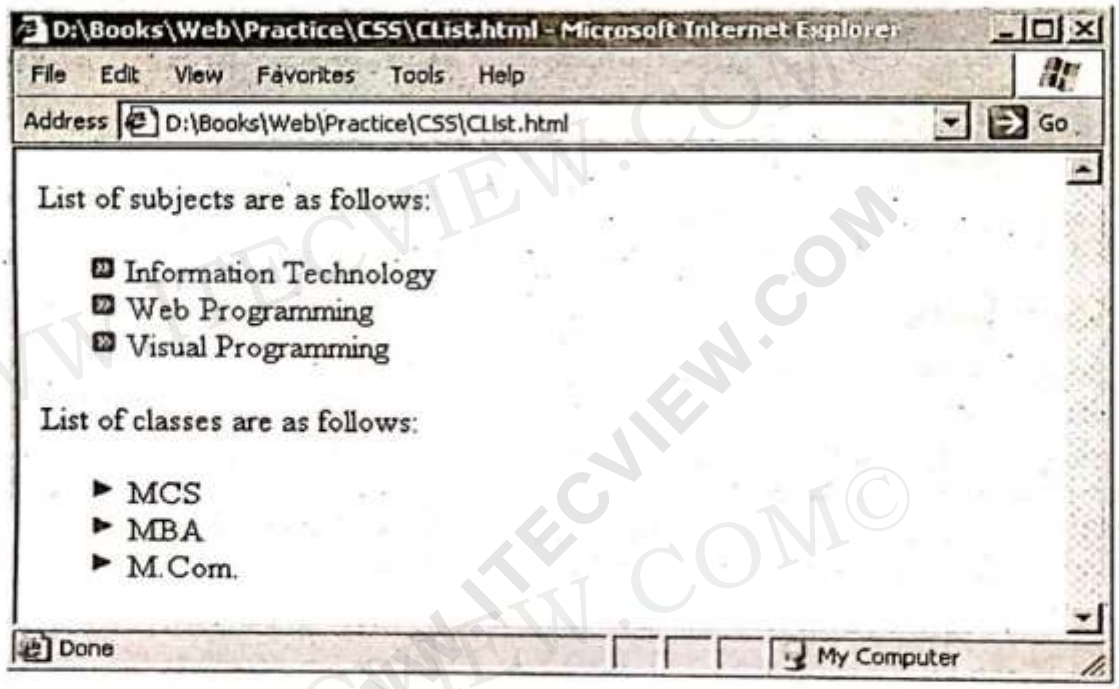
Create a web page that uses CSS code that defines two classes for unordered list. One class defines "Pic.gif" as list image and outside position for the list. The other defines "arrowright.gif" as list image and outside position for the list.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
UL.One {list-style-image: url(Pic.gif); list-style-position: outside;}
UL.Two {list-style-image: url(arrowright.gif); list-style-position: outside;}
</STYLE>
</HEAD>
<BODY>
List of subjects are as follows:
<UL class="One">
        <LI>Information Technology
        <LI>Web Programming
        <LI>Visual Programming
```

```
</UL>
List of classes are as follows:
<UL class="Two">
        <LI>MCS
        <LI>MBA
        <LI>M.Com.
</UL>
</BODY>
</HTML>
```

2. Save the file as "CList.html".
3. Double click the file. The web page will appear as follows:



Exercise Questions

1. Write CSS code that gives all H1 and H2 elements a padding of 0.5 ems, a grooved border style and a margin of 0.5 ems.
2. Write an HTML document that shows the names of colors in one column and colors in second column of a table. Use attributes to format width, border and cell spacing for the table.
3. Write CSS code that defines capital roman number for first level of list, capital alphabets for second level nested list and numbers for third level nested list.
4. Write CSS code that defines width to *medium*, color to *red* and style to *dotted* for top border of table. It defines width to *thick*, color to *blue* and style to *dashed* for bottom border. It defines style to *dashed*, width to *thin* and color to *green* for paragraph.
5. Write CSS code that defines "books.jpg" as background image for body, 30px left margin, 30px right margin, 50px top margin and 14pt font size for paragraphs.
6. Write CSS code that defines five classes of paragraphs width different background colors, margins, padding and border styles.

Short Questions

Q.1. What is DHTML?

DHTML stands for **Dynamic Hyper Text Markup Language.** It is simply a term that is used to describe the new technologies such as CSS, JavaScript etc.

Q.2. What is meant by document object model?

Document Object Model defines the contents of web page as objects. It considers fonts, graphics and tables as objects.

Q.3. What is style sheet?

Style sheet contains the characteristics of a document. Cascading style sheet contains the formatting details of each object in the document. Different style sheets can be used for applying different formatting to different objects.

Q.4. What is meant by scripting language?

Scripting language is used to manipulate the objects in the web page. A script can move, display, or change the appearance of an object as different events occur.

Q.5. List some benefits of DHTML.

The benefits of using DHTML technology are as following:

- **Dynamic Style & Content:** Dynamic HTML provides the facility of creating dynamic styles. The contents of a web page can also be changed dynamically.
- **Less Complexity:** Complex and full-featured Web applications can be developed with DHTML without writing complex applets.
- **Reusability:** DHTML code can easily be reused with multiple web pages.
- **Support of Multiple Browsers:** A single set of Web pages can be developed for all browsers that support the document object model for DHTML.
- **Use of Third-Party Tools:** DHTML supports a wide variety of Microsoft and third-party Web development and authoring tools.

Q.6. List some disadvantages of DHTML.

Some disadvantages of DHTML are as follows:

- **Incomplete Support:** Many DHTML effects require a **browser detection script** and two completely different sets of code to create the same effect in different browsers.
- **Expensive Tools:** Several tools are used to create DHTML effects. The use of different tools also increases the cost of web development.

Q.7. List some benefits of Style Sheet.

Some benefits of style sheets are as follows:

- **Override the Browser:** Style sheet can be used to override the browser's style and set a new style for the web page.
- **Page Layout:** Style sheets can be used to display fonts, change colors etc. without changing the structure of web page.
- **Style Reusability:** A style sheet can be used in many web documents by linking all web pages to a style sheet. It ensures that all web pages have a uniform appearance.
- **One time effort:** A style sheet can be linked to several documents. If the style sheet is modified, the changes will be reflected in all the pages that are linked to it.

Q.8. What are different parts of CSS?

A style sheet consists of the following three parts:

- **Selector:** It is normally the HTML element or tag to be defined.
- **Property:** It is the attribute of the tag to be changed
- **Value:** It is the value of that property.

Q.9. What is external style sheet?

An external style sheet is defined in a separate file stored with .css extension. An external style sheet is used to specify the style of an entire Web site in one file.

Q.10. What is internal style sheet?

An internal style sheet is inserted in the head section of a web page. It only affects the web page in which it is inserted. It is inserted in a web page by using the <STYLE> tag.

Q.11. What is inline style?

Inline style is applied to an individual tag. It modifies the attributes of a tag in the current occurrence of that tag. If the tag is used again, the default style of the tag will be used. The attributes are given in the opening tag.

Q.12. How font size is specified for text in CSS?

The size can be specified by using different units that are as follows:

- Units: points, pixels, inches, centimeters and millimeters
- Pre-defined keywords: xx-small, x-small, small, medium, large, x-large and xx-large
- Percentages

Multiple Choice

1. **DHTML is a term used to describe the combination of all EXCEPT:**
   a. HTML. b. JavaScript. c. CSS. d. Java.
2. **Cascading Style Sheets contain style definitions for specific formatting of Web page elements, such as:**
   a. Position of text b. Fonts and colors c. Image layout d. All
3. **DHTML is a term for a collection of:**
   a. HTML tags b. HTML options c. style sheet properties d. All
4. **Which tag is used to include a separately defined style sheet in web pages?**
   a. <style> b. <link> c. <body> d. <import>
5. **The style statements for an external style sheet should be entered into:**
   a. A section of HTML code b. A text file c. An image file d. All
6. **To define a custom class in CSS, the selector name must be preceded with:**
   a. An ampersand (&). b. Keyword "custom:" c. A period (.) d. Nothing
7. **In style sheet, options color, style and width can be controlled for _____ property.**
   a. background b. border c. font d. margin
8. **Which property of Web page refers to the amount of transparent space between elements on the page?**
   a. edge b. margin c. heading d. spacing
9. **Which tag must be inserted within <HEAD> tag to link it to external style sheet?**
   a. < external > b. < style > c. < link > d. < apply >
10. **In a style sheet, the options image, position, and type can be controlled for the _____ property.**
    a. List b. text c. font d. margin
11. **Which property-value pair is used to change italic or oblique text back to display in the default style?**
    a. normal: font-style b. font-style: normal
    c. font-style: default d. default: font-style

12. All of the following are pseudo classes for the "a" selector (anchor tag) EXCEPT:
a. visited. b. hover. c. link. d. color.

13. _____ are ideal to apply the same formats to all of the Web pages in a Web site.
a. Inline style sheets b. Embedded style sheets
c. External style sheets d. Foreign style sheets

14. The style statements that comprise an embedded style sheet appear:
a. in the body of Web page
b. between < style > and < /style > tags
c. immediately before specific elements that are named in the style statement
d. all of the above

15. Which value for TYPE property is used to link a style sheet to a Web page?
a. "link" b. "text/css" c. "style" d. "external"

16. CSS can be included in your HTML pages in all of the following ways EXCEPT:
a. Linked from an external file. b. In the document head using the <css> tag.
c. Inline as an HTML attribute d. Embedded via the <style> tag.

17. The part of the style statement that identifies the page element(s) is called the:
a. attribute b. style name c. selector d. declaration

18. The text-indent property allows you to specify a fixed value in _____, or a relative value using percentage.
a. Points b. inches or centimeters c. pixels d. All

19. With a style sheet, you can change things such as:
a. font family and size b. margins c. link specifications d. All

20. Examples of property-value pairs include:
a. font-family: Garamond b. font-color: navy c. font-size: 32pt d: All

21. All of the following positioning methods are supported by CSS EXCEPT:
a. static. b. relative. c. absolute. d. incremental.

22. Setting the text-decoration property to _____ will place a line above the text.
a. blink b. line-through c. overline d. underline

23. CSS provides support for _____ types of style sheets.
a. two b. three c. four d. five

24. CSS properties that contain two words are written as:
a. All lowercase with a hyphen between words b. All lowercase as one word.
c. All uppercase as one word.
d. One word where the first letter of the second word is capitalized.

25. The CSS attribute font-size has a companion property in JavaScript called:
a. Font. b. FontFamily. c. FontStyle. d. FontSize.

26. Which of the following is NOT changeable via the use of CSS properties?
a. Margins b. Borders c. Scroll bars d. File control.

27. Which of the following is a CSS font property:
a. Font-family b. Font-size c. Font-style d. All

28. A CSS (Style Sheet) consists of a list of:
a. Tags b.Rules c. Comments d. None

29. Which of the following are valid CSS properties:
a. String b. integer c. URL d. All -

Answers

1. D	2. d	3. d	4. b
5. b	6. c	7. b	8. b
9. c	10. a	11. b	12. d
13. c	14. b	15. b	16. b
17. c	18. d	19. d	20. d
21. d	22. c	23. b	24. a
25. d	26. d	27. d	28. b
29. d			

Fill in the Blanks

1. DHTML stands for _____
2. Cascading Style Sheets were introduced in late _____.
3. _____is a collection of formatting styles that can be applied to a web page
4. A style sheet consists of the _____parts
5. An external style sheet is defined in a separate file stored with _____extension
6. The external style sheet file is then linked to the web page using the ____tag
7. An _____is used to specify the style of an entire Web site in one file
8. An internal style sheet is inserted in a web page by using the _____tag
9. _____is used to specify the font-family of the text.
10. _____.is used to remove the underlines and other effects of the text
11. _____specifies the type of bullet displayed with each item in the list
12. The repetition of image in the background can be controlled by using _____property
13. The type attribute of a style tag for CSS should be set to _____ .
14. The CSS rule to set the text size to 12 points is _____.
15. The _____ attribute can be used when changing the background cell color of a table

Answers

1. Dynamic Hyper Text Markup Language	2. 1996
3. Style sheet	4. three
5. .css	6. <link>
7. external style sheet	8. <STYLE>
9. Typeface	10. text-decoration
11. list-style-type	12. background-repeat
13. text/css	14. {font-size: 12 pt;}
15. background-color	

WWW.ITECVIEW.COM©

True / False

1. To create a style class we use the period . as a prefix.
2. DHTML stands for Dynamic Hyper Text Markup Language.
3. DHTML is not a standard defined by the World Wide Web Consortium (W3C).
4. A style sheet consists of the following five parts.
5. An external style sheet is defined in a separate file stored with .css extension.
6. The external style sheet file is then linked to the web page using the <hplink> tag.
7. Internal style sheet is inserted in a web page by using the <STYLE> tag.
8. Inline style is applied to an individual tag.
9. Typeface is used to specify the font-family of the text.
10. There are twenty keywords that work similar to the <font size> values.
11. CSS command for italics is font-style.
12. The property that controls the boldness of text is font-varient .
13. CSS can be used to control the lists and their styles.
14. The property to define the style of the list is list-style.
15. CSS provides the facility to add an image as a background to any part of the page.
16. The repetition of image in the background can be controlled by using background-repeat property.
17. Border-width, border-color and border-style are important properties used to change the borders .
18. Border width can be specified in pixels or one of the standard values small, medium and large.
19. CSS does not provide the facility to define two or more styles for the same tag.
20. Netscape and Explorer support CSS and DOM but apply it to HTML document elements using different tags.

Answers

1. T	2. T	3. T	4. F
5. T	6. F	7. T	8. T
9. T	10. F	11. T	12. F
13. T	14. T	15. T	16. T
17. T	18. F	19. F	20. T

WWW.ITECVIEW.COM©

Part - 4

JavaScript

Includes:

Chapter 7	Introduction to JavaScript
Chapter 8	Conditions and Loops
Chapter 9	Functions and Arrays
Chapter 10	JavaScript Object Model

WWW.ITECVIEW.COM©

7

Introduction to JavaScript

7.1 Scripts

Script is an interpreted program that runs on the **client** computer. It is written in a scripting language. An interpreted language used to write scripts is called scripting language.

HTML is used to exchange information over the Internet. It provides little or no control over a web page once it is displayed in the browser. It is not possible to interact with the user in HTML. Moreover, the contents of an HTML document are static. **Scripting** is a method that can be used to create dynamic HTML pages and provide user interaction. There are several methods that can be used to create dynamic pages:

- Server-based scripts
- Client-side scripts

7.1.1 Server-Side Scripts

Server-side scripts are stored and executed on servers. These scripts are used to provide user interaction. Data from the client machine is sent to the Web server. Server-side scripts process the request and send the required page to the client browser.

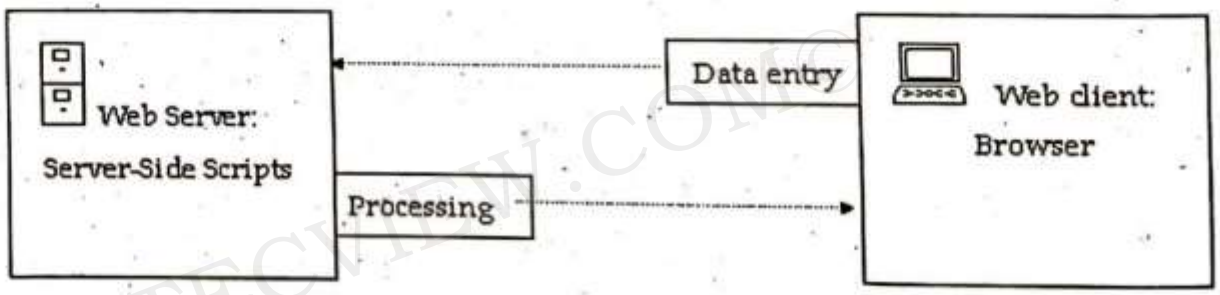


Figure: Server-side processing

7.1.2 Client-Side Scripts

Client-side scripts are executed on client computers. The browsers support the use of scripts. These scripts can be embedded in HTML documents. The browser interprets and executes the code embedded in the page.

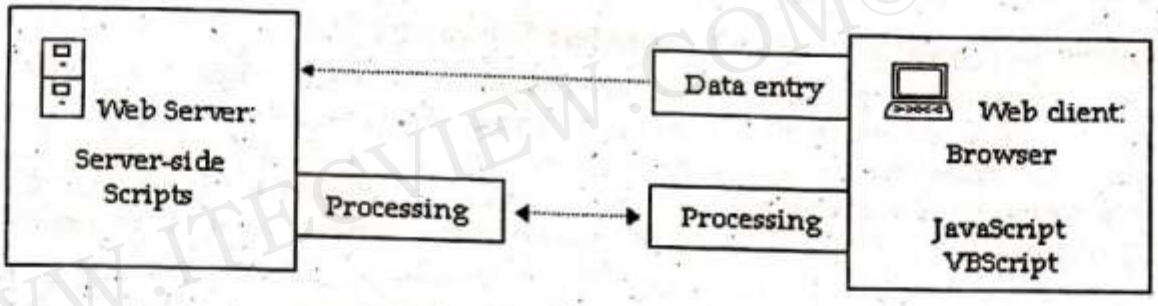


Figure: Client-side processing

Scripts can be used to change the content or style of a Web page in response to user input. If the application is database-oriented, scripts can be used to validate the data entered by the user before it is sent to the server for processing. For example, a client-side script can verify if the user has entered data in all the required fields before the form is submitted to the server for processing.

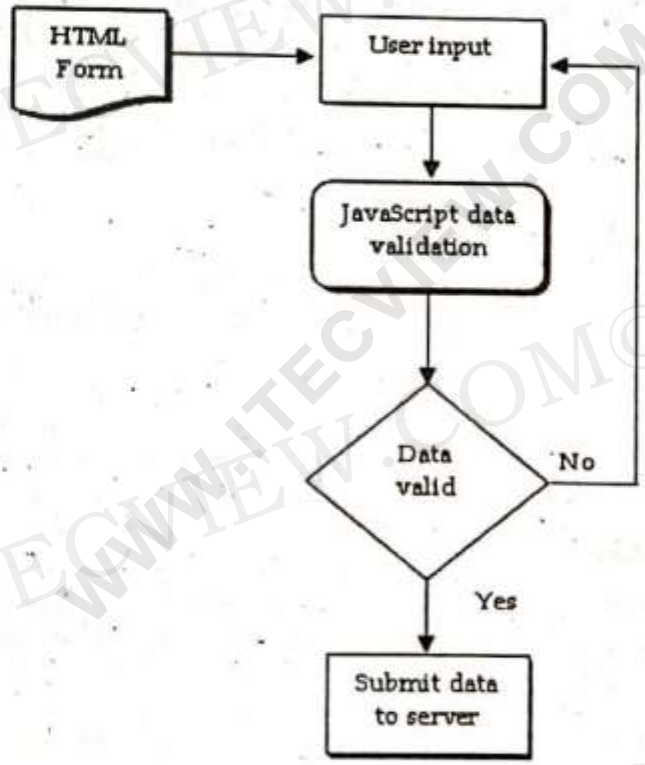


Figure: Validating data

7.2 JavaScript

JavaScript is an interpreted language that is used to add dynamic contents in Web pages. Interpreted means that the entire web page is downloaded to the browser and the JavaScript code is executed when an event is triggered. When the code is executed it is interpreted one line at a time. There are a number of events that will trigger the execution of a JavaScript such as click on a form button or the completion of a web page loading etc.

JavaScript is used to include messages, scrolling text, animations, menus, pop-up windows etc. JavaScript is directly embedded in HTML.

JavaScript was developed by Sun Microsystems and Netscape Communicator. It is an open language. It can be used without purchasing. Netscape supports JavaScript and Internet Explorer supports JScript. JScript is a subset of JavaScript.

Interpreted Programs versus Compiled Programs

An interpreted programming language is simpler to program but slower to execute. A program is interpreted line by line each time it is executed. Compiled programming language has a more complex syntax and requires more strict programming practices. A program written in compiled programming language is given to compiler to produce an executable binary program. A compiler usually generates platform specific program with **.exe** extension.

JAVA is a compiled language that is platform independent whereas JavaScript is an interpreted language. The browser provides the platform independence for JAVA through its **JAVA Virtual Machine** and the interpreter for JavaScript.

7.2.1 Advantages of JavaScript

Some advantages of using JavaScript are as follows:

1. Interpreted Language

JavaScript is an interpreted language. It requires no compilation process so no compiler is required. The browser interprets JavaScript as it interprets HTML tags.

2. Easy to Learn

The syntax of JavaScript is very easy. Any person can learn it very easily and use it to develop dynamic and interactive websites.

3. Easy to Debug and Test

JavaScript code is interpreted line by line. The errors are indicated along with line number. It is very easy to find error in the code, correct it and test it again.

4. Event-Based Programming

JavaScript is an event-based language. It means that different code segments are executed when certain event occurs. For example, a code segment may execute when the user clicks a button or moves mouse over an object etc.

5. Procedural Capabilities

JavaScript provides all capabilities of a procedural language. It provides condition checking, loops and branching facilities that can be executed in a web page.

6. Platform Independence

JavaScript is platform-independent language. Any JavaScript-enabled browser can understand and interpret JavaScript code. Any JavaScript code is executed on different types of hardware. A JavaScript program written for

7.2.2 Uses of JavaScript

Some important uses of JavaScript are as follows:

1. Input Validation

JavaScript can be used to validate the input. Data entered in forms should be validated before it is processed. For example, if a user enters an email, it must be a valid email. JavaScript code can ensure its validity by checking that it contains @ symbol and dot (.).

2. Mouse Rollover Effects

JavaScript can be used to create different buttons with interesting mouse rollover effects. When the user moves the mouse over a button, different effects are generated. It makes browsing more interesting and attractive.

3. Popup Windows

JavaScript can be used to create popup windows. These windows are normally used to display important announcements, offers and news etc. Popup windows can be created of different sizes.

4. Dynamic Contents

JavaScript can be used to generate dynamic contents in a website. For example, the current date and time can be displayed on web page. Different HTML tags can be generated based on the user input etc.

5. User Interaction

JavaScript can be used to interact with the user. The input entered by the user can be processed and proper message can be displayed to the user. The interactive capability of a website makes it more interesting and productive for the users.

7.3 Running JavaScript

JavaScript is written in an HTML file. A special tag is used to indicate that JavaScript statements have started. The browser executes these statements in different ways than simple HTML tags. It interprets JavaScript statements to generate HTML tags and then displays the result of these tags. The browser uses built-in JavaScript engine to interpret these statements.

<SCRIPT> Tag

The <SCRIPT> tag is used to inform browser about the start of JavaScript statements. The <SCRIPT> tag indicates the beginning of scripting code and </SCRIPT> tag indicates the end of scripting code.

Attributes

Some important attributes of <SCRIPT> tag are as follows:

Attribute	Description
LANGUAGE	Specifies the scripting language used for writing the code. The possible values are JavaScript or VBScript.
SRC	Specifies the JavaScript source file. This attribute is used if JavaScript is stored in another file with ".js" extension.

Example

```
<SCRIPT LANGUAGE="JavaScript">
        JavaScript statements...
</SCRIPT>
```

7.4 Data Types

JavaScript provides the following data types:

Numeric Data

Numeric data consists of numbers. It may be whole number or real number. Numeric data may consist of the following:

- Numbers (0 to 9)
- Decimal point

There are two types of numeric data in JavaScript that are as follows:

1. Integer

Integer values can be represented in decimal, hexadecimal and octal form.

- **Decimal:** Decimal numbers are from 0 to 9.
- **Hexadecimal:** Hexadecimal numbers are from 0 to 15. The numbers from 10 to 15 are represented by letters A to F.
- **Octal:** Octal numbers are from 0 to 7. These numbers have 0 as prefix.

2. Floating-Point Numbers

Floating point values are the values that require decimal point such as 25.32 or exponential notation such as 2E3 etc. These values can be positive or negative. Some examples are floating-point numbers are as follows:

- 25.33
- -12.41
- 5E-2
- .22

Logical or Boolean

Boolean data type includes **true** and **false**. It is normally used to indicate a condition. For example, check a number if it is even or odd or check if the student is pass or fail etc.

Strings

String data type includes all values that are written in single or double quotes such as "IT Series", "Pakistan", '25-A', "99" etc. String may consist of the following:

- Numbers (0 to 9)
- Alphabets (A to Z and a to z)
- Special characters such as @,), $ etc.

Mathematical operation cannot be performed on string data.

Null

A special keyword **null** indicates a null value. It is used to indicate a reference that is empty or does not exist.

7.5 Variables

Variables are symbolic names that refer to different values. Variables are created in main memory. JavaScript is case sensitive and it differentiates between upper and lower case letters. A variable name **Temp** is different from variable name **temp**. Some examples of variable names are my_age, No2 and _name.

7.5.1 Variable Declaration

Variation declaration is a statement that tells JavaScript to create a variable in memory. A variable should be declared before it is used. A script may have many variables.

Syntax

The syntax of declaring a variable is as follows:

```
var variable_name;
```

where **var** is a keyword that is used to declare a variable and **variable_name** is the name of variable to be declared.

Rules for Declaring Variables

Following are some rules for declaring variables:

- The first character of variable must be a letter or underscore _.
- Blank spaces are not allowed in variable names.
- Variable may include letters, numbers and underscore (_).
- Both upper and lower cases are allowed.
- Special symbols cannot be used as variable name.
- Reserved word cannot be used as variable name.

Examples

Some examples of variables are as follows:

Marks sub1 _test f_name

7.5.2 Variable Initialization

The process of assigning a value to a variable at the time of declaration is known as **variable initialization**. The equal sign = is used to initialize a variable. Variable name is given on left side and value is given on the right side of equal sign.

Examples

Some examples of variable initialization are as follows:

```
var n = 100;
var x = 50, y = 75;
var average = 50.73;
var grade = 'A';
var name = "Usman";
```

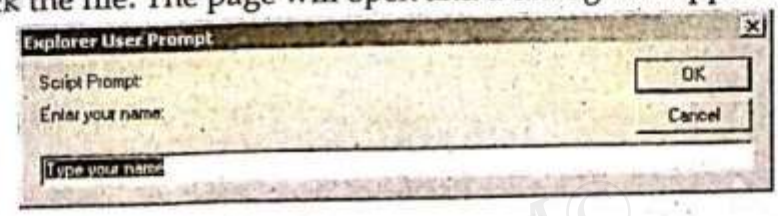
Practice 7.1

Write JavaScript code that inputs your name and displays it in the browser.

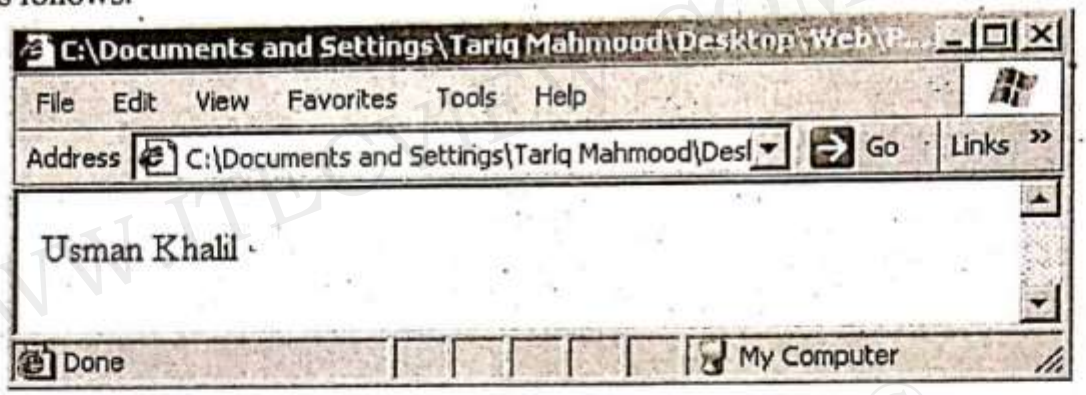
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = prompt("Enter your name:","Type your name");
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
document.write(n);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as file as "P01.html".
3. Double click the file. The page will open and a dialog will appear as follows:



4. Enter your name and click **OK** button. The entered name will appear in the browser as follows:



In the above example, **prompt** method is used to get input from the user and stores it in variable **n**. The **document.write** method writes the value of **n** in the body of the document.

7.6 Literals

Literals can be used to represent values in JavaScript. Literals are fixed values that cannot be changed during execution of the code. Different types of literals are as follows:

7.6.1 Floating-Point Literals

A floating-point literal can have the following parts:

- A decimal integer
- A decimal point (".")
- A fraction (another decimal number)
- An exponent

The exponent part is **e** or **E** followed by integer. It can be signed positive or negative.

Examples

Some examples of floating-point literals are as follows:

- 3.1415
- -3.1E12
- .1e12
- 2E-12.

7.6.2 Integer Literals

Integers can be expressed in decimal, hexadecimal and octal. A decimal integer literal consists of digits without a leading. A leading zero on an integer literal indicates that it is in octal form. Octal integers can include only the digits 0-7. A leading 0x or 0X indicates hexadecimal. Hexadecimal integers can include digits (0-9) and the letters a-f and A-F.

Examples

Some examples of integer literals are 42, 0xFFF and -345.

7.6.3 String Literals

A string literal is zero or more characters enclosed in double (") or single (') quotes.

Examples

Some examples of string literals are "Allah", 'Allah' and "1234".

Special Characters

A string literal may also contain some special characters. These characters are written along with backslash "\". It is known as **escape character.** Some important special characters are as follows:

Character	Meaning
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Tab
\'	Apostrophe or single quote
\"	Double quote
\\	Backslash character (\)

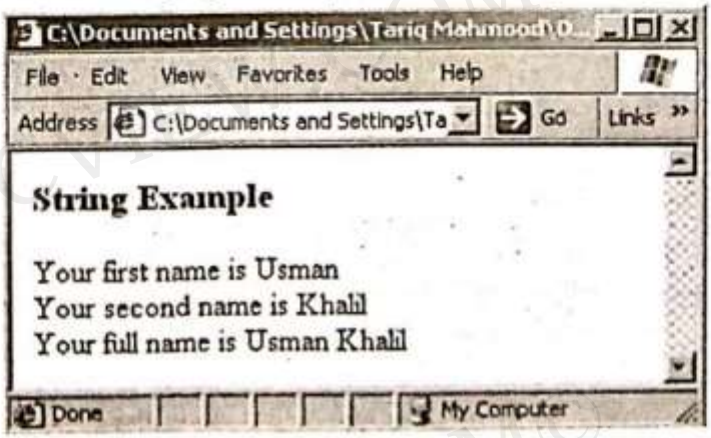
Practice 7.2

Write JavaScript code that inputs first and second name from the user. It dispalys first name in first line, second name in second line and full name in third line.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  var f, s, full;
  f = prompt("Enter your first name");
  s = prompt("Enter your second name");
  full = f + " " + s;
  document.write("<H3>String Example</H3>");
  document.write("Your first name is " + f);
  document.write("<BR>");
  document.write("Your second name is " + s);
  document.write("<BR>");
  document.write("Your full name is " + full);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as file as "String.html".
3. Double click the file. The page will open and a dialog will appear as follows:



7.7 Type Conversion

JavaScript is a dynamically typed language. It means that it is not necessary to specify the data type of a variable during declaration. The data types are converted automatically as needed during script execution. Suppose a variable is defined as follows:

```
var n = 100;
```

The variable n contains an integer value 100. In the next statement, the same variable can be assigned a string value as follows:

```
n = "Hello World";
```

In the above line, n contains a string value. JavaScript automatically converts the type of variable according to the values assigned to it.

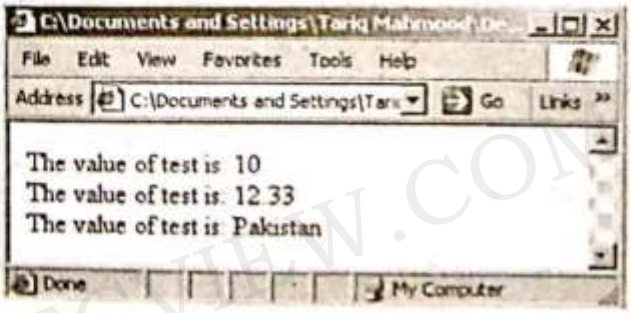
Practice 7.3

Write JavaScript code that stores different types of values in the same variable and displays these values.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  var test;
  test = 10;
  document.write("The value of test is: " + test);
  document.write("<BR>");
  test = 12.33;
  document.write("The value of test is: " + test);
  document.write("<BR>");
  test = "Pakistan";
  document.write("The value of test is: " + test);
</SCRIPT>
</HEAD>
</HTML>
```

2. Save the file as file as "Convert.html".
3. Double click the file. The page will open and a dialog will appear as follows:



7.8 Expressions

A statement that evaluates to a value is called an **expression**. An expression gives a single value. An expression consists of **operators** and **operands**. An operator is a symbol that performs some operation. Operand is the value on which the operator performs some operation. Operand can be a constant, variable or function. The evaluated value can be a number, string or a logical value.

JavaScript has the following types of expressions:

- **Arithmetic Expressions:** These expressions evaluate to a number such as 3.141.
- **String Expressions:** These expressions evaluate to a character string such as "Usman".
- **Logical Expressions:** These expressions evaluate to **true** or **false.**

7.9 Operators

An **operator** is a symbol that performs some operation. The value on which the operator performs operation is called **operand**. There are two categories of operators:

- **Binary Operator:** A type of operator that works with two operands is known as binary operator.
- **Unary Operator:** A type of operator that works with one operand is known as unary operator.

Some important types of operators in JavaScript are as follows:

- Assignment Operators
- Comparison Operators
- Arithmetic Operators
- Logical Operators
- String Operators

7.9.1 Assignment Operators

An assignment operator used to assign the value of right operand to left operand. A statement that assigns a value to a variable is known as **assignment statement**. The basic assignment operator is equal (=). The name of variable is written on the left side of the assignment operator and the value is written on the right side of assignment operator. The value can be a literal, variable, expression or function. For example, the expression **x = y** assigns the value of variable **y** to variable x.

An assignment statement that assigns a value to many variables is known as **compound assignment** statement. The assignment operator = is used in this statement.

Examples

```
A = B;
X = 100;
x = y = z = 100;
m = n = 50;
```

Compound Assignment

JavaScript provides **compound assignment operators** that combine assignment operator with arithmetic operators. Compound assignment operators are used to perform mathematical operations more easily. These operators are shorthand for standard operations as shown in the following table.

Operator	Meaning
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y

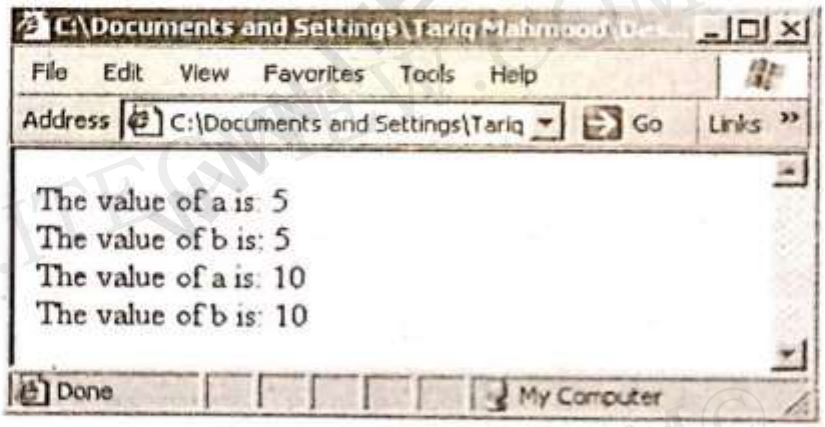
Practice 7.4

Write JavaScript code that inputs two numbers in variables from the user and displays them. It then doubles the values of these variables and displays these values again.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  var a, b;
  a = b = 0;
  a = prompt("Enter a number");
  b = a;
  document.write("The value of a is: " + a);
  document.write("<BR>");
  document.write("The value of b is: " + b);
  document.write("<BR>");
  a*=2;
  b*=2;
  document.write("The value of a is: " + a);
  document.write("<BR>");
  document.write("The value of b is: " + b);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as file as "Assign.html".
3. Double click the file. The page will open and a dialog will appear as follows:



7.9.2 Arithmetic Operators

Arithmetic operators process numerical values and return a single numerical value. The standard arithmetic operators are addition (+), subtraction (-), multiplication (*) and division (/). JavaScript provides the following arithmetic operators:

Operator	Description	Example
+	It is a binary operator. It adds two numbers and returns the result.	5 + 3 returns 8.
-	It is a binary operator. It subtracts two numbers and returns the result.	5 – 3 returns 2.
*	It is a binary operator. It multiplies two numbers and returns the result.	10 * 2 returns 20.

/	It is a binary operator. It divides left operand by right operand and returns the result.	10 / 2 returns 5.
%	Modulus operator is a binary operator. It returns integer remainder of dividing the two operands.	12 % 5 returns 2.
++	It is a unary operator. It adds 1 to operand.	If x is 3, then ++x sets x to 4 and returns 4.
--	It is a unary operator. It subtracts 1 from operand.	If x is 3, then --x sets x to 2 and returns 2.

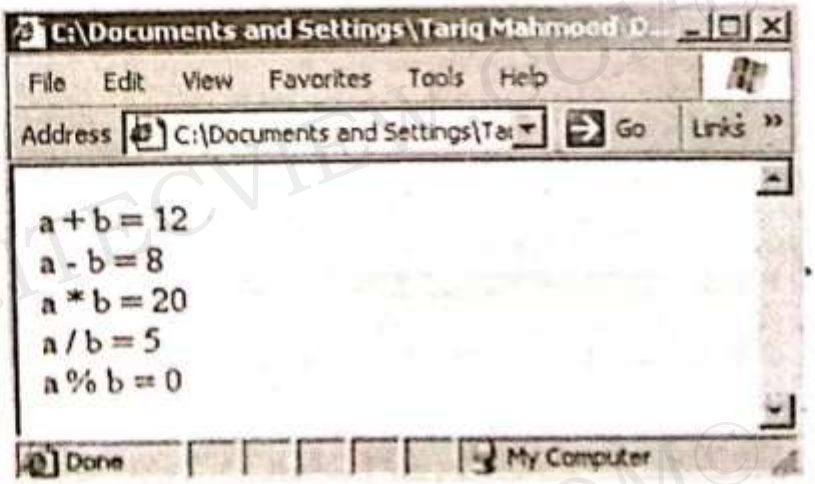
Practice 7.5

Write JavaScript code that stores two numbers in two variables. It performs all arithmatic operations on these variables and displays the results.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  var a, b;
  a = 10;
  b = 2;
  document.write("a + b = " + (a+b));
  document.write("<BR>");
  document.write("a - b = " + (a-b));
  document.write("<BR>");
  document.write("a * b = " + (a*b));
  document.write("<BR>");
  document.write("a / b = " + (a/b));
  document.write("<BR>");
  document.write("a % b = " + (a%b));
  document.write("<BR>");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Arithmatic.html".
3. Double click the file. The page will appear in browser with the following dialog box:



Increment and Decrement Operators

The increment operator (++) and decrement operator (– –) can be used in two ways i.e. before the operand and after the operand. If it is used before the operand such as ++x, it is called **prefix notation.** If it is used after the operand such as x++, it is called **postfix notation.** Prefix notation increments the operand by 1 and then returns the value of operand. Postfix notation returns the value of operand and then increments the value of operand by 1.

Example

```
A = 5;
B = ++A;
C = A++;
```

The first line assigns 5 to variable **A**. The second line increments the value of **A** by 1 and then assigns its value to **B**. The third line assigns the value of **A** to **C** and then increments its value by 1. After executing all statements, A contains 7, B contains 6 and C contains 6.

Practice 7.6

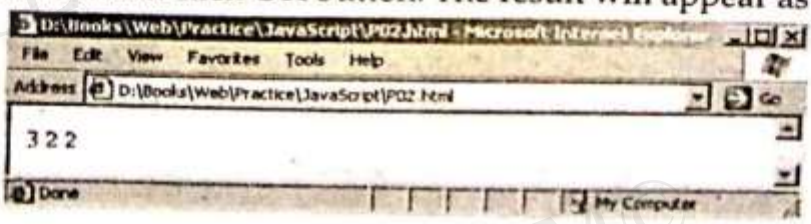
Write JavaScript code that inputs a number in a variable from the user, performs prefix and postfix increment operator on this variable and dipslays the result.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
var n, a, b;
n = prompt("Enter a number:","1");
a = ++n;
b = n++;
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
document.writeln(n);
document.writeln(a);
document.write(b);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P02.html".
3. Double click the file. The page will appear in browser with the following dialog box:

4. Enter any number and click **OK** button. The result will appear as follows:



7.9.3 Operator Precedence

The order in which different types of operators in an expression are evaluated is known as **operator precedence**. It is also known as **hierarchy of operators**.

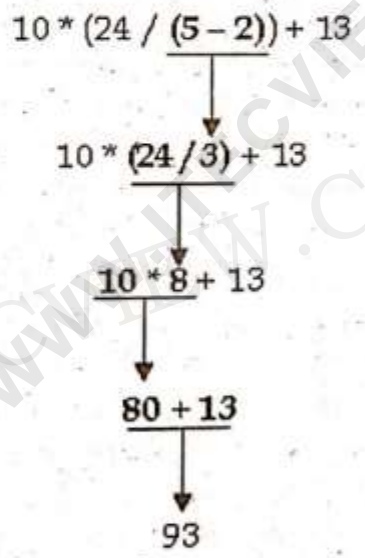
Each operator has its own precedence level. If an expression contains different types of operators, the operators with higher precedence are evaluated before the operators with lower precedence. The order of precedence in JavaScript is as follows:

- Any expression given in parentheses is evaluated first.
- Then multiplication * and division / operators are evaluated.
- Then plus + and minus - operators are evaluated.
- In case of parentheses within parentheses, the expression of the inner parentheses will be evaluated first.

Example

The expression **10 * (24 / (5 – 2)) + 13** is evaluated in the following order:

- First of all, the expression **5–2** will be evaluated. It gives a value **3**.
- Secondly, **24** will be divided by the result of last line i.e. **24 / 3** giving value **8**.
- Thirdly, **10** will be multiplied by **8** i.e. giving a result **80**.
- Finally, **80** will be added in **13** and the last result will be **93**.



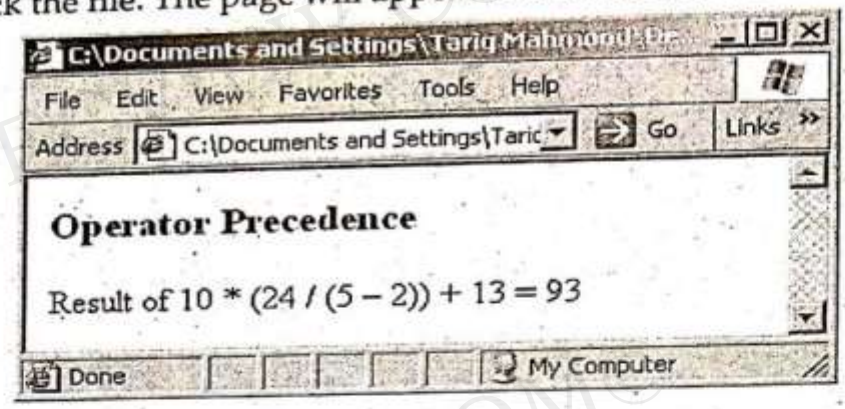
Practice 7.7

Write JavaScript code that displays the result of **10 * (24 / (5 – 2)) + 13**

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
 var n;
 document.write("<H3>Operator Precedence</H3>");
 n = 10*(24/(5-2))+13;
 document.write("Result of 10 * (24 / (5 – 2)) + 13 = " + n);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Prece.html".
3. Double click the file. The page will appear in browser with the following dialog box:



7.9.4 Operator Associativity

The order in which operators of same precedence are evaluated is known as **operator associativity**. If an expression contains some operators that have same precedence level, the expression is evaluated either from **left-to-right** or **right-to-left**.

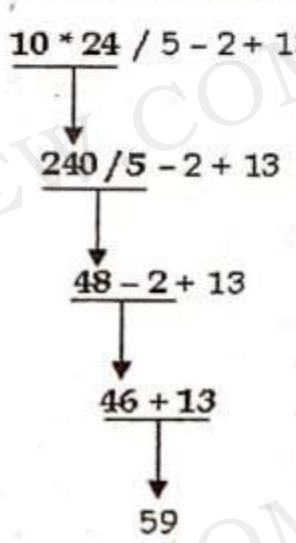
Operator associativity in JavaScript is as follows:

Operators	Associativity
() ++ (postfix) - - (postfix)	Left-to-right
+ (unary) – (unary) ++ (prefix) – – (prefix)	Left-to-right
* / %	Left-to-right
+ –	Left-to-right
= += –= *= /=	Right-to-left

Example

The expression 10 * 24 / 5 – 2 + 13 contains four operators. Two operators * and / have equal precedence. These expressions is evaluated from left to right. Similarly, two operators + and - also have equal precedence. So the expression will be evaluated in the following order:

- First of all, the expression 10 * 24 is evaluated. It gives a value **240**.
- Secondly, **240** will be divided by **5** i.e. 240 / 5 giving a value **48**.
- Thirdly, **2** will be subtracted from **48** i.e. 48 – 2 giving a result **46**.
- Finally, **46** will be added in **13** and the last result will be **59**.



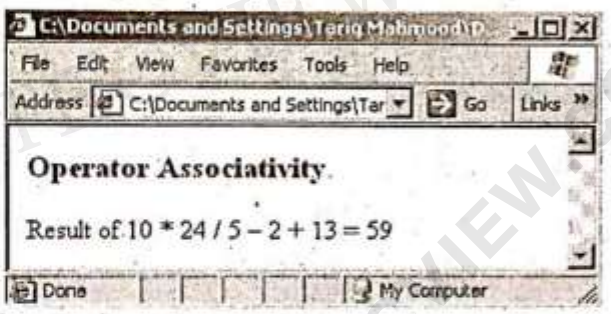
Practice 7.8

Write JavaScript code that displays the result of **10 * 24 / 5 – 2 + 13**

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
 var n;
 document.write("<H3>Operator Precedence</H3>");
 n = 10*24/5-2+13;
 document.write("Result of 10 * 24 / 5 – 2 + 13 = " + n);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Assoc.html".
3. Double click the file. The page will appear in browser with the following dialog box:



7.10 Type Casting

The process of converting the data type of a value during execution is known as **type casting**. Type casting can be performed in two ways:

- Implicit type casting
- Explicit type casting

7.10.1 Implicit Type Casting

Implicit type casting is performed automatically by JavaScript. The operands in arithmetic operation must be of similar types. If the data types of operands are different, the value with lower data type is converted into higher data type.

Example

Suppose **x = "Hello"** and **y = 10** and the following expression is evaluated:

```
x + y
```

The above expression will return a string value "Hello10". The value of **x** will be converted into string during the evaluation of expression.

7.10.2 Explicit Casting

Explicit casting is performed by programmer. It is performed by using built-in function such as parseInt.

Example

Suppose **x = "10"** and **y = "20"**. The expression **x + y** will return 1020. The expression **parseInt(x) + parseInt(y)** will return 30.

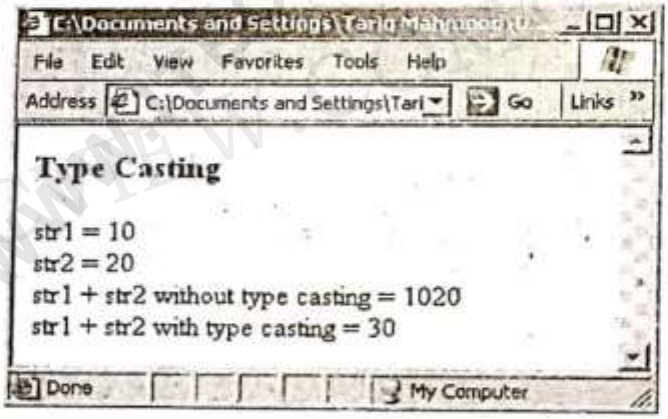
Practice 7.9

Write JavaScript code that stores "10" and "20" in two variables. It adds these values without type casting and with type casting and then displays the results.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
 var str1, str2, res;
 str1 = "10";
 str2 = "20";
 document.write("<H3>Type Casting</H3>");
 document.write("str1 = " + str1);
 document.write("<BR>");
 document.write("str2 = " + str2);
 document.write("<BR>");
 res = str1 + str2;
 document.write("str1 + str2 without type casting = " + res);
 document.write("<BR>");
 res = parseInt(str1) + parseInt(str2);
 document.write("str1 + str2 with type casting = " + res);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Casting.html".
3. Double click the file. The page will appear in browser with the following dialog box:



Exercise Questions

1. Write JavaScript code that inputs temperature in degrees Celsius from the user. It converts the temperature to Fahrenheit using the formula **((9/5) * Temp) + 32** and then displays the temperature in Fahrenheit.
2. Write JavaScript code that inputs base and height from the user and calculates the area of a triangle by using the formula **Area = ½ * Base * Height**.
3. Write JavaScript code that inputs your age in years and displays the age in days and months.
4. Write JavaScript code that inputs radius from the user and calculates area and circumference of circle using Area = πR^2 and Circumference = $2\pi R$.

Conditions and Loops

8.1 Conditional Statements

A statement or set of statements that is executed when a particular condition is **true** and ignored when the condition is **false** is called **conditional statement**. JavaScript supports two conditional statements: **if...else** and **switch**.

Suppose a script displays **Pass** if the student gets 40 or more than 40 marks. It displays **Fail** when the marks are below 40. The program checks the marks before displaying the message. This process is known as **decision-making** or **selection**. The condtional statements use relational operators for evaluating a condition.

8.1.1 Relational Operators

A comparison operator compares its operands and returns a logical value based on whether the comparison is **true** or **false**. The operands can be numerical or string values. The following table describes the comparison operators.

Operator	Description
>	**Greater than operator** returns **true** if the value on left side of > is greater than the value on the right side. Otherwise returns **false**.
<	**Less than operator** returns **true** if the value on left side of < is less than the value on right side. Otherwise returns **false**.
==	**Equal to operator** returns **true** if the values on both sides of = are equal. Otherwise returns **false**.
>=	**Greater than or equal to operator** returns **true** if value on left side of >= is greater than or equal to the value on right side. Otherwise returns **false**.
<=	**Less than or equal to operator** returns **true** if the value on left side of <= is less than or equal to the value on right side. Otherwise returns **false**.
!=	The not equal to operator. Returns **true** if the value on the left side of <> is not equal to the value on the right. Otherwise returns **false**.

8.1.2 Relational Expression

A relational expression is a statement that uses relational operators to compare two values. The result of a relational expression can be **true** or **false**. Both sides of a relational expression can be a **literal, variable** or **arithmetic expression**.

Examples

Suppose **x = 5** and **y = 10**.

Relational Expression	Result
x == y	False
x != y	True
x > y	False
x >= y	False
x < y	True
x <= y	True

8.2 if Statement

if statement is a decision-making statement. It is used to execute or skip a statement or set of statements by checking a **condition**.

The condition is given as a relational expression. If the condition is **true**, the statement or set of statements after **if statement** is executed. If the condition is **false**, the statement or set of statements after **if statement** is not executed.

Syntax

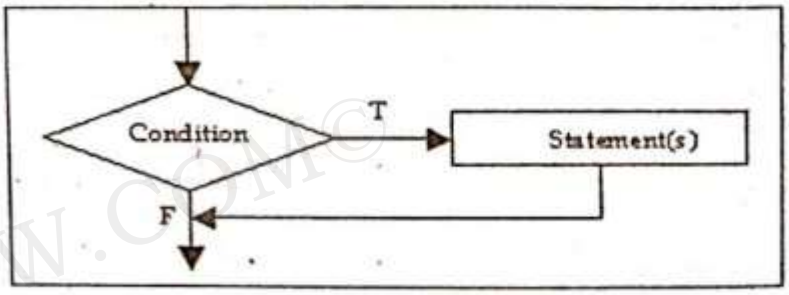
The syntax of if statement is as follows:

```
if (condition)
        statement;
```

The above syntax is used for single statement. A set of statements can also be made conditional. In this case, these statements are written in curly brackets { }. The set of statements is also called **compound statements**. The syntax for compound statements in **if statement** is as follows:

```
if (condition)
{
        statement 1;
        statement 2;
          :
        statement N;
}
```

Flowchart



Practice 8.1

Write JavaScript code that inputs a number from the user. It displays congratulation message if the number is 40 or above.

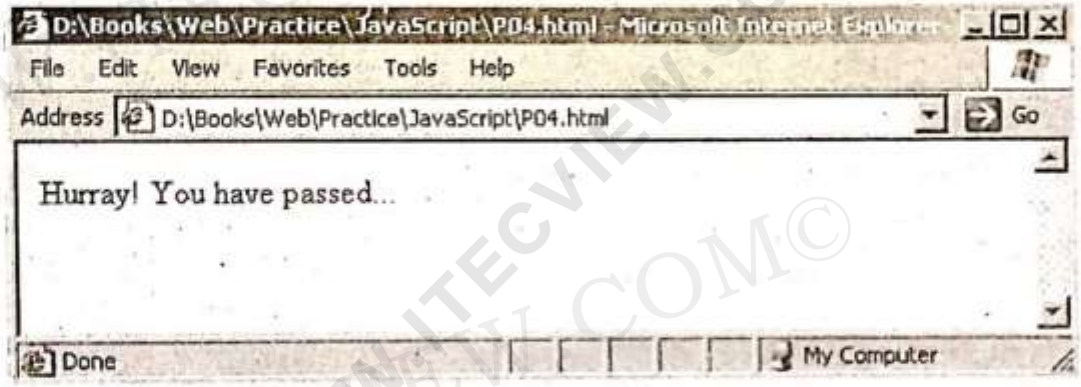
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
```

```
var num;
num = prompt("Enter a number:","1");
if(num >= 40)
document.write("Hurray! You have passed...");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P04.html".
3. Double click the file. The page will appear with the following dialog box:

4. Enter any number and click OK button. The result will appear as follows:



Limitation of "if" Statement

if statement is the simplest selection structure but it is very limited in its use. The statement or set of statements is executed if the condition is **true**. But if the condition is **false** then nothing happens.

For example, a script should display **"Pass"** if the student gets 40 or more marks. It should display **"Fail"** if the student gets less then 40 marks. Simple **if** statement cannot be used to handle this situation.

8.3 if...else Statement

if else statement is another type of **if** statement. It executes one block of statement(s) when the **condition** is **true** and the other when it is **false**. In any situation, one block is executed and the other is skipped. In **if else** statement, both blocks of statement can never be executed and both blocks of statements can never be skipped.

Syntax

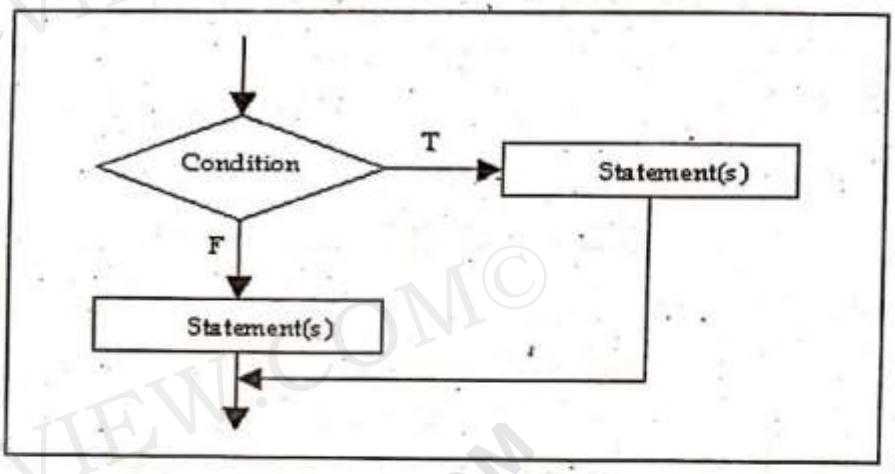
Its syntax is as follows:

```
if (condition)
        statement;
else
        statement;
```

Two or more statements are written in curly brackets { }. The syntax for compound statements in **if else** statement is as follows:

```
if (condition)
{
        statement 1;
        statement 2;
:
:
        statement N;
}
else
{
        statement 1;
        statement 2;
:
:
        statement N;
}
```

Flowchart



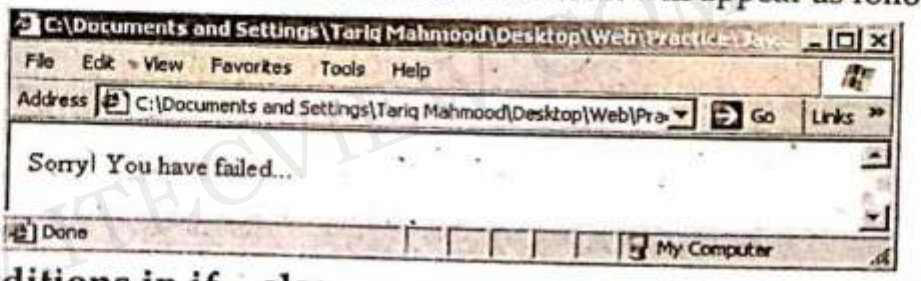
Practice 8.2

Write JavaScript code that inputs a number from the user. It displays congratulation message if the number is 40 or above. It displays sorry message otherwise.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var num;
num = prompt("Enter a number:","1");
if(num >= 40)
  document.write("Hurray! You have passed...");
else
  document.write("Sorry! You have failed...");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P04a.html".
3. Double click the file. The page will appear with a dialog box:
4. Enter any number and click **OK** button. The result will appear as follows:



Multiple Conditions in if...else

if-else statement can be used to choose one block of statements from many blocks of statements. This type of **if-else** is used when there are many options and only one block of statements should be executed on the basis of a condition.

Syntax

The syntax of this structure is:

```
if (condition)
{
        block 1;
}
else if (condition)
{
        block 2;
}
else if (condition)
{
        block 3;
}
.
.
.
else
{
        block N;
}
```

Practice 8.3

Write JavaScript code that inputs marks from the user. It displays the grade according to the following criteria:

80 or above	Grade A
60 – 79	Grade B
40 – 59	Grade C
below 40	Grade F

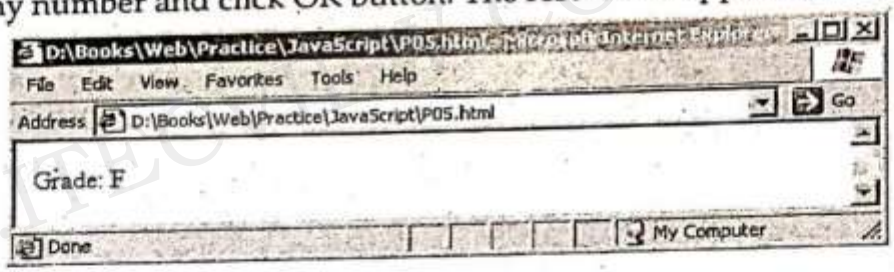
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var marks;
marks = prompt("Enter marks:","0");
if(marks >= 80)
  document.write("Grade: A");
else if(marks >= 60)
  document.write("Grade: B");
else if(marks >= 40)
  document.write("Grade: C");
else
  document.write("Grade: F");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P05.html".

WWW.ITECVIEW.COM©

3. Double click the file. The page will appear with a dialog box.
4. Enter any number and click OK button. The result will appear as follows:

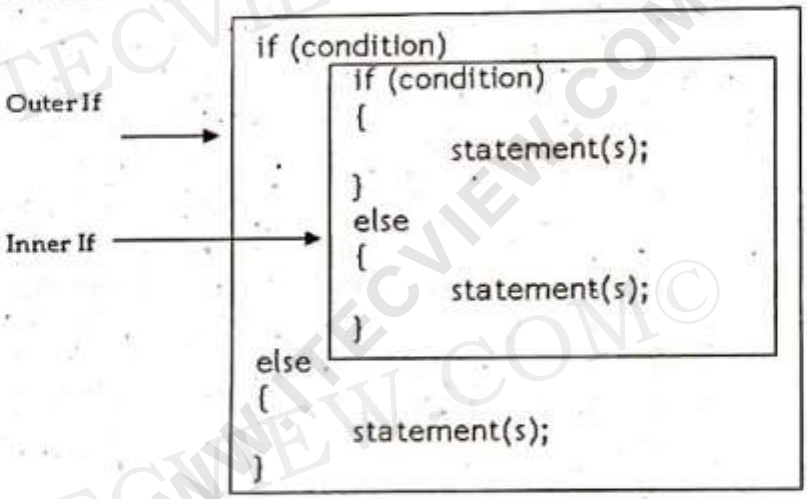


8.4 Nested "if" Statement

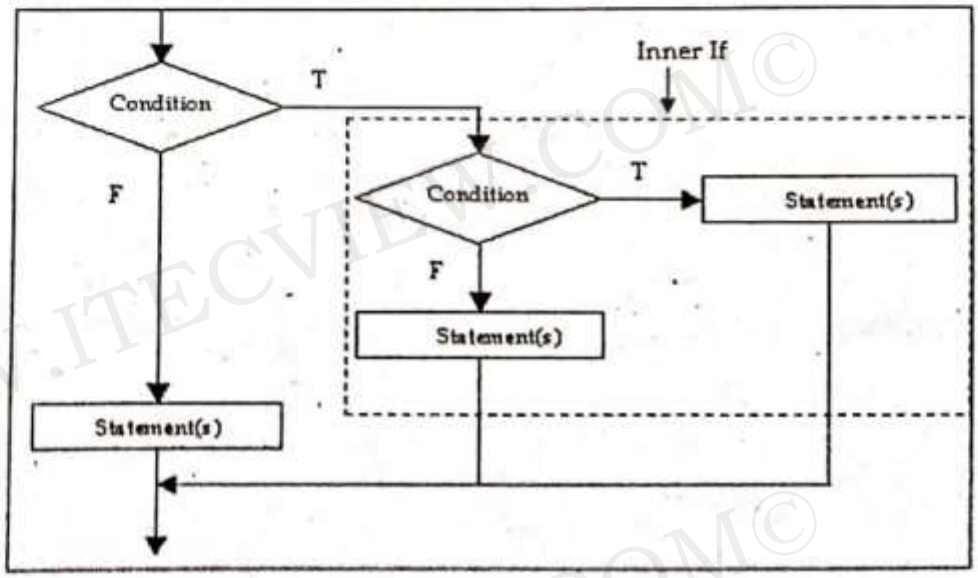
An if statement within an if statement is called **nested if** statement. In nested structure, the control enters into the **inner if** only when the outer condition is **true**. Only one block of statements are executed and the remaining blocks are skipped automatically.

Syntax

The syntax of nested If is as follows:



In **nested if** statement, the condition of outer if is evaluated first. If it is **true**, the control enters in the **inner if** block. If the condition is **false**, the **inner if** is skipped and control directly moves to the **else** part of **outer if**. If **outer if** is **true** then control enters in the **inner if** statement. The **inner if** evaluated according to **simple if** statement.



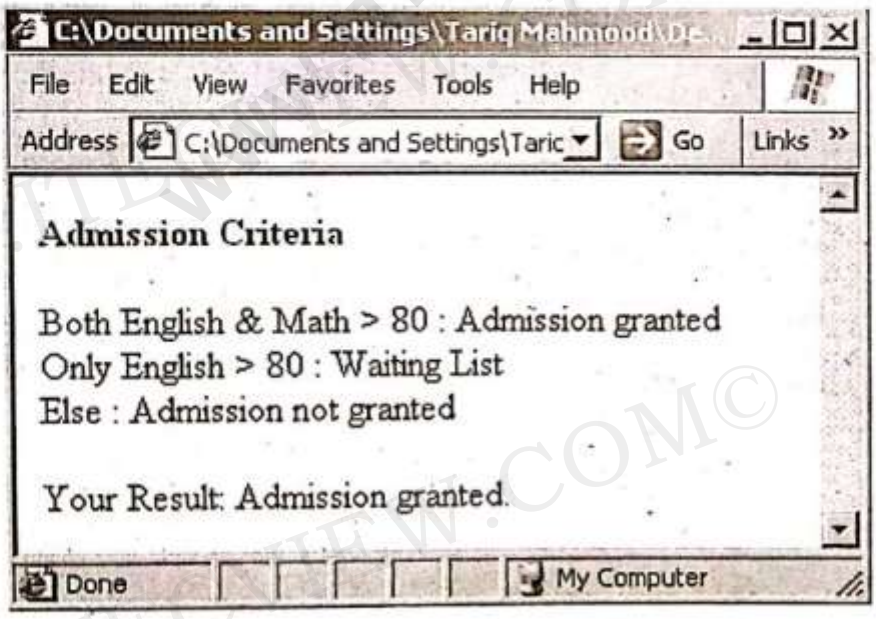
Practice 8.4

Write JavaScript code that inputs marks of English and Math. It displays "Admission granted" if both marks are above 80, displays "Waiting list" if only English marks are above 80 and displays "Admission not granted" otherwise using nested if.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var eng, math;
eng = prompt("Enter marks of English","0");
math = prompt("Enter marks of math","0");
if(eng>80)
  if(math>80)
    document.write("Admission granted.");
  else
    document.write("Waiting list.");
else
  document.write("Admission not granted.");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "NestIf.html".
3. Double click the file. Dialog box will appear twice to get marks of English and Math.
4. Enter marks and click OK button. The result will appear as follows:



8.5 Logical Operators

Logical operators are used to evaluate compound conditions. A type of comparison in which more than one conditions are evaluated is called **compound condition**. It is used to execute a statement or set of statements by testing many conditions. The logical operators in JavaScript are AND, OR and NOT.

1. AND Operator (&&)

The symbol used for **AND** operator is **(&&)**. It is used to evaluate two conditions. It returns **true** result if both conditions are **true**. It returns **false** result if any condition is **false**.

Condition 1	Operator	Condition 2	Result
False	&&	False	False
False	&&	True	False
True	&&	False	False
True	&&	True	True

Example

Suppose A = 100 and B = 50. The compound condition **(A>10) && (B>10)** is **true**. It contains two conditions and both are **true**. So the whole compound condition is also **true**.

The compound condition **(A>50) && (B>50)** is **false**. It contains two conditions. One condition **(A>50)** is **true** and second condition **(B>50)** is **false**. So the whole compound condition is **false**.

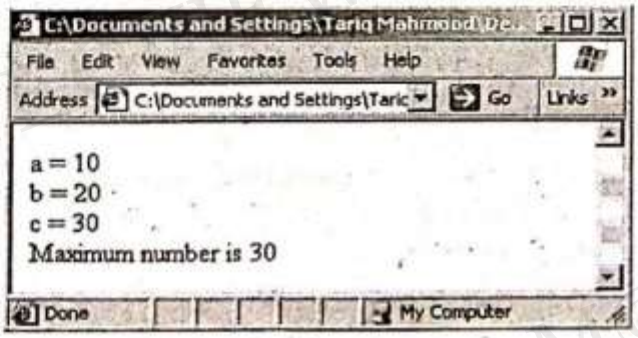
Practice 8.5

Write JavaScript code that inputs three numbers from the user and displays the maximum number by using AND operator.

1. Open Notepad and type the following code:

```
<HTML><BODY>
<SCRIPT LANGUAGE="JavaScript">
var a, b, c;
a = prompt("Enter a number","0");
b = prompt("Enter a number","0");
c = prompt("Enter a number","0");
document.write("a = " + a);
document.write("<BR>b = " + b);
document.write("<BR>c = " + c);
document.write("<BR>");
if(a>b && a>c)
  document.write("Maximum number is " + a);
else if(b>a && b>c)
  document.write("Maximum number is " + b);
else
  document.write("Maximum number is " + c);
</SCRIPT>
</BODY></HTML>
```

2. Save the file as "AND.html".
3. Double click the file and input three numbers. The result will appear as follows:



2. OR Operator (| |)

The symbol used for **OR** operator is (| |). It is used to evaluate two conditions. It returns **true** result if either condition is **true**. It returns **false** if both conditions are **false**.

Condition 1	Operator	Condition 2	Result
False	||	False	False
False	||	True	True
True	||	False	True
True	||	True	True

Example

Suppose A = 100 and B = 50. The compound condition **(A>50) || (B>50)** is **true**. It contains two conditions and one condition **(A>50)** is **true**. So the whole compound condition is also **true**. On the other hand, the compound condition **(A>500) || (B>500)** is **false** because both conditions are **false**.

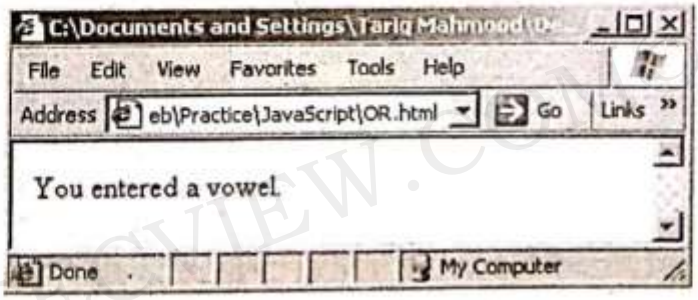
Practice 8.6

Write JavaScript code that inputs a character from the user and displays whether it is a vowel or not using OR operator.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var c;
c = prompt("Enter a character","a");
if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
  document.write("You entered a vowel.");
else
  document.write("You entered a consonant.");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "OR.html".
3. Double click the file and input a character. The result will appear as follows:



3. NOT Operator (!)

The symbol used for **NOT** operator is (!). It is used to reverse the result of a condition. It produces **true** result if the condition is **false**. It produces **false** result if the condition is **true**.

Operator	Condition	Result
!	True	False
!	False	True

Example

Suppose we have two variables A = 100 and B = 50. The condition !(A==B) is true. The result of (A==B) is false but NOT operator converts it into true. The condition !(A>B) is false. The condition (A>B) is true but NOT operator converts it into false.

Practice 8.7

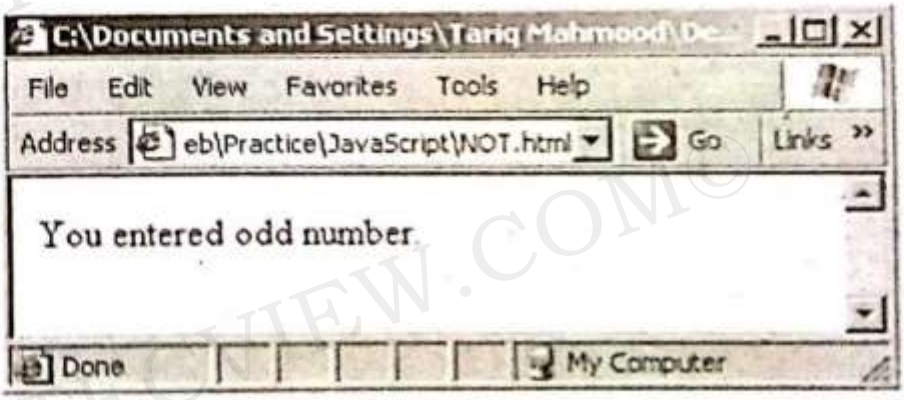
Write JavaScript code that inputs a number from the user and displays whether it is odd or even using NOT operator.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = prompt("Enter a number","1");
if(!(n%2==0))
  document.write("You entered odd number.");
else
  document.write("You entered even number.");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "NOT.html".
3. Double click the file. The following message will appear:

4. Enter 3 and click **OK** button. The result will appear as follows:



8.6 switch Statement

The **switch** statement is another conditional structure. It is a good alternative of **nested if-else**. It can be used easily when there are many choices available and only one should be executed. Nested if becomes very difficult in such situation.

Syntax

The syntax for writing this structure is as follows:

```
switch (expression)        → Integer or character variable
{
  case constant 1:         → Integer or character constant
    statement(s);
    break;                 → First case body → Causes exit from case body
  case constant 2:
    statement(s);
    break;                 → Second case body
      :
      :
  case constant N:
    statement(s);
    break;                 → N case body
  default:
    statement(s);          → Default body
}
```

Figure: Syntax of switch statement

The **default** block appears at the end of all cases. It is executed only when the result of **expression** does not match with any case. Its use is optional.

The **break** statement in each case block is used to exit from **switch** body. It is used at the end of each case block. When the result of the expression matches with a case, the corresponding statements are executed. The **break** statement comes after these statements and the control exits from **switch** body. If **break** is not used, all case blocks that come after the matching case, will also be executed.

Practice 8.8

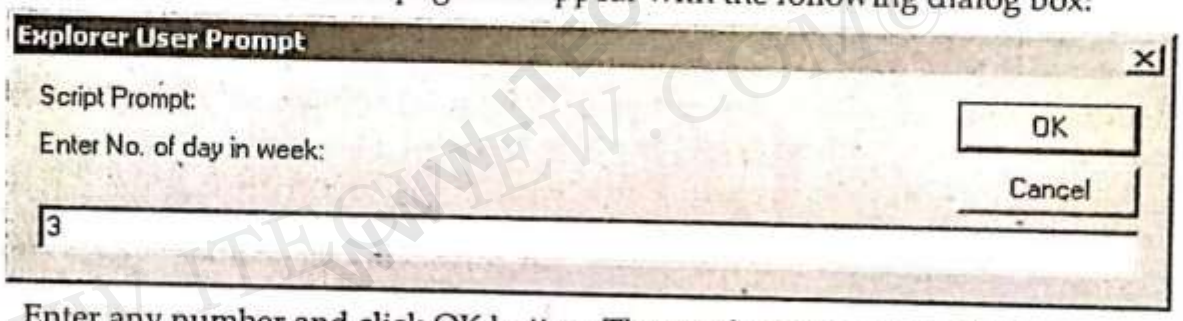
Write JavaScript code that inputs the day of week as number from the user and displays the name of the day starting from Friday. For example if the user enters 3, it displays Sunday.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = parseInt(prompt("Enter No. of day in week:","1"));
switch(n)
{
  case 1:
    document.write("Friday");
    break;
```

```
  case 2:
    document.write("Saturday");
    break;
  case 3:
    document.write("Sunday");
    break;
  case 4:
    document.write("Monday");
    break;
  case 5:
    document.write("Tuesday");
    break;
  case 6:
    document.write("Wednesday");
    break;
  case 7:
    document.write("Thursday");
    break;
  default:
    document.write("Invalid day");
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P06.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter any number and click OK button. The result will appear as follows:

8.7 Conditional Operator

The conditional operator is the only JavaScript operator that takes three operands. It is also called ternary operator. The operator can have one of two values based on a condition.

Syntax

The syntax is as follows:

```
condition ? n1 : n2;
```

If the **condition** is **true**, the expression will use the value of **n1**. If the condition is **false**, the expression will use the value of **n2**. n1 and n2 may be a variable, literal, arithmetic expression or simple input or output statement.

Example

```
result = (marks >= 40 ? "pass" : "fail");
```

The statement assigns the value "pass" to the variable **result** if the value of marks is 40 or above. Otherwise, it assigns the value "fail" to **result**.

Practice 8.9

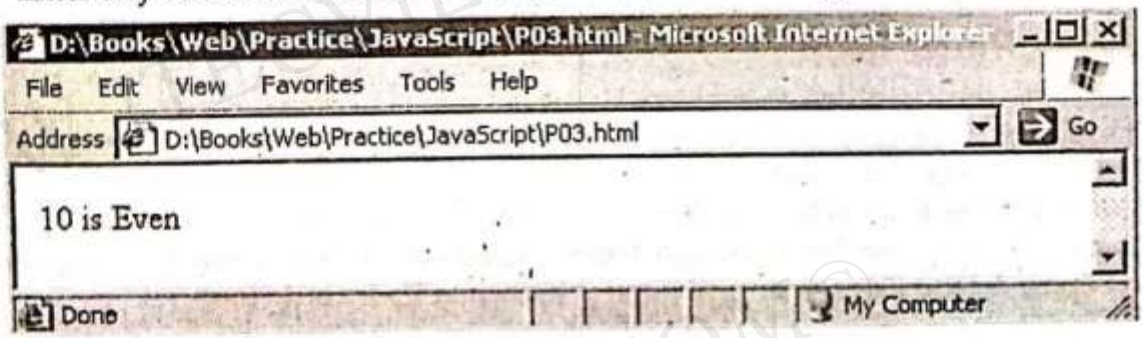
Write JavaScript code that inputs a number from the user and displays whether it is even or odd using conditional operator.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
var n, r;
n = prompt("Enter a number:","1");
r = (n%2 == 0 ? "Even" : "Odd");
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
document.write(n + " is " + r);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P03.html".
3. Double click the file. The page will appear with the following dialog box:

4. Enter any number and click OK button. The result will appear as follows:



8.8 Loops

A statement or a set of statements that is executed repeatedly is known as **loop**. The structure that repeats a statement(s) is known as **iterative, repetitive** or **looping structure.** Loops are basically used for two purposes:

- To execute a statement or number of statements for a specified number of times. For example, a user may display his name on screen for 10 times.
- To use a sequence of values. For example, a user may display a set of natural numbers from 1 to 10.

There are three types of loops available in JavaScript:

- while loop
- do-while loop
- for loop

8.9 while Loop

while loop is the simplest loop in JavaScript. This loop executes one or more statements while the given condition remains **true**.

Syntax

The syntax of **while** loop is as follows:

```
while (condition)
    statement;
```

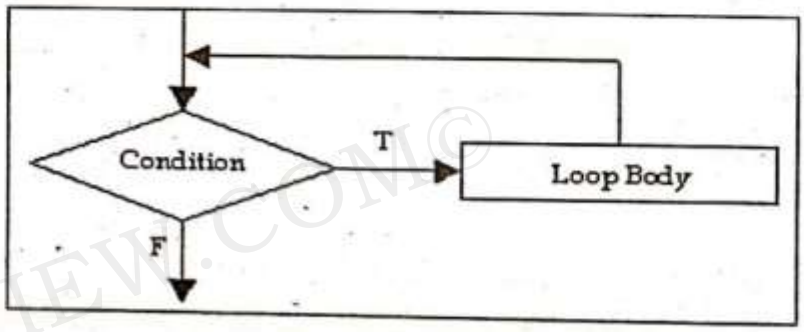
Condition: The condition is given as a relational expression. The statement is executed only if the given condition is **true**. If the condition is **false**, the statement is never executed.

Statement: Statement is the instruction that is executed when the condition is **true**. If two or more statements are used, these are given in braces { }. It is called the **body** of the loop.

The syntax for compound statements is as follows:

```
while (condition)
{
    statement 1;
    statement 2;
:
:
    statement N;
}
```

Flowchart



Working of "while" Loop

First of all, the condition is evaluated. If it is **true**, the control enters the body of the loop and executes all statements in the body. After executing the statements, it again moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains **true**. When the condition becomes **false**, the loop is terminated. **while** loop terminates only when the condition becomes **false**. If the condition remains **true**, the loop never ends. A loop that has no end point is known as an **infinite loop**.

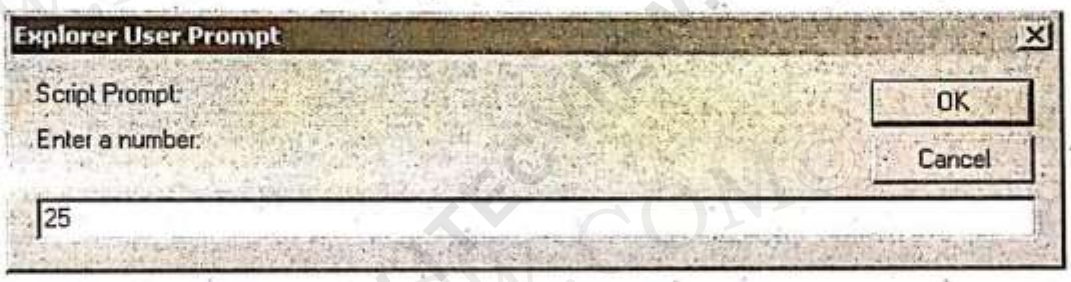
Practice 8.10

Write JavaScript code that displays numbers from 1 to 25 using **while** loop.

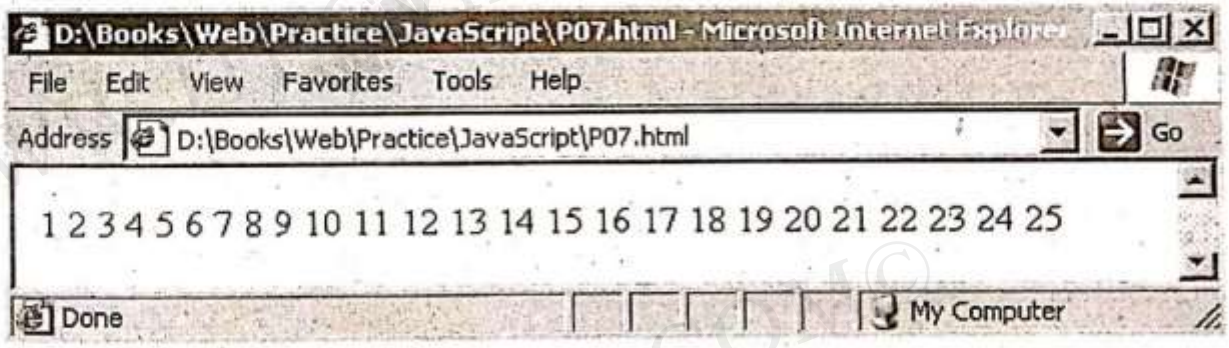
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n, i;
n = prompt("Enter a number:","10");
i = 1;
while(i<=n)
{
   document.writeln(i);
   i = i + 1;
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P07.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter any number and click OK button. The result will appear as follows:



Practice 8.11

Write JavaScript code that inputs a number from the user and displays it using **while** loop. The loop terminates if the user enters –1.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = 1;
```

```
while(n!=-1)
{
  n = prompt("Enter a number:","1");
  document.writeln("You entered " + n + "<BR>");
}
document.writeln("Bye...");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P07a.html".
3. Double click the file. A message will appear to get input. Enter different numbers and then -1 to end the loop. The result will appear as follows:

8.10 do-while Loop

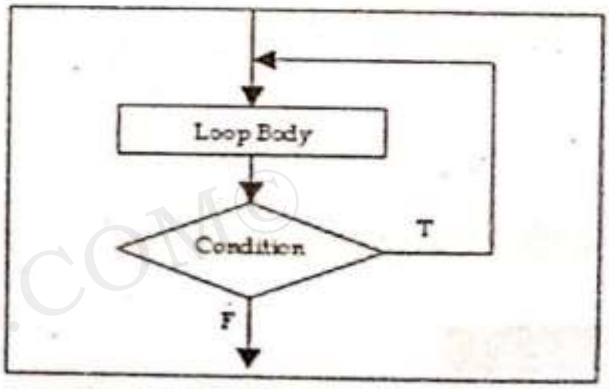
do-while is similar to **while** loop. The only difference is that the condition is checked after executing the body of the loop. The condition comes after the body of the loop. It means that the body of **do-while** loop executes at least once.

Syntax

The syntax of while loop is as follows:

```
do
{
        statement 1;
        statement 2;
:
:
        statement N;
}
while (condition);
```

Flowchart



do: It is the keyword that indicate the beginning of the loop.

Condition: The condition is given as a relational expression. The loop continues only if the given condition is **true**. If the condition is **false**, the loop is terminated.

Statement: Statement is the instruction that is executed when the condition is **true**. If two or more statements are used, these are written in braces {}. It is called the **body** of the loop.

Working of "do-while" Loop

First of all, the body of loop is executed. After executing the statements in loop body, the condition is evaluated. If it is **true**, the control again enters the body of the loop and executes all statements in the body again. This process continues as long as the condition remains **true**. The loop terminates when the condition becomes **false**. This loop is executed at least once even if the condition is **false** in the beginning.

Practice 8.12

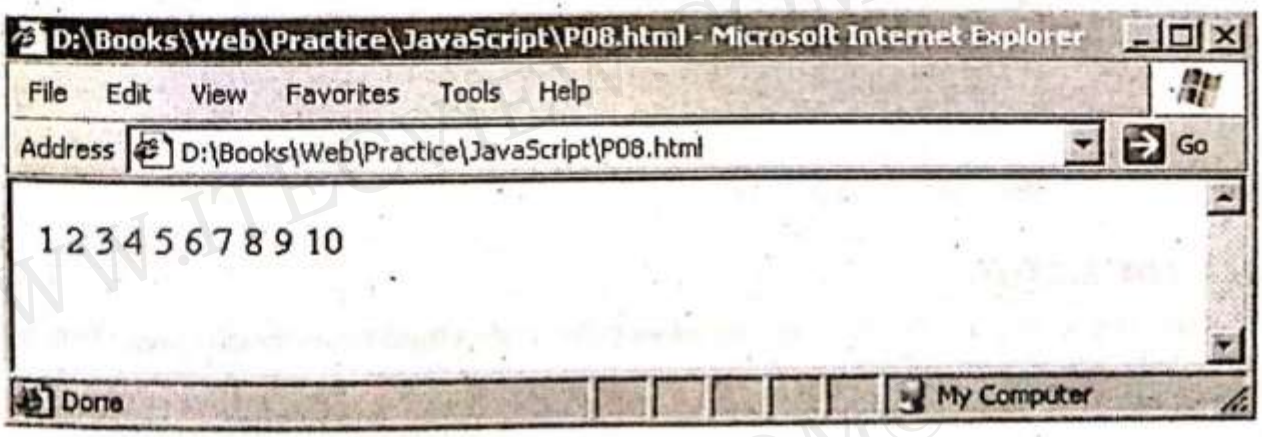
Write JavaScript code that displays numbers from 1 to 10 using **do-while** loop.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n, i;
n = prompt("Enter a number:","10");
i = 1;
do
{
   document.writeln(i);
   i = i + 1;
}
while(i<=n);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P08.html".
3. Double click the file. The page will appear with the following dialog box:

4. Enter any number and click OK button. The result will appear as follows:



Practice 8.13

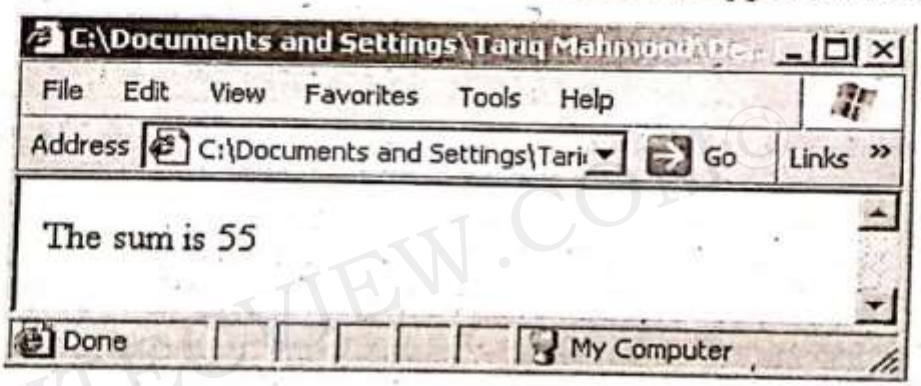
Write JavaScript code that inputs a number from the user and displays the sum of all numbers from 1 to that number using **do-while** loop.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n, i, s;
n = prompt("Enter a number:","10");
i = 1;
s = 0;
do
{
   s = s + i;
   i = i + 1;
}
while(i<=n);
document.write("The sum is " + s);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P08a.html".
3. Double click the file. The page will appear with the following dialog box:

4. Enter any number and click OK button. The result will appear as follows:



8.11 for Loop

for loop executes one or more statements for a specified number of times. This loop is also called **counter-controlled loop**.

Syntax

The syntax of this loop is as follows:

```
for (initialization; condition; increment/decrement)
{
        statement 1;
        statement 2;
:
:
        statement N;
}
```

Initialization: It specifies the starting value of counter variable. One or many variables can be initialized in this part. To initialize many variables, each variable is separated by comma.

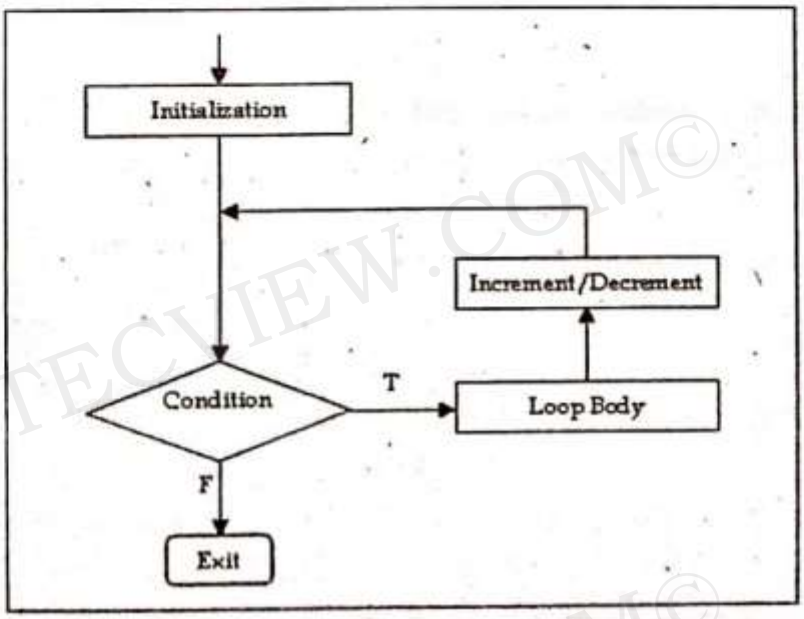
Condition: The condition is given as a relational expression. The statement is executed only if the given condition is **true**. If the condition is **false**, the statement is never executed.

Increment/decrement: This part of loop specifies the change in counter variable after each execution of the loop. To change many variables, each variable must be separated by comma.

Statement: Statement is the instruction that is executed when the condition is **true**. If two or more statements are used, these are given in braces { }. It is called the **body** of the loop.

Working of "for" Loop

The number of iterations depends on the initialization, condition and increment/decrement parts. The initialization part is executed only once when the control enters the loop. After initialization, the given condition is evaluated. If it is **true**, the control enters the body of loop and executes all statements in it. Then the increment/decrement part is executed that changes the value of counter variable. The control again moves to condition part. This process continues while the condition remains **true**. The loop is terminated when the condition becomes **false**.



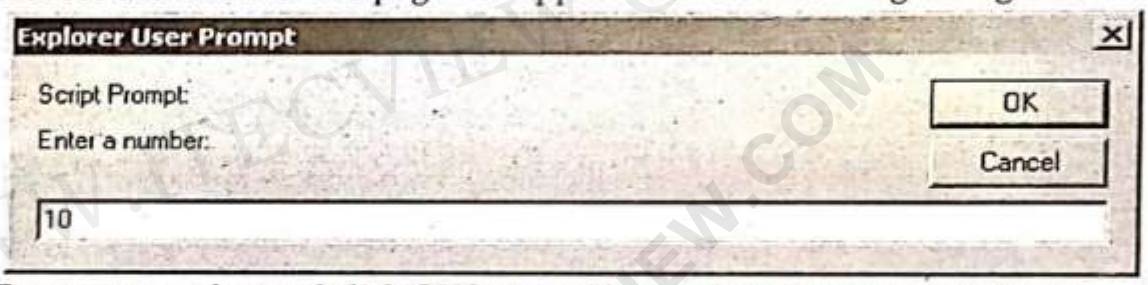
Practice 8.14

Write JavaScript code that displays numbers from 1 to 10 using **for** loop.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n, i;
n = prompt("Enter a number:","10");
for(i=1; i<=n; i++)
  document.writeln(i);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P09.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter any number and click OK button. The result will appear as follows:

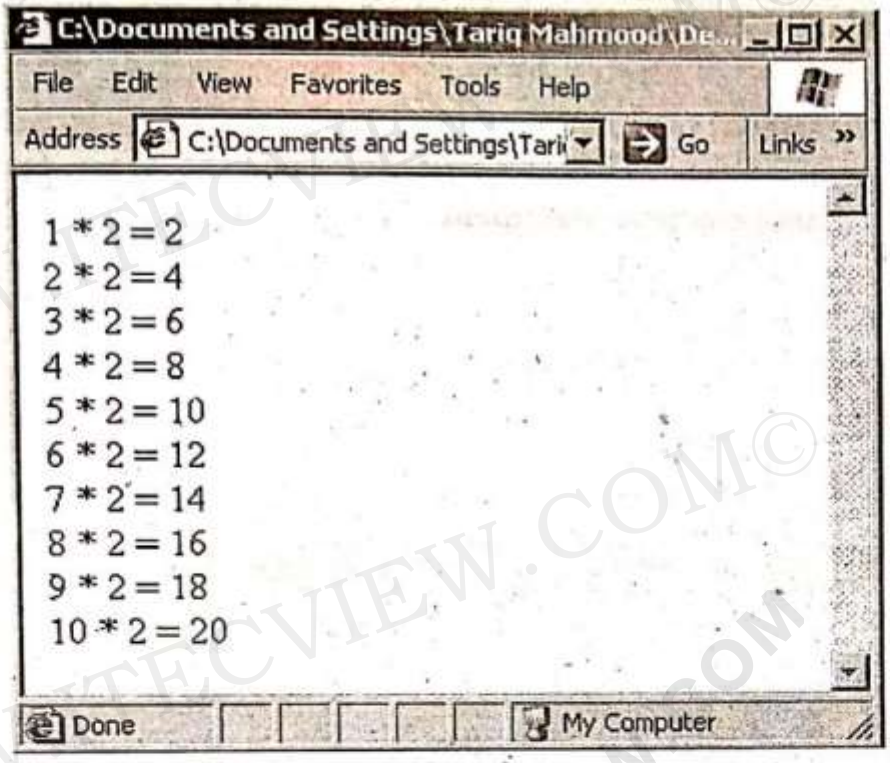
Practice 8.15

Write JavaScript code that displays table of 2 using **for** loop.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var i;
for(i=1; i<=10; i++)
{
  document.writeln(i + " * 2 " + " = " + (2*i));
  document.writeln("<BR>");
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P09a.html". The result will appear as follows:



8.12 "continue" Statement

The **continue** statement is used in the body of the loop. It is used to move the control to the start of the loop body. When this statement is executed in the loop body, the remaining statements of current iteration are not executed. The control directly moves to next iteration.

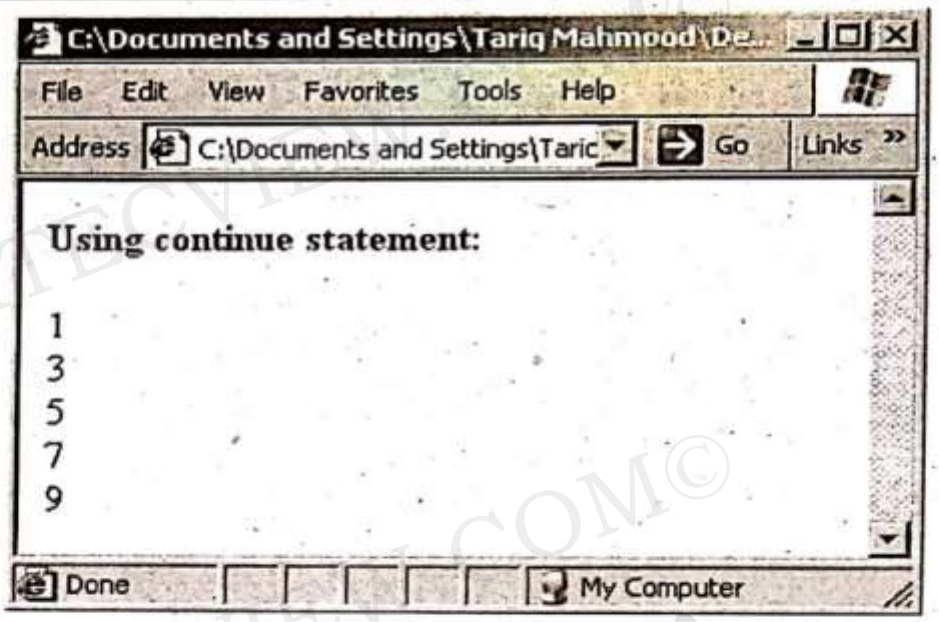
Practice 8.16

Write JavaScript code that displays first five odd numbers executing **for** loop from 1 to 10. Use **continue** statement to skip the even numbers.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var i;
document.write("<H4>Using continue statement:</H4>");
for(i=1; i<=10; i++)
{
  if(i%2==0)
    continue;
  document.write(i + "<BR>");
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Continue.html". The page will appear as follows:



How it Works?

The loop in the above practice executes from 1 to 10 but it displays only odd numbers between 1 and 10. When the value of counter variable **i** is even, the condition **if(i%2==0)** becomes **true** and the control moves to the next iteration without executing the statement **document.write(i + "<BR>");**

8.13 "break" Statement

The **break** statement is used in the body of loop to exit from it. When this statement is executed in the loop body, the remaining iterations of the loop are skipped. The control directly moves outside the body and the statement that comes after the body is executed.

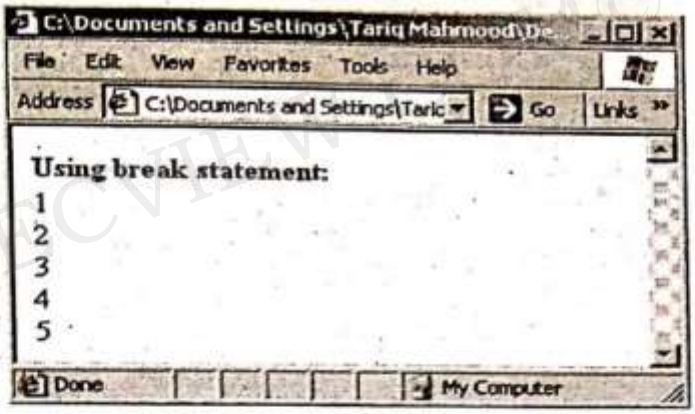
Practice 8.17

Write JavaScript code that displays numbers from 1 to 5 using **for** loop. The loop should execute from 1 to 10. Use **break** statement to skip the remaining numbers.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var i;
document.write("<H4>Using break statement:</H4>");
for(i=1; i<=10; i++)
{
  if(i>5) break;
  document.write(i + "<BR>");
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Break.html". The page will appear as follows:



How it Works?

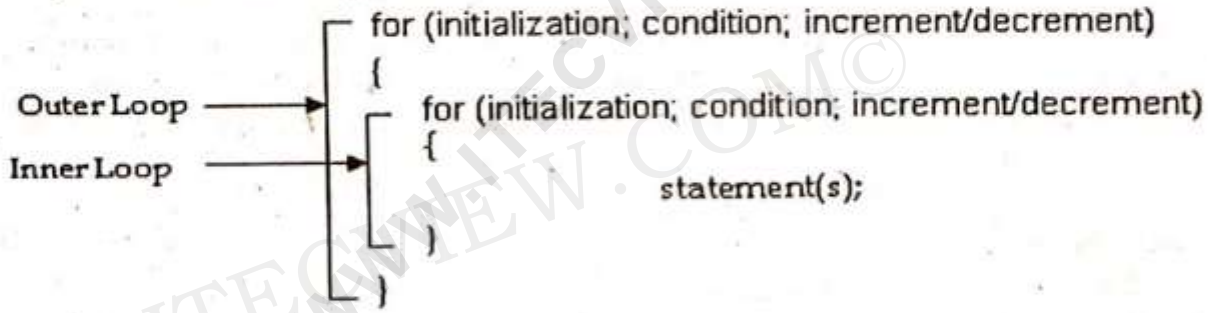
The loop in the above practice executes from 1 to 10 but it displays only first five numbers. When the value of counter variable i is 6, the condition **if(i>5)** becomes **true** and the loop is terminated without executing the remaining iterations.

8.14 Nested Loop

A loop within a loop is called **nested loop**. In nested loops, the inner loop is executed completely with each change in the value of counter variable of outer loop.

Syntax

The syntax of nested loop is as follows:



Practice 8.18

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var i, j;
document.write("<H4>Nested Loop:</H4>");
document.write("i ====== j<BR>");
for(i=1; i<=3; i++)
{
  for(j=1; j<=4; j++)
    document.write(i + " ===== " + j + "<BR>");
  document.write("<BR>");
}
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "NestLoop.html". The page will appear as follows:

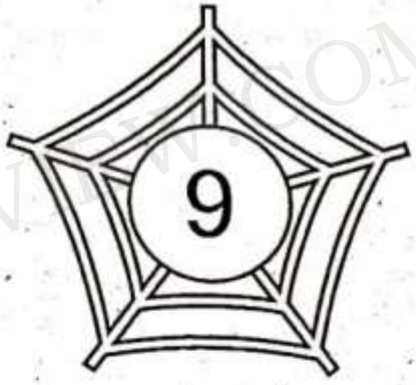
Exercise Questions

1. Write JavaScript code that inputs two numbers and finds if both are equal.
2. Write JavaScript code that inputs three numbers and displays only those numbers which are above 100.
3. Write JavaScript code that inputs three numbers and displays the minimum number.
4. Write JavaScript code that inputs a year and displays whether it is a leap year or not.
5. Write JavaScript code that inputs marks of a student and assigns scholarship on the following criteria:

Marks	Scholarship
700 and above	Rs. 3000
650 – 699	Rs. 2500
600 – 649	Rs. 2000

6. Write JavaScript code that inputs a number from 1 to 10 and displays it in words using **switch** statement. For example, if the user enters 5, it displays "Five".
7. Write JavaScript code that inputs a number and displays whether it is divisible by 3 or not by using conditional operator.
8. Write JavaScript code that displays "Pakistan Zindabad" five times using **while** loop.
9. Write JavaScript code that inputs starting number and ending number. It displays all odd numbers between starting and ending numbers using **do-while** loop.
10. Write JavaScript code that inputs a number from the user and displays table of that number using **for** loop.
11. Write JavaScript code that displays the following using nested **for** loop:

```
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
```



Functions and Arrays

9.1 Functions

A function is a named block of code that performs some action. The statements written in a function are executed when it is **called** by its name. Each function has a unique name. JavaScript provides the following types of functions:

- User-defined Functions
- Built-in Functions

1. User-defined Functions

A type of function written by the programmer is known as **user-defined function.** User-defined function has a unique name. A script may contain many user-defined functions.

2. Built-in Functions

A type of function that is available as a part of JavaScript is known as **built-in function.** Built-in functions make scripting faster and easier. JavaScript provides many built-in functions to solve different problems.

9.1.1 Function Declaration

Function declaration consists of the **function** keyword, followed by:

- The name of the function.
- A list of arguments to function in parentheses and separated by commas.
- JavaScript statements that define the function enclosed in curly braces { }.

Syntax

The syntax for declaring a function is as follows:

```
function function_name(parameters)
{
  JavaScript code
}
```

function: It is the keyword that is used to declare a function in JavaScript.

Function-name: It indicates the name of the function. The rules for specifying a function name is similar to the rules for declaring a variable. Each function should have a unique name.

Parameters: Parameters are the values that are provided to a function when the function is called. Parameters are given in the parentheses. If there are many parameters, these are separated by commas. If there is no parameter, empty parentheses are used.

Example

The following code defines a function **msg**:

```
function msg()
{
  write("Welcome to function...");
}
```

9.1.2 Function Call

The statement that activates a function is known as **function call**. A function is called with its name. Function name is followed by necessary parameters in parentheses. If there are many parameters, these are separated by commas. If there is no parameter, empty parentheses are used. When a function is called, the following steps take place:

1. The control moves to the function that is called.
2. All statements in the function body are executed.
3. The control returns back to the calling function.

When the control returns back to the calling function, the remaining statements in the calling function are executed.

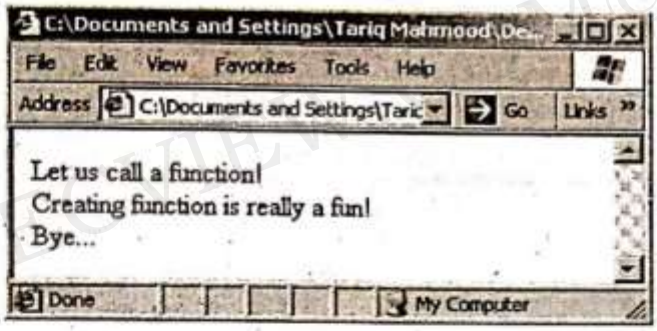
Practice 9.1

Write JavaScript code that displays a simple message by using a function.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function fun1( )
{
  document.write("Creating function is really a fun!<BR>");
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
document.write("Let us call a function!<BR>");
fun1( );
document.write("Bye...");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Fun1.html".
3. Double click the file. The page will appear in the browser.



9.1.3 Passing Parameters to Functions

Parameters are the values that are provided to a function when the function is called. Parameters are given in the parentheses. If there are many parameters, these are separated by commas. If there is no parameter, empty parentheses are used.

The number of parameters in function call must be similar to the number of parameters in function declaration.

- Parameters in function call are called **actual parameters.**
- Parameters in function declaration are called **formal parameters.**

When a function call is executed, the values of actual parameters are copied to formal parameters.

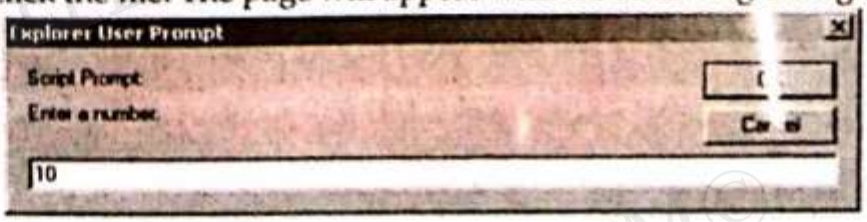
Practice 9.2

Write JavaScript code that inputs a number and passes it to a function. The function displays the square of that number.

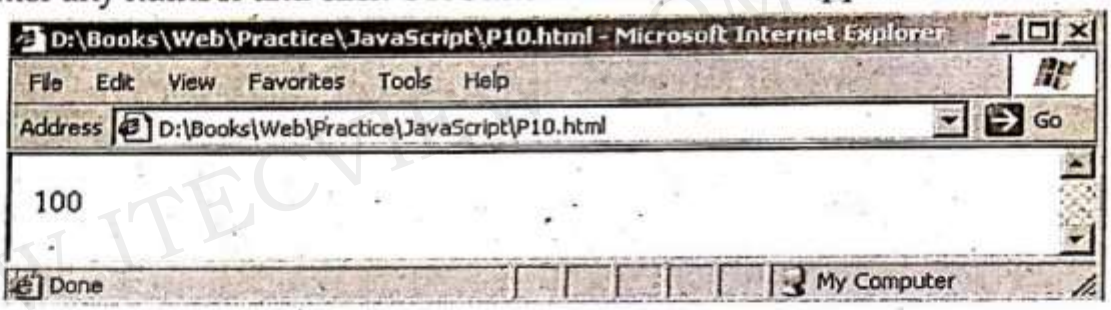
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function num(a)
{
  document.write(a*a);
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = prompt("Enter a number:","10");
num(n);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P10.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter any number and click **OK** button. The result will appear as follows:



How it Works?

The above example uses a function **num** that accepts a number as parameter and displays its square. Function is declared in HEAD part of the document. The user inputs a number in BODY part of the document that is stored in variable **n**. The statement num(n); calls the function and passes the value of **n** to the function as parameter. This value is stored in variable **a**. The function displays the square of this value.

9.1.4 Returning Value

A function can return a single value. The keyword **return** is used to return the value back to the calling function. When the **return** statement is executed in a function, the controls moves back to the calling function along with the returned value.

Syntax

The syntax for returning a value is as follows:

```
return expression;
```

expression: It can be a variable, constant or an arithmetic expression whose value is returned to the calling function.

Practice 9.3

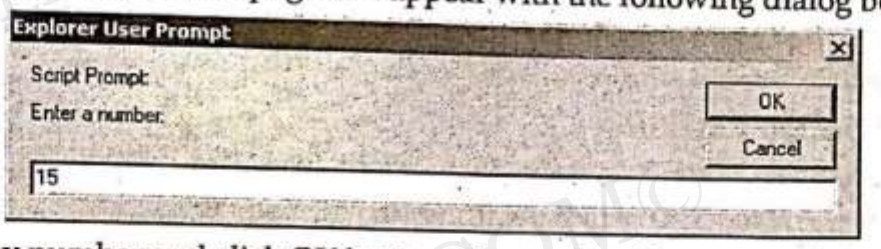
Write JavaScript code that inputs a number from the user and passes it to a function. The function checks it whether it is even or odd and returns the result. The script then displays the result.

1. Open Notepad and type the following code:

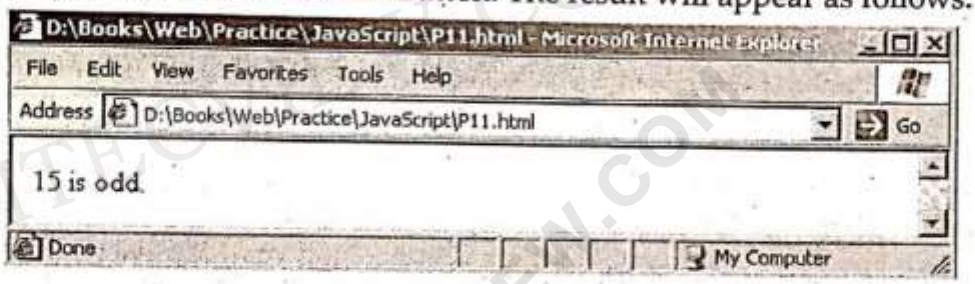
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function check(n)
{
  if(n%2 == 0)
    return "even";
  else
    return "odd";
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var num, r;
num = prompt("Enter a number:","10");
```

```
r = check(num);
document.write(num + " is " + r);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P11.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter any number and click OK button. The result will appear as follows:



The function check in the above example accepts a number as parameter. It checks the number and returns "even" if the number is even. Otherwise it returns "odd".

9.2 Local Variables

A variable declared inside a function is known as **local variable.** A function can declare as many local variables as required for completing a task.

Scope of Local Variable

The area where a variable can be accessed is known as **scope of variable.** Local variable can be used only in the function in which it is declared.

Lifetime of Local Variable

The time period for which a variable exists in the memory is known as the **lifetime of variable**. Different types of variables have different lifetimes.

The lifetime of local variable starts when the control enters the function in which it is declared. Local variable is automatically destroyed when the control exits from the function and its lifetime ends. When the lifetime of a local variable ends, the value stored in this variable also becomes inaccessible.

Example

```
<SCRIPT LANGUAGE="JavaScript">
Function show( )
{
 var m;
}
</SCRIPT>
```

In the above example, the function **show()** declares a local variable **m** in its body. It is a local variable and cab be used only in this function.

9.3 Global Variables

A variable declared outside any function is known as **global variable.** Global variables can be used by all functions in the program. The values of these variables are shared among different functions. If one function changes the value of a global variable, this change is also available to other functions.

Scope of Global Variable

Global variable can be used by all functions in the program. It means that these variables are globally accessed from any part of the script.

Lifetime of Global Variable

Global variables exist in the memory as long as the script is running. These variables are destroyed from the memory when the program terminates.

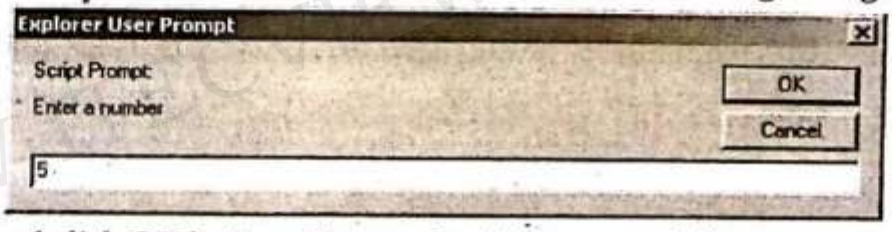
Practice 9.4

Write JavaScript code that inputs a number from the user and passes it a function. The function calculates its square and cube in two local variables and then display them.

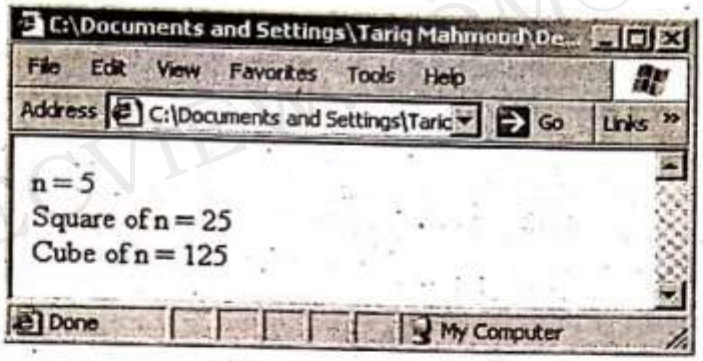
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function fun1(n)
{
  var s, c;
  s = n * n;
  c = n * n * n;
  document.write("n = " + n + "<BR>");
  document.write("Square of n = " + s + "<BR>");
  document.write("Cube of n = " + c + "<BR>");
}
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var num;
num = prompt("Enter a number","1");
fun1(num);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Local.html".
3. Double click the file. The page will appear with the following dialog box:



4. Enter 5 and click **OK** button. The result will appear as follows:



How it Works?

The above practice declares a function **fun1** with two local variables **s** and **c**. The variables are used to calculate the square and cube of the number **n**.

9.4 Built-in Functions

A type of function that is provided by JavaScript is called built-in function. JavaScript provides many built-in functions to write scripts easily and speadily.

9.4.1 Dialog Boxes

Dialog boxes are used to interact with the user. Some dialog boxes are used to get input from the user and to display information to the user.

Some important dialog boxes in JavaScript are as follows:

Alert Dialog Box

The alert dialog box is used to display information to user. It takes a string parameter and displays it in a dialog box. The dialog box contains OK button. The user has to press OK button to close it.

Syntax

```
alert("Message");
```

Practice 9.5

Write JavaScript code that displays a simple message in a message box.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  alert("Welcome to JavaScript");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P13.html".
3. Double click the file. The page will appear with the following dialog box.

4. Click OK button to close the dialog box.

Prompt Dialog Box

Prompt dialog box is used to get input from the user. The input given by the user in prompt dialog box can be stored in a variable for further processing. Prompt dialog box displays the following:

- A message to the user that tells the user to input a value.
- A textbox where the user types the input.
- The OK and Cancel buttons.

If the user clicks OK button, the value in the textbox is passed to JavaScript code. If the user clicks Cancel button, NULL value is passed to JavaScript code.

Syntax

```
prompt("Message", "DefaultValue");
```

prompt: It is a method that displays prompt dialog box.

Message: It is the message that is displayed on the prompt dialog box.

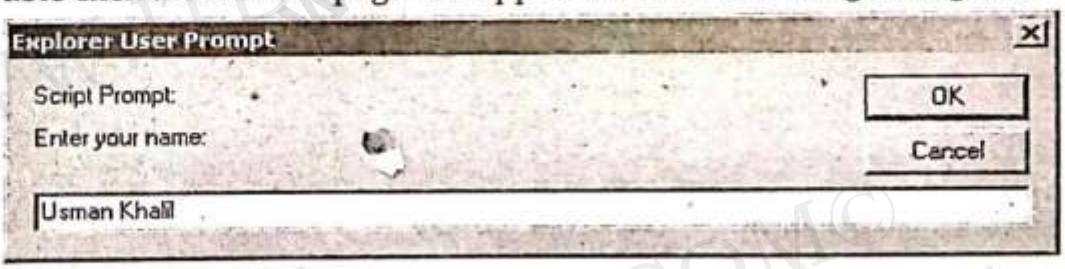
DefaultValue: It is the value that appears in the textbox as the prompt dialog box is displayed. The use of default value is optional.

Practice 9.6

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = prompt("Enter your name:","Name");
document.write("Hello <B>" + n + "! </B>Welcome to JavaScript...");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P14.html".
3. Double click the file. The page will appear with the following dialog box.



4. Enter your name and click OK button. The page will appear as follows:

Confirm Dialog Box

The confirm dialog box is used to confirm the action of user. It contains the following:

- A message to the user.
- The OK and Cancel buttons.

If the user clicks OK button, a **True** value is passed to JavaScript code. if the user clicks Cancel button, a **False** value is passed to JavaScript code.

Syntax

```
confirm("Message");
```

confirm: It is a method that displays confirm dialog box.

Message: It is the message that is displayed on the confirm dialog box.

Practice 9.7

Write JavaScript code that displays confirm dialog box and tells which button is clicked.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var r;
n = confirm("Click any button:");
if (n == true)
  document.write("You clicked OK.");
else
  document.write("You clicked Cancel.");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P15.html".
3. Double click the file. The page will appear with the following dialog box.

4. Click OK button. The page will appear as follows:

9.4.2 Conversion Functions

JavaScript provides different built-in functions to convert the values from one type to another type. Different types of conversion functions are as follows:

Integer Conversion

When the user types a value in a textbox, the value is considered to be string. The browser may not be able to identify its type exactly. The **parseInt()** function is used to analyze this value. Its syntax is as follows:

parseInt(Parameter);

The function is executed as follows:

- The interpreter will examine the parameter from left to right.
- If the value is a natural number, **parseInt()** returns that number.
- If the first character of the argument is not a digit, the function would return a value known as **NaN** (not a number).
- If the first character from left is a digit, the interpreter continues examining the other characters. When it finds a non-digit character, it stops. It then returns the digits found on the left side of the first non-digit character.

Examples

parseInt("124")

The above example will return 124.

parseInt("ABC")

The above example will return NaN.

parseInt("12A")

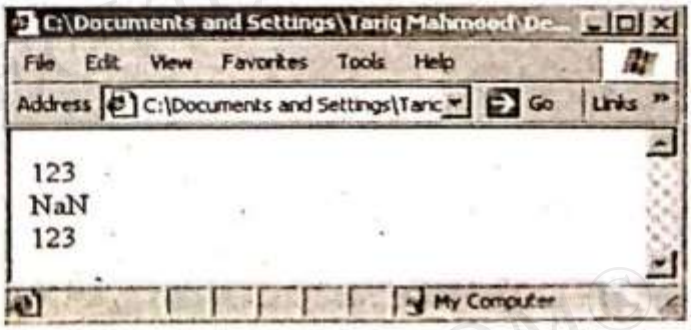
The above example will return 12.

Practice 9.8

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
        var n;
        n = "123";
        document.write(parseInt(n) + "<BR>");
        n = "M123";
        document.write(parseInt(n) + "<BR>");
        n = "123M";
        document.write(parseInt(n));
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "ParseInt.html".
3. Double click the file. The page will appear as follows:



Decimal Conversion

If a value is to be considered as a decimal number in a calculation, it should first be converted. This can be done by passing it to the **parseFloat()** function. Its syntax is as follows:

parseFloat(Parameter)

The parameter is passed as a string. The functions is executed as follows:

- The interpreter examines the argument.
- If all characters of the string are digits or can be evaluated to a decimal number, this function returns it.
- If some characters in the string are not digits and there is more than one decimal operator, the interpreter considers characters from left to right.

Examples

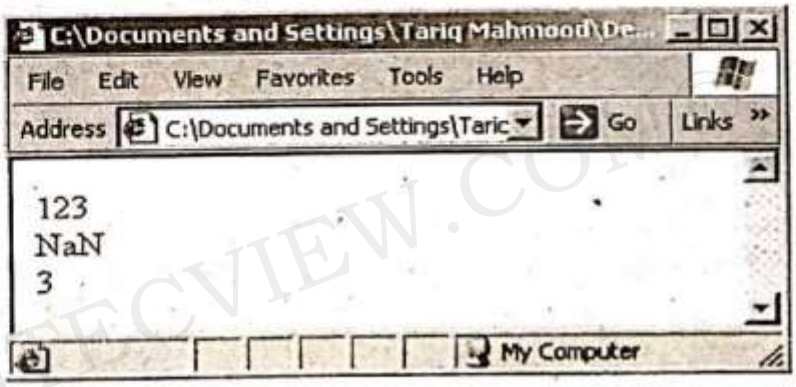
parseInt("10.5")	This example will return 10.
parseInt("ABC.9")	This example will return NaN.
parseInt("55.2.3")	This example will return 55.

Practice 9.9

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var n;
n = "123.3";
document.write(parseInt(n) + "<BR>");
n = "M123";
document.write(parseInt(n) + "<BR>");
n = "3.3.2";
document.write(parseInt(n));
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "ParseFloat.html".
3. Double click the file. The page will appear as follows:



Converting to String

A numeric value can be converted to a string by using **String()** function. The syntax of this function is as follows:

String(Parameter)

Example

```
String(123)
```

The above example will return "123".

Precision Conversion

When the interpreter performs some calculation, it produces results with all the digits it considers necessary. The number of characters used to display a number can be given by using **toPrecision()** function. Its syntax is as follows:

```
Value.toPrecision(number)
```

Example

```
N.toPrecision(3)
```

The above example will display the value of N using three characters.

Practice 9.10

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var a, b, c;
a = 1000;
b = 3;
c = a/b;
document.write(c + "<BR>");
document.write(c.toPrecision(2) + "<BR>");
document.write(c.toPrecision(3) + "<BR>");
document.write(c.toPrecision(4) + "<BR>");
document.write(c.toPrecision(5) + "<BR>");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Precision.html".
3. Double click the file. The page will appear as follows:

Convert to Fixed Decimal

JavaScript provides the **toFixed()** function to limit the number of digits of precision. Its syntax is:

```
Value.toFixed(numberOfDigits)
```

The parameter **numberOfDigits** is used to specify the number of digits to display on the right side of the decimal separator.

Example

```
var m = 10;
m.toFixed(2)
```

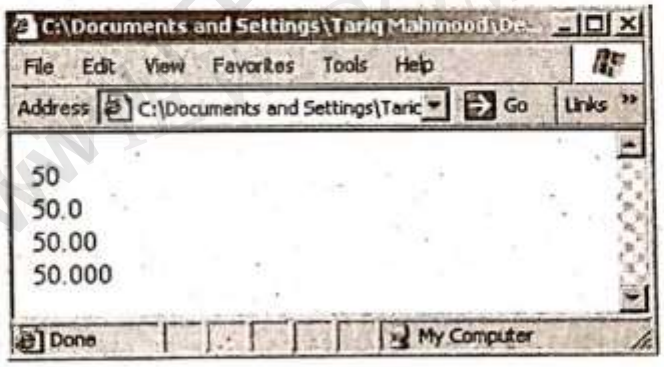
The above example will return 10.00

Practice 9.11

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var a, b, c;
a = 100;
b = 2;
c = a/b;
document.write(c + "<BR>");
document.write(c.toFixed(1) + "<BR>");
document.write(c.toFixed(2) + "<BR>");
document.write(c.toFixed(3) + "<BR>");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Fixed.html".
3. Double click the file. The page will appear as follows:



9.5 Arrays

An **array** is a collection of consecutive memory locations. Each memory location is called an **element** of array. The number of elements in an array is called the **length** of array. All elements have one collective name. An individual element in an array is accessed with the help of an **index**. Each element has a unique index. The index starts from 0. It means that the index of first element of an array is always 0.

The following example shows an array **marks** with five elements.

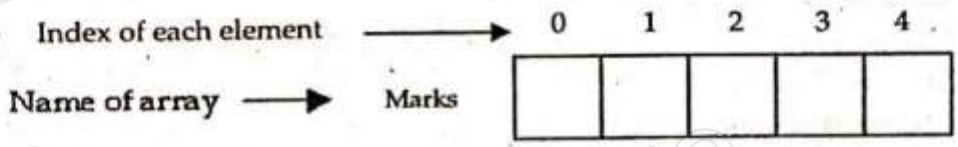


Figure: An array **Marks** with five elements

Arrays are used to store many values of similar type with single name. They provide an easy way of storing and processing a large amount of data. For example, 100 variables are required to store the marks of 100 students. It is difficult to process these variables individually. It is easier to use an array of length 100 instead of using 100 variables. It will be easier to process the marks.

9.5.1 Array Declaration

JavaScript provides two ways of declaring an array which are as follows:

Method 1

```
ArrayName = new Array(Length);
```

In the above example, ArrayName is the name of array and Length specifies the number of elements in the array.

Example

```
names = new Array(3);
```

The above example declares an array **names** that can store three values.

Method 2

```
ArrayName = new Array();
```

In the above example, ArrayName is the name of array. The length of the array is not defined. Its length is extended when values are stored in the array at run time.

9.5.2 Assigning Values to Array

The values are assigned to each element of the array by using assignment operator. An array in JavaScript can store different types of values.

Assigning String Values

The following example decalres an array of length 3 and stores three values in different elements of the array.

```
names = new Array(3);
names[0] = "Tariq Mahmood";
names[1] = "Imran Saeed";
names[2] = "IT Series";
```

Assigning Integer Values

The following example decalres an array of length 5 and stores five numeric values in different elements of the array.

```
marks = new Array(3);
marks[0] = 50;
marks[1] = 84;
marks[2] = 61;
marks[3] = 97;
marks[4] = 49
```

Assigning Character Values

The following example decalres an array of length 3 and stores three character values in different elements of the array.

```
grades = new Array(3);
grades[0] = 'A';
grades[1] = 'B';
```

```
grades[2] = 'C';
```

9.5.3 Retrieving Values from Arrays

The values stored in an array can be retrieved by accessing an individual element of the array directly.

Example

```
grades = new Array(3);
grades[0] = 'A';
grades[1] = 'B';
grades[2] = 'C';
ch = grades[1];
document.write(ch);
document.write(grades[2]);
```

The above example declares an array of length 3 and stores three characters in it. The statement **ch = grades[1];** retrieves value of second element of array and stores it in **ch**. The statement **document.write(grades[2]);** retirves value of third element of array and displays it.

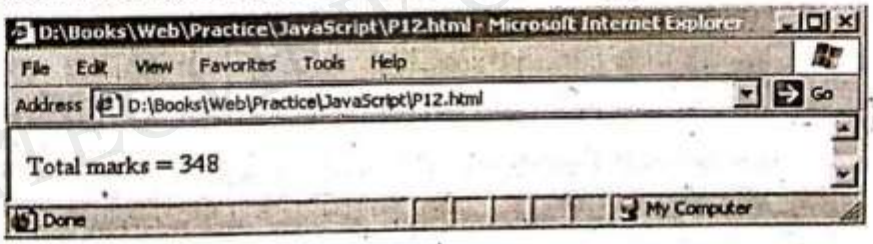
Practice 9.12

Write JavaScript code that inputs five marks from the user and stores them in an array. It calculates these marks and displays total marks.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var t, i;
marks = new Array(5);
t = 0;
for(i=0; i<5; i++)
{
  marks[i] = parseInt(prompt("Enter marks: ","0"));
  t = t + marks[i];
}
document.write("Total marks = " + t);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "P12.html".
3. Double click the file. A dialog box will appear for five times to get give marks.
4. Enter marks and click **OK** button. The result will appear as follows:



9.5.4 Passing Array to Function

The array can be passed to a function as parameter. The function can process the array and manipulate it elements.

Practice 9.13

Write JavaScript code that stores three numbers in an array and passes that array to a function. The function displays the values of the array.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function fun(a)
{
  var i;
  for(i=0; i<3; i++)
    document.write(a[i] + "<BR>");
}
arr = new Array(3);
arr[0] = 100;
arr[1] = 200;
arr[2] = 300;
fun(arr);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "ArrFun.html".
3. Double click the file. The page will display the values 100, 200 and 300.

9.6 Manipulating Arrays

JavaScript provides different built-in functions to manipulate arrays. These functions can be used to process array elements easily.

Finding Array Length

The **length** property of array is used to find the length of an array. The syntax of using this property is as follows:

ArrayName.length;

In the above line, ArrayName specifies the name of array whose length is to be found.

toString() Function

toString() function is used with arrays to convert an array to string. The values of different elements of array are separated with commas.

join Function

join function is also used to convert an array to string. It accepts a parameter that is used to specify the character to separate the values of array elements.

sort Function

sort function is used to sort the array.

reverse Function

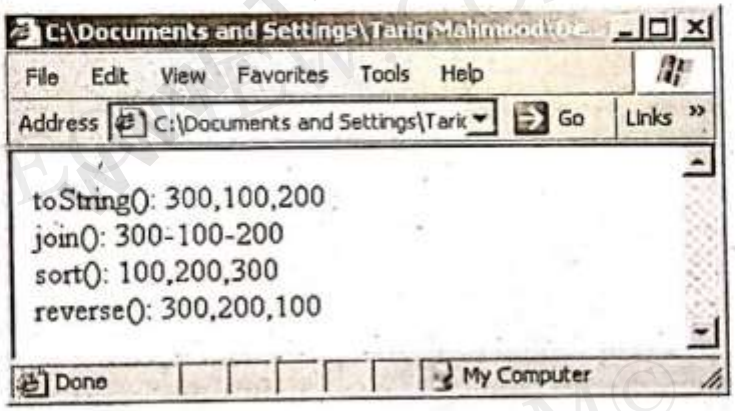
reverse function is used to reverse the elements of an array. The first element of the array becomes the last element and the last element becomes the first element.

Practice 9.14

1. Open Notepad and type the following code:

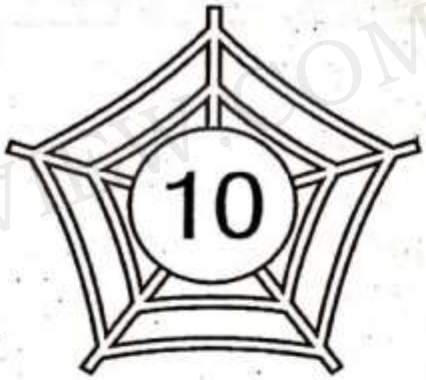
```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
var s1, s2;
arr = new Array(3);
arr[0] = 300;
arr[1] = 100;
arr[2] = 200;
s1 = arr.toString();
s2 = arr.join('-');
document.write("toString(): " + s1 + "<BR>");
document.write("join(): " + s2 + "<BR>");
arr.sort();
s1 = arr.toString();
document.write("sort(): " + s1 + "<BR>");
arr.reverse();
s1 = arr.toString();
document.write("reverse(): " + s1);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "ManArr.html".
3. Double click the file. The result will appear as follows:



Exercise Questions

1. Write JavaScript code that inputs two numbers and passes them to a function. The function displays the maximum number.
2. Write JavaScript code that inputs three marks and passes them to a function. The function calculates average marks and returns it. The script then displays the result.
3. Write JavaScript code that stores ten numbers in an array. It inputs a number from the user and tells whether the number is present in the array or not.
4. Write JavaScript code that displays the maximum and minimum numbers stored in an array.
5. Write JavaScript code that sorts an array in ascending order.



JavaScript Object Model

10.1 JavaScript Object Model

An HTML page is a collection of different **objects** such as FORM, TEXTBOX and BUTTON etc. A web browser works with different objects while displaying web pages. These objects have different properties and functions. JavaScript-enabled browsers use **Document Object Model (DOM)** to recognize these objects.

The HTML objects have a descending relationship with each other. The topmost object in DOM is called **Navigator**. It refers to the browsers itself. Different other objects in DOM hierarchy are as follows:

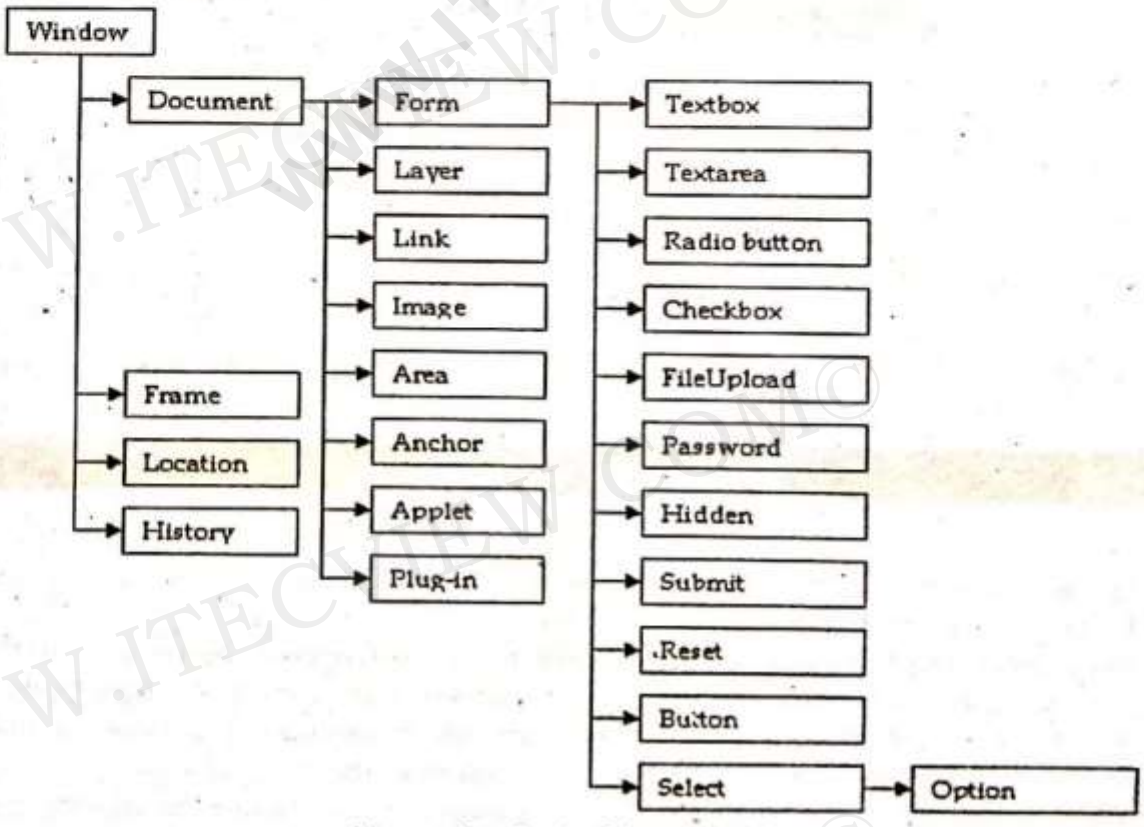


Figure: JavaScript Object Model

10.2 Objects

An object is a collection of **properties** and **methods**. A web page consists of many objects. HTML is used to create different objects such as textbox, radio buttons and list boxes.

Properties of Objects

An object may have different properties associated with it. The properties can be accessed as follows:

```
objectName.propertyName
```

Both the object name and property name are case sensitive. Suppose we have an object **Student** with some properties such as **Name, Marks** and **Gender**. These properties can be accessed as follows:

```
Student.Name = "Usman";
Student.Marks = 100;
Student.Gender = "Male";
```

Methods of Objects

Methods are used to perform actions. Objects have different types of methods. Some methods are used to set or get the values of different properties of the objects. These methods can be accessed as follows:

```
objectName.methodName
```

Both the object name and method name are case sensitive. Suppose we have an object **Student** with some methods such as **show** and **close**. These methods can be used as follows:

```
Student.show();
Student.close();
```

10.2.1 Browser Objects

Different browser objects are as follows:

Object	Use
Navigator	It is used to access information about the browser.
Window	It is used to access a browser window or frame.
Document	It is used to access the document loaded in a window.
Location	It refers to the URL the browser is visiting.
History	It is used to maintain the history of URLs accessed in the browser.
Screen	It is used to access the information about the computer screen.

10.2.2 Arrays of Objects

An HTML document consists of many objects. It may contain many objects of same type. For example, a document may have many images and many hyperlinks etc. The browser creates array to maintain the list of each type of objects. For example, if there are ten images in the document, an array called **Images** will be used to refer to these images. Each image will be referred by different index of the **Images** array. **Images[0]** will refer to the first image in the document, **Images[1]** will refers to the second image and so on.

Different arrays for maintaining different type of objects are as follows:

Arrays	Use
Images	It is used to access different images in the document.
Links	It is used to access different links in the document.

Area	It is used to access different areas in the document.
Frames	It is used to access different frames in the document.
Anchors	It is used to access different anchors in the document.
Applets	It is used to access different applets in the document.
Forms	It is used to access different forms in the document.

10.3 Form Elements

A form consists of different types of form objects. An array of objects is maintained for each type of form element. Different arrays for form elements are as follows:

Arrays	Use
Elements	It is used to access all elements in a form.
Text	It is used to access all textboxes in a form.
Textarea	It is used to access all text areas in a form.
Radio	It is used to access all radio buttons in a form.
Checkbox	It is used to access all checkboxes in a form.
Button	It is used to access all buttons in a form.
Submit	It is used to access all submit buttons in a form.
Reset	It is used to access all reset buttons in a form.
Select	It is used to access all select lists in a form.
Option	It is used to access all options of select list in a form.
Password	It is used to access all password fields in a form.
Hidden	It is used to access all hidden fields in a form.
Fileupload	It is used to access all file upload element in a form.

10.4 Events

An event occurs when something **happens** or an action is performed. Events are used to write JavaScript code that reacts to certain situations. Some events are as follows:

- The user clicking the mouse button.
- The Web page loading
- A form field being changed

10.4.1 Event Handlers

An **event handler** is a piece of code that is executed when an event occurs. Each even handler is associated with an event. All event handlers in JavaScript start with the word **on**.

Some important types of event handlers are as follows:

Event Handler	Applies To:	Triggered When:
onAbort	Image	The image loading is cancelled.
onBlur	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object loses focus.
onChange	FileUpload, Select, Text, TextArea	The data in the form element is changed by the user.
onClick	Button, Document, Checkbox, Link, Radio, Reset, Submit	The object is clicked.
onDblClick	Document, Link	The object is double-clicked.

onDragDrop	Window	An icon is dragged and dropped into the browser.
onError	Image, Window	A JavaScript error occurs.
onFocus	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object gains focus.
onKeyDown	Document, Image, Link, TextArea	The user presses a key.
onKeyPress	Document, Image, Link, TextArea	The user presses a key.
onKeyUp	Document, Image, Link, TextArea	The user releases a key.
onLoad	Image, Window	The page has finished loading.
onMouseDown	Button, Document, Link	The user presses a mouse button.
onMouseMove	None	The user moves the mouse.
onMouseOut	Image, Link	The user moves the mouse away from the object.
onMouseOver	Image, Link	The user moves the mouse over the object.
onMouseUp	Button, Document, Link	The user releases a mouse button.
onMove	Window	The user moves the browser window or frame.
onReset	Form	The user clicks the form's Reset button.
onResize	Window	The user resizes the browser window or frame.
onSelect	Text, Textarea	The user selects text in the field.
onSubmit	Form	The user clicks form's Submit button.
onUnload	Window	The user closes the page.

10.5 Forms and Events

Forms are used in web pages to get input from the user. The HTML form can also submit this data to the web server. The web server may process or store the data for future use. The data provided by the user is important and it must be validated before it is stored or processed. Forms use different functions to validate the data when the user clicks on **submit** button of the form. Different events can also be used to process and validate data in forms.

FORM object of HTML is used to create forms. A form consists of different elements. Each element has its own properties, methods and events handlers. Different elements of a form are as follows:

10.5.1 Button Element

Button is used in forms to process the form data. <BUTTON> tag is used to create a button in the form.

Methods

Some important methods of button element are as follows:

Method	Description
click()	It is used to press the button.

Events

Some important events of button element are as follows:

Event	Description
click()	It occurs when button element is clicked.

Event Handlers

Some important event handlers of button element are as follows:

Method	Description
onClick()	It is executed when button element is clicked.

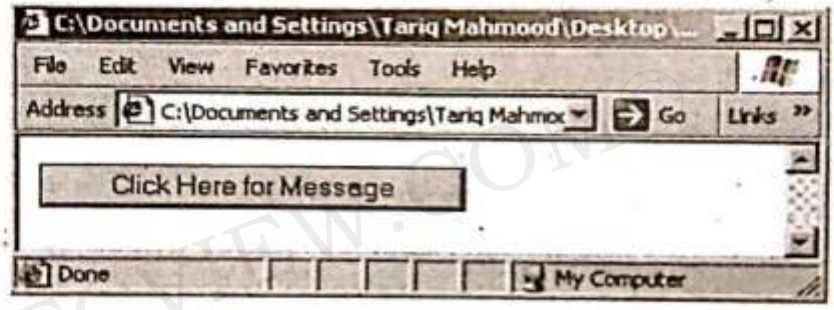
Practice 10.1

Write JavaScript code that displays a button on the page. A message should appear when the user click the button.

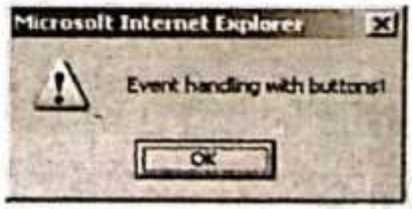
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function msg()
{
  alert("Event handling with buttons!");
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
<INPUT TYPE="button" VALUE="Click Here for Message" onClick="msg();">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Button.html".
3. Double click the file. The page will appear as follows:



4. Click on the button. The following message will appear on the screen:



10.5.2 Text Element

Text fields are used to get input from the user in the form of letters, numbers etc.

Methods

Some important methods of text element are as follows:

Method	Description
focus()	It is used to bring the focus to the text element.
blur()	It is used to bring the focus away from text element.
select()	It is used to select the contents of the text element.

Events

Some important events of text element are as follows:

Event	Description
focus()	It occurs when text element gains focus.
blur()	It occurs when text element loses focus.
select()	It occurs when the user selects the contents of text element.

Event Handlers

Some important event handlers of text element are as follows:

Method	Description
onFocus()	It is executed when text element gains focus.
onBlur()	It is executed when text element loses focus.
onSelect()	It is executed when the user selects the contents of text element.
onChange()	It is executed when the user changes the contents of text element.

Practice 10.2

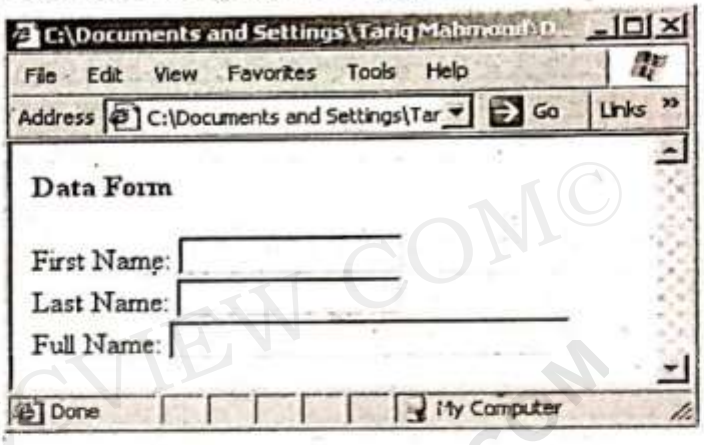
Write JavaScript code that displays three text boxes. If the user changes the contents of first two text boxes, the script combines the contents of first two text boxes and displays them in third text box.

1. Open Notepad and type the following code:

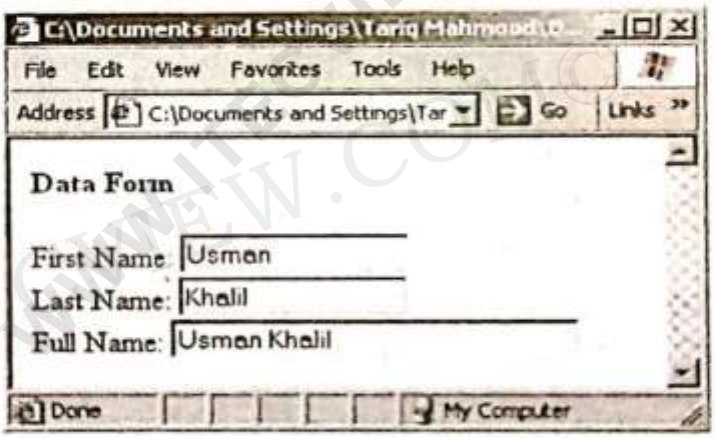
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function full()
{
  frm.txtFullName.value = frm.txtFirstName.value + ' ' + frm.txtLastName.value;
}
</SCRIPT>
</HEAD>
<BODY>
<H4>Data Form</H4>
<FORM NAME="frm">
First Name:
<INPUT TYPE="text" NAME="txtFirstName" SIZE="15" onChange="full();">
<BR>Last Name:
<INPUT TYPE="text" NAME="txtLastName" SIZE="15" onChange="full();">
<BR>Full Name:
```

```
<INPUT TYPE="text" NAME="txtFullName" SIZE="30">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Textbox.html".
3. Double click the file. The web page will appear as follows:



4. Enter first name and press TAB key.
5. Enter second name and press TAB key. Both first and second name will appear in the third text box as follows:



Practice 10.3

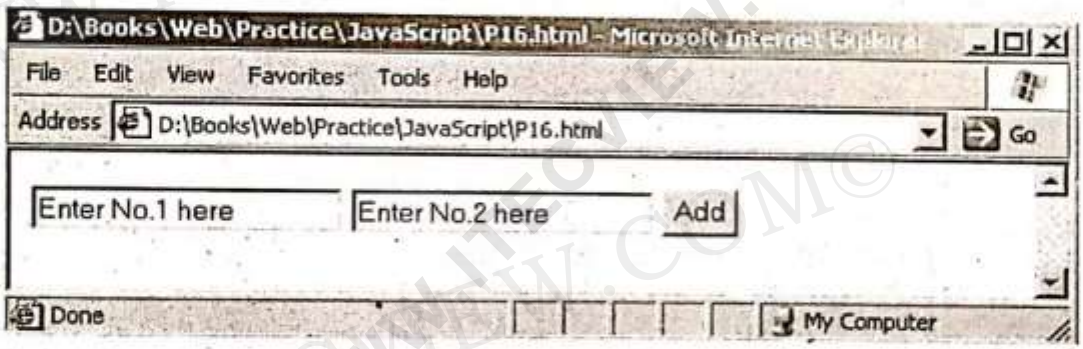
Write JavaScript code that displays two text boxes and a button. The default value of first text box should be "Enter first No. here" and default value of second text box should be "Enter second No. here". The script should add these numbers and display the result in a message box when the user clicks the button.

1. Open Notepad and type the following code:

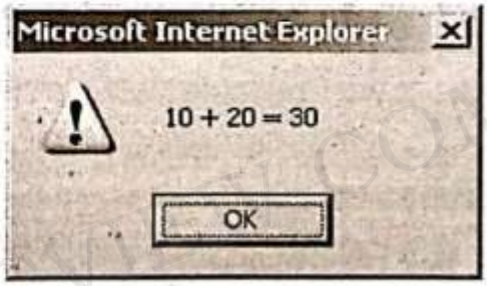
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function load()
{
  document.F.No1.value = "Enter No.1 here";
  document.F.No2.value = "Enter No.2 here";
}
```

```
function add()
{
  var a, b, c;
  a = parseInt(document.F.No1.value);
  b = parseInt(document.F.No2.value);
  c = a + b;
  alert(a + " + " + b + " = " + c);
}
</SCRIPT>
</HEAD>
<BODY onLoad="load();">
<FORM NAME="F">
<INPUT TYPE="text" NAME="No1" onFocus="document.F.No1.value=' ';">
<INPUT TYPE="text" NAME="No2" onFocus="document.F.No2.value=' ';">
<INPUT TYPE="button" VALUE="Add" NAME="Add" onClick="add();">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "P16.html".
3. Double click the file. The web page will appear as follows:



4. Click in first textbox. It will become blank.
5. Enter 10 in first textbox.
6. Click in second textbox. It will become blank.
7. Enter 20 in second textbox and click **Add** button. The following message will appear.



10.5.3 Password Element

The methods, events and event handlers of password element are same as text element.

10.5.4 Checkbox Element

Checkboxes are used to provide many options to the user. The user can choose one or multiple options from available choices. The methods, events and event handlers of checkbox element are same as button element.

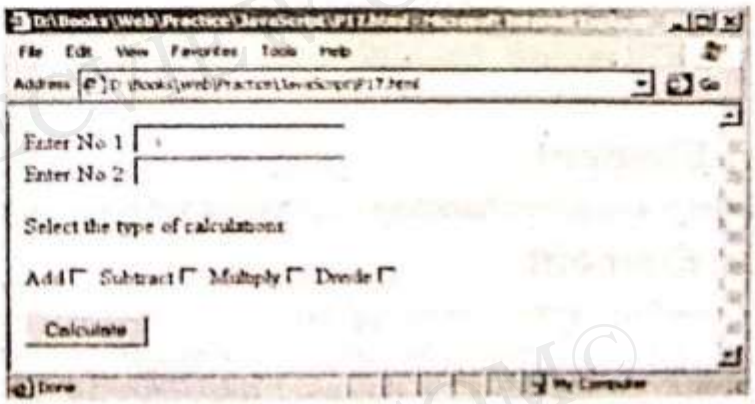
Practice 10.4

Write JavaScript code that displays two text boxes to input two numbers, four checkboxes to provide options for arithmatic operations and a button. The user enters two numbers and selects checkboxes to perform arithmatic operations.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function cal()
{
  var a, b, c;
  a = parseInt(document.F.No1.value);
  b = parseInt(document.F.No2.value);
  if(document.F.Add.checked)
    alert(a + " + " + b + " = " + (a+b));
  if(document.F.Sub.checked)
    alert(a + " - " + b + " = " + (a-b));
  if(document.F.Mul.checked)
    alert(a + " * " + b + " = " + (a*b));
  if(document.F.Div.checked)
    alert(a + " / " + b + " = " + (a/b));
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
Enter No.1: <INPUT TYPE="text" NAME="No1"><BR>
Enter No.2: <INPUT TYPE="text" NAME="No2"><BR>
<BR>Select the type of calculations: <P>
Add<INPUT TYPE="checkbox" NAME="Add">
Subtract<INPUT TYPE="checkbox" NAME="Sub">
Multiply<INPUT TYPE="checkbox" NAME="Mul">
Divide<INPUT TYPE="checkbox" NAME="Div"><P>
<INPUT TYPE="button" NAME="Cal" VALUE="Calculate" onClick="cal();">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "P17.html".
3. Double click the file. The web page will appear as follows:



4. Enter two numbers in textboxes and select **Add** and **Divide** checkboxes.
5. Click **Calculate** button. The result will appear in messages boxes according to the selection as follows:

10.5.5 Radio Element

Radio buttons are used when the user has to choose one option from multiple choices. The methods, events and event handlers of radio element are same as that of button element.

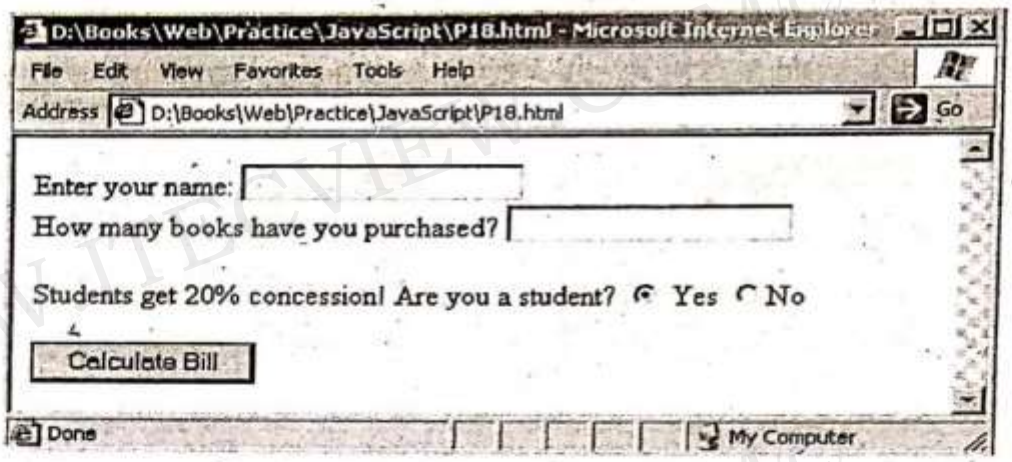
Practice 10.5

Write JavaScript code that inputs name and number of books in two text boxes. It displays two radio buttons to select whether the user is a student or not. It calculates the bill of books when the user clicks a button. If the user is a student, he gets 20% concession.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function cal()
{
  var n, books, bill;
  n = document.F.Name.value;
  books = parseInt(document.F.Books.value);
  if(document.F.std[0].checked)
    bill = books * 100 - ((books * 100) * 20 / 100);
  else if(document.F.std[1].checked)
    bill = books * 100;
  alert("Hi " + n + "! Your bill is " + bill);
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
Enter your name: <INPUT TYPE="text" NAME="Name"><BR>
How many books have you purchased? <INPUT TYPE="text" NAME="Books"><P>
Students get 20% concession! Are you a student?
<INPUT TYPE="radio" NAME="std" VALUE=1 CHECKED> Yes
<INPUT TYPE="radio" NAME="std" VALUE=0>No <P>
<INPUT TYPE="button" NAME="Cal" VALUE="Calculate Bill" onClick="cal();">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "P18.html".
3. Double click the file. The web page will appear as follows:



4. Enter your name and number of books.
5. Select the appropriate radio button and click **Calculate Bill** button. The following message will appear.

10.5.6 TextArea Element

Text area is similar to text box but it may consist of multiple rows. It is used to get lengthy input from the user like comments and suggestions etc. <TEXTAREA> tag is used to create text area in a form. The methods, events and event handlers of TextArea element are same as text element.

10.5.7 Select and Option Elements

Select field is used to provide predefined list of items to the user. The user can choose one or multiple options from the list depending upon the type of list. **<SELECT>** tag is used to create a list. **<OPTION>** tag is used to create an item in the list.

Methods

Some important methods of Select element are as follows:

Method	Description
blur()	It is used to bring focus away from select element.
focus()	It is used to bring focus to the select element.
change()	It is used to change the option of select element.

Events

Some important events of Select element are as follows:

Event	Description
blur()	It occurs when Select element loses focus.
focus()	It occurs when Select element gains focus.
change()	It occurs when the value of Select element is changed.

Event Handlers

Some important event handlers of button element are as follows:

Event	Description
onBlur()	It is executed when Select element loses focus.
onFocus()	It is executed when Select element gains focus.
onChange()	It is executed when the value of Select element is changed.

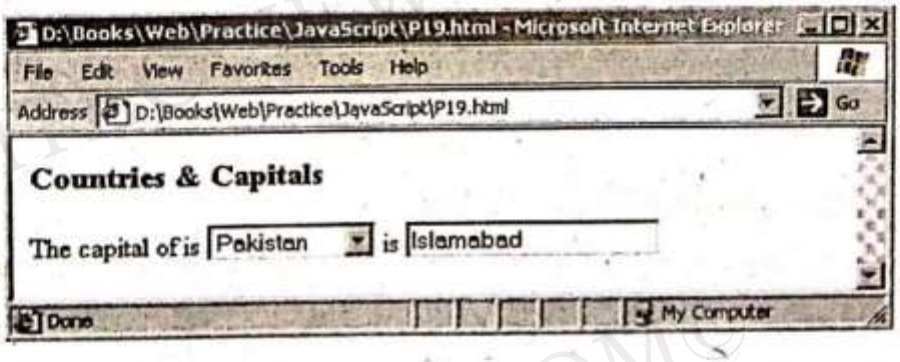
Practice 10.6

Write JavaScript code that displays a list with names of different countries. When the user selects a country from the list, the script displays the capital of that country in a text box.

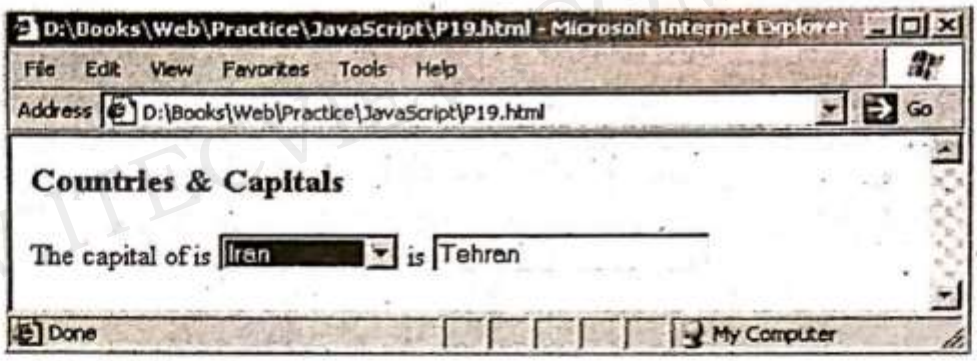
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function capital()
{
  c = new Array("Islamabad","Tehran","Kabul","Bejing","New Delhi");
  var i;
  i = document.F.Country.selectedIndex;
  document.F.Cap.value = c[i];
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
<H3>Countries & Capitals</H3><P>
The capital of is
<SELECT NAME="Country" onChange="capital()";>
<OPTION SELECTED>Pakistan
<OPTION>Iran
<OPTION>Afghanistan
<OPTION>China
<OPTION>India
</SELECT>
is <INPUT TYPE="text" NAME="Cap" VALUE="Islamabad"><BR>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "P19.html".
3. Double click the file. The web page will appear as follows:



4. Select Iran from the Select element. The page will appear as follows:



10.6 Form Validation

JavaScript is used to validate the contents of the forms. Form validation is a process of checking that proper information is entered in the form before it is submitted. For example, a field for entering email must be checked to ensure that the user has filled it. It should also be ensure that the user has entered a valid email address. Similarly, the validity of credit card number should also be checked.

10.6.1 Common Validation Types

Some important validation types are as follows:

1. Validating Email Addresses

The email validation is used to check the presence of a dot and an "@" sign in email address, It is also important to verify that there is a dot after the "@" sign.

2. Checking for Numeric Values

Many numeric values should be checked for specified format. For example, a credit card number can be checked for its specific length and format etc. Depending on the type of credit card, the number must contain either 13 or 16 digits and they must add up to a specified total. All characters must be numeric. The expiry date of credit card is also validated. Similarly, the value entered in the month field should be between 1 and 12 etc.

3. Checking for Blank Fields

A form may contain many fields. Some fields are compulsory to fill and some are optional. A validation process must check that all compulsory fields are filled by the user.

Practice 10.7

Write JavaScript code that displays a text box to input name and submit button. The script uses a function to validate the name.

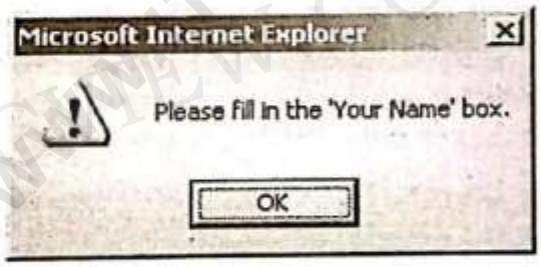
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function validate_form ( )
{
  valid = true;
  if ( document.contact_form.contact_name.value == "" )
  {
    alert ( "Please fill in the 'Your Name' box." );
    valid = false;
```

```
  }
  return valid;
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="contact_form" METHOD="post" ACTION="mailto:info@itseries.com.pk"
OnSubmit="return validate_form ( );">
<P>Your Name: <INPUT TYPE="text" NAME="contact_name"></P>
<P><INPUT TYPE="submit" NAME="send" VALUE="Send Details"></P>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form01.html".
3. Double click the file. The web page will appear as follows:

4. Press **Send Details** button without entering name. A message will appear as follows:



5. Now enter your name and press the button again. The form will be submitted.

How it Works?

The above form includes one text field and a **submit** button. The validation script ensures that the user enters name before the form is submitted. The page consists of a JavaScript function called **validate_form ( )** that performs the form validation.

The first part of the form is the **<FORM>** tag:

```
<FORM NAME="contact_form" METHOD="post" ACTION="mailto:info@itseries.com.pk"
OnSubmit="return validate_form ( );">
```

The form uses **post** method to send data. The **FORM** tag includes **OnSubmit** attribute to call JavaScript validation function **validate_form** when the form is submitted. The **return** statement is used to return **true** or **false** value from function to the browser. The **true** value means to carry on the submission process. The **false** value means to abort submission process. It means that the submission process can be aborted if the user has not filled the form properly.

The rest of the form prompts the user to enter name into a form field called **contact name** and adds a **Send Details** submit button:

The validate_form ( ) Function

The form validation function **validate_form ( )** is embedded in the **HEAD** tag. The first line of the function sets the value of **valid** variable to **true**.

```
valid = true;
```

The variable **valid** variable is used to keep track of whether the form is filled correctly or not. If checking process fails, the value of **valid** is set to **false** to abort submission process.

The next 5 lines check the value of **contact_name** field to verify that it has been filled. If the field is empty, a message appears in an **alert** box and variable **valid** is set to **false**. Finally, the value of **valid** is returned. If the value is **true**, the form will be submitted and if the value is **false**, the submission process will be aborted.

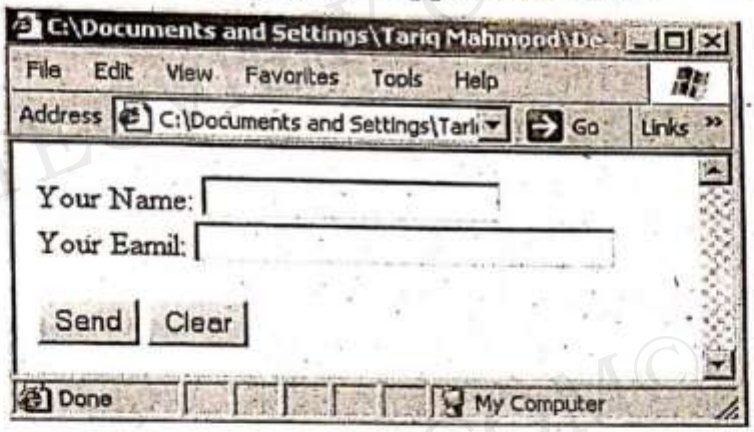
Practice 10.8

Write JavaScript code that displays a text box to input name and submit button. The script should use a function to validate name and email address.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function validate(formCheck)
{
  if (formCheck.name.value =="")
  {
    alert("Please provide your name:");
    formCheck.name.focus();
    return false;
  }
  var mail=formCheck.email.value
  if (mail.indexOf("@.") == -1)
  {
    alert("Please type a valid email:");
    formCheck.email.focus();

    return false;
  }
  return true;
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="form" ACTION="mailto:info@itseries.com.pk" METHOD="post">
  Your Name: <INPUT TYPE=text NAME="name" VALUE=""SIZE=20><BR>
  Your Eamil: <INPUT TYPE=text NAME="email" VALUE="" SIZE=30><P>
  <INPUT TYPE="submit" NAME="submit" VALUE="Send" onClick="return validate(form)"; >
  <INPUT TYPE="reset" NAME="reset" VALUE="Clear">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form02.html".
3. Double click the file. The web page will appear as follows:



4. Press **Send** button without entering name. A message will appear as follows:

5. Enter name and press **Send** button without entering email. A message will appear:

6. Enter an invalid email and press Send button. The above message will appear again.
7. Enter valid email and press the button. The form will be submitted.

How it Works?

The above practice uses a function *validate* which takes a parameter. The parameter represents the form name. It check name field using **if** statement. If the field empty, a message appear in alert box and the focus is set to this field. It also returns **false** to abort the submission process.

Secondly, email field is validated. The script checks the presence of @ and dot. If either is not present, a message appears, focus is set to email field and falise value is returned. If both fields are checked successfully, the function returns **true**. In this case, *onClick* event handler returns **true** and form is submitted.

Practice 10.9

Write JavaScript code that displays a calcultator.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
```

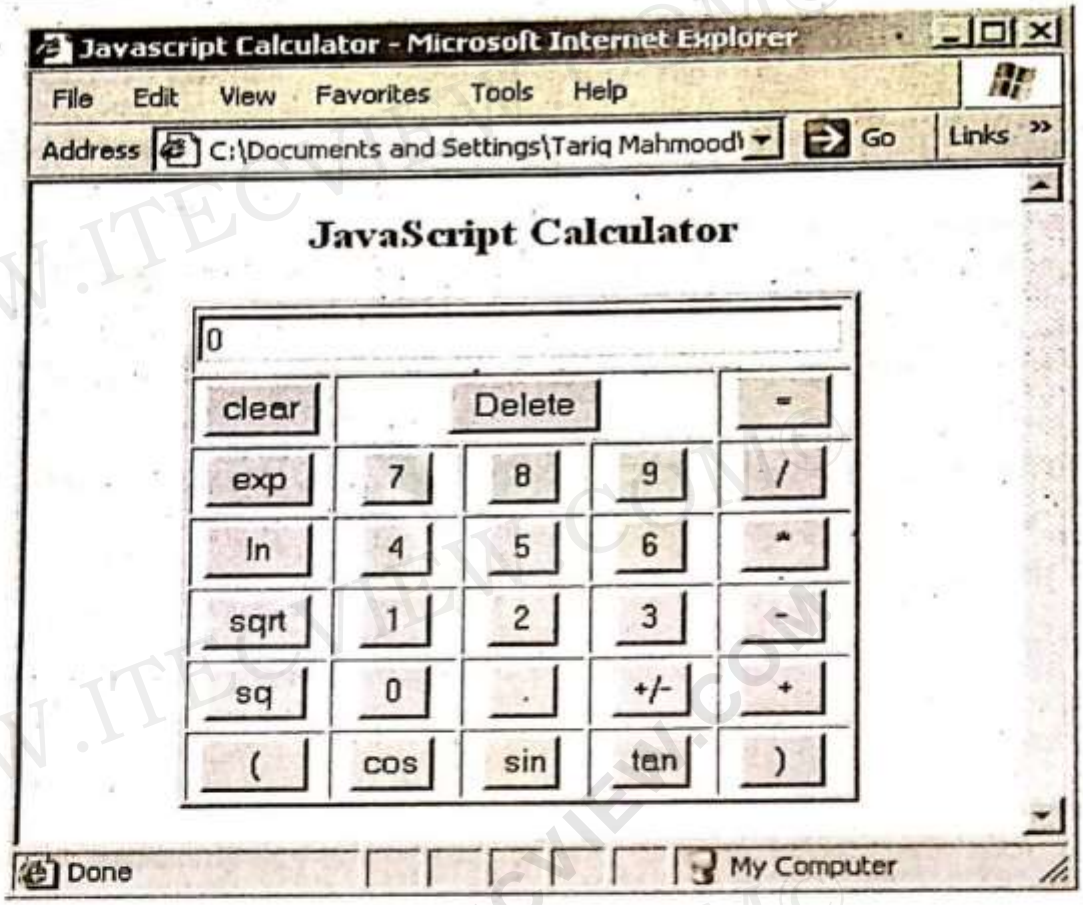
```
<TITLE>Javascript Calculator</TITLE>
<SCRIPT LANGUAGE="JavaScript">

function addChar(input, character)
{
  if(input.value == null || input.value == "0")
    input.value = character
  else
    input.value += character
}

function cos(form)
{
  form.display.value = Math.cos(form.display.value);
}

function sin(form)
{
  form.display.value = Math.sin(form.display.value);
}

function tan(form)
{
  form.display.value = Math.tan(form.display.value);
}

function sqrt(form)
{
  form.display.value = Math.sqrt(form.display.value);
}

function sq(form)
{
  form.display.value = eval(form.display.value) * eval(form.display.value)
}

function ln(form)
{
  form.display.value = Math.log(form.display.value);
}

function exp(form)
{
  form.display.value = Math.exp(form.display.value);
}

function changeSign(input)
{
  if(input.value.substring(0, 1) == "-")
    input.value = input.value.substring(1, input.value.length)
  else
    input.value = "-" + input.value
```

```
}
function equal(form)
{
  form.display.value = eval(form.display.value)
}

function delChar(input)
{
  input.value = input.value.substring(0, input.value.length - 1)
}

function chkValid(str)
{
  for (var i = 0; i < str.length; i++)
  {
    var ch = str.substring(i, i+1)
        if (ch < "0" || ch > "9")
        {
          if (ch != "/" && ch != "*" && ch != "+" && ch != "-" && ch != "." && ch != "(" && ch!=
")")
        {
          alert("invalid entry!")
          return false
        }
        }
  }
        return true
}
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3 align="center">JavaScript Calculator</H3>
<FORM NAME="calc"><center>
<TABLE CELLSPACING="1" width="100" height="50" border="2">
<TR>
 <TD COLSPAN=5 ALIGN="center" width=259 height=31>
  <INPUT NAME="display" VALUE=0 SIZE=38 MAXLENGTH=25>
 </TD>
</TR>
<TR>
 <TD ALIGN="center" width="55" height="31" >
  <INPUT TYPE="button" VALUE="clear" ONCLICK="this.form.display.value=0">
 </TD>
 <TD ALIGN="center" colspan="3" width="985" height="31">
  <INPUT TYPE="button" VALUE="Delete" ONCLICK="delChar(this.form.display)">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE="=" NAME="result"
ONCLICK="if(chkValid(this.form.display.value)) {equal(this.form) }"
 </TD>
</TR>
<TR>
```

```
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" exp " ONCLICK="if (chkValid(this.form.display.value)) {
exp(this.form)}">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 7 " ONCLICK="addChar(this.form.display, '7')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 8 " ONCLICK="addChar(this.form.display, '8')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 9 " ONCLICK="addChar(this.form.display, '9')">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE=" / " ONCLICK="addChar(this.form.display, '/')">
 </TD>
</TR>
<TR>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" ln " ONCLICK="if (chkValid(this.form.display.value)) {
ln(this.form) }">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 4 " ONCLICK="addChar(this.form.display, '4')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 5 " ONCLICK="addChar(this.form.display, '5')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 6 " ONCLICK="addChar(this.form.display, '6')">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE=" * " ONCLICK="addChar(this.form.display, '*')">
 </TD>
</TR>
<TR>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" sqrt " ONCLICK="if (chkValid(this.form.display.value)) {
sqrt(this.form) }">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 1 " ONCLICK="addChar(this.form.display, '1')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 2 " ONCLICK="addChar(this.form.display, '2')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 3 " ONCLICK="addChar(this.form.display, '3')">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE=" - " ONCLICK="addChar(this.form.display, '-')">
 </TD>
```

```
</TR>
<TR>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" sq " ONCLICK="if (chkValid(this.form.display.value)) {
sq(this.form)}">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" 0 " ONCLICK="addChar(this.form.display, '0')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" . " ONCLICK="addChar(this.form.display, '.')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" +/- " ONCLICK="changeSign(this.form.display)">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE=" + " ONCLICK="addChar(this.form.display, '+')">
 </TD>
</TR>
<TR>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" ( " ONCLICK="addChar(this.form.display, '(')">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE="cos" ONCLICK="if (chkValid(this.form.display.value)) {
cos(this.form) }">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" sin" ONCLICK="if (chkValid(this.form.display.value)) {
sin(this.form) }">
 </TD>
 <TD ALIGN="center" width="55" height="31">
  <INPUT TYPE="button" VALUE=" tan" ONCLICK="if (chkValid(this.form.display.value)) {
tan(this.form) }">
 </TD>
 <TD ALIGN="center" width="56" height="31">
  <INPUT TYPE="button" VALUE=" ) " ONCLICK="addChar(this.form.display, ')')">
 </TD>
</TR>
</TABLE>
</center>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Calculator.html".
3. Double click the file. The web page will appear as follows:



10.7 Advanced JavaScript

JavaScript provides advanced options to work with dates, images and windows etc.

10.7.1 Date Object

The Date object provides methods to manipulate data and time. The Date object can be created in two ways:

MyDate = new Date();

The above line creates a date object and assigns current data and time to it.

MyDate = new Date(Year, Month, Day, Hours, Minutes, Seconds, Milliseconds);

The above line creates a date object and assigns the specified values to it. The parameters must be given in the same order.

Methods of Date Object

Some important methods of Date object are as follows:

Method	Description
getDate()	Returns the day of the month from Date object.
getDay()	Returns the day of the week from Date object.
getMonth()	Returns the month of the year from Date object i.e. 0 for January, 1 for February and so on...
setMonth()	Sets the month of the year of Date object.
getYear()	Returns the year from Date object.

setYear()	Sets the year of Date object.
getFullYear()	Returns the year in four-digit form from Date object.
setFullYear()	Sets the year in four-digit form of Date object.
getTime()	Returns the time in milliseconds since midnight January 1, 1970 from Date object.
getHours()	Returns the hours from Date object.
getMinutes()	Returns the minutes from Date object.
getSeconds()	Returns the seconds from Date object.
setHours()	Sets the hours of Date object.
setMinutes()	Sets the minutes of Date object.
setSeconds()	Sets the seconds of Date object.

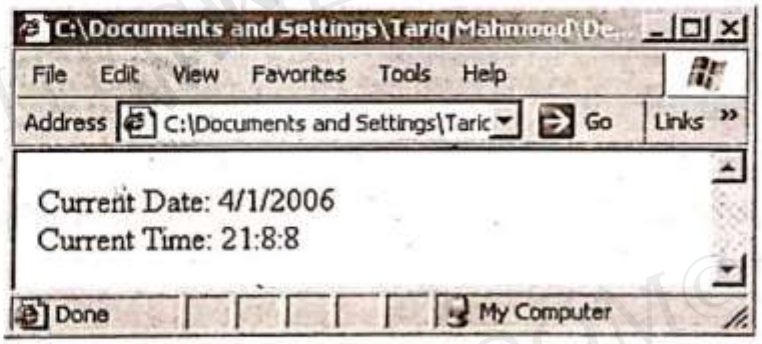
Practice 10.10

Write JavaScript code that displays current date and time using Date object.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
MyDate = new Date();
document.write("Current Date: " + MyDate.getDate() + "/" + MyDate.getMonth() + "/" +
MyDate.getYear() + "<BR>");
document.write("Current Time: " + MyDate.getHours() + ":" + MyDate.getMinutes() + ":" +
MyDate.getSeconds() + "<BR>");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Date.html".
3. Double click the file. The web page will appear as follows:



10.7.2 Image Object

The Image object is used to provide dynamic effect in web pages. The Image object can be created as follows:

```
myImage = new Image();
```

Properties of Image Object

Property	Description
src	Specifies the source of the image to be displayed.
complete	Specifies whether the image has completely loaded or not.
border	Specifies the border thickness of the image in pixels.

height	Specifies the height of image in pixels or percentage.
width	Specifies the width of image in pixels or percentage.
vspace	Specifies the size of image's left and right margins.
hspace	Specifies the size of image's top and bottom margins.
lowsrc	Specifies the URL of low-resolution image. High-resolution image takes time in loading. The low-resolution image can be displayed while high-resolution image is being loaded.

10.7.3 Window Object

The window object is used to open customized windows dynamically.

Opening Window

The **open()** method of window object is used to open a new window. The syntax of this method is as follows:

variable =.open(url, name, options)

where **url** specifies the file to be opened, **name** specifies the name of the new window and **options** specifies different attributes of the new opened window. Different attributes that can be specified by using **options** are as follows:

Option	Values	Description
toolbar	Yes/No	Specifies the toolbar of the window
menubar	Yes/No	Specifies the menu bar of the window
status	Yes/No	Specifies the status bar of the window
resizable	Yes/No	Specifies whether the window is resizable or not
width	Integer	Specifies the width of the window
height	Integer	Specifies the height of the window

Example

```
Open("http://www.itseries.com.pk", "IT Series", "menubar=no, toolbar=no, statusbar=no, width=300, height=200");
```

The values **Yes/No** can also be given as 0 and 1 as follows:

```
Open("http://www.itseries.com.pk", "IT Series", "menubar=0, toolbar=0, statusbar=0, width=300, height=200");
```

Closing Window

The **close()** method of window object is used to close the current window. The syntax of using this method is as follows:

```
window.close();
```

Practice 10.11

Write JavaScript code that displays a button on the page. When the user clicks the button, it displays another page in a popup window. The windows contains a button to close that window.

1. Open Notepad and type the following code:

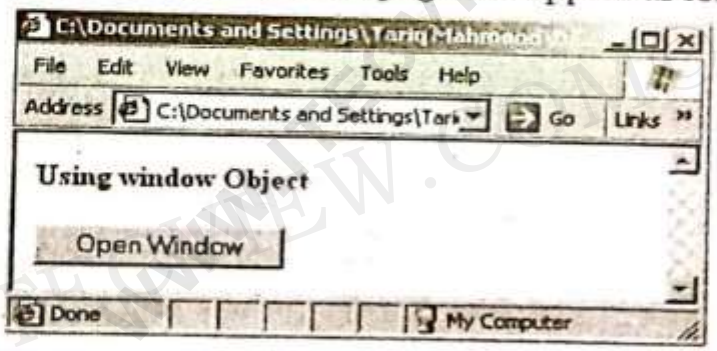
```
<HTML>
<HEAD>
```

```
<SCRIPT LANGUAGE="JavaScript">
function OpenNew()
{
window.open('win.html','PopUp','toolbar=0,location=0,directories=0,status=0,menubar=0,scroi
lbars=0,resizable=0,scrollbars=0,width=300,height=200,maximize=no');
}
</SCRIPT>
</HEAD>
<BODY>
<H4>Using window Object</H4>
<INPUT TYPE="button" VALUE="Open Window" onClick="OpenNew();">
</BODY>
</HTML>
```

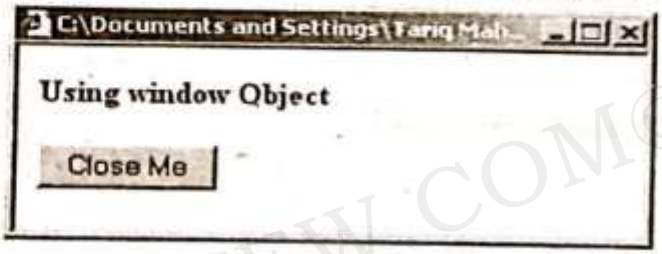
2. Save the file as "Open.html".
3. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Using window Object</H4>
<INPUT TYPE="button" VALUE="Close Me" onClick="window.close();">
</BODY>
</HTML>
```

4. Save the file as "win.html".
5. Double click "Open.html" file. The web page will appear as follows:



6. Click on **Open Window** button. A new window will appear as follows:



7. Click **Close Me** button to close the popup window.

Time Intervals

Time intervals can be used to execute a function repeatedly after specified time. The **setInterval()** function of window object is used to set time intervals. It executes an expression or function repeatedly for the specified number of milliseconds. The syntax of this function is as follows:

```
var = setInterval(Expression/Function, milliseconds);
```

var is the variable that stores the interval reference returned by **setInterval()** function.

Example

```
Ref = setInterval('alert("Hello World"), 10000);
```

The above example will display a message box after each ten seconds.

The **setInterval()** function executes an expression or function repeatedly after the specified time interval. It can be stopped by using **clearInterval()** function. The syntax of this function is as follows:

```
clearInterval(var);
```

var is the variable that stores the interval reference.

Example

```
Ref = setInterval('alert("Hello World"), 10000);
clearInterval(Ref);
```

The **setTimeout()** function is used to execute an expression or function after the specified time. Its syntax is similar to **setInterval()**. The difference between **setTimeout()** and **setInterval()** is that **setTimeout()** executes the expression or function only once.

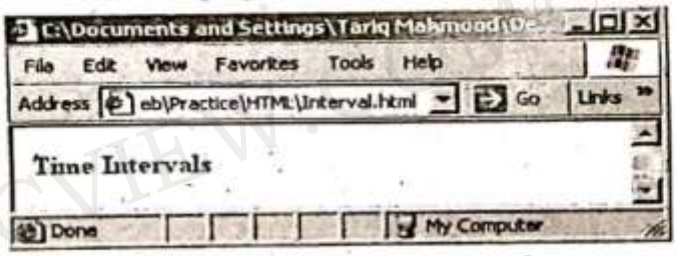
Practice 10.12

Write JavaScript code that displays a message in a message box after each five seconds.

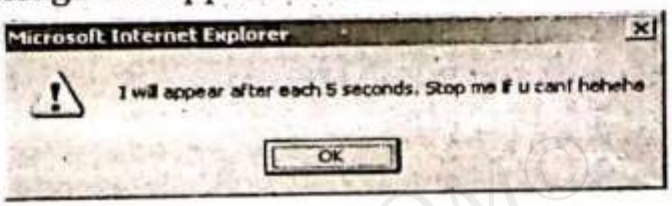
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Time Intervals</H4>
<SCRIPT LANGUAGE="JavaScript">
function msg()
{
  alert("I will appear after each 5 seconds. Stop me if u can! hehehe");
}

ref = setInterval('msg();', 5000);
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Interval.html".
3. Double click the file. The web page will appear as follows:



4. The following message will appear after each five seconds.



10.7.4 History Object

The history object refers to the list of documents that have been displayed by the current window or frame. It can be used to navigate back and forth through this list. It provides the following methods:

Method	Description	Example
back()	It loads the previous URL in the history object.	history.back()
forward()	It loads the next URL in the history object.	history.forward()
length	It specifies the length of history object.	history.length
go()	It loads the specified URL. It accepts a number as parameter. Positive number specifies the forward pages and negative number specifies the backward pages.	history.go(3) history.go(-2)

10.7.5 String Object

The string object is used to manipulate string. A string object can be declared in two ways as follows:

```
Str = "Hello Strings"
```

or

```
Str = new String("Hello Strings")
```

The string object provides the following methods:

Method	Description
toLowerCase()	It converts the string to lower case.
toUpperCase()	It converts the string to upper case.
substring(index)	It returns substring from string starting from specified index.
substring(index1, index2)	It returns substring from index1 to index2.
split(separator)	It separates the string into array of substrings based on specified separator character.
charAt(index)	It returns the character at specified index.
charCodeAt(index)	It returns the unicode of character at specified index.
indexOf(substring)	It returns the index at which the substring is located.

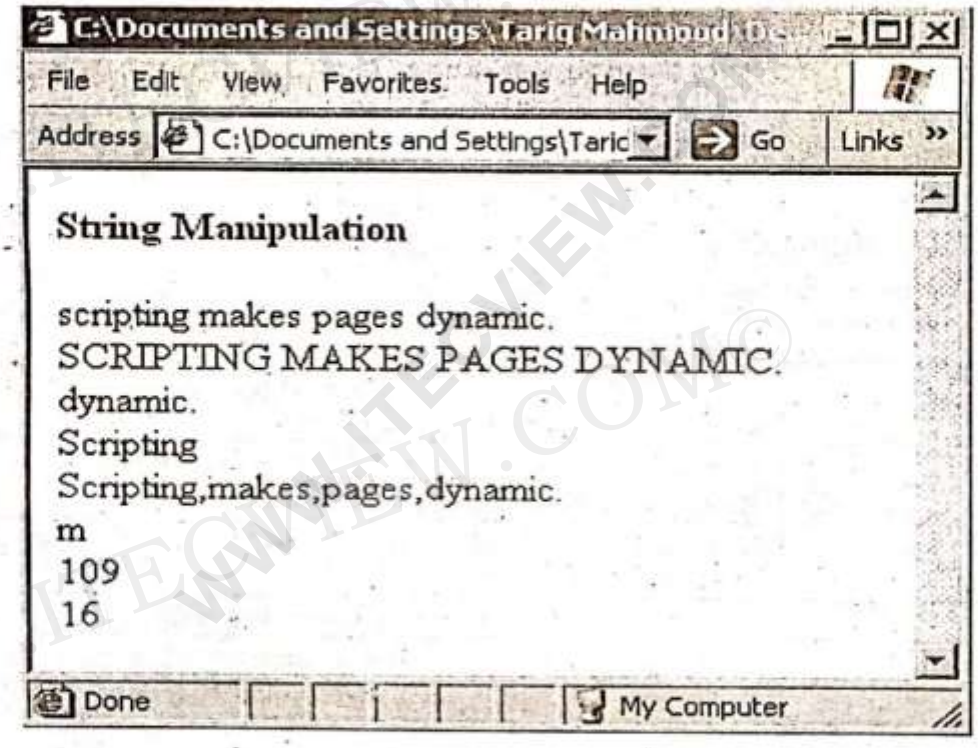
Practice 10.13

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>String Manipulation</H4>
<SCRIPT LANGUAGE="JavaScript">
str = new String("Scripting makes pages dynamic.");
document.write(str.toLowerCase());
document.write("<BR>");
document.write(str.toUpperCase());
document.write("<BR>");
document.write(str.substring(22));
document.write("<BR>");
document.write(str.substring(0,9));
```

```
document.write("<BR>");
document.write(str.split(' '));
document.write("<BR>");
document.write(str.charAt(10));
document.write("<BR>");
document.write(str.charCodeAt(10));
document.write("<BR>");
document.write(str.indexOf('pages'));
document.write("<BR>");
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "String.html".
3. Double click the file. The web page will appear as follows:



Exercise Questions

1. Write a Web page that displays four buttons to represent four colors Red, Green, Blue and White. The background color of document is changed according to the button clicked by the user.
2. Write a Web page that contains two text boxes and a button. When the user clicks the button, the contents of two text boxes should be swapped. The text boxes should appear on the same line. The text boxes and button should be in the center of page.
3. Write a Web page that displays a form to input name, email, phone and address from the user. A script should ensure that all fields are filled and the user has entered a valid email address. The length of phone number should not exceed 7 digits.

Short Questions

Q.1. Define script.

Script is an interpreted program that runs on client computer. It is written in scripting language.

Q.2. Differentiate between client-side and server-side scripts.

The difference between client-side and server-side scripts is as follows:

- **Server-Side Scripts:** Server-side scripts are stored and executed on servers. These scripts are used to provide user interaction. Data from the client machine is sent to the Web server. Server-side scripts process the request and send the required page to the client browser.
- **Client-Side Scripts:** Client-side scripts are executed on client computers. The browsers support the use of scripts. These scripts can be embedded in HTML documents. The browser interprets and executes the code embedded in the page.

Q.3. Why is it advantageous to combine JavaScript with other technologies?

JavaScript is the default scripting language of most browsers. It is almost universally supported across browser platforms. It has the ability to interact with HTML code. Combining JavaScript with other languages provides the advantages of both technologies.

Q.4. List some advantages of JavaScript.

Some advantages of using JavaScript are as follows:

- **Interpreted Language:** JavaScript is an interpreted language. It requires no compilation process so no compiler is required.
- **Easy to Learn:** The syntax of JavaScript is very easy. Any person can learn it very easily and use it to develop dynamic and interactive websites.
- **Easy to Debug and Test:** JavaScript code is interpreted line by line. The errors are indicated along with line number. It is very easy to find error in the code, correct it and test it again.
- **Event-Based Programming:** JavaScript is an event-based language. It means that different code segments are executed when certain event occurs.
- **Procedural Capabilities:** JavaScript provides all capabilities of a procedural language. It provides condition checking, loops and branching facilities that can be executed in a web page.
- **Platform Independence:** JavaScript is platform-independent language. Any JavaScript code is executed on different types of hardware.

Q.5. List some uses of JavaScript.

Some important uses of JavaScript are as follows:

- **Input Validation:** JavaScript can be used to validate the input entered by the user.
- **Mouse Rollover Effects:** JavaScript can be used to create different buttons with interesting mouse rollover effects.
- **Popup Windows:** JavaScript can be used to create popup windows. These windows are normally used to display important announcements, offers and news etc.
- **Dynamic Contents:** JavaScript can be used to generate dynamic contents in a website. For example, the current date and time can be displayed on web page.
- **User Interaction:** JavaScript can be used to interact with the user. The input entered by the user can be processed and proper message can be displayed to the user.

Q.6. What is a variable?

Variables are symbolic names that refer to different values. Variables are used to store values.

Q.7. What are different rules for declaring variables?

- The first character of variable must be a letter or underscore _.
- Blank spaces are not allowed in variable names.
- Variable may include letters, numbers and underscore (_).
- Both upper and lower cases are allowed.

- Special symbols cannot be used as variable name.
- Reserved word cannot be used as variable name.

Q.8. What is variable initialization?

The process of assigning a value to a variable at the time of declaration is known as variable initialization. The equal sign = is used to initialize a variable. Variable name is given on left side and value is given on the right side of equal sign.

Q.9. What are literals? List different types of literals.

Literals can be used to represent values in JavaScript. Literals are fixed values that cannot be changed during execution of the code. Different types of literals are as follows:

- **Floating-Point Literals:** 3.1415, -3.1E12, .1e12, 2E-12.
- **Integer Literals:** 42, 0xFFF and -345.
- **String Literals:** "Allah", 'Allah' and "1234".

Q.10. What is an expression?

A statement that evaluates to a value is called an **expression**. An expression gives a single value. An expression consists of **operators** and **operands**.

Q.11. List different types of expression in JavaScript.

JavaScript has the following types of expressions:

- **Arithmetic Expressions:** These expressions evaluate to a number such as 3.141.
- **String Expressions:** These expressions evaluate to a character string such as "Usman"
- **Logical Expressions:** These expressions evaluate to **true** or **false**.

Q.12. What is operator? List different categories of operators in JavaScript.

An **operator** is a symbol that performs some operation. The value on which the operator performs operation is called **operand**. There are two categories of operators:

- **Binary Operator:** It is a type of operator that works with two operands.
- **Unary Operator:** It is a type of operator that works with one operand.

Q.13. What is an assignment operator?

An assignment operator used to assign the value of right operand to left operand. The basic assignment operator is equal (=).

Q.14. What are different arithmetic operators?

Arithmetic operators process numerical values and return a single numerical value. The standard arithmetic operators are addition (+), subtraction (-), multiplication (*) and division (/).

Q.15. What do you mean by operator precedence?

The order in which different types of operators in an expression are evaluated is known as operator precedence. It is also known as hierarchy of operators.

Q.16. What is the order of precedence in JavaScript?

- Any expression given in parentheses is evaluated first.
- Then multiplication * and division / operators are evaluated.
- Then plus + and minus - operators are evaluated.
- In case of parentheses within parentheses, the expression of the inner parentheses will be evaluated first.

Q.17. What is meant by operator associativity?

The order in which operators of same precedence are evaluated is known as operator associativity. If an expression contains some operators that have same precedence level, the expression is evaluated either from left-to-right or right-to-left.

Q.18. Why is the order of precedence important in complex calculations?

The order of precedence determines which operations will take place first when calculating an equation. If the operations are completed in the wrong order, a wrong answer can easily be generated.

Q.19. What is type casting? List its types.

The process of converting the data type of a value during execution is known as **type casting**. Type casting can be performed in two ways:

- **Implicit Type Casting:** Implicit type casting is performed automatically by JavaScript. If the data types of operands are different, the value with lower data type is converted into higher data type.
- **Explicit Casting:** Explicit casting is performed by programmer. It is performed by using built-in function such as parseInt.

Q.20. What is conditional statement?

A statement or set of statements that is executed when a particular condition is true and ignored when the condition is false is called conditional statement.

Q.21. What are relational operators?

A comparison operator compares its operands and returns a logical value based on whether the comparison is true or false. The operands can be numerical or string values.

Q.22. What is relational expression?

A relational expression is a statement that uses relational operators to compare two values. The result of a relational expression can be true or false.

Q.23. What is an "if" statement?

if statement is a decision-making statement. It is used to execute or skip a statement or set of statements by checking a condition.

Q.24. What is an "if...else" statement?

if else statement is another type of if statement. It executes one block of statement(s) when the condition is true and the other when it is false. In any situation, one block is executed and the other is skipped.

Q.25. What is nested "if" statement?

An if statement within an if statement is called nested if statement. In nested structure, the control enters into the inner if only when the outer condition is true.

Q.26. What are logical operators?

Logical operators are used to evaluate compound conditions. A type of comparison in which more than one conditions are evaluated is called compound condition.

Q.27. List different types of logical operators in JavaScript?

The logical operators in JavaScript are as follows:

- **AND Operator:** The AND operator is (&&)is used to evaluate two conditions. It returns **true** result if both conditions are true. It returns false result if any condition is false.
- **OR Operator:** The OR operator (| |) is used to evaluate two conditions. It returns **true** result if either condition is **true**. It returns **false** if both conditions are **false**.
- **NOT Operator:** The NOT operator (!) is used to reverse the result of a condition. It produces **true** result if the condition is **false**. It produces **false** result if the condition is **true**.

Q.28. What is switch statement?

The switch statement is another conditional structure. It is a good alternative of nested if-else. It can be used easily when there are many choices available and only one should be executed. Nested if becomes very difficult in such situation.

Q.29. Why is it important to specify a default value for a switch statement?

Depending on how you define your case statements, it is entirely possible that no case will evaluate as true. When you create a default value, you can ensure that one or more statements will be executed, even if none of the cases evaluate as true.

Q.30. What is conditional operator?

The conditional operator is the only JavaScript operator that takes three operands. It is also called ternary operator. The operator can have one of two values based on a condition.

Q.31. What is loop? Why is it used?

A statement or a set of statements that is executed repeatedly is known as **loop**. The structure that repeats a statement(s) is known as **iterative, repetitive** or **looping structure**. Loops are basically used for two purposes:

- To execute a statement or number of statements for a specified number of times.
- To use a sequence of values such as numbers from 1 to 10.

Q.32. List different types of loops in JavaScript.

There are three types of loops available in JavaScript:

- while loop
- do-while loop
- for loop

Q.33. What is a function? List its types.

A function is a named block of code that performs some action. The statements written in a function are executed when it is called by its name. Each function has a unique name. JavaScript provides the following types of functions:

- **User-defined Functions:** It is a type of function written by the programmer.
- **Built-in Functions:** It is a type of function that is available as a part of JavaScript.

Q.34. What is meant by function declaration?

Function declaration consists of the **function** keyword, followed by the name of the function, a list of arguments to function in parentheses and separated by commas and the JavaScript statements that define the function enclosed in curly braces { }.

Q.35. What is function call?

The statement that activates a function is known as function call. A function is called with its name. Function name is followed by necessary parameters in parentheses.

Q.36. What happens when a function is called?

When a function is called, the control moves to the function that is called, all statements in the function body are executed and the control returns back to the calling function.

Q.37. How can a function return value?

A function can return a single value. The keyword return is used to return the value back to the calling function. When the return statement is executed in a function, the controls moves back to the calling function along with the returned value.

Q.38. What is meant by scope and lifetime of variable?

The area where a variable can be accessed is known as scope of variable. The time period for which a variable exists in the memory is known as the lifetime of variable.

Q.39. What is local variables?

A variable declared inside a function is known as local variable. Local variable can be used only in the function in which it is declared. The lifetime of local variable starts when the control enters the function in which it is declared. Local variable is automatically destroyed when the control exits from the function and its lifetime ends.

Q.40. What is global variable?

A variable declared outside any function is known as global variable. Global variable can be used by all functions in the program. Global variables exist in the memory as long as the script is running. These variables are destroyed from the memory when the program terminates.

Q.41. What is an array?

An array is a collection of consecutive memory locations. Each memory location is called an element of array. The number of elements in an array is called the length of array. All elements have one collective name.

Q.42. When are arrays useful? When is it better to use array instead of several variables?

Arrays are useful for managing lists or any elements that have a common identity. Arrays also work well because we can have a changing number of elements which are essentially recorded under

the same name. Arrays are often easier to manage than multiple variables because a number of array methods exist which make it easier to manipulate the information stored in the array.

Q.43. What do you about JavaScript Object Model?

A web browser works with different objects while displaying web pages. These objects have different properties and functions. JavaScript-enabled browsers use Document Object Model (DOM) to recognize these objects.

Q.44. What is an object?

An object is a collection of properties and methods. A web page consists of many objects. HTML is used to created different objects such as textbox, radio buttons and list boxes etc.

Q.45. List different types of browser objects.

Different browser objects are as follows:

- **Navigator:** It is used to access information about the browser.
- **Window:** It is used to access a browser window or frame.
- **Document:** It is used to access the document loaded in a window.
- **Location:** It refers to the URL the browser is visiting.
- **History:** It is used to maintain the history of URLs accessed in the browser.
- **Screen:** It is used to access the information about the computer screen.

Q.46. What is an event?

An event occurs when something happens or an action is performed. Events are used to write JavaScript code that reacts to certain situations.

Q.47. What is an event handler?

An event handler is a piece of code that is executed when an event occurs. Each even handler is associated with an event.

Q.48. List some important event handler in JavaScript.

Some important event handlers in JavaScript are onClick, onDblClick, onChange, onFocus, onMouseOver and onLoad etc.

Q.49. Why is form object useful in JavaScript?

We can create *jump menus* where users can click on an item in a pull down list and the page redirects automatically to another page. Form validation allows the server to do less work by only sending information when we are reasonably sure the user has entered the information correctly.

Q.50. Which methods can be used to submit or reset a form?

A designer can call the submit() or reset() methods of the related form object.

Q.51. Which property of checkbox indicates the value of the selected checkbox?

The value property

Q.52. Why is the image object useful in JavaScript? How can you use the image object in JavaScript to improve the usability of an HTML Web page?

The image object represents the images contained in the page. We can use the image object to create rollover effects, as one example. We could also preload images and reduce the amount of time needed to see new images when the user makes a decision or chooses a new option.

Q.53. What is meant by form validation?

Form validation is a process of checking that proper information is entered in the form before it is submitted. For example, a field for entering email must be checked to ensure that the user has filled it. It should also be ensure that the user has entered a valid email address.

Q.54. List some common validation types.

Some important validation types are as follows:

- **Validating Email Addresses:** The email validation is used to check the presence of a dot and an "@" sign in email address. It is also important to verify that there is a dot after the "@" sign.
- **Checking for Numeric Values:** Many numeric values should be checked for specified format. For example, a credit card number can be checked for its specific length and format etc.

- **Checking for Blank Fields:** Some fields are compulsory to fill and some are optional. A validation process must check that all compulsory fields are filled by the user.

Q.55. Describe two solutions where it is useful to validate data submitted in form.

It may be very important to get the user's phone number, in case we have problems filling the order. If the user does not enter their phone number correctly, it could lead to a bad customer experience. Another example: if a user doesn't enter his or her address information correctly the package cannot be delivered.

Q.56. Why JavaScript is used for validation because it is a client-side language?

Performing validation on the client is important for two reasons. First, by validating data before it is submitted to the server. It reduces the work of server. Validating data on the client rather than the server also reduces traffic on the network.

Q.57. When working with the history list to what does history.go(-1) refer?

The previous page in the history list, assuming you can go back.

Q.58. When working with the history list to what does history.go(1) refer?

The next page in the history list, assuming you can go forward.

Q.59. Name two techniques that you can use to reload the current page.

history.go(0) and window.location.reload()

Q.60. What is a confirm dialog box and how is it used?

With a confirm dialog box, you can display a message to the user with OK and Cancel buttons. This type of dialog box can be used to ask the user a question.

Q.61. What is a prompt dialog box and how is it used?

A prompt dialog box presents a message and allows users to enter information in response to the message. The prompt() method has two mandatory parameters. The first parameter is the message to display and the second parameter is a default reply.

Q.62. What is a window?

A window is the object at the top of the browser object model for all scriptable browsers.

Q.63. What is string data?

String data consists of an ordered set of characters. It resembles an array of characters because each character in the string can be accessed by an index.

Q.64. What is the opposite of the method toLowerCase()?

Opposite of the method toLowerCase() is the method toUpperCase(). The former converts a string to lower case, while the latter converts a string to upper case.

Q.65. State the order of evaluation of operators in each of the following statements, and show the value of a after each statement is performed.

a) y = 7 + 3 * 6 / 2 - 1;

Ans: * is first, / is second, + is third, and - is fourth. Value of y is 15.

b) a = 2 % 2 + 2 * 2 - 2 / 2;

Ans: % is first, * is second, / is third, + is fourth, - is fifth. Value of a is 3.

c) a = (3 * 9 * (3 + (9 * 3 / (3))));

Ans: a = (3 * 9 * (3 + (9 * 3 / (3))));

5 6 4 2 3 1

Value of a is 324.

Q.66. Identify and correct the errors in each of the following segments of code:

```
a) while ( c <= 5 ) {
product *= c;
++c;
```

Ans: Error: Missing the closing right brace of the while body.

Correction: Add closing right brace after the statement ++c;

```
b) if ( gender == 1 )
document.writeln( "Woman" );
else;
```

```
document.writeln( "Man" );
```

Ans: Error: The semicolon after else results in a logic error. The second output statement will always be executed.

Correction: Remove the semicolon after else.

Q.67. What is wrong with the following while repetition statement?

```
while ( z >= 0 )
sum += z;
```

The value of the variable z is never changed in the body of the while statement. Therefore,if the loop-continuation condition (z >= 0) is true, an infinite loop is created. To prevent the creation of the infinite loop, z must be decremented so that it eventually becomes less than 0.

Q.68. Identify and correct the errors in each of the following segments of code

```
a) if ( age >= 25 );
document.writeln( "Age greater than or equal to 25" );
else
document.writeln( "Age is less than 25" );
```

Ans: Error: The semicolon after if (age >= 25) will cause a dangling-else error. The semicolon ends the if...else statement prematurely, leaving the else clause out of the flow of execution.

Correction: Remove the semicolon after if (age >= 25).

```
b) var x = 1, total;
while ( x <= 10 ) {
total += x;
++x;
}
```

Ans: Error: total is not initialized, causing a logic error the first time we attempt to add x to total. The final value of total is thus NaN.

Correction: Assign total a value (e.g., total = 10) in the declaration or as a separate statement before the while statement.

Q.69. What is the output of the following code?

```
var a=10;
document. write(--a);
document. write(++a);
document. write(a++);
document. write(a--);
document. write(a++);
```

Ans.

```
9
10
10
11
10
```

Q.70. What will be the output of the following statements?

a) 10 + 1010s

Ans: Error

b) 10 + "1010s"

Ans: "101010s"

c) "123.45" + "12.3'

Ans: "123.4512.3"

d) "a" + "b" + "=" + "c"

Ans: "ab=c"

Q.71. What is displayed in the message box when each of the given JavaScript statements is performed? Assume that a = 2 and b = 3.

a) window.alert("a = " + a);

Ans: a = 2

b) window.alert("The value of a + a is " + (a + a));

Ans: The value of a + a is 4

c) window.alert("a =");

Ans: a =

d) Window.alert((a + b) + " = " + (b + a));

Ans: 5 = 5

Q.72. What are the values contained in the variables a, b, c and d?

```
var a, b, c, d;
a = 10;
b = a* 5;
c = b / 10;
d = c –1;
```

Ans.

a=10

b=50

c=5

d=4

Q.73. Find the errors in the following codes:

```
a) if ( a > b );
   a = b;
else
   b = a;
b) if ( a > b )
      a = b;
else
   b = a;
c) if (a > b)
   a = b
else
   b = a;
d) if ( x !=0 )
      a = a / x
```

Ans. a) Semicolon is not allowed after condition

b) Correct

c) Semicolon missing after the statement a=b

d) Semicolon missing after the statement a=a/x

Q.74. Do the following expressions evaluate to true or false?

```
(3<4) || (3>4)
(3!=3) && (4 == 4)
(15>=15) && (16==16) || (14<2)
(15>=15) && ((16==16) || (14<2))
((15>=15) && (16==16)) || (14<2)
```

Ans.

True

False

True

True

True

Q.75. Consider the following statement

X = (((10*5)*(7-7)) > 4) ? 1 : 0

What is the value assigned to the variable X ?

Ans. Value 0 is assigned to the variable X.

WWW.ITECVIEW.COM©

Q.76. Consider the following code

```
while ( i < 15 )
{
        i + =2;
}
```

How many times will the while loop get executed?

Ans. 8 times

Q.77. Consider the following code

```
cnt = 0
for (i=0; i <3; i++)
{
for(j=0; j < 2; j++)
{
cnt++
}
}
```

What is the value of the variable cnt, after executing the above piece of code?

Ans : Value of cnt is 6.

Q.78. What is wrong with the following while repetition statement?

```
while ( z >= 0 )
sum += z;
```

The value of the variable z is never changed in the body of the while statement. Therefore,if the loop-continuation condition (z >= 0) is true, an infinite loop is created. To prevent the creation of the infinite loop, z must be decremented so that it eventually becomes less than 0.

Q.79. Point out the errors if any in the code given below. If there are errors, then make the required changes.

```
var i=0, j=0;
while (i<15)
{
document.write("Hello World");
j=i+1;
}
```

Ans. i value never gets changed and hence the execution of the loop will never get terminated.

Q.80. What will be the contents of myArray, after executing the code given below?

```
myArray=new Array("M", "A", "L", "A", "Y", "A", "L", "A", "M");
myArray.reverse();
```

Ans. Same as myArray originally

Q.81. What will be the output of the following code?

```
<HEAD>
<SCRIPT LANGUAGE = "JavaScript">
num = 0;
function handler()
{
        if(num==1)
        {alert("Welcome")}
        else
        {alert("Hello World")}
}
</SCRIPT>
</HEAD>
<BODY>
```

```
<A Href = "Hello.html" onMouseOver='num++; handler()'> Place Your Mouse Cursor Over
This </A>
</BODY>
```

Ans. The message Hello.html gets displayed as an alert box message. When the mouse cursor is brought over it the message is replaced by Hello.html since the value of num is 0.

Q.82. How can you disable the click event associated with link elements?

Ans. By specifying <A HREF="linkname.html" onClick='return false'>

Multiple Choice

1. **JavaScript is which of the following?**
   a. Interpreted b. Object-based c. Client-side programming d. All
2. **When JavaScript code is embedded in HTML file, responses to users are initiated:**
   a. On client browser b. On server c. Both a and b d. None
3. **JavaScript syntax is:**
   a. Case-sensitive b. Space sensitive c. Both a and b d. None
4. **How many JavaScript statements can be embedded between tags?**
   a. One b. Ten c. Eighty d. Unlimited
5. **How does JavaScript interpreter identify that a script is being used in Web page?**
   a. <JavaScript> tag b. <script> tag c. <JAVAscript> tag d. <HTMLScript> tag
6. **Which of the following is written correctly?**
   a. <script language="JavaScript"> b. <script language="JAVAScript">
   c. <script language="javascript"> d. All
7. **What is meant by the // symbol in JavaScript?**
   a. It will create an event handler. b. It will create a single line comment.
   c. It will create a multi-line comment. d. Both b and c
8. **In *document.write("Hello");* which part of the statement is a method?**
   a. "Hello" b. write () c. document d. Both b and c.
9. **A reserved word is a:**
   a. Keyword in JavaScript language b. Variable name that is already declared
   c. Both a and b d. Neither a nor b
10. **Which of the following is NOT a reserved word?**
    a. dynamic b. return c. true d. static
11. **Which of the following is NOT a valid opening comment element?**
    a. <!-- b. // c. /* d. @@
12. **A statement should be terminated by which character?**
    a. : b. ; c. * d. +
13. **JavaScript applications on the client side are largely:**
    a. Function-driven. b. Action-driven.
    C. Event-driven. d. Executed without user interaction.
14. **Which of the following types of values is recognized by JavaScript?**
    a. null b. Null c. NULL d. All
15. **A memory location with some data that will change is:**
    a. Constant. b. Variable. c. Named constant. d. Symbolic constant.
16. **Which is true about a variable?**
    a. Name and data value can both change b. Name can change, but data value cannot
    c. Name cannot change, but data value can d. Name and data value both cannot change

17. Which is not a rule for naming variables?
a. Use a descriptive name for the variable
b. Start the name of a variable with a letter.
c. Use nothing but letters, digits, or the underscore character.
d. All of the above

18. Variable names can not begin with a(n):
a. Number. b. Underscore. c. Upper-case letter. d. Lower-case letter.

19. Which of the following are valid variable names?
a. If b. Var c. weather d. All

20. Which statement is true about a data type?
a. A data type determines the kind of data a variable can store.
b. A data type has no effect on how data is stored in memory.
c. A data type has no impact on how fast your application will run.
d. A data type is always handled by the computer.

21. Which statement is true about data types?
a. Each data type has no memory requirements
b. Each data type has different memory requirements.
c. Each data type has the same memory requirements.
d. None of the above

22. Strings can include which of the following characters?
a. a b. @ c. + d. All

23. Which of the following is a data type?
a. String b. 12 c. = d. city

24. Concatenations of strings use which of the following symbols?
a. + b. - c. & d. ;

25. JavaScript determines a variable's "type" when:
a. Declaration is made b. Program starts c. Value is assigned to it d. All

26. In JavaScript, these are fixed values, not variables.
a. Operators b. Functions c. Expressions d. Literals

27. In JavaScript, which of the following is NOT a type of literal value?
a. Integers b. Floating points c. Strings d. Variables

28. The process of giving a variable its starting value is called:42
a. Declaring b. Initializing c. Naming d. None

29. An expression consists of:
a. Operators b. Operand c. Both a and b d. None

30. An expression can be a:
a. Constant. b. Variable.
c. Combination of constants, variables and arithmetic operators d. All of the above.

31. All of the following are valid expressions EXCEPT:
a. Sales – Revenues b. Mpg, Gallons. c. Pi * Radius d. M/n.

32. An assignment statement will:
a. Perform a calculation. b. Store the results of a calculation.
c. Display the results of a calculation. d. Both a and b.

33. Which of the following is arithmetic operators except
a. + b. - c. % d. >

34. Which of the following operators is used to assign a value to a variable?
a. > b. + c. = d. *

35. Which of the following is correct syntax for writing an assignment statement?
a. Variable = right_side b. Constant = right_side c. Constant = variable d. None

36. Which of the following is valid assignment statement
a. a = 100; b. c = a + b; c. x = c – d + 10; d. All of these

37. Which of the following variable string assignments is NOT valid?
a. name = "ITSERIES"; b. name ="IT"+"SERIES";
c. name = 'ITSERIES'; d. name='ITSERIES";.

38. The statement: var xyz = 10; causes the type of the variable xyz to be:
a. String. b. Numeric. c. Integer. d. Real.

39. In statement: *var xyz = "Pakistan";* the type of variable xyz is interpreted as:
a. Numeric. b. Character. c. String. d. Boolean.

40. Which of the following are not valid types of expressions in JavaScript?
a. Arithmetic expressions b. String expressions
c. Logical expressions d. Null expressions

41. In 5+3, the symbol + is called:
a. Expression b. Operator c. Assignment d. Argument

42. In JavaScript, what is needed in an expression to relate values together?
a. Operator b. Conjunctive c. Function d. Action

43. Which of the following is NOT a basic JavaScript operator?
a. Logical b. Comparison c. Evaluation d. Arithmetic

44. Which of the following operators works with one operand
a. Unary b. Binary c. Ternary d. None

45. Which of the following operators works with two operand
a. Unary b. Binary c. Ternary d. None

46. Which of the following variable assignments is NOT valid?
a. amount=70; b. amount = 110; c. amount = 50p; d. amount = "70";.

47. The statement I += 3 has the same effect as:
a. I = I+3. b. I = 3. c. I-3 = I. d. I3.

48. Which of the following operator is used to increase the value of a variable?
a. ++ b. +- c. >> d. None

49. Which of the following operator is used to decrease the value of a variable?
a. — b. -+- c. << d. None

50. Which choice has the highest order of precedence when computing an expression?
a. () b. * c. / d. +

51. What is the value of the expression ((1+4+4)/(3*3)-(3+6)) ?
a. –8 b. 9 c. 12 d. 16

52. What is the value of the expression ((5+1)/((2+2+2)/2))
a. 2 b. 6 c. 12 d. 9

53. What is the value of the expression (1*2*3*4+5)
a. 29 b. 30 c. 33 d. 40

54. Which escape sequence can be used to begin a new line in JavaScript?
a. \a b. \b c. \m d. \n

55. The three programming structures are:
a. Sequence, decision, and repetition. b. Process, decision, and alternation.
c. Sequence, definition, and process. d. Relation, comparison, and process.

56. Which programming structure will execute your program statements in order?
a. Relation b. Decision c. Sequence d. Repetition

57. Another term for a computer making a decision is:
a. Sequential. b. Selection. c. Repetition. d. Iteration.

58. You can use a decision statement to:
a. Run a series of statements if a test fails.
b. Test a series of conditions.
c. Test whether a condition is true or false.
d. All of the these

59. Operators that are used to compare operands and decide whether the relation is true or false are called:
a. Arithmetic operators
b. Logical operators.
c. Syntax operators.
d. Comparison operators

60. A relation expression produces results as:
a. true
b. false
c. Either a or b
d. Any.value

61. Which of the following relation operator represents Greater-then-or-Equal-to?
a. >=
b. =>
c.>==
d. <==

62. Which of the following relation operator represents Not equal to?
a. !=
b. =!
c.#==
d. <=

63. Which of the following is a relational operator?
a. !=
b. ||
c. &&
d. +

64. All of the following are relational operators except:
a.!=.
b. ||.
c. =
d. >=.

65. Which of the following returns True if A = 25 and B = 35:
a. A+B
b. A>=B
c. A!=B
d. A=B

66. For A = 4 and B = 4 which evaluates as true?
a. A <> B
b. A < B
c. A > B
d. A >= B

67. The conditional portion of the If statement can contain:
a. Any valid constant.
b. Any expression that can be evaluated to a Boolean value.
c. Any valid variable.
d. Any valid constant or variable.

68. Which is not part of an IF statement?
a. A condition that evaluates as a Boolean
b. A condition that evaluates as true or false.
c. A true block.
d. A false block.
e. All of the above

69. An IF statement inside the true block of another IF statement is called:
a. a nested IF statement.
b. a branched IF statement.
c. a conditional operator.
d. a relational operator.

70. What does a compound condition use to join two conditions?
a. Relational Operator
b. Logical Operator
c. Relational Results
d. Logical Results

71. Operators that perform operations on Boolean data are:
a. Arithmetic operators.
b. Logical operators.
c. Relational operators.
d. Math operators.

72. When using logical operators to join multiple conditions, "AND" operator means:
a. If both conditions are True, the compound condition is True.
b. If either condition is True, the compound condition is True.
c. If neither condition is True, the compound condition is True.
d. If either condition is False, the compound condition is True.

73. When using logical operators to join multiple conditions, "OR" operator means:
a. If both conditions are True, the compound condition is True.
b. If either condition is True, the compound condition is True.
c. If neither condition is True, the compound condition is True.
d. If either condition is False, the compound condition is True.

74. The purpose of the "NOT" logical operator is to
a. Perform a logical conjunction on two Boolean expressions.
b. Perform a logical disjunction on two Boolean expressions.
c. Perform a logical disfunction on two Boolean expressions.
d. Perform a logical negation on two Boolean expressions.

75. When combined with logical operators, If statements can
a. Change multiple conditions in compound statements.
b. Test multiple conditions in compound statements.
c. Combine multiple conditions in compound statements.
d. Separate multiple conditions in compound statements.

76. Which of the following symbol is used for OR operator:
a. || b. && c. ! d. None

77. Which of the following symbol is used for AND operator?
a. || b. && c. ! d. None

78. Which of the following symbol is used for NOT operator?
a. || b. && c. ! d. None

79. All of the following are logical operators except:
a. NOT b. AND c. OR d. =

80. For A = 3 and B = 7 which evaluates to true?
a. !(A < B) b. !(B < 6) c. (A = 3) && (B = 5) d. (A = B) || (A > B)

81. Which is an example of multiple branches from single expression?
a. If statement b. switch statement c. while loop d. All

82. What is the difference between the expression x=1 and x==1?
a. They are the same
b. x==1 adds one to x
c. x=1 is assignment whereas x==1 is equivalence
d. x==1 is assignment whereas x=1 is equivalence

83. Will document.write(++x) give a different result to document.write(x++)?
a. Yes b. No c. Only if x > y d. Only if x is positive

84. If x=8, y=4, then what is x%y?
a. 8 b. 4 c. 2 d. 0

85. In If statement, the code to be performed if the conditional statement evaluates to True is enclosed in:
a. ( ) b. [ ] c. { } d. < >

86. The first line in the switch block contains:
a. The value of the first criterion b. Statement to be executed if first criterion is true.
c. The expression to be evaluated d. Statement to be executed if no criteria is true

87. The Case block ends with:
a. End Select. b. End Case. c. Break; d. Case Else

88. Which of the following is equivalent to (p >= q)
a. p < q b. !(p < q) c. p > q d. !p < q

89. The conditional statement a < b is true if:
a. a is greater than b b. a is less than b
c. a is greater than or equal to b d. a is less than or equal to b

90. Another term for a conditional operator is:
a. Ternary b. Binary c. Byte d. Iteration.

91. The purpose of a repetition structure is to:
a. Execute a sequence of tasks once. b. Execute a procedure.
c. Repeat the execution of program statements. d. Repeat an If statement once.

92. As a computer processes instructions, one execution of a loop is known as a(n):
a. Cycle. b. Duration. c. Iteration. d. Test.

93. Which loop structure always executes at least once?
a. do-while b. for c. While d. None

94. In this loop, condition comes before the body of the loop:
a. While loop b. do-while loop c. both a and b d. Neither a nor b

95. In this loop, condition comes after the body of the loop:
a. while loop b. do-while loop c. for loop d. b and c

96. Which of the following loop is called counter loop?
a. for b. while c. do-while d. None

97. A loop within a loop is called:
a. Nested loop b. Complex loop c. Outer loop d. Inner loop

98. The loop which never ends is called:
a. Infinite loop b. Running loop c. Continuous loop d. a and c

99. A JavaScript procedure that performs a specific task is called a(n):
a. Syntax. b. Function. c. operator. d. Event.

100. In JavaScript, a function is a(n):
a. Procedure b. Fundamental building block c. Object. d. Both A and B.

101. Where can a JavaScript function definition appear?
a. Only in <head> b. Only in <body> c. In <head> or <body> d. None

102. A JavaScript user-defined function name may not contain any:
a. Spaces b. Punctuation c. JavaScript reserved word d: All

103. Which of the following is not included in the syntax of a JavaScript function?
a. List of arguments b. function keyword c. Function name d. List of operators

104. Specifying function name and its working is called:
a. Describing a function b. Calling a function
c. Defining a function d. Handling a function

105. In JavaScript, having a function perform the specified actions is called:
a. Handling a function. b. Calling a function.
c. Defining a function. d. Executing a function.

106. Actions that occur as a result of user activity are called:
a. Events. b. calls. c. Functions. d. All

107. Specific directions in JavaScript that call for an action or event are called:
a. Action operators b. Executable operators c. Action handlers d. Event handlers

108. In JavaScript, when a user clicks the button of a defined _____, it makes the browser call the function to be executed.
a. Function b. Event handler c. Program d. Attribute

109. Which method is used to ensure that user enters correct data into an input box?
a. Data validation b. Input validation d. Data check d. Field check

110. Input validation can be created by inserting which specific event handler?
a. onClick b. onEntry c. onInput d. onChange

111. Which JavaScript event handler can be used to check if an input box is empty?
a. onClick b. onEntry c. onInput d. onChange

112. Which JavaScript event handler pertains to actions enabled by mouse clicks?
a. onInput b. onClick c. onDrag d. onEntry

113. The selection function is used for designing a(n):
a. Input field. b. Input form. c. radio button. d. Drop-down menu.

114. Which JavaScript event handler checks to see if an input form is empty?
a. onChange b. onClick c. onDrag d. onEntry

115. Which operator is needed to access an object's properties or methods?
a. Comma b. Splash c. Bang d. Dot

116. Which of the following is the first object in an array?
a. Marks[0] b. Marks [1] c. Marks [2] d. Marks [3]

117. If an array has ten elements, which of the following is the highest subscript?
a. 0 b. 1 c. 9 d. 10

118. A new array can be declared by which of the following?
a. Student = new Array; b. Array = new Student;
c. Student = new Array (); d. Array = new Student ();

119. Which of the following objects is used to represent the HTML anchor tag (<a>)?
a. Document Object b. Image Object c. Form Object d. Link Object

120. Objects in the DOM represent:
a. Variables. b. Images. c. Classes. d. HTML elements.

121. In JavaScript programming, an object is a:
a. Person b. Place c. Thing d. All

122. What does DOM stand for?
a. Document Object Mode b. Document Object Map
c. Document Object Model d. Document Object Mole

123. All of the following events can be generated by a mouse EXCEPT:
a. OnClick b. OnMouseOver c. OnMouseOut d. OnChange

124. All of the following HTML tags may use the OnLoad event EXCEPT:
a. <frameset>. b. <input> c. <body> d. <link>

125. The onsubmit event handler is usually bound to the _____ HTML tag.
a. <p> b. <form> c. <input> d. <body>

126. Which types of dialog boxes have OK and Cancel buttons?
a. Alert only b. Confirm only c. Prompt only d. Alert and confirm

127. How should a script defined in the current window reference a window property?
a. window.propertyName b. propertyName c. self.propertyName d. All

128. What property of Image object is used to determine if image has loaded?
a. src b. complete c. border d. alt

129. Which of the following are valid event sources for JavaScript
a. Browser Window b. Web Page (document) c. Hyperlink d. All

130. Which of the following is/are correct about JavaScript
a. Enhances interaction with WebPages b. Reduces client server traffic
c. Can process form validation d. All

131. The first image in an HTML Web page can be referred as ____ in JavaScript code.
a. image[0] b. image[1] c. image1 d. images[1]

132. What does the method property of the form object represent in JavaScript?
a. The name of the form
b. The file name of the server-side script where the form information is sent
c. Whether the form will be submitted by the post or get method
d. The type of operating system the user is currently using

133. Why is JavaScript usually used for form validation?
a. Because most server-side processing scripts also use JavaScript
b. Because no other language contains similar tools for validation
c. Because the processing can be done before the form is submitted to the server
d. JavaScript is not normally used for form validation.

134. How does the getDay() method of the date object return information?
a. As a day of the week, represented by a text string (e.g., « Saturday »)
b. As a day of the week, represented by a number (e.g., « 2 »)
c. As a day of the current month (e.g., « 25th »)
d. None of the above

135. Which method can you use to clear a repeating loop of actions?
a. setInterval() b. setTimeout() c. clearInterval() d. clearTimeout()

136. Which method can you use to create a repeating loop of actions?
a. setInterval() b. setTimeOut() c. clearInterval() d. toLocaleString()

137. In the statement var myAr=newArray(30), what does the number "30" represent?
a. The value assigned to the first element in the array
b. The number of elements to be used in the array
c. The number of dimensions created in a multi-dimensional array
d. All of the above

138. Which value is returned by indexOf() method if the search is not successful?
a. 0 b. –1 c. undefined d. code 3

139. Which value is returned by lastIndexOf() method if the search is not successful?
a. 0 b. –1 c. undefined d. code 3

140. Which of the following is true concerning a history's methods?
a. A user can read web history
b. A user can move to any item in list
c. User should be worried with signed scripts
d. User needs to be worried with permissions

141. The history.go(-1) does which of the following?
a. The history.go(-1) does the same thing as history.forward().
b. The history.go(-1) does the same thing as history.backward().
c. The history.go(-1) goes to the bottom page on the list.
d. The history.go(-1) goes to the top page on the list.

142. When a user selects a cancel button, which of the following occurs?
a. The confirm() method returns false.
b. The confirm() method returns true.
c. The alert() method returns false.
d. The alert() method returns true.

143. Which of the following is the valid syntax for initialization of a String object?
a. Var Student = "Jane Kwok";
b. Student = new String("Jane Kwok");
c. Var Student = new String("Jane Kwok");
d. Var Student = new Student("Jane Kwok");

144. In string "turtle", the index of "u" is:
a. 1 b. 2 c. 3 d. 4

145. Which method is used to change String's value into number with decimal values?
a. parse() b. parseFloat() c. parseInt() d. parseNumber ()

146. What is the correct syntax to get a sub-string of a String started in 10th position for a total of 10 characters?
a. Student.substring(10)
b. Student.substring(10,20)
c. Student.substring(10,10)
d. Student.substring(10,0)

147. Which of the following method returns string in lower case?
a. LowerCase() b. ToLowerCase() c. ToUpperCase() d. UpperCase()

148. Which of the following method returns the string in upper case?
a. LowerCase() b. ToLowerCase() c. ToUpperCase() d. UpperCase()

149. A push button's form, name, type, and value are which of the following?
a. Events b. Methods c. Properties d. All

150. A checkbox allows a user to do which of the following?

a. Make zero selections
b. Make one selection
c. Make more than one selection
d. All

Answers

1. d	2. a	3. c	4. d	5. b	6. d
7. b	8. b	9. a	10. a	11. d	12. b
13. c	14. a	15. b	16. c	17. d	18. a
19. c	20. a	21. b	22. d	23. a	24. a
25. c	26. d	27. d	28. b	29. c	30. d
31. b	32. c	33. d	34. c	35. a	36. d
37. d	38. b	39. c	40. d	41. b	42. a
43. c	44. a	45. b	46. c	47. a	48. a
49. a	50. a	51. a	52. a	53. a	54. d
55. a	56. c	57. b	58. d	59. d	60. c
61. a	62. a	63. a	64. b	65. c	66. d
67. b	68. e	69. a	70. b	71. b	72. a
73. b	74. d	75. b	76. a	77. b	78. c
79. d	80. b	81. b	82. c	83. a	84. d
85. c	86. c	87. c	88. b	89. b	90. a
91. c	92. c	93. a	94. a	95. b	96. a
97. a	98. a	99. b	100. d	101. c	102. d
103. d	104. c	105. b	106. a	107. d	108. b
109. b	110. d	111. d	112. b	113. d	114. a
115. d	116. a	117. c	118. c	119. d	120. d
121. d	122. c	123. d	124. b	125. b	126. d
127. d	128. b	129. d	130. d	131. d	132. c
133. c	134. b	135. c	136. a	137. b	138. b
139. b	140. b	141. b	142. a	143. b	144. a
145. b	146. c	147. b	148. c	149. c	150. d

Fill in the Blanks

1. A page is displayed within the Browser window – known as the _____ object.
2. Web-based applications use both server-side and _____ scripts for dynamic pages
3. JavaScript was jointly developed by Netscape Communications and _____.
4. JavaScript is mainly used for _____
5. The _____ tag allows JavaScript to be inserted into an HTML document.
6. In JavaScript, a(n) _____ object represents the browser window.
7. _____ comparison operator is used to determine if one value is greater than another.
8. The operator consisting of two equal signs is known as _____ comparison operator.
9. The comparison operator used to determine the equality of two values is _____.
10. The "logical and operator" is written as the characters_____
11. The _____ method is similar to the alert() method, but it also allows the user to respond to a question being asked.
12. _____ operator returns *true* if either or both of the variables being compared is *true*.
13. Variables that can only have the values of *true* or *false* are called _____ variables.

WWW.ITECVIEW.COM©

14. _____ method is used to interactively ask user simple *true/false* types of questions.
15. A confirm box always returns a value of type _____.
16. The operators &&, ||, ^ and ! can only be used with variables of type _____.
17. The operator ! returns the _____ of the variable on which it is used
18. All of the logical operators except _____ require two operands.
19. _____ structure executes a statement or number of statements repeatedly.
20. In _____ loop, condition is checked before the execution of the loop body.
21. While loop continues as long as the _____ remains true.
22. _____ loop also called counter-controlled loop.
23. _____ is similar to "continue" except that it exits a loop.
24. The term _____ is used when describing the characteristic of an object.
25. Functions are normally triggered by _____ in the HTML document.
26. OnunLoad event handler can be used with both the <body> tag and _____ tag.
27. A(n) _____ event is generated by an object when it loses the focus.
28. To handle event generated when a form is submitted, <form> should be aimed for ____ event.
29. The script tag includes the _____ attribute to identify the scripting language.
30. The write(" ") portion of the document.write(" ") statement is called a(n) _____.
31. The biggest hurdles in developing effective JavaScript are _____ compatibility issues.
32. String values can be inserted into string literals using the _____ operator
33. To _____ a variable, you need to choose a name that represents the variable.
34. Abstract, Boolean, extends, and int are examples of _____words.
35. A(n) _____ are objects of related items.
36. An if and an if...else are _____ statements.
37. Functions are the basis of _____ programming.
38. The select() method selects all text within a text field that has _____.
39. If a user tries to close the main window, users will see a(n) _____ box.
40. When a variable is declared and no value is assigned to it, its initial value is_____.
41. A(n) _____ dialog box is more versatile than an alert dialog box..
42. A user creates an alert dialog box with the _____ method.
43. _____ counts the number of items in the history list.
44. _____ operator is used to find the integer remainder that is left after a division.
45. In a switch statement, a(n) _____ follows a case statement.
46. The process of finding and correcting error in code is known as _____.
47. The _____ value for a switch statement will execute if no case statements evaluate as true.
48. The values of the _____ statement use double quotes.

Answers

1. Document	2. client-side	3. Sun Microsystems
4. scripting	5. <script>	6. Window
7. >	8. Equality	9. >=
10. &&	11. Prompt	12. OR (||)
13. Boolean	14. confirm(...)	15. Boolean
16. Boolean	17. NOT	18. ! (NOT)
19. loop	20. while	21. condition
22. for	23. break	24. Property
25. event handler	26. <frameset>	27. Blur
28. OnSubmit	29. type	30. Method
31. cross-platform	32. + (addition)	33. Declare
34. Reserved	35. Array	36. Conditional
37. Modular	38. Focus	39. confirm dialog
40. null	41. confirm	42. window.alert()
43. Length	44. Modulus	45. Colon
46. Debugging	47. Default	48. Case

True / False

1. JavaScript was developed by Netscape and Sun Microsystems.
2. The scripting approach to developing Web-based spreadsheet applications uses JavaScript or VBScript.
3. JavaScript is space-sensitive.
4. JavaScript objects can contain properties, methods, and event handlers.
5. JavaScript statements must end with a semi-colon.
6. The write () method requires a single string as a parameter.
7. JavaScript applications on the client side are largely event-driven.
8. An example of a logical value recognized in JavaScript is "Welcome to JavaScript."
9. Netscape displays windows created with the alert () method as error windows.
10. JavaScript requires var in front of a variable name.
11. The default text in a prompt window is null.
12. The window created with confirm () method contains OK, Clear and Cancel buttons.
13. The + operator can be used to concatenate strings.
14. JavaScript recognizes NULL and null as the same value.
15. Variables are symbolic names for values.
16. Literals are the same thing as global variables.
17. An event handler should return a true or false value.
18. The pow () method performs exponentiation.
19. A syntax error results from a violation of a JavaScript grammar rule.
20. The mod () method returns the remainder after an integer division.
21. The unary minus and the subtraction operators have the same precedence.
22. The addition operator is a binary operator.
23. A prefix operator places the operand before the operator.
24. When the division operator is applied to two integers, the result is a floating-point number.
25. A literal is needed in an expression to relate the values together.
26. The precedence of the logical operators is: &&, ||, !
27. Floating-point numbers are part of the ordinal type.
28. JavaScript interprets false as anything producing a zero.
29. The else clause of an if statement is optional".
30. If we wish to execute more than one statement within a case statement we must use a compound statement.
31. The switch statement uses an optional else clause to handle unmatched values.
32. JavaScript implements the inclusive logical or, but not the exclusive logical or.
33. A block of code is a set of statements enclosed in braces.
34. The relational operators have a higher precedence than all of the logical operators.
35. A selection statement executes the statements in the body zero or one times.
36. The for statement header must include initial, test, and change expressions.
37. A while statement executes the body of the statement at least once.
38. A do . . . while statement repeats until a specified condition evaluates to true.
39. A do..while statement checks for loop termination only after the statements in the body have been executed.
40. The for statement is the most efficient loop when adding or counting.
41. A do..while statement is the best statement to use when creating an error loop.
42. A for statement executes the statements in the loop body zero or more times.
43. A while statement executes the statements in the body as long as the expression in the header is true.
44. A function definition can contain none, one, or several return statements.
45. A return statement always returns a value.
46. When an object is passed as a parameter to a function, JavaScript uses pass by reference.

47. If we have several different function definitions in the same script, we must use different names for the local variables.
48. A global variable is available anywhere in the current document.
49. Defining a function names the function and specifies what is to be done when the function is called.
50. A function is a JavaScript operator that performs a specific task.
51. A JavaScript function can have only one argument.
52. Parameters are used only to provide input to a function.
53. It is preferable to use global variables rather than passing parameters.
54. If a user creates several objects in the same script, the methods associated with the objects cannot have duplicate names.
55. The numbering of JavaScript array subscripts begins at one.
56. The shift() method removes the rightmost array element.
57. The slice() method removes a portion of an array.
58. The default sort order for a numeric array is lexicographic order.
59. The join() method creates a string from an array .
60. length() is a method of the Array object.
61. To create an array, we use new Array().
62. The sort() method has an optional parameter that specifies the sort order.
63. The push() method adds an element to the right end of an array .
64. The onChange event handler is used in conjunction with the input validation method.
65. The input validation method is used to authenticate user name and password.
66. The onChange event handler can be used to check whether or not an input form is empty.
67. The onChange event handler is used in conjunction with the automatic output method.
68. Physical wrap is the default style for text areas.
69. The POST method is used to email form results.
70. Radio buttons let the user select none, one, or several buttons in a group.
71. The default mode for select lists allows multiple selections.
72. Check boxes and multiple select lists can be used interchangeably.
73. Radio buttons and check boxes can be used interchangeably.
74. Actions enabled by mouse clicks are controlled by the onControl event handler
75. The calculation function can be written in JavaScript code and then embedded in HTML file.
76. Users signal form completion in JavaScript by pressing the Submit button.
77. Assume that we have a form with one single select list. We use the selectedIndex property to determine which option was selected.
78. The focus() method brings a checkbox into focus.
79. Checkboxes use an onClick event handler.
80. The concat() method appends the string passed as a parameter to the string invoking the method.

Answers

1. T	2. T	3. T	4. T	5. F	6. T	7. T	8. F
9. T	10. F	11. F	12. F	13. T	14. F	15. T	16. F
17. T	18. T	19. T	20. T	21. F	22. T	23. F	24. T
25. F	26. F	27. F	28. T	29. T	30. F	31. F	32. T
33. T	34. F	35. T	36. F	37. F	38. F	39. T	40. T
41. F	42. T	43. T	44. T	45. F	46. T	47. F	48. T
49. T	50. F	51. F	52. F	53. F	54. F	55. F	56. F
57. F	58. T	59. T	60. F	61. T	62. T	63. T	64. T
65. F	66. T	67. T	68. F	69. T	70. F	71. F	72. T
73. F	74. F	75. F	76. F	77. T	78. F	79. T	80. T

Part - 5

VBScript

Includes:

Chapter 11	Introduction to VBScript
Chapter 12	Conditions and Loops
Chapter 13	Procedures, Functions and Arrays
Chapter 14	Events and String Handling

WWW.ITECVIEW.COM©

Introduction to VBScript

11.1 VBScript

VBScript supports several scripting languages. **VBScript** is the default language. It is a scripting language developed by Microsoft. VBScript is a subset of Visual Basic language. It supports the syntax and structure of Visual Basic. VBScript code is embedded in a Web page. The code is translated by Web browser when the web page is viewed by the user.

VBScript can be used as follows:

- **Web pages:** VBScript can be used as client-side scripting. It can be embedded in HTML documents to add interactive functionality to Web pages. It can be used for data validation and user interaction at the client-end.
- **Active Server Pages:** VBScript can also be used as server-side scripting in **IIS (Internet Information Server)** in **Active Server Pages**. A web page can be generated and modified at server-end before it is sent to the client through browser.

The applications that use VBScript are called **host applications**. The most common host applications are as follows:

- Internet Explorer
- Internet Information Server
- Outlook uses VBScript as its macro language
- Windows Scripting Host

VBScript language works similarly on all host applications. However, the objects available with each application may differ. Some objects available with Internet Explorer may not be available with Internet Information Server. Similarly, the objects available with Internet Information Server may not be supported by Internet Explorer.

11.1.1 Advantages of VBScript

Some advantages of using VBScript are as follows:

1. Interpreted Language

VBScript is an interpreted language. It requires no compilation process so no compiler is required. The browser interprets VBScript as it interprets HTML tags.

2. Easy to Learn

The syntax of VBScript is very easy. Any person can learn it very easily and use it to develop dynamic and interactive websites.

3. Easy to Debug and Test

VBScript code is interpreted line by line. The errors are indicated along with line number. It is very easy to find error in the code, correct it and test it again.

4. Event-Based Programming

VBScript is an event-based language. It means that different code segments are executed when certain event occurs. For example, a code segment may execute when the user clicks a button or moves mouse over an object etc.

5. Procedural Capabilities

VBScript provides all capabilities of a procedural language. It provides condition checking, loops and branching facilities that can be executed in a web page.

6. Platform Independence

VBScript can be used to interact with the user. The input entered by the user can be processed and proper message can be displayed to the user. The interactive capability of a website makes it more interesting and productive for the users.

11.1.2 Uses of VBScript

Some important uses of VBScript are as follows:

1. Input Validation

VBScript can be used to validate the input entered by the user. The format of data entered in forms should be validated before it is processed. For example, if the user enters his email address, it must be a valid email. VBScript code can ensure its validity by checking that it contains @ symbol and dot (.).

2. Mouse Rollover Effects

VBScript can be used to create different buttons with interesting mouse rollover effects. When the user moves the mouse over a button, different effects are generated. It makes browsing more interesting and attractive.

3. Popup Windows

VBScript can be used to create popup windows. These windows are normally used to display important announcements, offers and news etc. Popup windows can be created of different sizes.

4. Dynamic Contents

VBScript can be used to generate dynamic contents in a website. For example, the current date and time can be displayed on web page. Different HTML tags can be generated based on the user input etc.

5. User Interaction

VBScript can be used to interact with the user. The input entered by the user can be processed and proper message can be displayed to the user. The interactive capability of a website makes it more interesting and productive for the users.

11.1.3 Structure of VBScript

VBScript code is embedded into an HTML document using the **SCRIPT** tag. The **<SCRIPT>** tag is used to inform browser about the start of VBScript statements. The **<SCRIPT>** tag indicates the beginning of scripting code and **</SCRIPT>** tag indicates the end of scripting code.

Example

```
<SCRIPT LANGUAGE="VBScript">
        VBScript statements...
</SCRIPT>
```

The section between opening **<Script>** tag and closing **</Script>** tag is called **body** of the script. All statements are written in this part of the script. VBScript is not case sensitive. It means that **SCRIPT, script** and **Script** are the same.

11.1.4 The Document Object

VBScript uses an object called **Document**. This object manages many instructions of VBScript. It can be used is to display a string on the screen. A **string** is the text that is diaplayed in the browser. It is displayed by using the function **Write**. The syntax of **Write** function of **Document** object is as follows:

```
Document.Write(String)
```

The **String** can be a constant written in double-quotes or a variable. An example of **Document.Write()** function is as follows:

```
<SCRIPT LANGUAGE="VBScript">
        Document.Write("VBScript Statments")
</SCRIPT>
```

Practice 11.1

Write a VBScript code that displays a message "Cool VBScript!" in the browser.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>VBScript Example</TITLE>
<SCRIPT LANGUAGE = "VBScript">
        Document.Write ("Cool VBScript! ")
</SCRIPT>
</HEAD>
</HTML>
```

2. Save the file as file as "Vb01.html".
3. Double click the file. The message "Cool VBScript!" will appear in the browser.

11.2 Variables

A quantity whose value can be changed during program execution is called **variable**. It is a named memory location. It is used to store program's input data and its computational results during execution. The value of a variable may change during the execution of program. However, the name of variable cannot be changed.

11.2.1 Rules for Variable Names

The name of a variable is given by the user at design time. It may consist of alphabetic and numeric digits. The name of the variable should be meaningful and simple. It helps the user to understand the purpose of variable. For example, a variable to store the marks of student should be **Marks**. Following are some rules for declaring variables:

- A variable name may consist of maximum 255 characters.
- A variable name may include letters, numbers and underscore (_).
- The first character of variable must be a letter. The variables 9minute, #home and 2kg are invalid.
- Blank spaces are not allowed in variable names. The variables **my var** and **your car** are invalid.
- Both upper and lower cases are allowed in variable name.
- A reserved word cannot be used as variable name.

Examples

Some examples of valid variables are as follows:

Marks sub1 f_name MySalary

11.2.2 Variable Declaration

The variables in VBScript are declared using **Dim** keyword. Dim stands for **dimension**. The syntax for declaring variable is as follows:

```
Dim VarName
```

Here, **VarName** is the name of the variable. VBScript has only one data type called **variant**. It can store different types of data such as strings, integers, floating point numbers etc. VBScript interprets a variant according to the value stored in it. For example, a variant is treated as a number if it contains numeric value. Similarly, a variable is treated as string if it contains string value. The data type is not specified when a variable is declared in VBScript because all variables are variant.

Option Explicit

The **Option Explicit** is used to ensure that all variables are declared before they are used. The Option Explicit statement is included before any HTML text or script commands. A misspelled variable name is declared as a new variable if this option is not used that may result in an error in the script.

Example

```
<SCRIPT LANGUAGE="VBScript">
        Option Explicit
        Dim myVar
</SCRIPT>
```

Practice 11.2

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
        n = 10
        Document.Write (n)
</SCRIPT>
```

2. Save the file as "vb02.html".

3. Double click the saved file. The page will appear with the value 10.
4. Change the VBScript code as follows:

```
<SCRIPT LANGUAGE="VBScript">
        Option Explicit
        n = 10
        Document.Write (n)
</SCRIPT>
```

5. Save changes made in file and refresh the page in browser. The output will disappear due to the error.
6. Again change the VBScript code as follows:

```
<SCRIPT LANGUAGE="VBScript">
        Option Explicit
        Dim n
        n = 10
        Document.Write (n)
</SCRIPT>
```

7. Save changes and open the file again in the web browser. The error will be removed and the result will appear again correctly.

11.3 Mathematical Operators

Operators are mathematical symbol that is used to perform some operation on data. Mathematical operators are the symbols that are used to perform arithmetic operations on data. VBScript provides different mathematical operators to manipulate data and perform calculations that are as follows:

Operation	Symbol	Description
Addition	+	Adds two values
Subtraction	-	Subtracts one value from another value
Multiplication	*	Multiplies two values
Division	/	Divides one value by another value
Integer Division	\	Divides one value by other giving integer result
Exponentiation	^	Raises a value to a power
Modulus	Mod	Gives the remainder

Table: Mathematical operators

Following are some examples that explain the use of mathematical operators:

Operation	Example	Result
Addition	7 + 2	9
Subtraction	6 – 4	2
Multiplication	8 * 4	32
Division	17 / 5	3.4
Integer division	17 \ 5	3
Exponentiation	3 ^ 2	9
Modulus	7 mod 4	3

Table: Examples of Mathematical operators

Practice 11.3

Write a VBScript code that performs all mathematical operations on two variables.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
  Dim a, b
  a = 10
  b = 2
  Document.write("a + b = " & (a+b))
  document.write("<BR>")
  document.write("a - b = " & (a-b))
  document.write("<BR>")
  document.write("a * b = " & (a*b))
  document.write("<BR>")
  document.write("a / b = " & (a/b))
  document.write("<BR>")
  document.write("a mod b = " & (a mod b))
  document.write("<BR>")
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb03.html".
3. Double click the file.

The symbol & is used as concatenation operator in the above example.

11.3.1 Operator Precedence

The order in which different types of operators in an expression are evaluated is known as **operator precedence**. It is also known as **hierarchy of operators**.

Each operator has its own precedence level. If an expression contains different types of operators, the operators with higher precedence are evaluated before the operators with lower precedence. The order of precedence in VBScript is as follows:

- First of all expression in parentheses is evaluated.
- Then exponentiation (^) is performed.
- Then multiplication (*) and division (/) is performed.
- Then Integer division (\) is performed.
- Then Modulus (Mod) is performed.
- Then Addition (+) and subtraction (-) are performed.

If an expression contains similar operator more than once, the expression is evaluated from left to right.

Example

The expression **10 * (24 / (5 – 2)) + 13** is evaluated in the following order:

1. First of all, the expression **5–2** will be evaluated. It gives a value **3**.
2. Secondly, **24** will be divided by the result of last line i.e. **24 / 3** giving value **8**.
3. Thirdly, **10** will be multiplied by **8** i.e. giving a result **80**.
4. Finally, **80** will be added in **13** and the last result will be **93**.

11.4 Concatenation

Concatenation is a process of combining two or more items together. The symbols + and & are used as concatenation operators. Suppose 100 is stored in **n**. The following statement will display "The number is 100":

```
Document.Write ( "The number is " & n )
```

The above line uses & symbol to combine a string and a variable. The string is printed as given in the statement. But **n** is replaced with its value in the memory.

11.5 Comments

Comments are the part of script that is not executed. Comments are actually notes about different lines of code that explain the purpose of the code.

Comments can be added anywhere in the code. In VBScript, comments starts with an single quote (). A single quote (') is used to tell the browser to ignore statements that follow the symbol while processing the code. Following line is an example of comments:

```
' Hello! Comments are important to maintain things easily...
```

11.6 Input and Output

The process of giving something to computer is known as **input**. The input is mostly given by keyboard. A program may need certain inputs from the user for working properly. The process of getting something from computer is known as **output**. The output is mostly displayed on monitor. The result of a program is the output of that program.

- You can use built-in **InputBox** function to get input from the user.
- You can use built-in **MsgBox** function to display the result in message boxes.

11.6.1 Input Box

InputBox is used to get input from the user during execution. A predefined InputBox function is used to display input box.

Syntax

The syntax of using this function is as follows:

```
Variable = InputBox(Message, Title, Default, PositionX, PositionY)
```

Here,

Variable	is the variable to store the value input by the user.
InputBox	is the predefined function for getting input.
Message	is the text displayed on the input box for the user.
Title	is the title of the input box.
Default	is the default value displayed in the input box.
PositionX	is the position of input box on x-axis and is given in twips.
PositionY	is the position of input box on y-axis and in given in twips.

Practice 11.4

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE> Use of Input Box</TITLE>
</HEAD>
```

```
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n
n = inputbox("Enter any number","Hello","0")
Document.Write (n)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb04.html".
3. Double click the file. Following box will appear. It contains two buttons **OK** and **Cancel**. When the user click on **OK**, the value enter by the user is assigned to the variable on the left side. If the user clicks **Cancel**, **NULL** value is assigned.

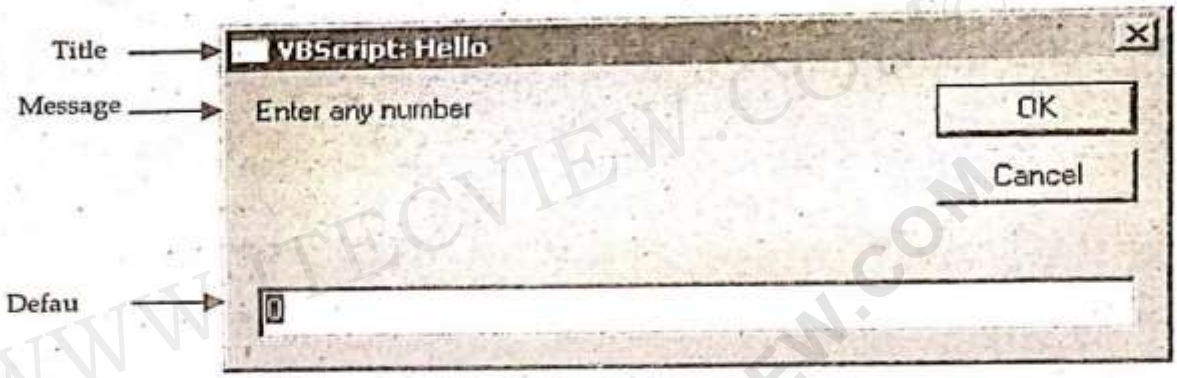
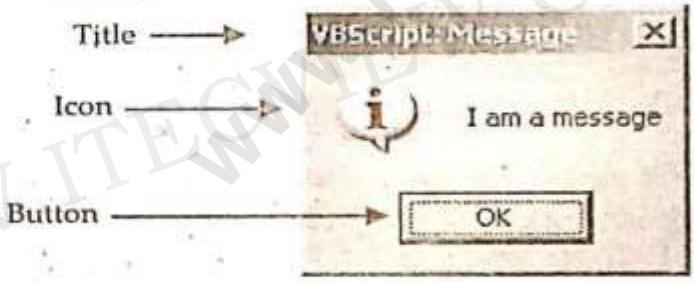


Figure: Input box

11.6.2 Message Box

MsgBox is a predefined function used to display output to the user. It can be used to simply display a message to the user.



Simple message box

Syntax

The syntax of using this function is as follows:

MsgBox (Message, Style, Title)

Where,

MsgBox	is the name of the predefined function to display message.
Message	is the message displayed to the user.
Style	is used to display different buttons and icons on message box.
Title	is a string displayed on the title bar of the message box.

Practice 11.5

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
```

```
        MsgBox "I am a message", vbInformation, "Message"
</SCRIPT>
```

2. Save the file as "vb05.html".
3: Double click the file. The message box will appear.

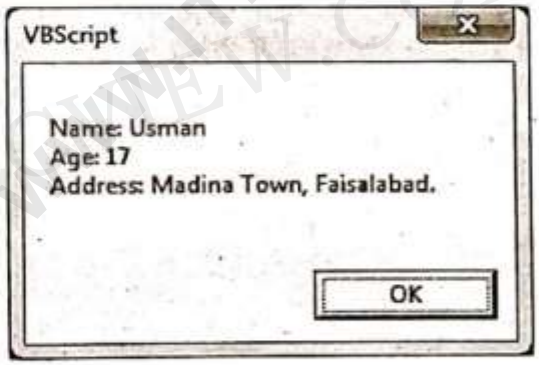
Practice 11.6

Write a VBScript code that gets user name, age and address in input box and displays this information in message box.

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        dim name,address, age
        name = inputbox("Enter your name: ","Name")
        age = inputbox("Enter your age: ","Age")
        address = inputbox("Enter your Address:", "Address")
        Msgbox "Name: " & name & vbNewLine _
              & "Age: " & age & vbNewLine _
              & "Address: " & address
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb06.html".
3. Double click the file.
4. Enter name, age and address in the input boxes. The entered information will appear in the message box as follows:



Practice 11.7

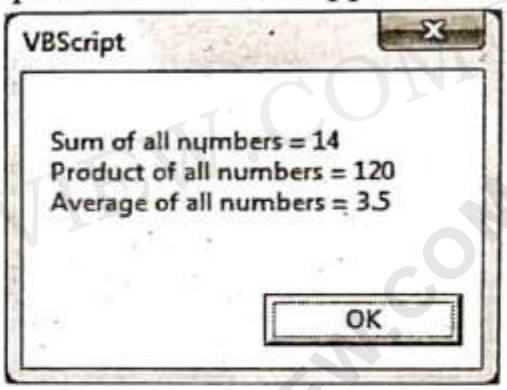
Write a VBScript code that gets four integer numbers from the user and calculates and displays sum, average and product of all the numbers.

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim N1, N2, N3, N4, Sum, Mul, avg
        N1 = Cint(InputBox("Enter 1st number: ", "Number 1", "0"))
        N2 = Cint(InputBox("Enter 2nd number: ", "Number 2", "0"))
        N3 = Cint(InputBox("Enter 3rd number: ", "Number 3", "0"))
        N4 = Cint(InputBox("Enter 4th number: ", "Number 4", "0"))
        Sum = N1 + N2 + N3 + N4
```

```
        Mul = N1 * N2 * N3 * N4
        Avg = Sum / 4
        MsgBox "Sum of all numbers = " & Sum & vbNewLine _
            & "Product of all numbers = " & Mul & vbNewLine _
            & "Average of all numbers = " & Avg
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb07.html".
3. Double click the file.
4. Enter 2, 3, 4 and 5 as input. The result will appear as follows:



Practice 11.8

Write a VBScript code that gets temperature from the user in Celsius and converts it into Fahrenheit using the Formula F = 9/5*C+32.

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim Cel, Faren
        Cel = InputBox("Enter temperature in Celsius:")
        Faren = 9 / 5 * Cel + 32
        MsgBox " Temperature in Fahrenheit is: " & Faren
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb08.html".
3. Double click the file.

Practice 11.9

Write a VBScript code that inputs base and height from the user and calculates area of a triangle by using the formula Area = ½ * Base * Height.

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim Base, Height, Area
        Base = InputBox("Enter base: ", "Base", "0")
        Height = InputBox("Enter height ", "Height", "0")
        Area = 0.5 * Base * Height
```

```
        MsgBox "Triangle area= " & Area
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb09.html".
3. Double click the file.

Practice 11.10

Write a VBScript code that inputs radius of sphere from the user. Calculates its volume and surface area using the formula Area = $4\pi R^2$ and circumference = $4/3\pi R^3$ where π=3.14

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim r, area, cir
        r = InputBox("Enter radius: ", "Sphere Radius", "0")
        area = 4.0 * 3.14 * r * r
        cir = (4.0/3.0) * 3.14 * r * r * r
        MsgBox "Area = " & area & " and Circumference" & cir
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb10.html".
3. Double click the file.

Practice 11.11

Write a VBScript code that gets a 3-digit number from the user and displays it in reverse order. For example, if the user enters 123, it should display it as 321.

1. Open Notepad and type the following code

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim n , r,  a , b , c
        n = inputbox("Enter three digit numbers")
        a = n \ 100
        n = n Mod 100
        b = n \ 10
        n = n Mod 10
        MsgBox "Number is reverse order is " & n & b & a
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb11.html".
3. Double click the file.

Practice 11.12

Write VBScript code that inputs seconds from the user and displays hours, minutes and seconds. For example, if the user enters 5300 seconds, it displays 1:28:20

1. Open the notepad and type the following code.

```
<HTML>
```

```
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim sec, s, m, h
        sec = InputBox("Enter time in seconds :", "Time", "0")
        h = sec \ 3600
        sec = sec Mod 3600
        m = sec \ 60
        s = sec Mod 60
        MsgBox " Time in HH-MM-SS Format  : " & h & ":" & m & ":" & s
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb12.html".
3. Double click the file.

Practice 11.13

Write VBScript code that inputs and integer and displays the sum of its digits. For example, if the user enters 941, it displays 14.

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim number, last_digit, next_digit, total
        number = CInt(InputBox("Enter a integer number", "sum of digit", "0"))
        last_digit = number Mod 10
        total = last_digit
        next_digit = (number \ 10) Mod 10
        total = total + next_digit
        next_digit = (number \ 100) Mod 10
        total = total + next_digit
        next_digit = (number \ 1000) Mod 10
        total = total + next_digit
        next_digit = (number \ 10000) Mod 10
        total = total + next_digit
        MsgBox "The sum of the digits of the entered number is:" & total
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb13.html".
3. Double click the file.

Practice 11.14

Write VBScript code that inputs 5-digit number from the user and displays its reverse. For example, if the user enters 72519, it dispalys 91527 in message box.

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Dim n, a , b, c ,d
        n = Inputbox("Enter a 5-digit number:","Reverse")
        a = n \ 10000
```

```
        n = n mod 10000
        b = n \ 1000
        n = n mod 1000
        c = n \ 100
        n = n mod 100
        d = n \ 10
        n = n mod 10
        Msgbox ("Number in reverse order is" & n & d & c & b & a)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb14.html".
3. Double click the file.

Practice 11.15

Write VBScript code that will prompt the user to enter number of hours. Compute and display the number of weeks, days and hours within the input number of hours.
(e.g. 4000 hrs = 23, weeks, 5 days, 16 hrs)

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        dim hrs,w ,d
        hrs = CInt(inputbox("Enter hours"))
        w = hrs \ 168
        hrs = hrs mod 168
        d = hrs \ 24
        hrs = hrs mod 24
        Msgbox("Weeks: " & w & " Days: " & d & " Hours: " & hrs)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb15.html".
3. Double click the file to execute it.

Practice 11.16

Write VBScript code that inputs pounds and converts it into kilograms.

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        dim pounds, kilograms
        pounds = inputbox("Enter the number of pounds: ")
        kilograms = pounds/2.205
        document.write( pounds & " pounds = " & kilograms & " kilograms " )
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb16.html".
3. Double click the file to execute it.

Practice 11.17

Write VBScript code that accepts distance in meters and converts it into kilometers.

1000 meter = 1 kilometer

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
dim m, k
m = inputbox("Enter distance in meters: ")
k = m / 1000
document.write("The distance in kilometers = " & k)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb17.html".
3. Double click the file to execute it.

Practice 11.18

Write VBScript code that gets temperature from the user in Fahrenheit and converts it into Celsius degrees using the Formula C = 5/9(F-32).

1. Open the notepad and type the following code.

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim f, c
f = InputBox("Enter temperature in Fahrenheit: ")
c = (f - 32) * 5 / 9
MsgBox "Temperature in Celcius is " & c
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb18.html".
3. Double click the file to execute it.

Exercise Questions

1. Write VBScript code that inputs Basic Salary and calculates 35% dearness allowance, 25% house rent and then displays the gross salary.
2. Write VBScript code that inputs age in years and displays age in days and months.
3. Write VBScript code that inputs obtained and total marks from the user and prints its percentage value.

WWW.ITECVIEW.COM©

12

Conditions and Loops

12.1 Control Structure

A statement used to control the flow of execution in a program is called **control structure**. The instructions in a program can be organized in three kinds of control structures to control execution flow. The control structures are used to implement the program logic.

12.1.1 Types of Control Structures

Different types of control structures are as follows:

1. Sequence

In **sequential structure**, the statements are executed in the same order in which they are specified in program. The control flows from one statement to other in a logical sequence. All statements are executed exactly once. It means that no statement is skipped and no statement is executed more than once

Example

Suppose a program inputs two numbers and displays average on screen. The program uses two statements to input numbers, one statement to calculate average and one statement to display average number. These statements are executed in a sequence to find the average number. All statements are executed once when the program is executed. It is an example of sequence control structure.

Figure: Execution of Sequential Statements

WWW.ITECVIEW.COM©

2. Selection

A **selection structure** selects a statement or set of statements to execute on the basis of a condition. In this structure, statement or set of statements is executed when a particular condition is **true** and ignored when the condition is **false.** There are different types of selection structures in VBScript. These are **IF...THEN , IF...THEN...ELSE** and **SELECT CASE.**

Example

Suppose a program inputs the marks of a student and displays a message on screen whether the student is pass or fail. It displays **Pass** if the student gets 40 or more than 40 marks. It displays **Fail** when the marks are below 40. The program checks the marks before displaying the message. It is an example of selection structure.

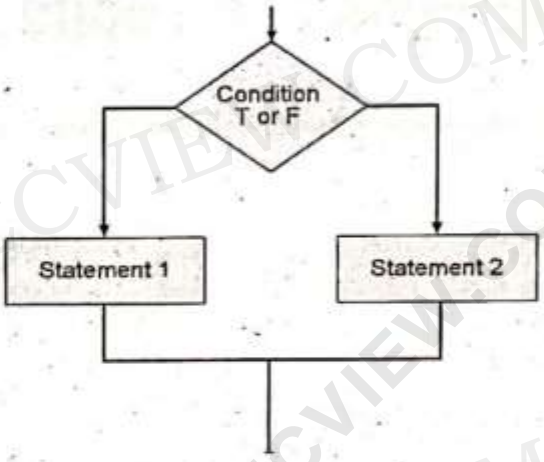


Figure: Execution of Conditional Statements

3. Repetition

A **repetition structure** executes a statement or set of statements repeatedly. It is also known as **iteration structure** or **loop**. There are different types of repetition structures in VBScript. These are **FOR...NEXT , WHILE...WEND** and **DO WHILE...LOOP** etc.

Example

Suppose we want to display a message "I love Pakistan" on screen for 100 times. It is time consuming to perform this task using control sequence structure. However, it is very easy to perform this task using repetition structure. A single loop statement can display the message for 100 times.

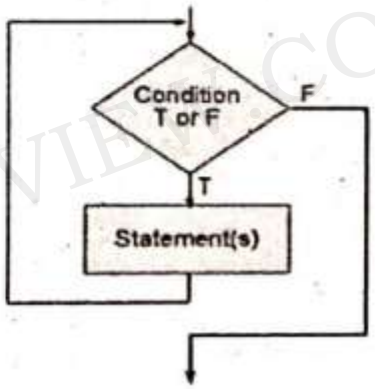


Figure: Execution of Iterative Statements

12.2 Relational Operators

The relational operators are used to specify conditions in programs. A relational operator compares two values. It produces result as **true** or **false**. The relational operators are sometimes called the **conditional operators** or **comparison operators** as they test conditions that are either **true** or **false**. VBScript provides the following relational operators:

Operator	Description
>	**Greater than operator** returns **true** if the value on left side of > is greater than the value on the right side. Otherwise returns **false**.
<	**Less than operator** returns **true** if the value on left side of < is less than the value on right side. Otherwise returns **false**.
=	**Equal to operator** returns **true** if the values on both sides of = are equal. Otherwise returns **false**.
>=	**Greater than or equal to operator** returns **true** if value on left side of >= is greater than or equal to the value on right side. Otherwise returns **false**.
<=	**Less than or equal to operator** returns **true** if the value on left side of <= is less than or equal to the value on right side. Otherwise returns **false**.
<>	The not equal to operator. Returns **true** if the value on the left side of <> is not equal to the value on the right. Otherwise returns **false**.

Table: Relational Operators

Relational Expression

A relational expression is a statement that uses relational operators to compare two values. The result of a relational expression can be **true** or **false**. Both sides of a relational expression can be a **constant, variable** or **arithmetic expression**.

Examples

Following are some examples that explain the use of different relational operators.

Relation Expression	Result
100 > 15	True
25 < 5	False
30 <= 12	False
40 >= 20	True
"Visual Basic" = "Java"	False
0 >= 0	True
0 <= 0	True
1 <> 2	True
5 <> 5	False

12.3 Conditional Statements

VBScript provides different selection or conditional structures for making decisions. These structures check the given condition before executing a statement or set of statements. The selection structures in VBScript are as follows:

- IF...THEN
- IF...THEN...ELSE
- SELECT...CASE

12.3.1 If...Then Structure

It is the simplest form of selection constructs. It is used to execute or skip a statement or set of statements by checking a **condition**. The condition is given as a relational expression. If the condition is **true**, the statement or set of statements after **THEN** keyword is executed. If the condition is **false**, the statement or set of statements after **THEN** keyword is not executed.

Syntax

```
IF condition THEN
      Statement(s)
END IF
```

IF	It indicates the start of IF structure.
condition	It indicates a condition that is given as a relational expression.
statement(s)	It indicates the statement or set of statements that will be executed if the condition is **true**. Any number of statements can be written between **THEN** and **END IF**. This area is called the **body** of **IF** statement.
END IF	It indicates the end of IF structure.

Example

```
<SCRIPT LANGUAGE="VBScript">
      If result = 1 Then
        Document.Write "Good"
</SCRIPT>
```

If only a single statement is to be executed in **Then** block, the complete statement can be written in one line as follows:

```
If result = 1 Then Document.Write "Good"
```

If there are multiple statements to execute in **Then** block, the first statement should begin on the next line after **Then**.

Practice 12.1

Write VBScript code that inputs a number from the user. It displays congratulation message if the number is 40 or above

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim num
num = Inputbox("Enter a number:","1")
if num >= 40 then
  document.write("Hurray! You have passed...")
end if
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb19.html".
3. Double click the file.

Practice 12.2

Write VBScript code that inputs five integer numbers and displays the maximum and minimum number from the group.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim a, b, c, d, e, max, min
  a = CInt(InputBox("Enter first integers: "))
  b = CInt(InputBox("Enter second integers: "))
  c = CInt(InputBox("Enter third integers: "))
  d = CInt(InputBox("Enter four integers: "))
  e = CInt(InputBox("Enter five integers: "))
  max = a
  min = max
  If b < min Then min = b
  If c < min Then min = c
  If d < min Then min = d
  If e < min Then min = e
  If b > max Then max = b
  If c > max Then max = c
  If d > max Then max = d
  If e > max Then max = e
  document.write("Largest number is " & max & "<BR>")
  document.write("Smallest number is " & min)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb20.html".
3. Double click the file.

12.3.2 If...Then...Else Structure

IF...THEN...ELSE is two-way decision making statement that depends on a given condition. If the condition is **true**, it executes the statement or set of statement after **THEN** keyword. If the condition is **false**, it executes the statement or set of statement after **ELSE** keyword. In **IF...THEN...ELSE** structure:

- Both blocks of statement can never be executed.
- Both blocks of statements can never be skipped.

Syntax

The syntax of **IF...THEN...ELSE** structure is as follows:

```
IF condition THEN
  Statement(s)1
ELSE
  Statement(s)2
END IF
```

IF	It indicates the start of **IF...THEN...ELSE** structure.
condition	It indicates the condition that is specified as a relational expression.
statement(s)1	It indicates a statement or set of statements to be executed if condition is **true**.
statement(s)2	It indicates a statement or set of statements to be executed if the condition is **false**.
END IF	It indicates the end of **IF...THEN...ELSE** structure.

Flowchart

The flowchart of **IF...THEN...ELSE** structure is as follows:

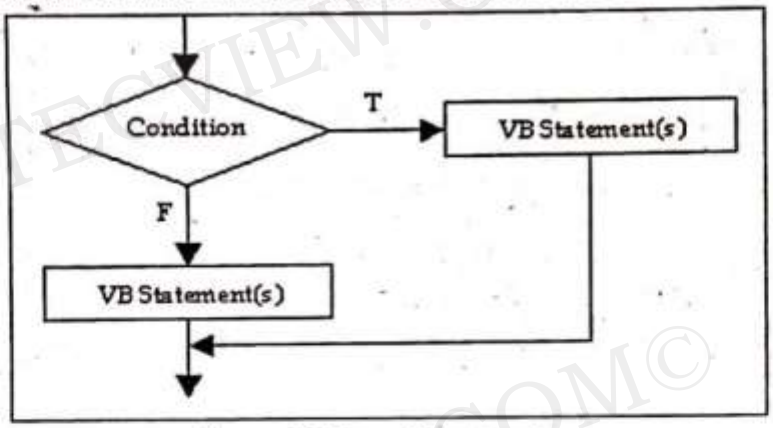


Figure: If-Then-Else structure

Practice 12.3

Write a VBScript code to get a number from the user and display whether it is even or odd

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n
n = InputBox("Enter any number: ", "Number", "0")
if n Mod 2 = 0 then
   MsgBox "Number is even"
else
   MsgBox "Number is odd"
end If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb21.html".
3. Double click the file.

Practice 12.4

Write VBScript code that inputs two numbers and finds whether the second number is multiple of first number or not.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n1, n2
n1 = CInt(InputBox("Enter first number: ", "Number1", "0"))
n2 = CInt(InputBox("Enter second number: ", "Number2", "0"))
If n2 Mod n1 = 0 Then
MsgBox n2 & "is multiple of " & n1
Else
MsgBox n2 & " is not multiple of " & n1
End If
```

```
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb22.html".
3. Double click the file.

Practice 12.5

Write VBScript code that inputs a string from user and checks the string is a palindrome or not. A palindrome is a string that reads the same backwards as forwards such as MADAM and MOM.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim nam , rev1
nam = InputBox("Enter the word : ")
rev1 = StrReverse(nam)
If nam = rev1 Then
MsgBox "The Word Is Palindrome"
Else
MsgBox "The Word Is Not Palindrome"
End If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb23.html".
3. Double click the file.

Practice 12.6

Write VBScript code that inputs salary and grade. It adds 50% bonus if grade is greater than 15 and 25% bonus if grade is 15 or less. Finally, the program displays total salary.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim sal, grade , bonus
sal = inputBox("Enter your salary ", "Salary", "0")
grade = inputBox("Enter your grade ", "Grade", "0")
if grade > 15 then
   bonus = sal * 50 / 100
else
bonus = sal * 25 / 100
end If
sal = sal + bonus
MsgBox "Your total salary = " & sal
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb24.html".
3. Double click the file.

12.3.3 Extended If...Then...Else Structure

Extended **IF...THEN...ELSE** structure is used to choose one block of statements from many. This structure is used when there are many options and only one block of statements should be executed on the basis of condition.

Syntax

The syntax of this structure is:

```
IF condition 1 THEN
        Statement(s)
ELSEIF condition 2 THEN
        Statement(s)
ELSEIF condition n THEN
        Statement(s)
ELSE
        Statement(s)
END IF
```

Working of Extended IF...THEN...ELSE Structure

The test conditions in **IF...THEN...ELSE** with multiple alternatives are executed in a sequence until a **true** condition is found. If a condition is **true**, the block of statements following the condition is executed. The remaining blocks are skipped. If a condition is **false**, the block of statements following the condition is skipped. The statement after the last **ELSE** are executed if all conditions are **false**.

Flowchart

The flowchart of this structure is as follows:

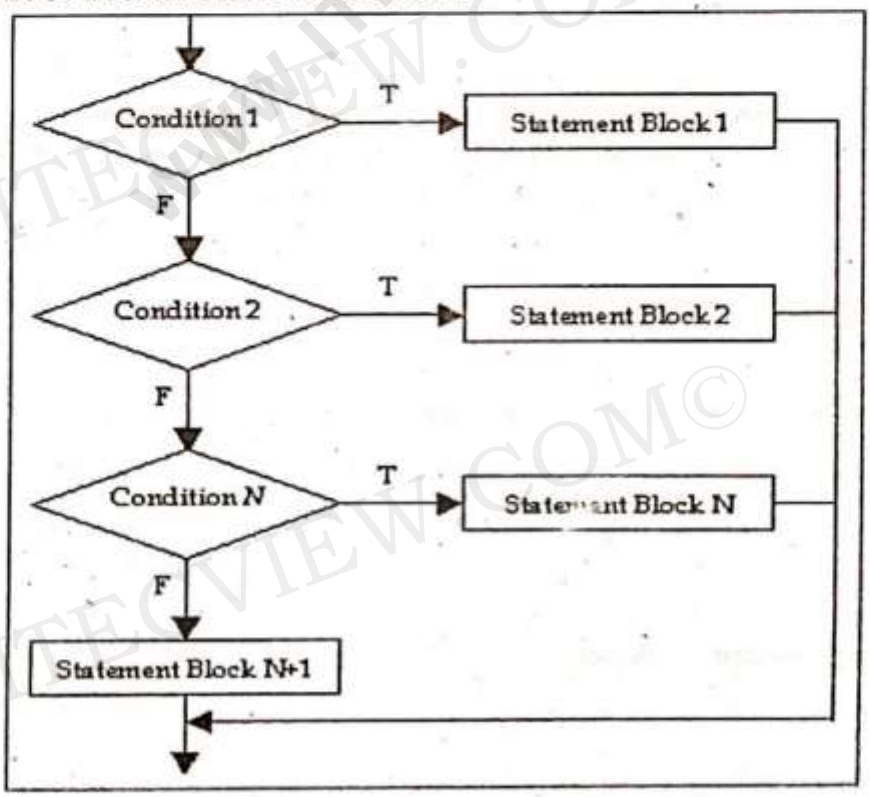


Figure: Extended IF...THEN...ELSE structure

Practice 12.7

Write VBScript code that inputs marks from the user. It displays the grade according to the following criteria:

80 or above	Grade A
60 – 79	Grade B
40 – 59	Grade C
below 40	Grade F

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim marks
marks = Inputbox("Enter marks:","0")
if marks >= 80 then
   document.write("Grade: A")
elseif marks >= 60 then
   document.write("Grade: B")
elseif marks >= 40 then
   document.write("Grade: C")
else
   document.write("Grade: F")
end if
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb25.html".
3. Double click the file.

Practice 12.8

Write VBScript code that inputs radius. It calculates area of circle if user enters 1 as choice. It calculates circumference if user enters 2 as choice. For example, if the user enters 3 as radius and 2 as choice, the program will display the area of circle as 28.269.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim radius, choice , area
choice = InputBox("Enter 1 for Area and 2 for Circum calc", "Calc", "0")
radius = InputBox("Enter Radius", "Radius", "0")
if choice = 1 then
   area = radius * radius * 3.141
   MsgBox "Area of circle = " & area
elseIf Choice = 2 then
   circum = 2 * 3.141 * radius
   MsgBox "Circumference of circle = " & circum
else
   MsgBox "Invalid Choice"
end If
</SCRIPT>
```

```
</BODY>
</HTML>
```

2. Save the file as " Vb26.html ".
3. Double click the file.

12.3.4 Compound Comparison

A type of comparison in which more than one conditions are evaluated is called **compound condition.** It is used to execute a statement or set of statements by testing many conditions.

Example

For example, a program inputs two numbers. It displays **OK** if one numbers is greater than 100 and second number is less than 100. Compound condition is executed by using logical operators..

12.3.5 Logical Operators

Logical operators are used to evaluate more than one condition in single conditional structure. There are three logical operators:

Operator	Usage	Description
And	If (A > 50) And (B > 50)	It produces True if both conditions are true. It produces False result if any one condition is false.
Or	If (A > 50) Or (B < 50)	It produces True if either condition true. It produces False result if both conditions are false.
Not	If Not (N = 100)	It produces the opposite true or false result.

Table: Logical operators

Suppose, we have two variables A = 100 and B = 50. The use of logical operators is explained with the following examples:

Example	Result
If A > 50 And B > 50	False
If A > 50 Or B > 50	True
If A > 100 And B < 100	False
If A > 100 Or B < 100	True
If Not (A = 100)	False
If Not (B < 50)	True

Table: Logical operators examples

Practice 12.9

Write VBScript code that inputs three numbers and displays the maximum number using Logical operators.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim a, b, c
a = Cint(inputbox("Enter a number","0"))
b = Cint(inputbox("Enter a number","0"))
c = Cint(inputbox("Enter a number","0"))
document.write("a = " & a)
```

```
document.write("<BR>b = " & b)
document.write("<BR>c = " & c)
document.write("<BR>")
if a>b AND a>c then
  document.write("Maximum number is " & a)
elseif b>a AND b>c then
  document.write("Maximum number is " & b)
else
  document.write("Maximum number is " & c)
end if
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb27.html".
3. Double click the file.

Practice 12.10

Write a VBScript code that inputs a character and displays whether it is a vowel or not

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim ch
ch = InputBox("Enter character: ", "Char", "")
if (Ch = "A") Or (Ch = "a") Or (Ch = "E") Or _
(Ch = "e") Or (Ch = "I") Or (Ch = "i") Or _
(Ch = "O") Or (Ch = "o") Or (Ch = "U") Or (Ch = "u") then
  MsgBox "You entered a vowel"
else
  MsgBox "Entered character is not a vowel"
end If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb28.html".
3. Double click the file.

Practice 12.11

Write VBScript code that allows user to enter character through keyboard and determines whether it is capital letter, small case letter, digit number or a special symbol.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim ch
ch = InputBox("Enter any character: ", "Enter character")
  If ((ch >= "A") And (ch <= "Z")) Then
    document.write ("The character " & ch & " is a capital letter.")
  ElseIf ((ch >= "a") And (ch <= "z")) Then
    document.write ("The character " & ch & " is a small case letter.")
```

```
  ElseIf ((ch >= "0") And (ch <= "9")) Then
    document.write ("The character " & ch & " is a digit.")
  Else
    document.write("The character " & ch & " is a symbol.")
End If

</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb29.html".
3. Double click the file.

Practice 12.12

Write VBScript code that inputs a number from the user and displays whether it is odd or even using NOT operator.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n
n = InputBox("Enter no.: ", "Number", "0")
if (NOT (n Mod 2 = 0)) then
  MsgBox "Number is odd"
else
  MsgBox "Number is Even"
end If
</SCRIPT>
</BODY>
</HTML>
```

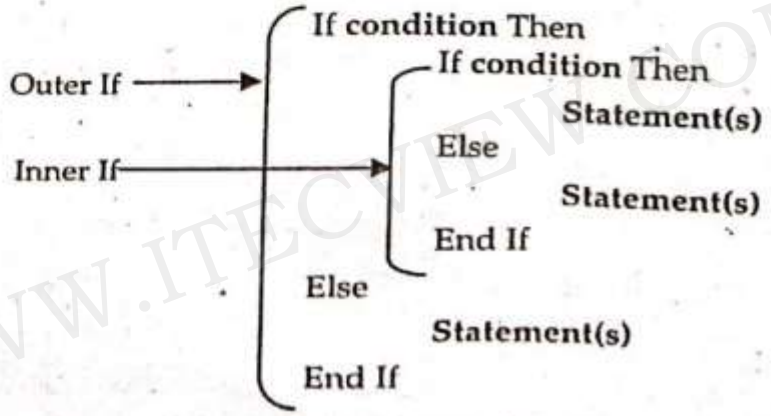
2. Save the file as "Vb30.html".
3. Double click the file.

12.3.6 Nested If...Then...Else Structure

An IF structure within an IF structure is called nested IF structure.

Syntax

The syntax of nested If is as follows:



Working of Nested If

In **nested IF** statement, the condition of outer IF is evaluated first. If it is **true**, the control enters in the inner **IF** block. If the condition is **false**, the inner **IF** is skipped and control directly moves to the **ELSE** part of outer **IF**. If outer **IF** is **true** then control enters in the inner **IF** statement. The inner **IF** evaluated according to simple **IF** statement.

Practice 12.13

Write a VBScript code that inputs three numbers and displays smallest number by using Nested If Condition

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n1,n2,n3
n1 = inputBox("Enter first number: ", "Number1", "0")
n2 = inputBox("Enter second number: ", "Number2", "0")
n3 = inputBox("Enter third number: ", "Number3", "0")
If n1 < n2 then
  If n1 < n3 Then
    MsgBox n1 & " is smallest number"
  else
    MsgBox n3 & " is smallest number"
  end If
else
  If n2 < n3 Then
    MsgBox n2 & " is smallest number"
  else
    MsgBox n3 & " is smallest number"
  end If
end If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb31.html".
3. Double click the file.

Practice 12.14

Write a VBScript code that inputs three numbers and displays whether all the numbers are equal or not using Nested If Condition

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n1, n2, n3
n1 = inputBox("Enter first number: ", "Number1", "0")
n2 = inputBox("Enter second number: ", "Number2", "0")
n3 = inputBox("Enter third number: ", "Number3", "0")
If n1 = n2 Then
  If n1 = n3 Then
    MsgBox "All numbers are equal"
```

```
  else
     MsgBox "Numbers are different"
  end If
else
     MsgBox "Numbers are different"
end If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb32.html".
3. Double click the file.

12.3.7 Select...Case Structure

Select...Case statement is another conditional structure. It is used for checking the result of a single expression with multiple values. The expression is evaluated at the top of the structure and its result is compared with different values. If the result matches with any value, the corresponding block of statements are executed.

The syntax for writing this structure is as follows:

```
Select Case Expression
  Case value
    One or more VBSCRIPT statements
  Case value
    One or more VBSCRIPT statements


  Case Else
    One or more VBSCRIPT statements
End Select
```

Here, Select Case, Case Else and End Select are keywords. Expression can be any valid expression that results in integer or string value. You can use any number of "Case value" statements. Each "Case value" represents a choice.

This structure is a good alternative of nested If structure. It can easily be used when there are many choices available. It is very simple and easy to understand.

Practice 12.15

Write VBScript code that inputs the day of week as number from the user and displays the name of the day starting from Friday. For example if the user enters 3, it displays Sunday.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n
n = inputbox("Enter No. of day in week:","1")
select case n
 case 1
  document.write("Friday")
 case 2
  document.write("Saturday")
 case 3
  document.write("Sunday")
```

```
  case 4
   document.write("Monday")
  case 5
   document.write("Tuesday")
  case 6
   document.write("Wednesday")
  case 7
   document.write("Thursday")
  Case else
   document.write("Invalid day")
end select
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb33.html".
3. Double click the file.

Practice 12.16

Write VBScript code that uses the following categories of movies:

A	for Adventure Movies
C	for Comedy Movies
F	for Family Movies
H	for Horror Movies
S	for Science Fiction Movies

The program inputs code for movie type and displays its category. For example, if the user enters "H", it displays "Horror Movies".

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim movieCh
movieCh = InputBox("Enter Category Code A, C, F, H or S", _
"CategoryCode", "")
select case movieCh
  case "A"
    MsgBox "Your selection is Adventure Movies"
  case "C"
    MsgBox "Your selection is Comedy Movies"
  case "F"
    MsgBox "Your selection is Family Movies"
  case "H"
    MsgBox "Your selection is Horror Movies"
  case "S"
    MsgBox "Your selection is Science Fiction Movies"
  case Else
    MsgBox "Please select any valid category from A, C, F, H or S"
end Select
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb34.html".
3. Double click the file.

12.4 Loops

VBScript provides a control structure for repeating the same statement again and again. This is known as **iterative, repetitive** or **looping** structure. There are different types of loops available in VBscript. Loops are basically used for two purposes:

- To execute a statement or number of statements for a specified number of times. For example, you may want to display a message for 100 times etc.
- To use a sequence of values. For example, you may want to display first 50 natural numbers etc.

12.4.1 While...Wend Loop

WHILE...WEND is the simplest loop of Visual BASIC language. It executes one or more statements while the given condition remains **true**.

Syntax

The syntax of **WHILE...WEND** loop is as follows:

```
WHILE condition
   statement(s)
WEND
```

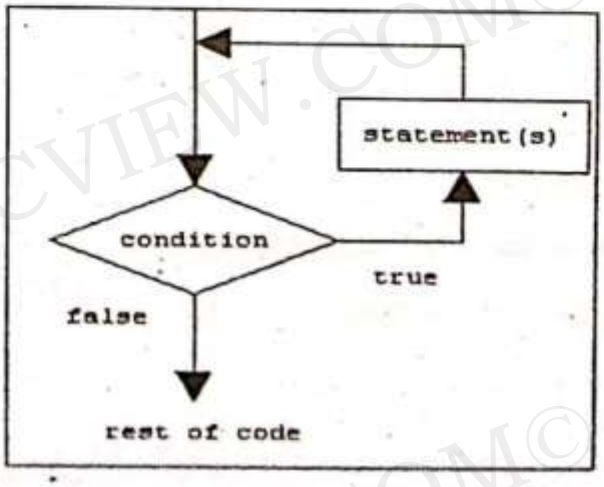
WHILE	It indicates the start of loop.
condition	It is given as relational expression. The loop executes as long as the condition remains true.
statements(s)	It indicates the statement or set of statements to be repeated by the loop.
WEND	It indicates the end of loop. The statements written between WHILE and WEND are repeated by the loop.

Working of WHILE...WEND Loop

First of all, the condition is evaluated. If it is **true**, the control enters the body of the loop and executes all statements. After executing the statements, it again moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains **true**. When the condition becomes **false**, the loop is terminated.

Flowchart

The flowchart of **while** loop is as follows:



Practice 12.17

Write VBScript code that prints "Pakistan" five times on form using while loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i
i = 1
While i <= 5
  document.write( "Pakistan"  & "<BR>")
  i = i + 1
Wend
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb35.html".
3. Double click the file.

Practice 12.18

Write a VBScript code that displays first five numbers with their sum using While loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i, Sum
i = 1
Sum = 0
While i <= 5
  Document.write ( I & "<BR>")
  Sum = Sum + i
  i = i + 1
Wend
document.write ("Sum = " & Sum)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb36.html".
3. Double click the file.

Practice 12.19

Write a VBScript code that inputs starting and ending number from the user and then displays all even numbers in the given range using While loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim st, en, n
st = Cint(InputBox("Enter range start: ", "Range Start", "0"))
en = Cint(InputBox("Enter range end: ", "Range End", "0"))
n = st
```

```
While n <= en
   If n Mod 2 = 0 Then
      document.write ( n & "<BR>" )
   End If
   n = n + 1
Wend
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb37.html".
3. Double click the file.

Practice 12.20

Write VBScript code inputs a number from the user and displays a table of that number using while loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, count
n = InputBox("Enter any number: ", "Number", "0")
count = 1
While count <= 10
   document.write( n & " * " & count & " = " & n * count & "<BR>")
   count = count + 1
Wend
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb38.html".
3. Double click the file.

12.4.2 Do...While...Loop

This loop is similar to while...wend loop. It executes one or more statements while the given condition is true.

Syntax

The syntax of **do...while** loop is as follows:

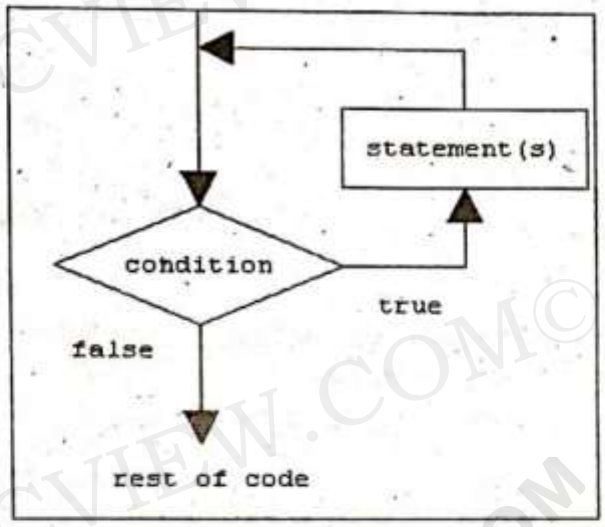
```
Do While condition
        Statement(s)
Loop
```

Working of do...while Loop

First of all, the condition is evaluated. If it is true, the control enters the body of the loop and executes all statements. After executing the loop body, it again moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains true. The keyword **Loop** indicates the end of loop body.

Flowchart

The flowchart of **do...while** loop is as follows:



Practice 12.21

Write VBScript code that displays back-counting from 10 to 1 using do-while loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i
i = 10
Do While i >= 1
   document.write ( i )
   document.write("<BR>")
   i = i - 1
Loop
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb39.html"
3. Double click the file.

Practice 12.22

Write VBScript code that gets two numbers from the user and displays the result of first number raise to the power second number

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n1, n2, pow, count
n1 = Cint(InputBox("Enter 1st number: ", "Number1", "0"))
n2 = Cint(InputBox("Enter 2nd number: ", "Number2", "0"))
count = 1
pow = 1
```

```
Do While count <= N2
   pow = pow * n1
   count = count + 1
Loop
document.write ( "Result = " & pow )
document.write("<BR>")
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb40.html" .
3. Double click the file.

Practice 12.23

Write VBScript code that inputs a number from user and displays all fibonacci numbers up to that number. In Fibonacci sequence, sum of two successive terms gives the third term.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim a,b,nxt,n,count
n = CInt(InputBox("How many fibonnacci terms required:"))
a = 0
b = 1
document.write ("Fibonacci terms are" & "<BR>" & a & "<BR>" & b & "<BR>")
count = 2
Do While count < n
   nxt = a + b
   document.write nxt & "<BR>"
   count = count + 1
   a = b
   b = nxt
Loop
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb41.html" .
3. Double click the file.

Practice 12.24

Write VBScript code that displays first ten odd numbers using do-while loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i
i = 1
Do
   If i Mod 2 <> 0 Then
      document.write( i)
      document.write("<BR>")
   End If
```

```
  i = i + 1
Loop While i <= 20
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb42.html".
3. Double click the file.

Practice 12.25

Write VBScript code that inputs a number. It then displays the sum of all odd numbers and the sum of all even numbers from 1 to the number entered by the user. For example, if the user enters 5, the sum of odd numbers is 9 and sum of even numbers is 6.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, oddsum, evensum
  oddsum = 0
  evensum = 0
  n = CInt(InputBox("Enter a number to calculate sum:"))
  Do While n >= 0
    If n Mod 2 = 0 Then
      evensum = evensum + n
    Else
      oddsum = oddsum + n
    End If
    n = n - 1
  Loop
  document.write "The sum of even digits is: " & evensum & "<BR>"
  document.write "The sum of odd digits is: " & oddsum
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb43.html".
3. Double click the file.

Practice 12.26

Write VBScript code that continuously inputs positive integer values from the user. The user enters a 0 to show that he has no more values to enter. The program should finally display the largest and smallest number entered.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim num,sum,min,max,count
count = 0
num = CInt(Inputbox("Enter positive number:"))
min = num
max = num
do while num>0
  count = count+1
```

```
  if num > max then
    max = num
  Elseif num<min then
     min = num
  End if
  Num = Inputbox("Enter number (o to exist): ")
Loop
if count = 0 then
 document.write( "No positive number entered. ")
else
 Msgbox "Largest number: " & max & vbNewLine _
      & "Smallest number: " & min
End if
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb44.html".
3. Double click the file.

Practice 12.27

Write VBScript code that inputs a number from user and prints lowest and highest digit in the number. For example, if the user enters 219, the highest digit is 9 and lowest digit is 1.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n , high , low , num , i , reman
   n = CInt(InputBox("Enter a number:"))
   num = n
   high = n Mod 10
   low = n Mod 10
   n = n \ 10
   i = n
   Do While i >= 1
      reman = i Mod 10
      If reman > high Then
         high = reman
      End If
      If reman < low Then
         low = reman
      End If
      i = i \10
  Loop
  document.write " The highest digit of " & num & " is " & high & "<BR>"
  document.write " The lowest digit of " & num & " is " & low
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb45.html".
3. Double click the file.

12.4.3 Do...Until...Loop

This loop executes one or more statements while the given condition remains false.

Syntax

The syntax of this loop is as follows:

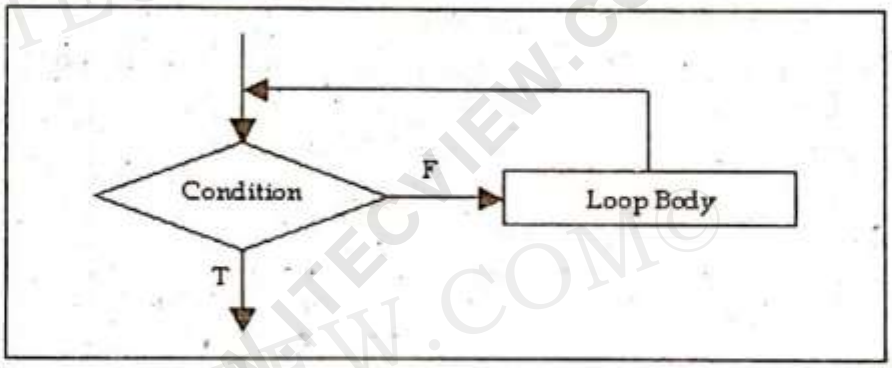
```
Do Until condition
        Statement(s)
Loop
```

Working of do...until Loop

First of all, the condition is evaluated. If it is false, the control enters the body of the loop and executes all statements. After executing the loop body, it moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains false. The keyword Loop indicates the end of loop body.

Flowchart

The flowchart of **do...until** loop is as follows:



Practice 12.28

Write VBScript code that inputs a number from the user and finds its factorial.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, c
c=1
f=1
n=CInt(inputbox("Enter a number :"))
Do Until c>n
  f=f*c
  c=c+1
Loop
Msgbox "Factorial of " & n & " is " & f
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb46.html".
3. Double click the file.

12.4.4 Do...Loop...While

This loop executes one or more statements while the given condition is true. This loop is similar to **Do...while...Loop** except that it first executes the loop body and then checks the condition. It means that the body of the loop is executed once even if the condition is false. So this loop should be used when you want to execute a statement at least once.

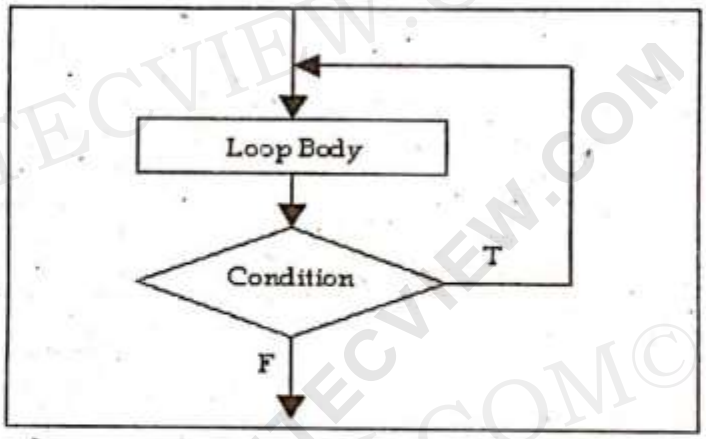
Syntax

The syntax of this loop is as follows:

```
Do
        Statement(s)
Loop While condition
```

Flowchart

The flowchart of do loop...while is as follows:



Practice 12.29

Write VBScript code that inputs a number from the user and displays factorial of that number using While loop in another

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, fact, temp
fact = 1
temp = 1
n = Cint(InputBox("Enter no: ", "Fact Number", "0"))
Do
   fact = fact * temp
   temp = temp + 1
Loop While temp <= n
document.write ( " Factorial of " & n & " = " & fact )
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb47.html".
3. Double click the file.

Practice 12.30

Write VBScript code to display table of 13.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, i
i=1
do
 document.write(13 & " * " & i & " = " & 13 * i & "<BR>")
  i=i+1
Loop while i<=10
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb48.html".
3. Double click the file.

Practice 12.31

Write VBScript code that continuously inputs positive integer values from the user. The user enters a zero to show that he has no more values to enter. The program shoudl finally display the second largest number entered.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
dim numb,first,second
first=numb
second=numb
do
 if numb>first then
    second=first
    first=numb
 elseif numb>second then
    second=numb
 end if
 numb=CInt(Inputbox("Enter number(Zero to exit):"))
loop while numb<>0
Msgbox "Second Largest Number:" & second
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb49.html".
3. Double click the file.

Practice 12.32

Write VBScript code that inputs a number and finds whether it is a palindrome or not. A palindrome is a number that reads the same backward and forward such as 62526 and 4994.

1. Open Notepad and type the following code:

```
<HTML>
```

```
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n , num , digit , rev
rev = 0
num = CInt(InputBox("Enter a positive number: "))
   n = num
   Do
      digit = num Mod 10
      rev = (rev * 10) + digit
      num = num \ 10
  Loop While (num <> 0)
      document.write("The reverse of the number is : " & rev)
   If (n = rev) Then
      document.write( " The number is a palindrome")
   Else
      document.write( " The number is not a palindrome")
   End If
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb50.html".
3. Double click the file.

12.4.5 Do...Loop...Until

This loop executes one or more statements while the given condition is **false**. This loop is similar to **Do...Until...Loop** except that it first executes the loop body and then checks the condition. It means that the body of the loop is executed once even if the condition is **true**. So this loop should be used when you want to execute a statement at least once.

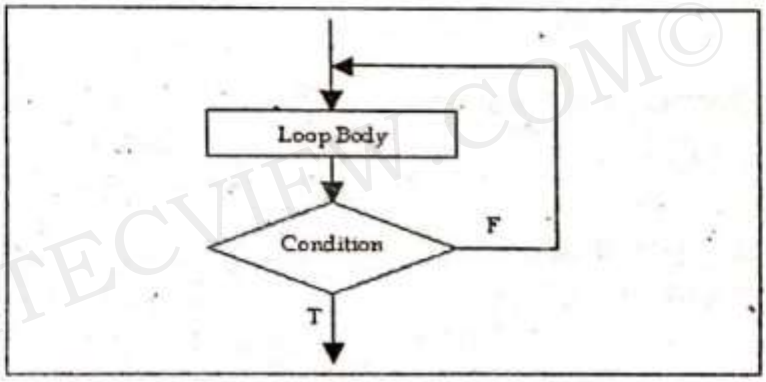
Syntax

The syntax of this loop is as follows:

```
Do
        Statement(s)
Loop Until condition
```

Flowchart

The flowchart of **do...loop...until** is as follows:



Practice 12.33

Write VBScript code that inputs starting and ending number from the user and then displays all odd numbers in the given range using do-loop until

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim st, en, n
st = Cint(InputBox("Enter range start: ", "Range Start", "0"))
en = Cint(InputBox("Enter range end: ", "Range End", "0"))
n = St
Do
  If n Mod 2 <> 0 Then
     document.write( n)
     document.write("<BR>")
  End If
  n = n + 1
Loop Until n > en
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb51.html".
3. Double click the file.

12.4.6 For...Next Loop

This loop executes one or more statements for a specified number of times.

Syntax

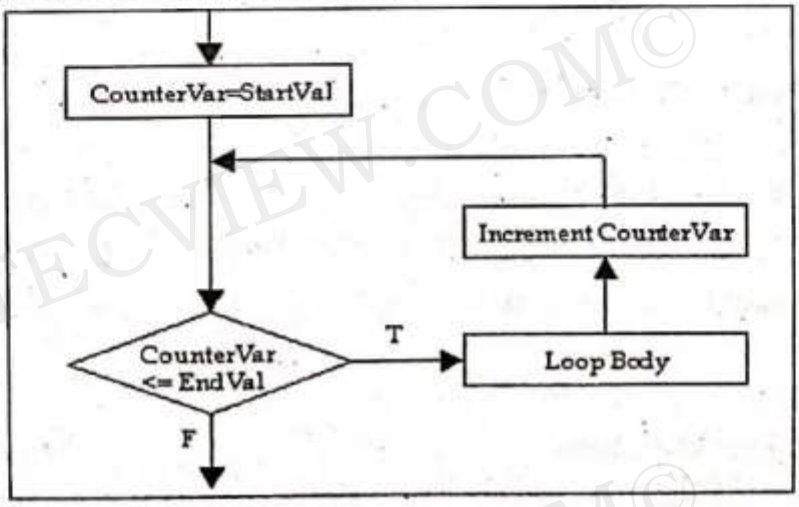
The syntax of this loop is as follows:

```
For <counter variable> = <start value> To <end value> Step <increment>
        Statement(s)
Next
```

The number of iterations is determined by the start value and end value of counter variable. By default, the counter variable is incremented by 1 after each iteration. But you can change this value by using **Step** clause, which is optional. The keyword **Next** indicates the end of loop body.

Flowchart

The flowchart of **For...Next** is as follows:



Practice 12.34

Write VBScript code that displays the sum of first five numbers using For...Next loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n
For n = 1 To 10
  document.write(n)
  document.write("<BR>")
Next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb52.html".
3. Double click the file.

Practice 12.35

Write a VBScript code that displays first five odd numbers using For...Next loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        Document.Write ("First five odd numbers:<BR>")
        for i = 1 to 10 step 2
          Document.Write (i & "<BR>" )
        next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb53.html".
3. Double click the file.

Practice 12.36

Write VBScript code to calculate and display product of all odd numbers from 1 to 15.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim product,c
product=1
for c = 1 to 15 step 2
  product = product * c
next
document.write("Result is " & product)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb54.html".
3. Double click the file.

Practice 12.37

Write a VBScript code that inputs a number from the user to display its table and length up to which table should be displayed using For Next loop

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, l, count
n = InputBox("Enter any number: ", "Number", "0")
l = InputBox("Enter table length: ", "Table Length", "0")
For count = 1 To l
   document.write( n & " * " & count & " = " & n * count)
   document.write("<BR>")
Next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb55.html" .
3. Double click the file.

Practice 12.38

Write a VBScript code that finds the sum of squares of the integers from 1 to n where n is a positive integer value entered by the user (i.e. Sum = $1^2 + 2^2 + 3^2 + \ldots n^2$).

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, count, sum
sum = 0
n = Cint(InputBox("Enter any number: ", "Number", "0"))
For Count = 1 To n
   sum = sum + (count * count)
Next
document.write ( "The sum of squares " & Sum)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb56.html" .
3. Double click the file.

Practice 12.39

Write VBScript code that takes n numbers as input. It outputs the frequency of positive and negative numbers. Supposes the user inputs 12 in variable n and enters the following numbers:

40 0 -1 59 -37 283 61 387 -91 -103 341 523

The result should be as follows:

Frequency of positive numbers: 8

Frequency of negative numbers: 4

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n,num,pos,neg
n=Inputbox("How many numbers to input? ", "Number", "0")
for i=1 to n
   num=Inputbox("Enter a number:","Number","0")
   if (num<0) Then
     neg=neg+1
   Else
     pos=pos+1
   End if
Next
document.write("Total positive numbers:" & pos & "<BR>")
document.write("Total negative numbers:" & neg)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb57.html" .
3. Double click the file.

Practice 12.40

Write VBScript code that displays first five odd numbers in reverse order.

9 7 5 3 1

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Area of circle </TITLE>
 <SCRIPT LANGUAGE="VBSCRIPT">
        dim n
        for n = 9 to 1 step -2
                document.write(n & " ")
        next
 </SCRIPT>
 </HEAD>
 <BODY>
 </BODY>
</HTML>
```

2. Save the file as "Vb58.html" .
3. Double click the file.

Practice 12.41

Write a VBScript code that enters an integer from the user and displays the factorial of that number

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n, fact, temp
fact = 1
```

```
n = Cint(inputbox("enter any number"))
For temp = 1 To n
   fact = fact * temp
Next
document.write (" The factorial is" &  fact)
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb59.html" .
3. Double click the file.

12.4.7 Early Termination of Loops

Normally, the termination of the loop depends on the given condition. But you can terminate a loop earlier, without looking at the condition. **Exit For** statement is used in the body of **For...Next** loop to terminate it abnormally. **Exit Do** statement is used with Do loops for the same purpose.

Practice 12.42

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
        for i = 1 to 10
          Document.Write (i & "<BR>" )
          if I >= 3 then exit for
        next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb60.html".
3. Double click the file.

Practice 12.43

Write VBScript code that inputs a number from the user and displays all prime numbers up to the entered number.

1. Open Notepad and type the following code:

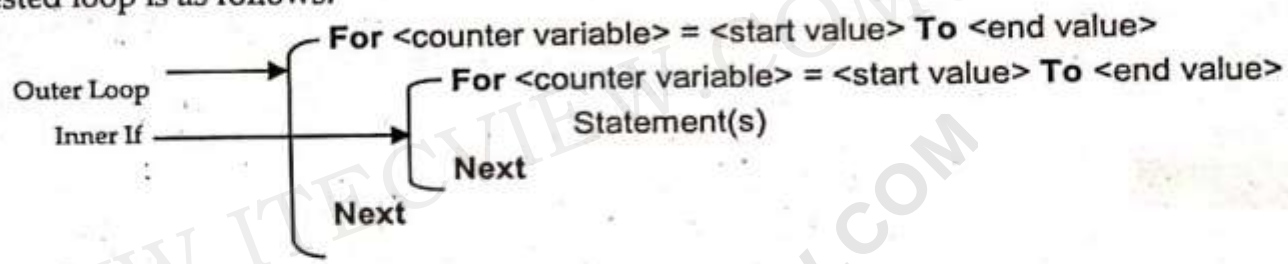
```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i ,n,p,j
n = InputBox("Enter a number: ")
For i = 1 To n
   p = 1
      For j = 2 To i - 1
         If i Mod j = 0 Then
            p = 0
            Exit For
         End If
      Next
      If p = 1 Then
         document.write i & "<BR>"
```

```
    End If
Next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb61.html".
3. Double click the file.

12.4.8 Nested Loops

A loop within a loop is called nested loop. In nested loops, the inner loop is executed completely with each change in the value of counter variable of outer loop. The syntax of nested loop is as follows:



Practice 12.44

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim i, j
    for i = 1 to 5
      for j = 1 to 10
        Document.Write ("* ")
      next
      Document.Write ("<BR>")
    next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb62.html".
3. Double click the file.

Exercise Questions

1. Write VBScript code that inputs the salary of an employee from the user. It deducts the income tax from the salary on the following basis :
   20% income tax if the salary is above Rs. 30000.
   15% income tax if the salary is between Rs. 20000 and Rs. 30000.
   10% income tax if the salary is below Rs. 20000.
   The program finally displays salary, income tax and the net salary.
2. Write VBScript code to get three numbers from user for integer variables a, b and c. If a is not zero, find out whether it is the common divisor of b and c.
3. Write VBScript code that gets a century from the user and displays all leap years in that century.
4. Write VBScript code to get an integer from the user and display whether it is prime or not.
5. Write VBScript code to display first five numbers with their squares and cubes.
6. Write VBScript code that inputs starting and ending numbers and option from user. If the option is "E", it displays all even numbers in the given range and their sum. If the option is "O", it displays all odd numbers in the given range and their sum.
7. Write VBScript code that inputs the number of students in the class. It then inputs the marks of these students and displays the highest and second highest marks.

WWW.ITECVIEW.COM©

Procedures, Functions and Arrays

13.1 Procedures

A procedure is a block of code that performs some operation. The code written is executed only when it is **called**. A simple procedure can be called any number of times by just writing its name. When a user-defined procedure is called, the control moves in that procedure. All statements of the procedure are executed and then the control again moves back to the point where the procedure was called.

Syntax

The basic syntax for a procedure is as follows:

```
Sub procedure_name
        Statement(s)
End Sub
```

Sub and **End Sub** are the keywords that indicate the start and end of procedure body. procedure_name is the name of procedure. The rules for procedure name are similar to the rules for declaring a variable.

Example

```
Sub Message()
Reponse.Write ("Hello World")
End Sub
```

Practice 13.1

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
        sub show()
          Document.write("I'm a procedure.<br>")
        end sub
        Document.Write ("Let us call a procedure...<br>")
        call show
        Document.Write ("Let us call it again...<br>")
        call show
</SCRIPT>
```

2. Save the file as "Vb63.html".
3. Double click the file.

13.1.1 Passing Parameters to Procedures

Parameters are the values, which are passed to a procedure when the procedure is called. Parameters are given in the parentheses. The number of parameters and their type is decided during procedure coding.

Practice 13.2

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
    Dim a, b
    a = 10
    b = 20
    call sum(a, b)
    sub sum(x, y)
      Document.write("Sum = " & x+y)
    end sub
</SCRIPT>
```

2. Save the file as "Vb64.html".
3. Double click the file.

13.2 Functions

A function is a block of code that performs some action. It is similar to the procedure. The difference is that it can return a value back to the position where it is called. The statements written in a function are executed only if it is called by its name. When a user-defined function is called, the control moves in that function. All statements of the procedure are executed and then the control again moves back to the point where the function was called along with the returned value.

Syntax

The basic syntax for a function is as follows:

```
Function function_name
  Statement(s)
End Function
```

Function and **End Function** are the keywords that indicate the start and end of function body. function_name is the name of function. The rules for function name are similar to the rules for declaring a variable.

Example

```
Function Msg()
  Document.Write("Hello Function!")
End Function
```

13.2.1 Returning a Value from Function

A function can return a value back to the calling point. You can return a value by simply assigning that value to the name of the function. Assume that the name of the function is a variable. Any value assigned to the name of the function is actually returned. In the above figure, 10 is returned to the point from where the function was called.

Practice 13.3

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
        Dim a, b, c
        a = 100
        b = 200
        c = sum(a, b)
        Document.Write("Sum = " & c)

        function sum(x, y)
          Dim s
          s = x+y
          sum = s
        end function
</SCRIPT>
```

2. Save the file as "Vb65.html".
3. Double click the file.

13.3 Built-in Functions

Visual Basic provides many built-in functions for different purposes. These functions save a lot of time. The use of these functions makes Visual Basic programming easier and faster. There are different types of built-in functions:

- Date & Time Functions
- Informational Functions
- Numeric Functions
- Conversion Functions

13.3.1 Date & Time Functions

VBScript provides different built-in function to manipulate dates and times.

Date

Date function is used to display current system date. The following example will display current date on browser in "mm/dd/yyyy" format.

```
document.write(Date)
```

Time

Time function is used to display current system time. The following example will display the current time on the browser.

```
document.write(Time)
```

Now

Now function is used to display both current date and time. The following example will display the current date and time on the browser.

```
document.write(Now)
```

Day

Day function returns the day part of the date. It uses a parameter of type date. The following example will display 22 on the browser.

```
D = "01/22/2003"
document.write(Day(D))
```

Month

Month function returns the month part of the date. It uses a parameter of type date. The following example will display 1 on the browser.

```
D = "01/22/2003"
Document.write Month(D)
```

Year

Year function returns the year part of the date. It uses a parameter of type date. The following example will display 2003 on the browser.

```
D = "01/22/2003"
Document.write Year(D)
```

Hour

Hour function returns the Hour part of the time. It accepts a parameter of type time. The following example will display 19 on the browser. If you change "PM" to "AM" then it will display 7. You can also use "Now" as parameter.

```
T = "7:30:45 PM"
Document.write Hour(T)
```

Minute

Minute function returns the Minute part of the time. It accepts a parameter of type time. The following example will display 30 on the browser. You can also use "Now" as parameter.

```
T = "7:30:45 PM"
Document.write Minute(T)
```

Second

Second function returns the Second part of the time. It accepts a parameter of type time. The following example will display 45 on the browser. You can also use "Now" as parameter.

```
T = "7:30:45 PM"
Document.write Second(T)
```

WeekDay

WeekDay function returns the day number of the week. The number starts from 1 indicating Sunday. It uses a parameter of type date. The following example will display 4 on the browser. The number 4 indicates 4th day i.e. Wednesday.

```
D = "01/22/2003"
Document.write WeekKey(D)
```

WeekDayName

WeekDayName function returns the name for the day of the week. The first parameter is an integer representing the day of the week. The second parameter determines if you want an abbreviated weekday name. A value of **True** will abbreviate the name. The following example will display "Wednesday" on the browser:

```
D = "01/22/2003"
Document.write WeekDayName(WeekDay(D), False)
```

The following example will display "Wed" on the browser.

```
Document.write WeekDayName(WeekDay(D), True)
```

DateAdd

DateAdd function is used to add or subtract from a date. It has three parameters. The first parameter indicates the part of date to be added or subtracted. The second parameter indicates the amount to be added or subtracted. The third parameter is any valid date to be processed. The following example will display "1/22/2008" on browser. It adds 5 years in date stored in D. A negative number is used to subtract an amount from the date.

```
D = "01/22/2003"
Document.write DateAdd("yyyy", 5, D)
```

Option	Description
yyyy	Add or subtract years
m	Add or subtract months
d	Add or subtract days
h	Add or subtract hours
n	Add or subtract minutes
s	Add or subtract seconds

Tables: Options for **DateAdd** function

DateDiff

DateDiff function is used to find the difference between two dates. It has three parameters. The first parameter indicates the format for displaying the difference. The second and third parameters are any valid dates on which the process will be applied. The following example will display 1 on the browser as the difference between two dates is of 1 year.

```
D1 = "01/22/2003"
D2 = "01/22/2004"
Document.write DateDiff("yyyy", D1, D2)
```

Following example will display 365 as the difference between two dates is of 365 days.

```
Document.write DateDiff("d", D1, D2)
```

13.3.2 Numeric Functions

Numeric functions work on numbers. These functions are used frequently to perform different calculations on numbers.. Following are some important numeric functions.

Abs

Abs function is used to find the absolute value of a number. The absolute value of both -87 and 87 is 87. The following line will display 5:

```
Document.write Abs(-5)
```

Cos

Cos function is used to determine the cosine value of a number.

Log

Log function is used to determine the natural logarithm of a number.

Sin

Sin function is used to determine the sine value of a number.

Tan

Tan function is used to determine the tangent value of a number.

Round

Round function is used to round a numeric value with decimal points to the specified digits. The following line will display 5.333

```
Document.write Round(5.33333, 3)
```

The following line will display 5.667

```
Document.write Round(5.66666, 3)
```

Rnd

Rnd function is used to generate and random number that is greater than 0 and less than 1. The following line will display a random number between 0 and 1.

```
Document.write Rnd
```

The above statement is not very useful individually. It is used with other statements. The following line will display a whole number between 1 and 100.

```
Document.write Int(Rnd * 100) + 1
```

Sqr

Sqr function is used to find square root of a number. The following line will display 10.

```
Document.write Sqr(100)
```

Randomize

Rnd function may generate the same random number again and again. **Randomize** statement is used before **Rnd** function to get a different random each time. The following example will display a whole number between 1 and 100. This time the random numbers will be different each time.

```
Randomize
Document.write Int(Rnd * 100) + 1
```

13.3.3 Informational Functions

Visual Basic provides different informational functions as follows:

VarType

VarType function is used to find the type of a variable. It returns an integer value that indicates a particular type of variable. The following example will display 2 that indicates that the variable is of Integer type.

```
S = 10
Document.write VarType(S)
```

The following table shows all numbers for different type of variables.

Value	Meaning	Value	Meaning
0	Empty: Variable not initialized	7	Date
1	Null	8	String
2	Integer	9	OLE object
3	Long Integer	10	Error
4	Single	11	Boolean
5	Double	12	Variant
6	Currency	17	Byte

TypeName

TypeName function is used to find the type name of a variable. It returns a string value that indicates a particular type of variable. The following example will display **Integer** that indicates that the variable is of Integer type:

```
S = 10
Document.write TypeName(S)
```

IsNumeric

IsNumeric function is used to check whether the parameter is a numeric value or not. It returns **True** if the value of the parameter is numeric or can be converted to numeric form. The following example will display **True** that indicates that the value is numeric.

```
S = 10
Document.write IsNumeric(S)
```

IsDate

IsDate function is used to check whether the parameter is a date/time value or not. It returns **True** if the value of the parameter is date/time or can be converted to date/time form. The following example will return **True** because it can be converted to date value.

```
S = "1/1/2000"
Document.write IsDate(S)
```

IsArray

IsArray function is used to check whether the parameter is an array or not. It returns **True** if the parameter is array. The following example will display **False** that indicates that the value is not an array.

```
S = 10
Document.write IsArray(S)
```

IsEmpty

IsEmpty function is used to check whether the parameter is empty or not. A variable is empty when it is created but not initialized. It returns **True** if the parameter is empty. The following example will display **True** that indicates that the variable is empty.

```
Document.write IsEmpty(n)
```

IsNull

IsNull function is used to check whether the parameter is Null or not. It returns **True** if the parameter is null. Null is normally used in databases. The following example will display **False** that indicates that the value is null.

```
S = 10
Document.write IsNull(S)
```

13.3.4 Conversion Functions

VBScript provides different built-in function to convert different values.

Int

Int function is used to drop fractional part of positive number. The following example will display 5.

```
Document.write Int(5.5)
```

Fix

Fix function is used to drop fractional part of negative number. The following example will display -5.

```
Document.write Fix(-5.5)
```

CInt

CInt function is used to round a numeric value with decimal points to the nearest whole number. The following example display 9:

```
Document.write CInt(8.9)
```

The following example will display 8:

```
Document.write CInt(8.1)
```

The difference between Int, Fix and CInt functions is as follows:

Original Number	Int	Fix	CInt
3.14	3	3	3
4.7	4	4	5
-1	-1	-1	-1
-3.7	-3	-4	-4
-4.2	-4	-5	-4
4.5	4	4	4
1.5	1	1	2

CBool

CBool function is used to convert an expression into a **Boolean** value. It returns **false** if the value is 0 and **true** for all other numbers. The following example will display **true**:

```
Document.write CBool(5)
```

CByte

CByte converts an expression to **Byte**. It returns byte value of number from 0 to 255. It gives error for any other number. The following example will display 5:

```
Document.write CByte(5)
```

The following example will generate an error:

```
Document.write CByte(256)
```

CDate

CDate function is used to convert and expression into a date or time value. It returns 12/31/1899 if parameter is 1. The following example will display 1/1/1900:

```
Document.write CDate(2)
```

CDbl

CDble function is used to convert an expression into double number. It gives error if the expression is not a number.

CCur

CCur function is used to convert an expression into a currency value.

WWW.ITECVIEW.COM©

CLng

CLng function is used to convert an expression into a long integer number.

CSng

CSng function is used to convert an expression into a flotting-point (decimal) number.

CStr

CStr function is used to convert an expression into a string.

Chr

Chr function is used to associate a character with its ASCII character equivalent. It could be used to convert a number to a character. The following example will display **a**:

```
Document.write Chr(97)
```

13.4 Arrays

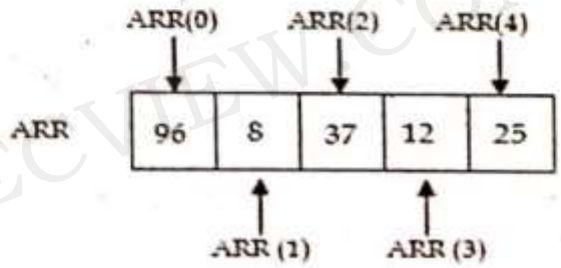
An **array** is a set of variables that can store similar type of data. This set of variables consists of different memory locations. These locations are adjacent in memory and are indicated with same name. Each memory location can store one value and is called element of array. The elements of the array are accessed with reference to their position or location number. This position number is called **index** or **subscript**.

The position of each element in array is different. The first element of array has an index value of 0. The index is incremented by 1 for each next element. If an array has five elements, the index numbers of these elements will be 0, 1, 2, 3 and 4. The index value or subscript is written in parentheses after the name of array.

The basic purpose of using array is to store a large number of similar types of data. For example, hundred variables are required to store the marks of hundred students. The use of individual variables will take a lot of time. It is also very difficult and time-consuming task to perform different processing on these variables. This task can be performed more easily by using arrays. The marks of 100 students can be stored in one array with 100 elements.

Example

Suppose an array **ARR** can store five integer values. In this case, the name of the array is **ARR** and its five elements are **ARR(0)**, **ARR(1)**, **ARR(2)**, **ARR(3)** and **ARR(4)**. The values written in parentheses along with array name indicate subscript values. These values indicate the locations of different array elements in the memory where the values of these elements are stored. It is shown in the following figure:



13.4.1 Advantages / Uses of Arrays

Some advantages of arrays are as follows:

- Arrays can store a large number of values with single name.

- Arrays are used to process many values easily and quickly.
- The values stored in an array can be sorted easily.
- A search process can be applied on arrays easily.

13.4.2 Declaring One-Dimensional Arrays

One-dimensional array is used to store different values of similar type as a single row or column. The syntax of declaring one-dimensional array is as follows:

```
Dim <array name> (length)
```

The following example declares an array to store the marks of five students:

```
Dim Marks(4)
```

This statement will create an array in the memory that consists of five memory locations numbered from 0 to 4. We can store five integers in this array.

Each block in the array is called **cell** or **element**. Each element is assigned a unique number in the array. This number is called the **index** or **subscript** of that element. The index is used to access an individual element of the array. Normally, index starts from 0. So, **Marks(0)** refers to the first element of the array, **Marks(1)** refers to the second element.

Practice 13.4

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
    Dim n(3)
    n(0) = 10
    n(1) = 20
    n(2) = 30
    n(3) = 40
    for i = 0 to 3
      Document.Write (n(i) & "<BR>")
    next
</SCRIPT>
```

2. Save the file as "Vb66.html".
3. Double click the file.

13.4.3 Initializing Array

The array can be initialized with the values as follows:

```
Dim variable_name = Array (list_of_values)
```

Dim and **Array** are keywords. variable_name is the name of array and list_of_values indicates the values stored in array.

Example

```
Dim a = Array(1,2,3)
```

Practice 13.5

Write a VBScript code that initializes an array with four integers. It then calculates and displays the sum of these elements.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Sum of Array </TITLE>
```

```
<SCRIPT LANGUAGE="VBScript">
        Dim n, i, s
        s = 0
        n = Array(10,20,30,40)
        for i = 0 to 3
          s = s + n(i)
        next
        Document.Write ("Sum of array elements = " & s)
</SCRIPT>
</HEAD>
</HTML>
```

2. Save the file as "Vb67.html".
3. Double click the file.

Practice 13.6

Write a VBScript code that inputs values from the user and saves into 10-elements array and also prints them in browser using one-dimensional array.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Array of 10 elements </TITLE>
  <SCRIPT LANGUAGE="VBSCRIPT">
        dim n, arr(10)
        for n = 0 to 9
                arr(n) = inputbox("Enter a number: ")
        next
        document.write("You entered the following values: <BR>")
        for n = 0 to 9
                document.write(arr(n) & " ")
        next
  </SCRIPT>
  </HEAD>
</HTML>
```

2. Save the file as "Vb68.html".
3. Double click the file.

Practice 13.7

Write a VBScript code that uses two For...Next loops. The first loop gets five inputs from the user and stores it in an array. The second loop prints the values stored in that array.

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim Marks(4)
For i = 0 To 4
   Marks(i) = Inputbox("Enter marks:", "Marks", "0")
Next
For i = 0 To 4
   document.write "Value at index " & i & "is " & Marks(i)
Next
</SCRIPT>
```

```
</BODY>
</HTML>
```

2. Save the file as "Vb69.html" .
3. Double click the file.

Practice 13.8

Write a VBScript code that gets five numbers from the user and stores them in the array. It then displays the sum and average of all numbers in the array

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim num(4) , sum
For i = 0 To 4
   Num(i) = Cint(Inputbox("Enter marks:", "Number", "0"))
Next
For i = 0 To 4
   sum=sum+Num(i)
Next
document.write "The sum is" & sum
document.write("<BR>")
document.write "The average is" & sum/5
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb70.html".
3. Double click the file.

Practice 13.9

Write a VBScript code that gets ten numbers from the users in array and displays the minimum number

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n(9) , Min
For i = 0 To 9
   n(i) = InputBox("Enter No.")
Next
Min = n(0)
For i = 1 To 9
   If n(i) < Min Then
      Min = n(i)
   End If
Next
document.write "Minimum number is " & Min
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb71.html" .

3. Double click the file.

Practice 13.10

Write a VBScript code that gets five numbers from the users in array and displays total number of even and odd numbers stored in the array

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim n(4) , Even , Odd
For i = 0 To 4
   n(i) = InputBox("Enter No.")
   If n(i) Mod 2 = 0 Then
      Even = Even + 1
   Else
      Odd = Odd + 1
   End If
Next
document.write "Total even numbers are " & Even
document.write("<BR>")
document.write "Total odd numbers are " & Odd
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb72.html".
3. Double click the file.

Practice 13.11

Write a VBScript code that gets five numbers from the user and stores them in the array. It then displays the numbers in reverse order

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="VBScript">
Dim Num(4)
For i = 0 To 4
   Num(i) = InputBox("Enter No.")
   Next
document.write( "Numbers in original order:<BR>")
For i = 0 To 4
   document.write Num(i) & "<BR>"
Next
document.write( "Numbers in reverse order:<BR>")
For i = 4 To 0 Step -1
   document.write Num(i) & "<BR>"
Next
</SCRIPT>
</BODY>
</HTML>
```

2. Save the file as "Vb73.html" .
3. Double click the file.

13.4.4 Searching in Arrays

Searching is a process of finding the required data in the array. Searching becomes more important when the length of the array is very large. A simple way of search an array for the desired value is as follows:

1. Visit the first element of the array and compare its value with the required value.
2. If the value of array matches with the desired value, the search is complete.
3. If the value of array does not match, move to next element and repeat same process.

Loops are frequently used to visit the elements of array for searching a value. You can start the counter variable of loop from 0 and move it to the last index of the array. For example if the array to be searched has 10 elements, you will start the counter variable from 0 and move to 9.

Practice 13.12

Write a VBScript code that initializes an array and then search a number from array.

1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript">
    Dim arr, i, n, loc
    n = 30
    loc = -1
    arr = Array(10,20,30,40)
    for i = 0 to 3
      if n = arr(i) then
            loc = i
            exit for
      end if
    next
    if loc = -1 then
      Document.Write ("Number not found.")
    else
      Document.Write("Number found at element: " & loc)
    end if
</SCRIPT>
```

2. Save the file as "Vb74.html".
3. Double click the file.

Exercise Questions

1. Write VBScript code that inputs five numbers from the user in array and displays the maximum number.
2. Write VBScript code that inputs ten numbers from the user in array and displays all even numbers in the array.
3. Write VBScript code that inputs the age of 20 persons and counts the number of persons in the age group of 50 to 60.

WWW.ITECVIEW.COM©

Events and String Handling

14.1 Document Object Model

An HTML page is a collection of different **objects** such as form, textbox and button etc. A web browser works with different objects while displaying web pages. These objects have different properties and functions. VBScript-enabled browsers use **Document Object Model (DOM)** to recognize these objects. The HTML objects have a descending relationship with each other. The topmost object in DOM is called **Navigator**. It refers to the browsers itself. Different other objects in DOM hierarchy are as follows:

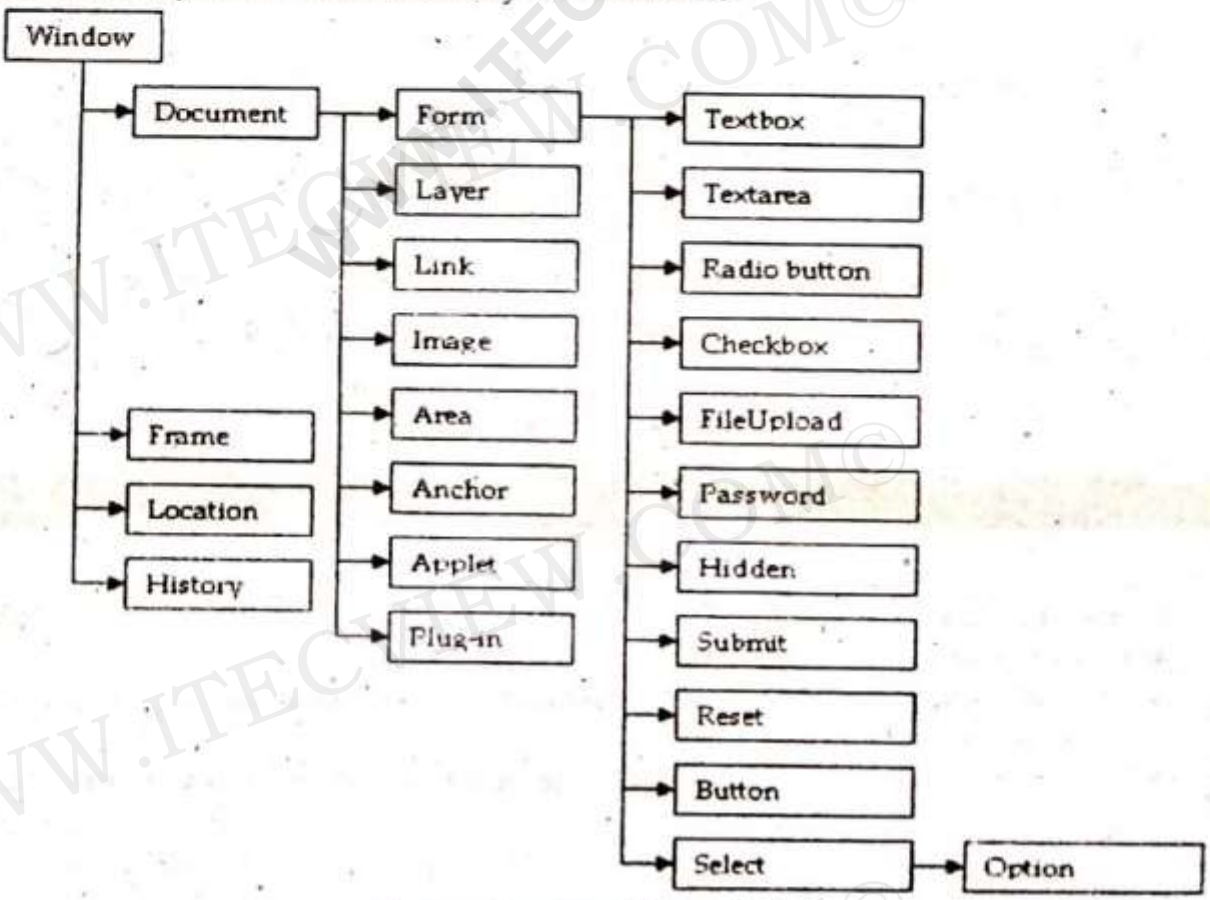


Figure: Document Object Model

14.2 Objects

An object is a set of **properties** and **methods**. A web page consists of many objects. HTML is used to created different objects such as textbox, radio buttons and list boxes etc.

Properties of Objects

An object may have different properties associated with it. The properties can be accessed as follows:

```
objectName.propertyName
```

Both the object name and property name are case sensitive. Suppose we have an object **Student** with some properties such as **Name, Marks** and **Gender**. These properties can be accessed as follows:

```
Student.Name = "Usman"
Student.Marks = 100
Student.Gender = "Male"
```

Methods of Objects

Methods are used to perform actions. Objects have different types of methods. Some methods are used to set or get the values of different properties of the objects. These methods can be accessed as follows:

```
objectName.methodName
```

Both the object name and method name are case sensitive. Suppose we have an object **Student** with some methods such as **show** and **close**. These methods can be used as follows:

```
Student.show()
Student.close()
```

14.2.1 Browser Objects

Different browser objects are as follows:

Object	Use
Navigator	It is used to access information about the browser.
Window	It is used to access a browser window or frame.
Document	It is used to access the document loaded in a window.
Location	It refers to the URL the browser is visiting.
History	It is used to maintain the history of URLs accessed in the browser.
Screen	It is used to access the information about the computer screen.

14.2.2 Arrays of Objects

An HTML document consists of many objects. It may contain many objects of same type. For example, a document may have many images and many hyperlinks etc. The browser creates array to maintain the list of each type of objects. For example, if there are ten images in the document, an array called **Images** will be used to refer to these images. Each image will be referred by different index of the **Images** array. **Images(0)** will refer to the first image in the document, **Images(1)** will refer to the second image and so on.

Different arrays for maintaining different type of objects are as follows:

Arrays	Use
Images	It is used to access different images in the document.
Links	It is used to access different links in the document.

Area	It is used to access different areas in the document.
Frames	It is used to access different frames in the document.
Anchors	It is used to access different anchors in the document.
Applets	It is used to access different applets in the document.
Forms	It is used to access different forms in the document.

14.3 Form Elements

A form consists of different types of form objects. An array of objects is maintained for each type of form element. Different arrays for form elements are as follows:

Arrays	Use
Elements	It is used to access all elements in a form.
Text	It is used to access all textboxes in a form.
Textarea	It is used to access all text areas in a form.
Radio	It is used to access all radio buttons in a form.
Checkbox	It is used to access all checkboxes in a form.
Button	It is used to access all buttons in a form.
Submit	It is used to access all submit buttons in a form.
Reset	It is used to access all reset buttons in a form.
Select	It is used to access all select lists in a form.
Option	It is used to access all options of select list in a form.
Password	It is used to access all password fields in a form.
Hidden	It is used to access all hidden fields in a form.
Fileupload	It is used to access all file upload element in a form.

14.4 Events

An event occurs when something happens or an action is performed. Events are used to write VBScript code that reacts to certain situations. Some events are as follows:

- The user clicking the mouse button
- The Web page loading
- A form field being changed

14.4.1 Event Handlers

An **event handler** is a piece of code that is executed when an event occurs. Each even handler is associated with an event. All event handlers in VBScript start with the word **on.**

Some important types of event handlers are as follows:

Event Handler	Applies To:	Triggered When:
onAbort	Image	The image loading is cancelled.
onBlur	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object loses focus.
onChange	FileUpload, Select, Text, TextArea	The data in the form element is changed by the user.
onClick	Button, Document, Checkbox, Link, Radio, Reset, Submit	The object is clicked on.
onDblClick	Document, Link	The object is double-clicked on.

onDragDrop	Window	An icon is dragged and dropped into the browser.
onError	Image, Window	A VBScript error occurs.
onFocus	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object gains focus.
onKeyDown	Document, Image, Link, TextArea	The user presses a key.
onKeyPress	Document, Image, Link, TextArea	The user presses a key.
onKeyUp	Document, Image, Link, TextArea	The user releases a key.
onLoad	Image, Window	The page has finished loading.
onMouseDown	Button, Document, Link	The user presses a mouse button.
onMouseMove	None	The user moves the mouse.
onMouseOut	Image, Link	The user moves the mouse away from the object.
onMouseOver	Image, Link	The user moves the mouse over the object.
onMouseUp	Button, Document, Link	The user releases a mouse button.
onMove	Window	The user moves the browser window or frame.
onReset	Form	The user clicks the form's Reset button.
onResize	Window	The user resizes the browser window or frame.
onSelect	Text, Textarea	The user selects text within the field.
onSubmit	Form	The user clicks the form's Submit button.
onUnload	Window	The user closes the page.

14.5 Forms and Events

Forms are used in web pages to get input from the user. The HTML form can also submit this data to the web server. The web server may process or store the data for future use. The data provided by the user is important and it must be validated before it is stored or processed. Forms use different functions to validate the data when the user clicks on **submit** button of the form. Different events can also be used to process and validate data in forms.

FORM object of HTML is used to create forms. A form consists of different elements. Each element has its own properties, methods and events handlers. Different elements of a form are as follows:

14.5.1 Button Element

Button is used in forms to process the form data. <BUTTON> tag is used to create a button in the form.

Methods

Some important methods of button element are as follows:

Method	Description
click()	It is used to press the button.

Events

Some important events of button element are as follows:

Event	Description
click()	It occurs when button element is clicked.

Event Handlers

Some important event handlers of button element are as follows:

Method	Description
onClick()	It is executed when button element is clicked.

Practice 14.1

Write VBScript code that displays a button on the page. A message should appear when the user click the button.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
function msg()
  alert("Event handling with buttons!")
end function
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
<INPUT TYPE="button" VALUE="Click Here for Message" onClick="msg()">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb75.html".
3. Double click the file. The page will appear with a button.
4. Click on the button. The following message will appear on the screen:

14.5.2 Text Element

Text fields are used to get input from the user in the form of letters, numbers etc.

Methods

Some important methods of text element are as follows:

Method	Description
focus()	It is used to bring the focus to the text element.
blur()	It is used to bring the focus away from text element.
select()	It is used to select the contents of the text element.

Events

Some important events of text element are as follows:

Event	Description
focus()	It occurs when text element gains focus.
blur()	It occurs when text element loses focus.
select()	It occurs when the user selects the contents of text element.

Event Handlers

Some important event handlers of text element are as follows:

Method	Description
onFocus()	It is executed when text element gains focus.
onBlur()	It is executed when text element loses focus.
onSelect()	It is executed when the user selects the contents of text element.
onChange()	It is executed when the user changes the contents of text element.

Practice 14.2

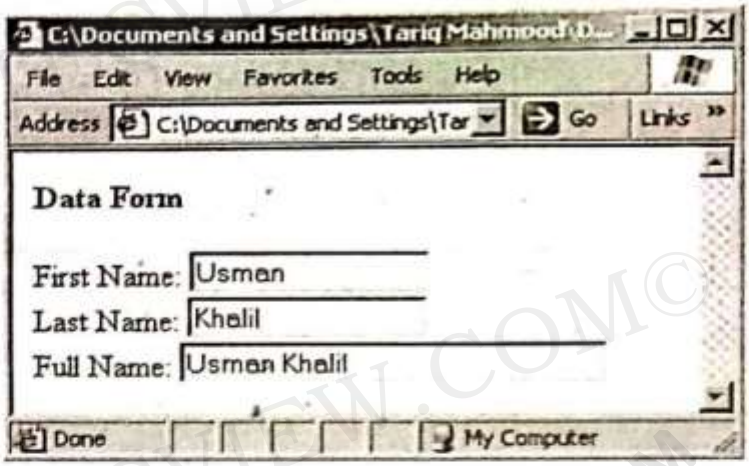
Write VBScript code that displays three text boxes. If the user changes the contents of first two text boxes, the script combines the contents of first two text boxes and displays them in third text box.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
sub full()
  frm.txtFullName.value = frm.txtFirstName.value & " " & frm.txtLastName.value
end sub
</SCRIPT>
</HEAD>
<BODY>
<H4>Data Form</H4>
<FORM NAME="frm">
First Name:
<INPUT TYPE="text" NAME="txtFirstName" SIZE="15" onChange="full()">
<BR>Last Name:
<INPUT TYPE="text" NAME="txtLastName" SIZE="15" onChange="full()">
<BR>Full Name:
<INPUT TYPE="text" NAME="txtFullName" SIZE="30">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb76.html".

3. Double click the file. The web page will appear:
4. Enter first name and press TAB key.
5. Enter second name and press TAB key. Both first and second name will appear in the third text box as follows:



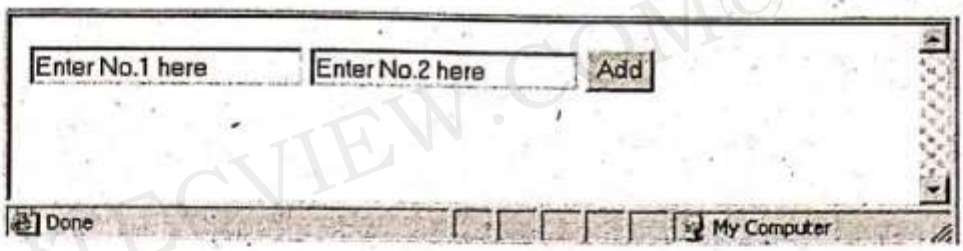
Practice 14.3

Write VBScript code that displays two text boxes and a button. The default value of first text box should be "Enter first No. here" and default value of second text box should be "Enter second No. here". The script should add these numbers and display the result in a message box when the user clicks the button.

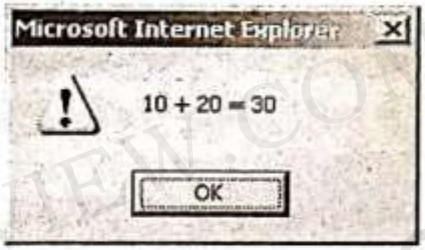
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
sub load()
  document.F.No1.value = "Enter No.1 here"
  document.F.No2.value = "Enter No.2 here"
end sub
sub Sum()
  dim a, b, c
  a = CInt(document.F.No1.value)
  b = CInt(document.F.No2.value)
  c = a + b
  msgbox(a & " + " & b & " = " & c)
end sub
</SCRIPT>
</HEAD>
<BODY onLoad="load()">
<FORM NAME="F">
<INPUT TYPE="text" NAME="No1" onFocus="document.F.No1.value="">
<INPUT TYPE="text" NAME="No2" onFocus="document.F.No2.value="">
<INPUT TYPE="button" VALUE="Add" NAME="Add" onClick="Sum()">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb77.html".
3. Double click the file. The web page will appear as follows:



4. Click in first textbox. It will become blank.
5. Enter 10 in first textbox.
6. Click in second textbox. It will become blank.
7. Enter 20 in second textbox and click **Add** button. The following message will appear.



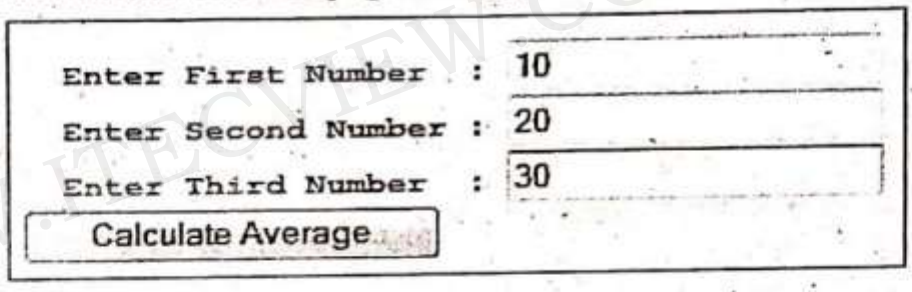
Practice 14.4

Write VBScript code that inputs three numbers in text boxes. It displays the average in a message box using function when the user clicks a button.

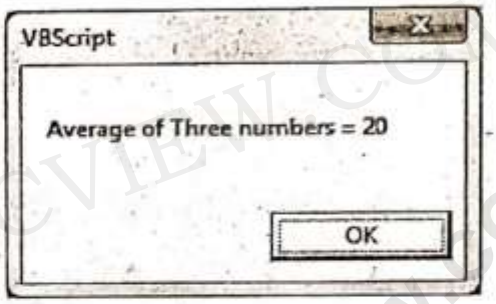
1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Average </TITLE>
 <SCRIPT LANGUAGE="VBSCRIPT">
  Option Explicit
  Function Find_avg(v1,v2,v3)
  Find_avg = (v1 + v2 + v3)/3.0
  End Function
  Sub cmdAvg_OnClick()
   Dim num1, num2, num3
   num1 = CInt(txtV1.Value)
   num2 = CInt(txtV2.Value)
   num3 = CInt(txtV3.Value)
   Dim Avg
   Avg = Find_avg(num1,num2,num3)
   MsgBox("Average of Three numbers = " & Avg)
  End Sub
 </SCRIPT>
 </HEAD>
 <BODY>
  <PRE>
   Enter First Number  : <INPUT TYPE=TEXT NAME=txtV1>
   Enter Second Number : <INPUT TYPE=TEXT NAME=txtV2>
   Enter Third Number  : <INPUT TYPE=TEXT NAME=txtV3>
  <INPUT TYPE=Button NAME=cmdAvg VALUE="Calculate Average">
  </PRE>
 </BODY>
</HTML>
```

2. Save the file as "vb78.html".
3. Double click the file. The web page will appear as follows:



4. Enter values in the textboxes.
5. Click **Calculate Average** button. The following message will appear.



Practice 14.5

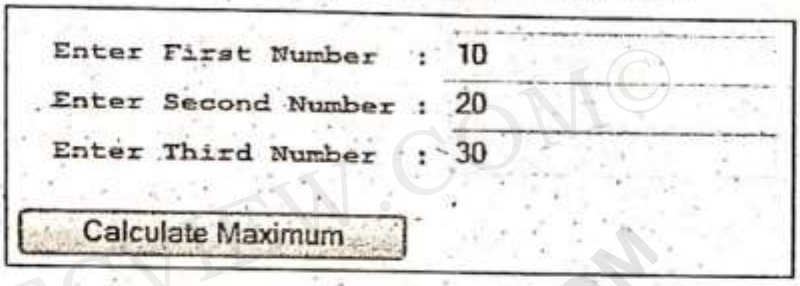
Write VBScript code that inputs three numbers in text boxes. It displays the maximum number in a message box using function when the user clicks a button.

1. Open Notepad and type the following code:

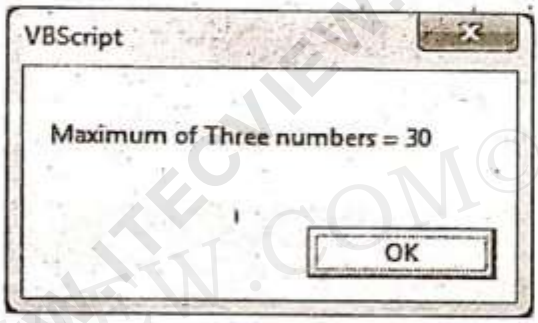
```
<HTML>
 <HEAD>
  <TITLE> Maximum </TITLE>
 <SCRIPT LANGUAGE="VBSCRIPT">
  Option Explicit
  Function Find_max(v1,v2,v3)
   dim max
   max = v1
   if v2 > max then max = v2
   if v3 > max then max = v3
   Find_max = max
  End Function

  Sub cmdMax_OnClick()
   Dim num1, num2, num3
   num1 = CInt(txtV1.Value)
   num2 = CInt(txtV2.Value)
   num3 = CInt(txtV3.Value)
   Dim Max
   Max = Find_max(num1,num2,num3)
   MsgBox("Maximum of Three numbers = " & Max)
  End Sub
  </SCRIPT>
 </HEAD>
 <BODY>
```

```
 <PRE>
  Enter First Number  : <INPUT TYPE=TEXT NAME=txtV1>
  Enter Second Number : <INPUT TYPE=TEXT NAME=txtV2>
  Enter Third Number  : <INPUT TYPE=TEXT NAME=txtV3>
 <INPUT TYPE=Button NAME=cmdMax VALUE="Calculate Maximum">
 </PRE>
</BODY>
</HTML>
```

2. Save the file as "vb79.html".
3. Double click the file. The web page will appear as follows:



4. Enter values in the textboxes.
5. Click **Calculate Average** button. The following message will appear.



14.5.3 Password Element

The methods, events and event handlers of password element are same as text element.

Practice 14.6

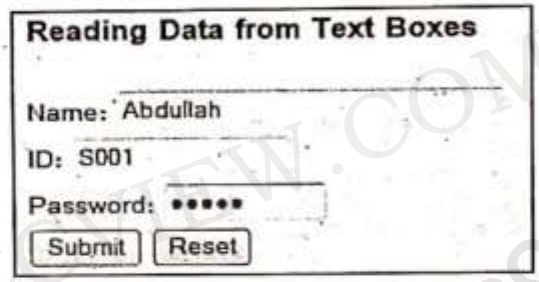
Write VBScript code that inputs name, ID and password in three text boxes. It displays the enter data in a message box when the user clicks a button.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Reading Data from Text Boxes </TITLE>
<SCRIPT LANGUAGE="VBSCRIPT">
 Option Explicit
 Sub cmdEnter_OnClick()
  MsgBox("Name = " & frmData.txtName.Value & vbCrLf & "ID = " & frmData.txtID.Value &
vbCrLf & "Password = " & frmData.txtPassword.Value)
 End Sub
 </SCRIPT>
</HEAD>
 <BODY>
  <H3 ALIGN="center">Reading Data from Text Boxes </H3>
```

```
<FORM NAME=frmData>
   Name: <INPUT TYPE="text" NAME="txtName" SIZE=30 MAXLENGTH=28><BR>
   ID:  <INPUT TYPE="text" NAME="txtID" SIZE=15 MAXLENGTH=12><BR>
   Password:  <INPUT TYPE="password" NAME="txtPassword" SIZE=10><BR>
   <INPUT TYPE="button" VALUE="Submit" NAME="cmdEnter">
   <INPUT TYPE="reset" NAME="cmdReset" VALUE="Reset">
 </FORM>
</BODY>
<HTML>
```

2. Save the file as "vb80.html".
3. Double click the file. The web page will appear as follows:



4. Enter values in the textboxes.
5. Click **Submit** button. The following message will appear.

14.5.4 Checkbox Element

Checkboxes are used to provide many options to the user. The user can choose one or multiple options from available choices. The methods, events and event handlers of checkbox element are same as button element.

Practice 14.7

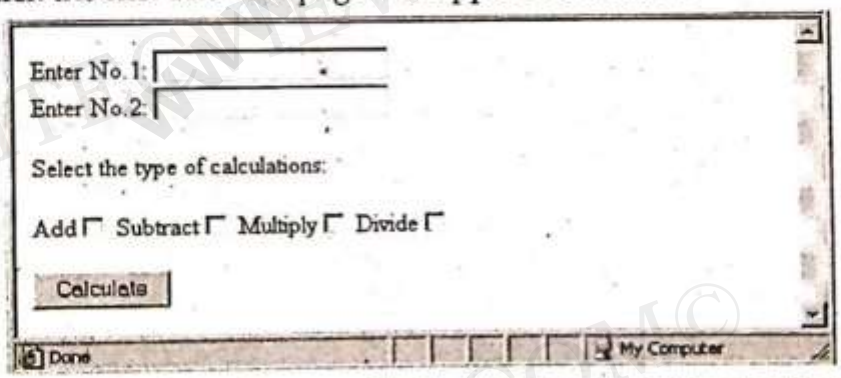
Write VBScript code that displays two text boxes to input two numbers, four checkboxes to provide options for arithmatic operations and a button. The user enters two numbers and selects checkboxes to perform arithmatic operations.

1. Open Notepad and type the following code:

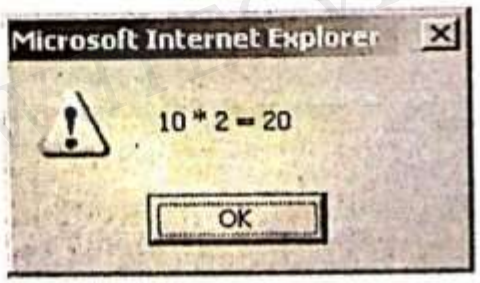
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
sub calc()
  dim a, b, c
  a = CInt(document.F.No1.value)
  b = CInt(document.F.No2.value)
```

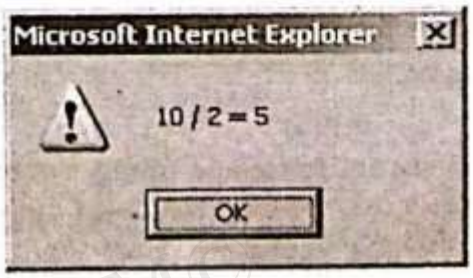
```
  if(document.F.Add.checked) then
    alert(a & " + " & b & " = " & (a+b))
  end if
  if(document.F.Sub.checked) then
    alert(a & " - " & b & " = " & (a-b))
  end if
  if(document.F.Mul.checked) then
    alert(a & " * " & b & " = " & (a*b))
  end if
  if(document.F.Div.checked) then
    alert(a & " / " & b & " = " & (a/b))
  end if
end sub
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
Enter No.1: <INPUT TYPE="text" NAME="No1"><BR>
Enter No.2: <INPUT TYPE="text" NAME="No2"><BR>
<BR>Select the type of calculations: <P>
Add<INPUT TYPE="checkbox" NAME="Add">
Subtract<INPUT TYPE="checkbox" NAME="Sub">
Multiply<INPUT TYPE="checkbox" NAME="Mul">
Divide<INPUT TYPE="checkbox" NAME="Div"><P>
<INPUT TYPE="button" NAME="Cal" VALUE="Calculate" onClick="calc()">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb81.html".
3. Double click the file. The web page will appear as follows:



4. Enter two numbers in textboxes and select **Add** and **Divide** checkboxes.
5. Click **Calculate** button. The result will appear in messages boxes according to the selection as follows:





14.5.5 Radio Element

Radio buttons are used when the user has to choose one option from multiple choices. The methods, events and event handlers of radio element are same as that of button element.

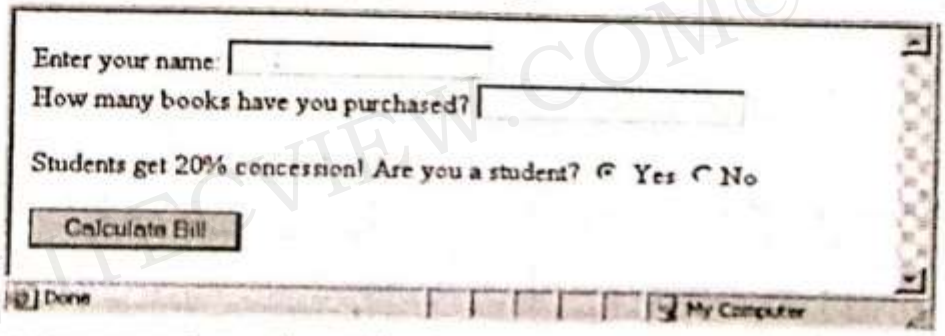
Practice 14.8

Write VBScript code that inputs name and number of books in two text boxes. It displays two radio buttons to select whether the user is a student or not. It calculates the bill of books when the user clicks a button. If the user is a student, he gets 20% concession.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
sub Bill()
  dim n, books, bill
  n = document.F.Name.value
  books = CInt(document.F.Books.value)
  if(document.F.std(0).checked) then
    bill = books * 100 - ((books * 100) * 20 / 100)
  elseif(document.F.std(1).checked) then
    bill = books * 100
  end if
  alert("Hi " & n & "! Your bill is " & bill)
end sub
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
Enter your name: <INPUT TYPE="text" NAME="Name"><BR>
How many books have you purchased? <INPUT TYPE="text" NAME="Books"><P>
Students get 20% concession! Are you a student?
<INPUT TYPE="radio" NAME="std" VALUE=1 CHECKED> Yes
<INPUT TYPE="radio" NAME="std" VALUE=0>No <P>
<INPUT TYPE="button" NAME="Cal" VALUE="Calculate Bill" onClick="Bill()">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb82.html".
3. Double click the file. The web page will appear as follows:



4. Enter your name and number of books.
5. Select the appropriate radio button and click **Calculate Bill** button. The following message will appear.

14.5.6 TextArea Element

Text area is similar to text box but it may consist of multiple rows. It is used to get lengthy input from the user like comments and suggestions etc. <TEXTAREA> tag is used to create text area in a form. The methods, events and event handlers of TextArea element are same as text element.

14.5.7 Select and Option Elements

Select field is used to provide predefined list of items to the user. The user can choose one or multiple options from the list depending upon the type of list. <SELECT> tag is used to create a list. <OPTION> tag is used to create an item in the list.

Methods

Some important methods of Select element are as follows:

Method	Description
blur()	It is used to bring focus away from select element.
focus()	It is used to bring focus to the select element.
change()	It is used to change the option of select element.

Events

Some important events of Select element are as follows:

Event	Description
blur()	It occurs when Select element loses focus.
focus()	It occurs when Select element gains focus.
change()	It occurs when the value of Select element is changed.

Event Handlers

Some important event handlers of button element are as follows:

Event	Description
onBlur()	It is executed when Select element loses focus.
onFocus()	It is executed when Select element gains focus.
onChange()	It is executed when the value of Select element is changed.

Practice 14.9

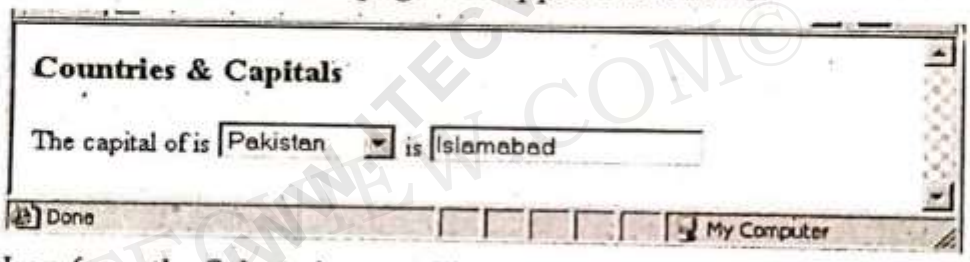
Write VBScript code that displays a list with names of different countries. When the user selects a country from the list, the script displays the capital of that country in a text box.

1. Open Notepad and type the following code:

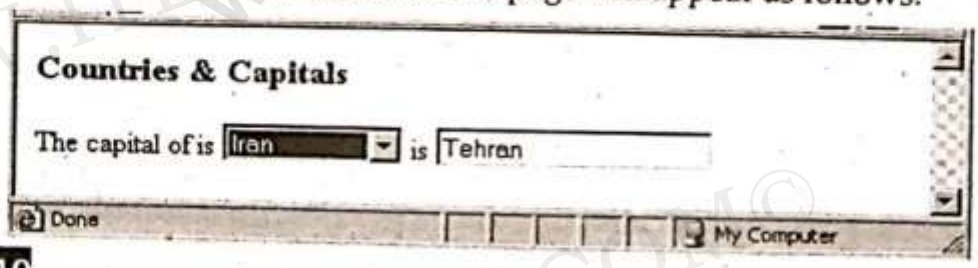
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
```

```
sub capital()
  dim i, c
  c = Array("Islamabad","Tehran","Kabul","Bejing","New Delhi")
  i = document.F.Country.selectedIndex
  document.F.Cap.value = c(i)
end sub
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="F">
<H3>Countries & Capitals</H3><P>
The capital of is
<SELECT NAME="Country" onChange="capital()">
<OPTION SELECTED>Pakistan
<OPTION>Iran
<OPTION>Afghanistan
<OPTION>China
<OPTION>India
</SELECT>
is <INPUT TYPE="text" NAME="Cap" VALUE="Islamabad"><BR>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "vb83.html".
3. Double click the file. The web page will appear as follows:



4. Select Iran from the Select element. The page will appear as follows:



Practice 14.10

Write VBScript function CircleArea that prompts the user for the radius of circle. It then calculates and prints the area of circle.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD> <TITLE> Area of circle </TITLE>
 <SCRIPT LANGUAGE="VBSCRIPT">
 Function Area()
  dim r, a
  r = inputbox("Enter radius")
  a = 3.14 * r * r
```

```
  msgbox("Area of circle = " & a)
 End Function
 </SCRIPT>
 </HEAD>
 <BODY>
  <INPUT TYPE=Button NAME=Ar VALUE="Calculate Area" onClick="Area()">
  </PRE>
 </BODY>
</HTML>
```

2. Save the file as "vb84.html".
3. Double click the file. The web page will appear in the browser with a button.
4. Click **Calculate Area** button. An input box will appear.
5. Enter the radius and click **OK** button. The result will appear in message box.

Practice 14.11

Write VBScript code that displays the multiple of integer 2 in the browser window. The loop should terminate on 1000.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> Multiple of 2 </TITLE>
 <SCRIPT LANGUAGE="VBSCRIPT">
 Sub ShowNum()
   dim n, c
   n = 1
   while (c<1000)
       c = n * 2
       document.write (c & " ")
       n = n + 1
   wend
  End Sub
 </SCRIPT>
 </HEAD>
 <BODY>
  <INPUT TYPE=Button NAME=Multiple VALUE="Show Numbers" onClick="ShowNum()">
  </PRE>
 </BODY></HTML>
```

2. Save the file as "vb85.html".
3. Double click the file. The web page will appear in the browser with a button.
4. Click **Show Number** button. The result will appear in the browser window.

14.6 String Handling

Strings are used frequently in web pages. VBScript provides many built-in functions to apply different function on strings. Some important string handling functions are as follows:

UCase

UCase function is used to convert a string to upper case letters.

```
Name1 = "Usman Khalil"
Name2 = UCase(Name1)
```

The value of Name2 variable will be "USMAN KHALIL".

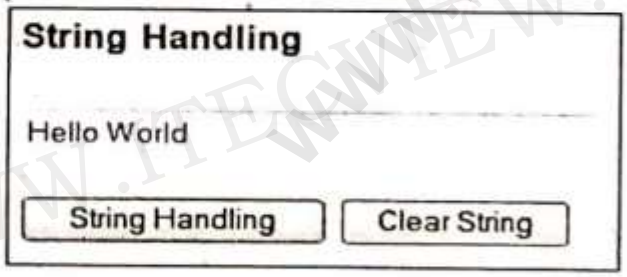
Practice 14.12

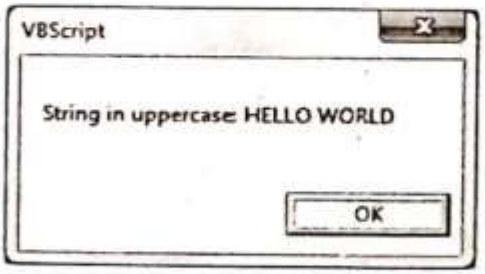
Write VBScript code that inputs a string in textobx and displays the string in uppercase in mesage box.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
sub process()
  dim s
  s = document.strings.s1.value
  msgbox("String in uppercase: " & UCase(s))
end sub
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
   <INPUT NAME="s1" VALUE="" SIZE=38 MAXLENGTH=25><P>
   <INPUT TYPE="button" VALUE="String Handling" onClick="process()">
   <INPUT TYPE="reset" VALUE="Clear String">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "UCase.html".
3. Double click the file. The web page will appear in the browser.
4. Enter any string in textbox and click **String Handling** button. The result will appear in the message box as follows:





LCase

LCase function is used to convert a string to lower case letters.

```
Name1 = "Usman Khalil"
Name2 = LCase(Name1)
```

The value of Name2 variable will be "usman khalil".

Practice 14.13

Write VBScript code that inputs a string in textobx and displays the string in lowercase in mesage box.

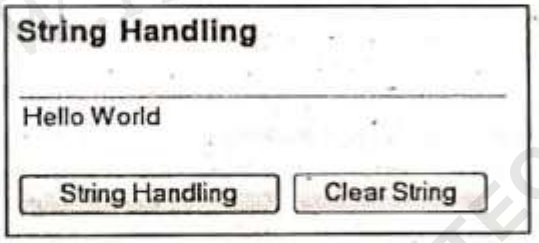
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
```

WWW.ITECVIEW.COM©

```
sub process()
  dim s
  s = document.strings.s1.value
  msgbox("String in lowercase: " & LCase(s))
end sub
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
  <INPUT NAME="s1" VALUE="" SIZE=38 MAXLENGTH=25><P>
  <INPUT TYPE="button" VALUE="String Handling" onClick="process()">
  <INPUT TYPE="reset" VALUE="Clear String">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "LCase.html".
3. Double click the file. The web page will appear in the browser.
4. Enter any string in textbox and click **String Handling** button. The result will appear in the message box as follows:



LTrim

LTrim function is used to remove any space from the left side of a string.

```
Name1 =  "   Usman Khalil"
Name2 = LTrim(Name1)
```

The value of Name2 variable will be "Usman Khalil".

RTrim

RTrim function is used to remove any space from the right side of a string.

```
Name1 = "Usman Khalil   "
Name2 = RTrim(Name1)
```

The value of Name2 variable will be "Usman Khalil".

Trim

Trim function is used to remove any space from both sides of a string.

```
Name1 =  "   Usman Khalil   "
Name2 = Trim(Name1)
```

The value of Name2 variable will be "Usman Khalil".

String

String function is used to display or return specified number of character.

```
MsgBox String(10, "+")
```

The above line will display 10 "+" symbols in a message box.

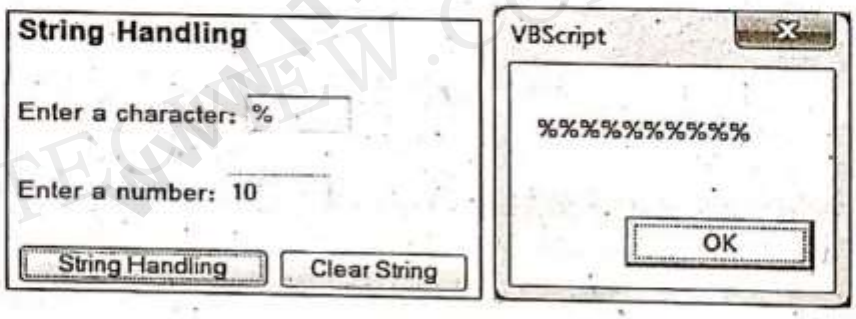
Practice 14.14

Write VBScript code that inputs a character and a number in textobxes. It displays the given character **n** number of times in mesage box where **n** is the number entered by the user.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD><TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
sub process()
  dim s, n
  s.= document.strings.t1.value
  n = document.strings.t2.value
  msgbox(String(n, s))
end sub
</SCRIPT></HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
   Enter a character: <INPUT NAME="t1" VALUE="" SIZE=5 MAXLENGTH=1><P>
   Enter a number: <INPUT NAME="t2" VALUE="" SIZE=5 MAXLENGTH=2><P>
   <INPUT TYPE="button" VALUE="String Handling" onClick="process()">
   <INPUT TYPE="reset" VALUE="Clear String">
</FORM></BODY></HTML>
```

2. Save the file as "String.html".
3. Double click the file. The web page will appear in the browser.
4. Enter any string in textbox and click **String Handling** button. The result will appear in the message box as follows:



Space

Space function is used to display or return specified number of spaces.

```
MsgBox "Hello" & Space(10) & "World"
```

The above line will display "Hello" then 10 spaces and then "World".

StrComp

StrComp function is used to compare two string values and returns an integer. The possible returned values are as follows:

Returned value	Description
1	S1 > S2
0	S1 = S2
-1	S1 < S2

S1 = "Usman"
S2 = "Khalil"

```
MsgBox StrComp(S1, S2)
```

The above line will display 1 because "Usman" is greater than "Khalil". The letters that come first in alphabetic are smaller than the letters that come later in alphabetic. For example "A" is smaller than "B". But capital letters are always greater than lower case letter. For example "A" is greater than "b".

Practice 14.15

Write VBScript code that inputs two strings in textobxes. It compares the two strings and displays the result in mesage box.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
sub process()
  dim a, b
  a = document.strings.s1.value
  b = document.strings.s2.value
  if StrComp(a, b) = 1 then
        msgbox(a & " is greater than " & b)
  elseif StrComp(a, b) = -1 then
        msgbox(b & " is greater than " & a)
  else
        msgbox(a & " and " & b & " are equal")
  end if
end sub
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
   Enter a string: <INPUT NAME="s1" VALUE="" SIZE=25 MAXLENGTH=25><P>
   Enter a string: <INPUT NAME="s2" VALUE="" SIZE=25 MAXLENGTH=25><P>
   <INPUT TYPE="button" VALUE="Compare" onClick="process()">
   <INPUT TYPE="reset" VALUE="Clear">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Compare.html".
3. Double click the file. The web page will appear in the browser.
4. Enter any string in textbox and click **Compare** button. The result will appear in the message box as follows:

Len

Len function is used to determine the total length of a string.

```
S = "Hello World"
MsgBox Len(S)
```

The above line will display 11 in a message box.

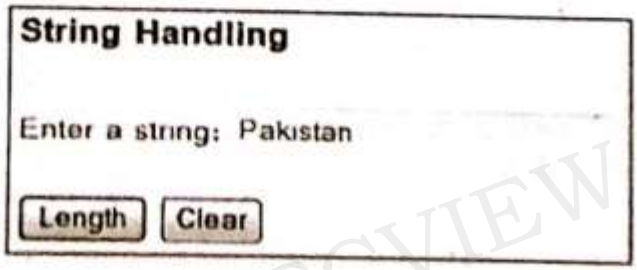
Practice 14.16

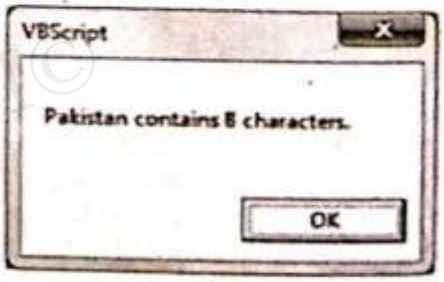
Write VBScript code that inputs a string in textobx and displays its length in a mesage box.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
sub process()
  dim s, n
  s = document.strings.s1.value
  n = Len(s)
  msgbox(s & " contains " & n & " characters.")
end sub
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
   Enter a string: <INPUT NAME="s1" VALUE="" SIZE=25 MAXLENGTH=25><P>
   <INPUT TYPE="button" VALUE="Length" onClick="process()">
   <INPUT TYPE="reset" VALUE="Clear">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Length.html".
3. Double click the file. The web page will appear in the browser.
4. Enter any string in textbox and click **Length** button. The result will appear in the message box as follows:





Left

Left function is used to extract specified number of characters from the left side of a string.

```
S = "Hello World"
MsgBox Left(S, 2)
```

The above line will display "He" in a message box.

Right

Right function is used to extract specified number of characters from the right side of a string.

```
S = "Hello World"
MsgBox Right(S, 3)
```

The above line will display "rld" in a message box.

Mid

Mid function is used to extract specified number of characters from any part of string.

```
S = "Hello World"
MsgBox Mid(S, 7, 5)
```

The above line will display "World" in a message box.

InStr

InStr function is used to search a string for a particular character(s). It returns the position of first occurrence of given character in string. It returns 0 if character is not found.

```
S = "Hello World"
MsgBox InStr(S, "e")
```

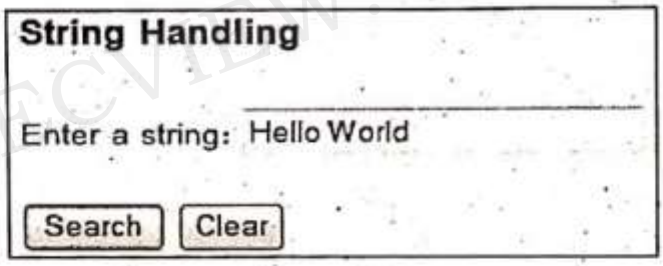
Practice 14.17

Write VBScript code that inputs a string in a textobx. It inputs a character or string in an input box and searches it in the string in the textbox. It displays the result of searching in a mesage box.

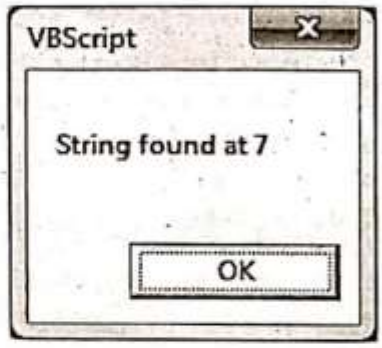
1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>String Handling</TITLE>
<SCRIPT LANGUAGE="VBScript">
sub process()
  dim s, c, n
  s = document.strings.s1.value
  c = inputbox("Enter a string to search: ")
  n = InStr(s, c)
  if n = 0 then
        msgbox("String not found.")
  else
        msgbox("String found at " & n)
  end if
end sub
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<H3>String Handling</H3>
<FORM NAME="strings">
   Enter a string: <INPUT NAME="s1" VALUE="" SIZE=25 MAXLENGTH=25><P>
   <INPUT TYPE="button" VALUE="Search" onClick="process()">
   <INPUT TYPE="reset" VALUE="Clear">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Search.html".
3. Double click the file. The web page will appear in the browser.



4. Enter any string in textbox and click **Search** button. An input box will appear.
5. Enter the required string or character in the input box and click OK. The result will appear in the message box as follows:



Practice 14.18

Write VBScript code that inputs full name in input box and displays the first name and last name separately in a mesage box.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD>
  <TITLE> String Handling </TITLE>
 <SCRIPT LANGUAGE="VBScript">
  Sub cmdEnter_OnClick()
    Dim s1, strfirst, strlast, strmiddle, posBlank, count
    s1 = InputBox("Enter your Full Name")
    posBlank = InStr(1,s1," ")
    count = Len(s1)-posBlank
    MsgBox("First Name = " & Left(s1,posBlank) & vbCrLf & _
           "Last Name = " & Right(s1,count))
  End Sub
</SCRIPT>
</HEAD>
 <BODY>
  <INPUT TYPE="button" NAME=cmdEnter VALUE="Enter Full Name">
 </BODY>
</HTML>
```

2. Save the file as "names.html".
3. Double click the file. The web page will appear in the browser with a button.
4. Click the button. An input box will appear.
5. Enter the name **Usman Khalil** and click **OK**. The result will appear in message box

14.7 Form Validation

VBScript is used to validate the contents of the forms. Form validation is a process of checking that proper information is entered in the form before it is submitted. For example, a field for entering email must be checked to ensure that the user has filled it. It should also be ensure that the user has entered a valid email address. Similarly, the validity of credit card number should also be checked.

14.7.1 Common Validation Types

Some important validation types are as follows:

1. Validating Email Addresses

The email validation is used to check the presence of a dot and an "@" sign in email address. It is also important to verify that there is a dot after the "@" sign.

2. Checking for Numeric Values

Many numeric values should be checked for specified format. For example, a credit card number can be checked for its specific length and format etc. Depending on the type of credit card, the number must contain either 13 or 16 digits and they must add up to a specified total. All characters must be numeric. The expiry date of credit card is also validated. Similarly, the value entered in the month field should be between 1 and 12 etc.

3. Checking for Blank Fields

A form may contain many fields. Some fields are compulsory to fill and some are optional. A validation process must check that all compulsory fields are filled by the user.

Practice 14.19

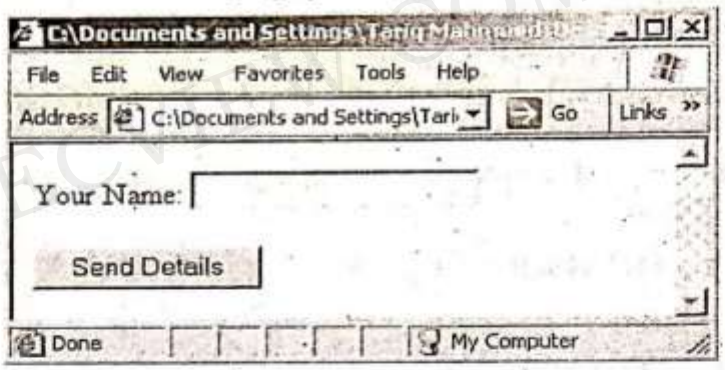
Write VBScript code that displays a text box to input name and submit button. The script uses a function to ensure that the name field is not emply.

1. Open Notepad and type the following code:

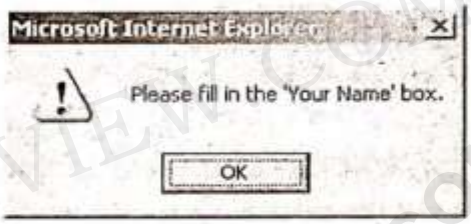
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
sub validate_form ()
  if (document.contact_form.contact_name.value = "" ) then
    alert ( "Please fill in the 'Your Name' box." )
  else
        document.contact_form.submit()
  end if
end sub
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="contact_form" METHOD="post" ACTION="mailto:info@itseries.com.pk">
<P>Your Name: <INPUT TYPE="text" NAME="contact_name"></P>
<P><INPUT TYPE="button" NAME="send" VALUE="Send Details"
onClick="validate_form()"></P>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Val01.html".

3. Double click the file. The web page will appear as follows:



4. Press **Send Details** button without entering name. A message will appear as follows:



5. Now enter your name and press the button again. The form will be submitted.

Practice 14.20

Write VBScript code that inputs name, age and email in three textboxes. It checks the following validations when the user clicks the button:

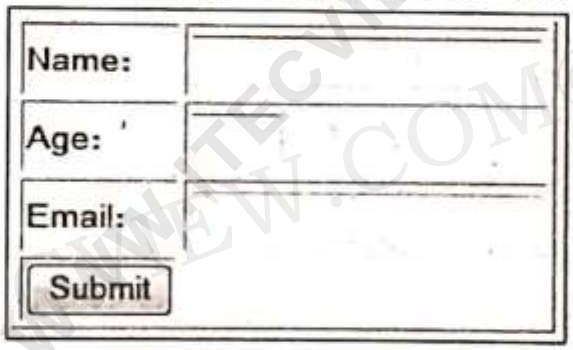
- Name cannot be blank or numeric value.
- Age cannot be blank. It must be numeric and two-digit value.
- Email cannot be blank. It must contain @ and dot character.

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "VBScript">
Sub cmdUserInfo_OnClick()
if len(form.txtName.value)<2 or IsNumeric(form.txtName.value)then
  msgbox("Type a valid name")
  form.txtName.focus
  form.txtName.select
elseif len(form.txtAge.value) < 2 or Not IsNumeric(form.txtAge.value) then
  msgbox("Type a valid age")
  form.txtAge.focus
  form.txtAge.select
elseif form.txtEmail.value="" then
  msgbox("Enter a valid email")
  form.txtEmail.focus
  form.txtEmail.select
elseif instr(form.txtEmail.value,"@") = 0 or instr(form.txtEmail.value,".") = 0 then
  msgbox("Type a valid email")
  form.txtEmail.focus
else
  msgbox"Success!"
end if
end sub
```

```
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME = "form" METHOD = "post">
<TABLE BORDER="1">
<TR>
 <TD>Name:</TD><TD><INPUT TYPE = "text" NAME = "txtName"></TD>
</TR>
<TR>
 <TD>Age:</TD><TD><INPUT TYPE = "text" NAME = "txtAge" SIZE="2"></TD>
</TR>
<TR>
 <TD>Email:</TD><TD><INPUT TYPE = "text" NAME ="txtEmail"></TD>
</TR>
<TR>
 <TD><INPUT TYPE = "button" NAME = "cmdUserInfo" VALUE = "Submit"></td>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Val02.html".
3. Double click the file. The web page will appear as follows:



4. Enter the values and click **Submit** button to check the validation.

Practice 14.21

Write VBScript code that has two textboxes for email ID and password. It also contains a Register button. It displays an error message if the email's lenght is lesser than 6 and email is other than hotmail.com

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "VBScript">
Sub Check()
 Dim e
 e = document.form.txtEmail.value
 if (Len(e) < 6) then
        msgbox("Invlid email length.")
 elseif (Instr(e, "hotmail")) = 0 then
        msgbox("Email must be hotmail")
 else
```

```
        msgbox("Valid input.")
 end if
end sub
</SCRIPT>
</head>
<BODY>
<FORM NAME = "form" METHOD = "post">
<TABLE BORDER="1">
<TR>
 <TD>Email ID:</TD><TD><INPUT TYPE = "text" NAME = "txtEmail"></TD>
</TR>
<TR>
 <TD>Password:</TD><TD><INPUT TYPE = "password" NAME = "txtPass"></TD>
</TR>
<TR>
<TD><INPUT TYPE="button" NAME="cmdInfo" VALUE="Register" onClick="Check()"></td>
</TR>
</TABLE></FORM>
</BODY>
</HTML>
```

2. Save the file as "Val03.html".
3. Double click the file. The web page will appear as follows:

4. Enter the values and click **Register** button to check the validation.

Practice 14.22

Write VBScript code to develop a simple calculator.

1. Open Notepad and type the following code:

```
<HTML>
 <HEAD><TITLE>VBScript Calculator </TITLE></HEAD>
 <BODY>
  <TABLE><COL SPAN = "4" ALIGN = "center" VALIGN = "middle">
   <TR><TD COLSPAN="4"> <INPUT TYPE="text" VALUE="0" NAME = "t"></TD></TR>
   <TR>
    <TD><INPUT TYPE = "button" VALUE = " 7 " NAME = "b7"></TD>
    <TD><INPUT TYPE = "button" VALUE = " 8 " NAME = "b8"></TD>
    <TD><INPUT TYPE = "button" VALUE = " 9 " NAME = "b9"></TD>
    <TD><INPUT TYPE = "button" VALUE = " / " NAME = "div"></TD>
   </TR>
   <TR>
    <TD><INPUT TYPE = "button" VALUE = " 4 " NAME = "b4"></TD>
    <TD><INPUT TYPE = "button" VALUE = " 5 " NAME = "b5"></TD>
    <TD><INPUT TYPE = "button" VALUE = " 6 " NAME = "b6"></TD>
    <TD><INPUT TYPE = "button" VALUE = " x " NAME = "mult"></TD>
   </TR>
```

```
<TR>
  <TD><INPUT TYPE = "button" VALUE = "  1  " NAME = "b1"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  2  " NAME = "b2"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  3  " NAME = "b3"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  -  " NAME = "subt"></TD>
</TR>
<TR>
  <TD><INPUT TYPE = "button" VALUE = "  0  " NAME = "b0"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  .  " NAME = "dot"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  =  " NAME = "eq"></TD>
  <TD><INPUT TYPE = "button" VALUE = "  +  " NAME = "add"></TD>
</TR>
<TR><TD COLSPAN="4"><INPUT TYPE="button" VALUE="Clear" NAME="clr"></TD>
</TR>
</TABLE>
<SCRIPT LANGUAGE = "VBscript">
  Dim Clr, n, op, opf
  Sub add_OnClick()
    op = "+"
    If Not opf Then
      n = CDbl( t.Value )
      t.Value = "0"
      opf = True
    End If
  End Sub

  Sub div_OnClick()
    op = "/"
  If Not opf Then
    n = CDbl( t.Value )
    t.Value = "0"
    opf = True
  End If
  End Sub

  Sub mult_OnClick()
    op = "*"
    if Not opf Then
      n = CDbl( t.Value )
      t.Value = "0"
      opf = True
    End If
  End Sub
  Sub subt_OnClick()
    op = "-"
    If Not opf Then
      n = CDbl( t.Value )
      t.Value = "0"
      opf = True
    End If
  End Sub
```

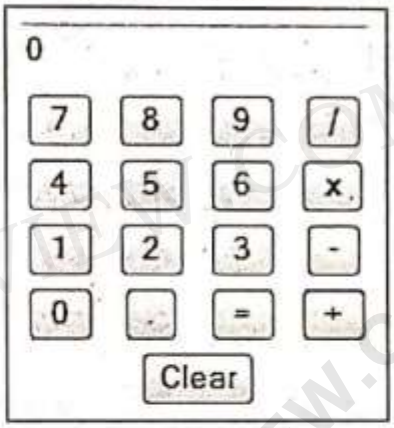
```
Sub eq_OnClick()
  Select Case op
    Case "+"
      t.Value = n + CDbl ( t.Value )
    Case "-"
      t.Value = n - CDbl ( t.Value )
    Case "*"
      t.Value = n * CDbl ( t.Value )
    Case "/"
      If t.Value <> 0 Then
        t.Value = n / CDbl ( t.Value )
      Else
        t.Value = "E"
      End If
  End Select
  Clr = True
  opf = False
  op = "0"
End Sub

Sub b0_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b0.Value )
    opf = False
  End If
End Sub

Sub b1_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b1.Value )
  Else
    t.Value = Trim( b1.Value )
  End If
  opf = False
End Sub

Sub b2_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b2.Value )
  Else
```

```
    t.Value = Trim( b2.Value )
  End If
  opf = False
End Sub

Sub b3_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b3.Value )
  Else
    t.Value = Trim( b3.Value )
  End If
  opf = False
End Sub

Sub b4_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b4.Value )
  Else
    t.Value = Trim ( b4.Value )
  End If
  opf = False
End Sub

Sub b5_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b5.Value )
  Else
    t.Value = Trim( b5.Value )
  End If
  opf = False
End Sub

Sub b6_OnClick()
  If Clr Then
    t.Value = "0"
    Clr = False
  End If
  If t.Value <> "0" Then
    t.Value = t.Value & Trim( b6.Value )
  Else
```

```
        t.Value = Trim( b6.Value )
      End If
      opf = False
    End Sub

    Sub b7_OnClick()
      If Clr Then
        t.Value = "0"
        Clr = False
      End If
      If t.Value <> "0" Then
        t.Value = t.Value & Trim( b7.Value )
      Else
        t.Value = Trim( b7.Value )
      End If
      opf = False
    End Sub

    Sub b8_OnClick()
      If Clr Then
        t.Value = "0"
        Clr = False
      End If
      If t.Value <> "0" Then
        t.Value = t.Value & Trim( b8.Value )
      Else
        t.Value = Trim ( b8.Value )
      End If
      opf = False
    End Sub

    Sub b9_OnClick()
      If Clr Then
        t.Value = "0"
        Clr = False
      End If
      If t.Value <> "0" Then
        t.Value = t.Value & Trim( b9.Value )
      Else
        t.Value = Trim ( b9.Value )
      End If
      opf = False
    End Sub

    Sub dot_OnClick()
      If Clr Then
        t.Value = "0"
      End If
      If InStr( t.Value, "." ) = 0 Then
        t.Value = t.Value & Trim( dot.Value )
        Clr = False
      End If
    End Sub
```

```
    Sub clr_OnClick()
      t.Value = "0"
    End Sub

  </SCRIPT>
 </BODY>
</HTML>
```

2. Save the file as "calculator.html".
3. Double click the file. The web page will appear as follows:



Exercise Questions

1. Write a Web page that displays four buttons to represent four colors Red, Green, Blue and White. The background color of document is changed according to the button clicked by the user.
2. Write a Web page that contains two text boxes and a button. When the user clicks the button, the contents of two text boxes should be swapped. The text boxes should appear on the same line. The text boxes and button should be in the center of page.
3. Write a Web page that displays a form to input name, email, phone and address from the user. A script should ensure that all fields are filled and the user has entered a valid email address. The length of phone number should not exceed 7 digits.

WWW.ITECVIEW.COM©

Short Questions

Q.1. What is VBScript?

VBScript is a scripting language developed by Microsoft. It is a subset of Visual Basic. It supports the syntax and structure of Visual Basic. VBScript code is embedded in a Web page. The code is translated by Web browser when the web page is viewed by the user.

Q.2. What is a variable?

A quantity whose value can be changed during execution of program is known as variable. The value of a variable is referred by variable name. Each variable represents a memory location. The data stored in the variables is automatically removed when program ends. The value of a variable can change during program execution but the name of variable cannot be changed.

Q.3. Explain the difference between a constant and a variable.

Variables and constants are temporary memory locations with a name and data type. The value stored in a variable can be changed during the execution of the project. The values stored in constants cannot change.

Q.4. What is concatenation and when would it be useful?

Concatenation is a process of combining two or more items together. The symbols + and & are used as concatenation operators.

Q.5. What value is assigned to the left-hand variables in each of the following?

A. ResultA = 1 + 2 * 3
B. ResultB = (1 + 2) * 3
C. ResultC = 2 ^ 4
D. Average = 10 + 20 + 30 / 3
E. Num = 10 - 3 * 2 + 4 / 2
F. Ans = 10 * 2 ^ 2

Answer:

A. 7
B. 9
C. 16
D. Parentheses is required around the addition to compute a true average.
E. 6
F. 40

Q.6. What is the difference between + and &?

+ is used for addition or string concatenation and & is used only for concatenation

Q.7. Explain the use of a message box?

A message box is used to display a message to the user.

Q.8. What are control statements or decision statements?

A statement or set of statements that is executed when a particular condition is true and ignored when the conditions is false is called conditional statement.

Q.9. What is a condition?

The test in an If statement is based on a condition. Six relational operators, >, <, =, <>, >= and <= are used to compare values in condition. The result of the comparison is either True or False.

Q.10. Explain the purpose of relational operators and logical operators.

The relational operators are used to compare values. Compound conditions test more than one condition. Logical operators are used to create compound conditions. Logical operators include And, Or, and Not.

Q.11. Explain the differences between a Do...Loop and a For...Next loop.

For...Next loop is used when the number of iterations are already known. The loop index determines the number of times the statements are executed. A Do...Loop terminates the loop only when a specified condition is satisfied. It is used when the exact number of iterations is unknown.

Q.12. What is the difference between Do While and Do-Loop While loop?

VBScript checks the relational test at the top of the loop using a Do While statement. Whereas it checks the relational test at the bottom of the loop using a Do-Loop While statement.

Q.13. What is the difference between a Do While and a Do Until loop?

Do While loop continues executing while a relational test is true. Do Until loop continues executing while a relational test is false.

Q.14. Can every For-Next loop be written as Do loop structure? Why or why not?

For-Next loops are used when the programmer knows the exact number of times a loop is executed. In Do loop, the programmer may or may not know the number of iterations. Therefore, every For-Next loop can be written as a Do loop.

Q.15. How many times does the following loop execute?

```
I = 10
Do While I > 1
I = I - 1
Loop
```

Answer. 9

Q.16. How many times does the following loop execute?

```
I = 10
Do While I >= 1
I = I - 1
Loop
```

Answer. 10

Q.17. What Step value is used if it is not specified in For...Next loop?

The default Step value used in For...Next loop is 1.

Q.18. Which statements terminate Do and For loops early?

Exit Do and Exit For statements are used to terminate loops before their natural termination.

Q.19. How does a programmer decide to use a While loop versus a For loop?

For loop is used when the number of iterations are known before the loop is entered. Suppose the user wants to calculate the salary of 100 employees. In this situation, For loop is the best option.

While loop is used when the number of iterations is not known at the beginning of loop. Suppose a loop inputs a number from the user and then displays its square. It then asks the user whether he wants to repeat it or not. In this situation, the iterations are decided by the user. So While loop is better.

Q.20. Name two types of subprograms.

Two types of subprograms are sub procedures and function procedures.

Q.21. What is the primary difference between sub procedure and function procedure?

A function procedure can return a value back to the calling procedure but sub procedure cannot return a value.

WWW.ITECVIEW.COM©

Multiple Choice

1. **Which operator is only used for string concatenation?**
   a. Plus sign (+). b. Underscore (_). c. "and" d. Ampersand (&).
2. **Comments are used to:**
   a. Help others read & understand program b. Make the program run faster
   c. Make program compile easier d. Increase the size of executable program
3. **Which of the following can be used to get input instead of having a text box on a form?**
   a. Format b. Inputbox c. Msgbox d. Ucase
4. **MsgBox is used to do which of the following:**
   a. Display a message b. Perform arithmetic operations
   c. Both of the above d. None of the above
5. **The MOD operator is used for:**
   a. Exponentiation b. Multiplication c. Division d. Remainder
6. **The expression 10 mod 3 has a value equal to:**
   a. 1 b. 3 c. 8 d. None
7. **What is the value of x after the following statements?**

```
Dim x, y, z
y = 10
z = 3
x = y * z + 3
```

   a. 12 b. 60 c. 30 d. 33
8. **What is the value of x after the following statements?**

```
Dim x
x = 0
x = x + 30
```

   a. 0 b. 30 c. 33 d. none
9. **What is the value of x after the following statements?**

```
Dim x
x = 15 Mod 4
```

   a. 15 b. 4 c. 3 d. 3.75
10. **All of the following are relational operators EXCEPT:**
    a. < > b. & c. = d. >=
11. **Which of the following returns True if A = 25 and B = 35:**
    a. A+B b. A>=B c. A<>B A=B
12. **Variable that holds a large group of homogeneous data is called:**
    a. Constant b. scalar value c. array d. multiple variable

Answers

1. d	2. a	3. b	4. a
5. d	6. a	7. d	8. b
9. c	10. b	11. c	12. c

WWW.ITECVIEW.COM©

Part - 6

Microsoft FrontPage

Includes:

Chapter 15 Web Designing using Microsoft FrontPage

WWW.ITECVIEW.COM©

Web Designing using Microsoft FrontPage

15.1 Microsoft FrontPage

Microsoft FrontPage is an application used as a tool for designing web pages. It was first developed by **Vermeer Technologies Incorporated (VTI)**. Later, **Microsoft** took the control of FrontPage. FrontPage is used to design web pages without remembering lot of HTML commands.

FrontPage provides many wizards and libraries of templates to create websites. It also supports **Cascading Style Sheets**, which are used to place things (images, text, etc) exactly where you want them to be on the page, using the distance in pixels from the top and the left of the browser window.

FrontPage is a best application for beginners for creating interactive small websites. It also works extremely nicely for the small business developers. FrontPage has a user-friendly environment to develop interactive and useful professional site. However, many extra-large website developers do not prefer to use Front Page because it produces very large inefficient code that does not comply with the **W3C** standards.

15.1.1 FrontPage Environment

FrontPage 2000 has a very attractive, integrated interface to design, develop and manage entire Web sites. By default, **Menu bar, Standard toolbar** and **Formatting toolbar** is displayed on the workspace, for providing easy access to most commonly used commands. To display additional toolbars: Click **View > Toolbars** and select any toolbar to display.

15.1.2 Screen Layout

The area where you will create your web pages is called **Work Space** or **Main Work Area.** Main working area has two panels:

1. Side Panel

The side panel is also called *Views* which is mainly used for Site Management and contains six different view options:

- **Page View:** It is the default view and provides a powerful environment to design and edit web pages. It displays text, images, tables etc as they will appear in browser.

- **Folders View:** It displays a list of all files and folders of your website.
- **Reports View:** It provides a site summary on the status of its files and hyperlinks. It indicates all faults with pages and links in the web such as broken hyperlinks etc.
- **Navigation View:** It provides navigation structure of website. It indicates how pages are linked together. You may change the order of pages to navigate site in a different sequence. You can create navigation bars to work on particular parts of the site.
- **Hyperlinks View:** It provides the list of all items that link together in your Website. It is used to organize the links in the web pages.
- **Tasks View:** It provides a list of all tasks needed to be completed. It also keeps some further information like their status, priority, association, etc.

Bottom Panel

Bottom panel is also called *View Tabs* which has three display options:

- **Normal View:** It is used to edit your page.
- **HTML View:** It is used to see and/or alter the source code.
- **Preview View:** It is used to check how the page will behave when viewed on Internet. Preview tab will not appear if Internet Explorer is not installed.

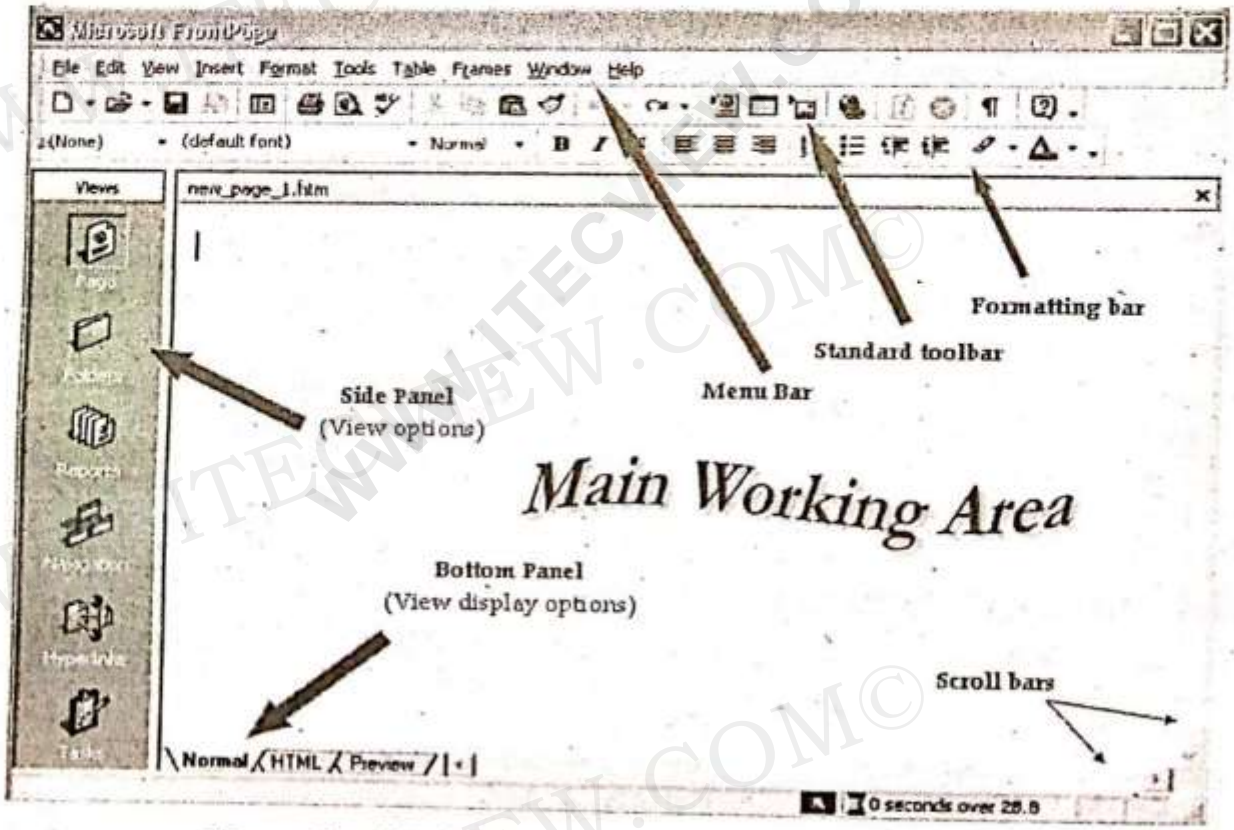


Figure: Graphical illustration of Front-Page environment

15.2 Designing the Documents

Web pages can be designed and developed using FrontPage simply by typing text on the blank document. We can also add pictures, tables, hyperlinks, etc to our documents.

Before beginning the actual development of site, you should create a directory on your hard disk to store all the web pages. It is important to put all images, files, documents in one directory for web publication.

First of all we will design document of our default page, working from page-view.

1. Create a folder WEB in any drive.
2. Click **Start -> Programs -> Microsoft FrontPage** to open FrontPage. If you are not in page view, click **Page** button on the **Views bar.** If normal tab is not selected then click the **Normal** tab at the bottom of the page.

Adding text

Now insert the cursor into the blank page and type any text as you type in a wordprocessor. An example is as below:

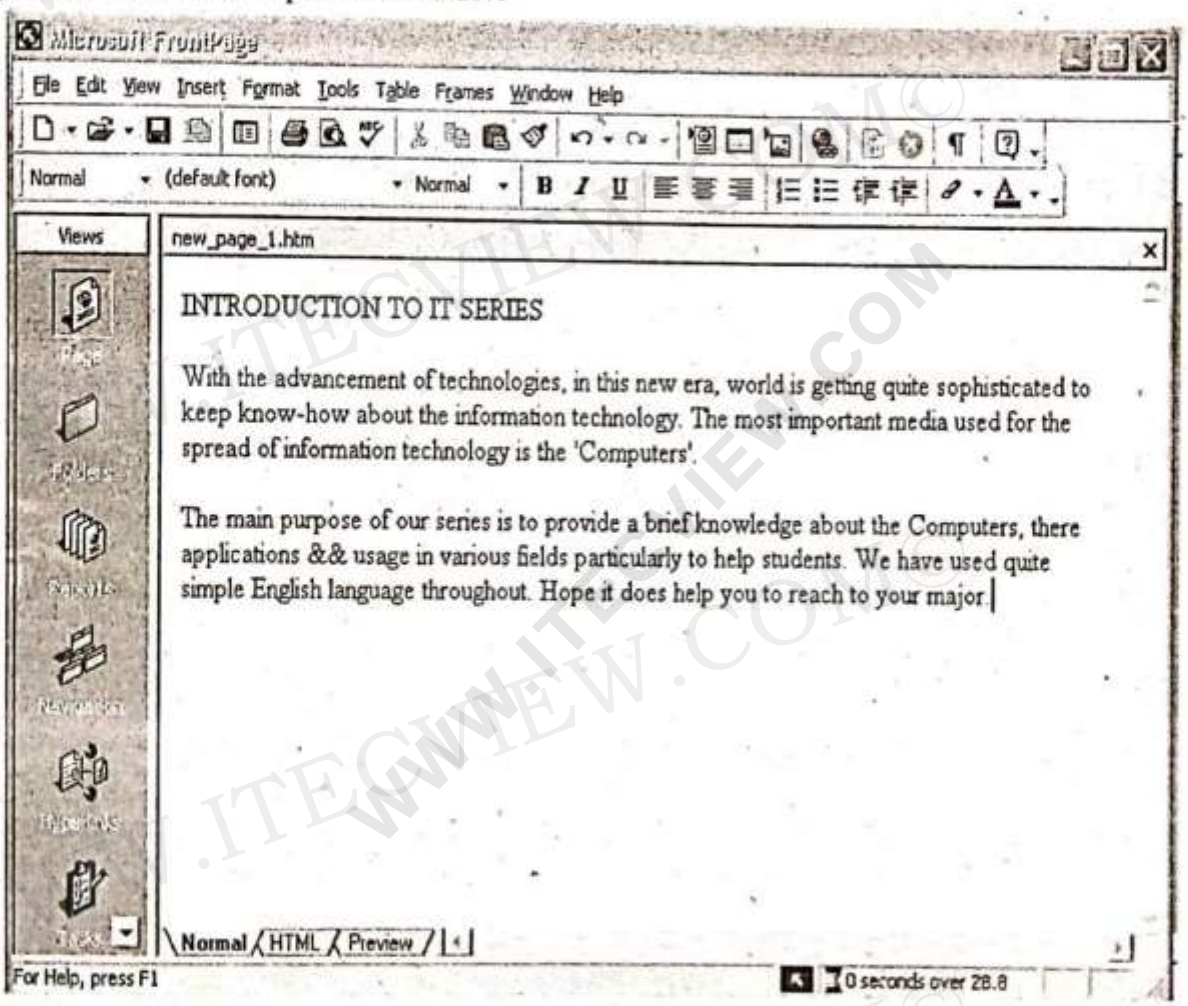
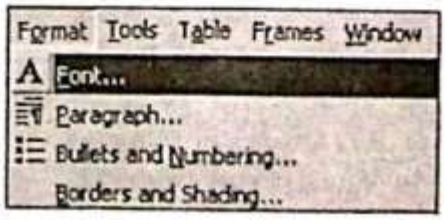


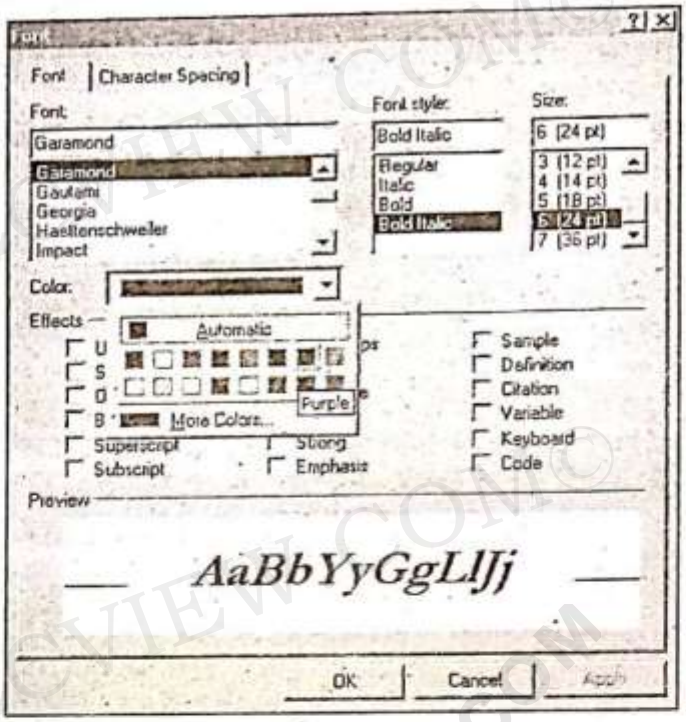
Figure: FrontPage with some typed text in Page-view

Formatting text

You can apply various effects to your text for making it more attractive. Here are some attributes that you may apply to text:



1. Select your desired text to format.
2. Click **Format >Font** on the menu bar.
3. Choose the desired Font under **Font** heading.
4. Choose the desired style under **Font Style** heading.
5. Choose the desired Font Size under **Size** heading.
6. Click **Color** drop down arrow and choose the desired Font Color.



7. Press **OK** to apply the formatting.

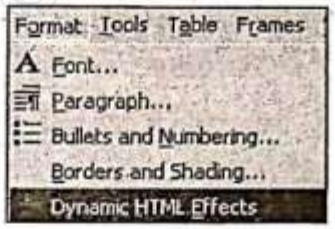
Applying Effects

1. Select your desired text (I selected last line only).
2. Click **Format >Font** on the menu bar.
3. Under **Effects** heading, click desired boxes (I clicked Capitalize & Definition boxes).

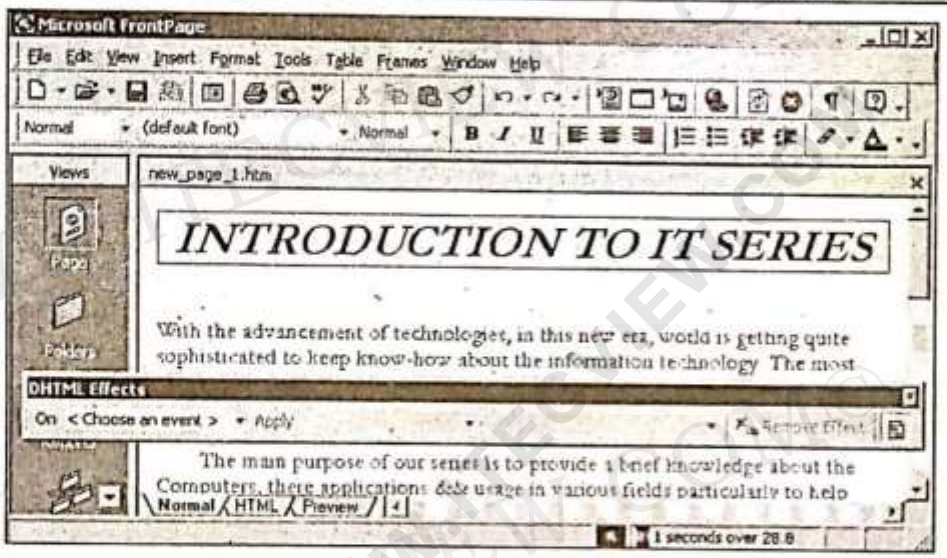
4. Press **OK** to apply the effect on selection.

Dynamic Text Effects

Dynamic text effects are used for applying some attractive effects to text such as animation.



1. Select the text to which you want to apply some dynamic effect (I selected the heading line *INTRDOUCTION TO IT SERIES*).
2. Click **Format >Dynamic HTML Effects** on the menu bar. **DHTML Effects** toolbar will appear.
3. Click **On** drop down arrow to choose the action at which DHTML effect will begin to appear (I selected **Page load**).
4. Now click on **Apply** drop down arrow to choose the effect to be applied on the selected text (I selected **Wave**). Text to which DHTML effect is applied will be highlighted automatically.

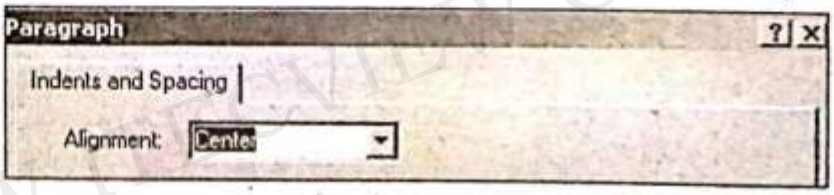


Note: Text is highlighted to distinct normal text from DHTML applied text. It is not highlighted in browser. To remove this highlighting in FrontPage as well:

5. Click Highlight Dynamic HTML Effects button on DHTML Effects toolbar.
6. Click anywhere in the text to which DHTML effect is applied.
7. Press Remove Effect button on the DHTML Effects toolbar.

Aligning the Text

1. Select your desired text (I selected first line only).
2. Click **Format->Paragraph** on the menu bar.
3. Click the **Alignment** drop down arrow and choose **Center** to align the text to center.

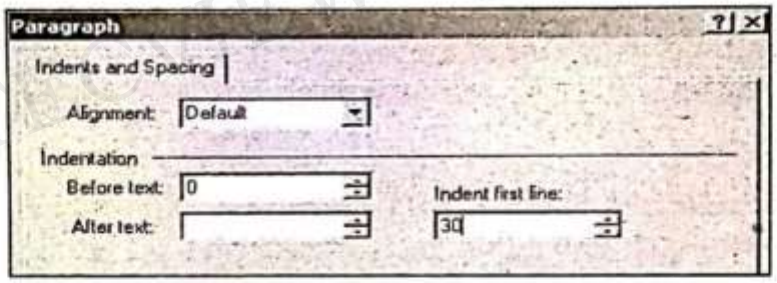


4. Press **OK** to apply the alignment effect on text.

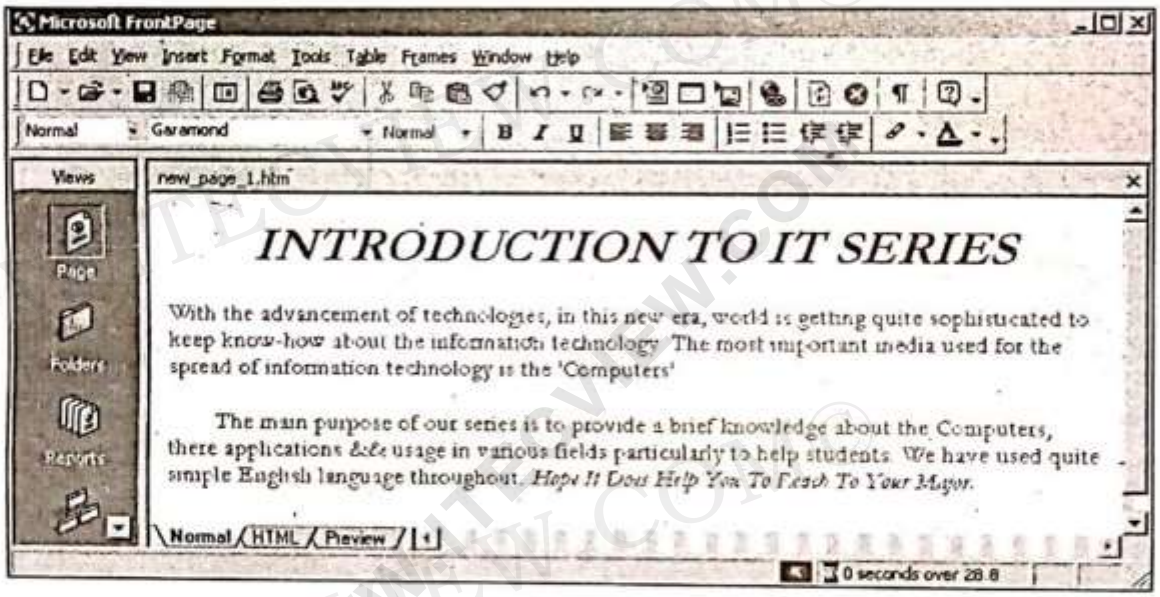
Paragraph Indents

1. Place the cursor at start of paragraph you want to indent (I placed at start of second paragraph).

2. Click **Format >Paragraph** on the menu bar.
3. Select or type your desired values in **Before text, After text** and **Indent first line** boxes (I typed 30 in **Indent first line** text box), under **Indention** heading.

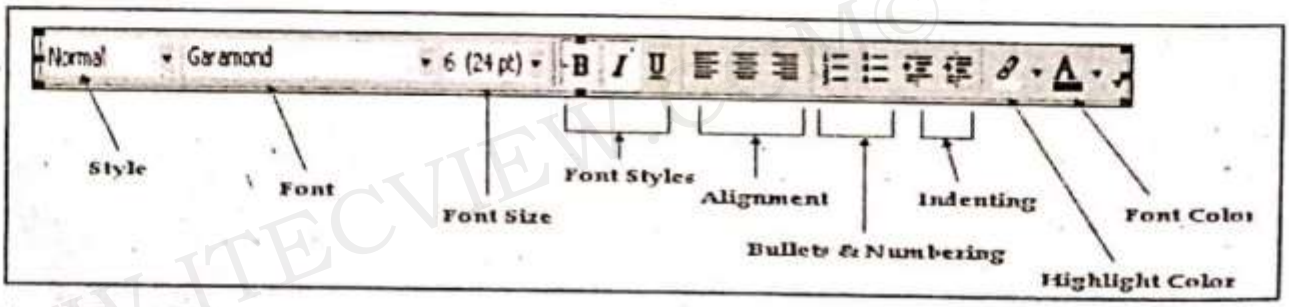


4. Press **OK** to set alignment of paragraph. Your page will look like as follows:



You can also use **Formatting toolbar** for changing font size, style, color, size, etc.

You can also use **Style** drop down list to change the heading styles of text. Heading styles in the **Style** list are based on universal HTML standards. A Level 1 heading is the largest possible text style for web pages while Level 6 is the smallest possible size. However, keep in mind, if you are going to use heading styles, they don't work properly in Netscape but do work when used in themes. The picture below shows various format options provided on the **Formatting** toolbar:



Copying Text Format

Format painter is used to apply the formatting of a text to some other text. It copies the format of one text to other text as it is. To copy the format:

1. Select the text whose format you want to copy.

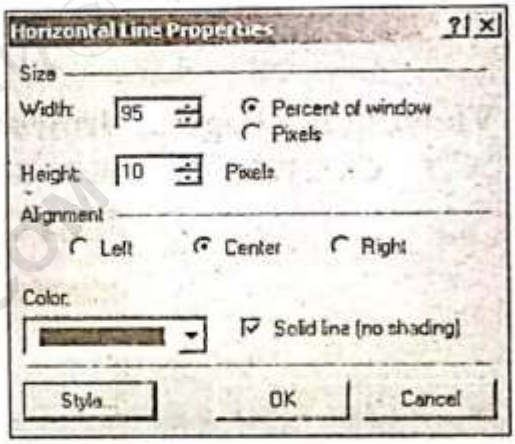
2. Click the **Format Painter** button on the Standard toolbar. The cursor will turn into a paintbrush.
3. Select all the text on which you want to apply the format.

Horizontal Lines

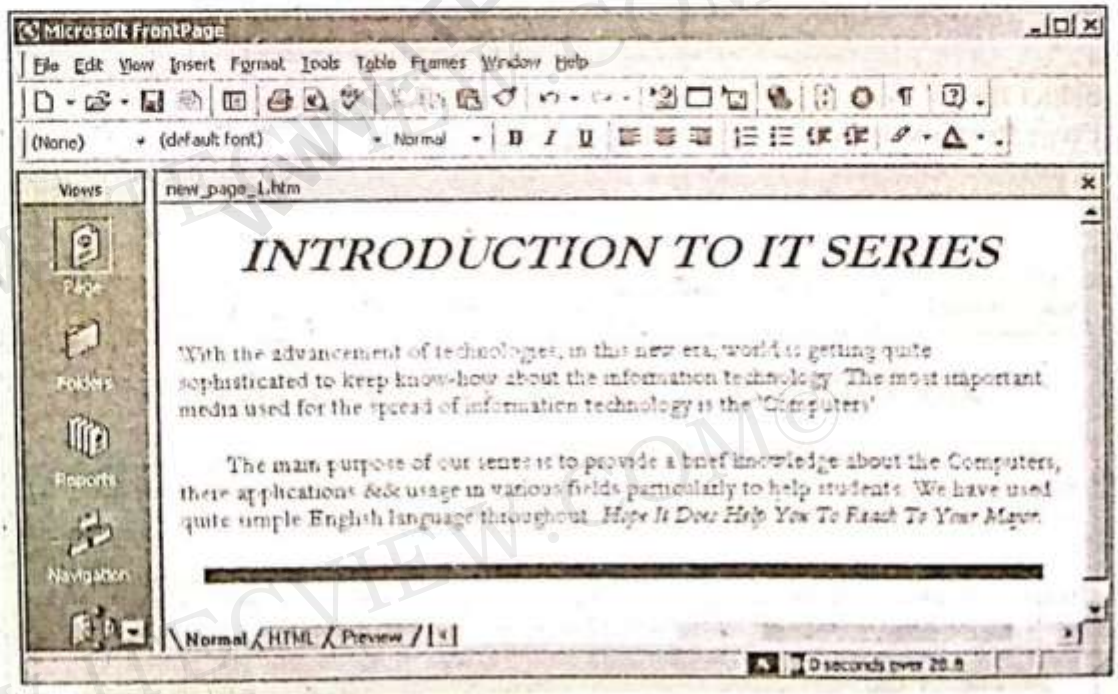
Horizontal lines are used to divide sections of text for better readability.

1. Place the cursor anywhere after which you want to insert horizontal line (I placed at the end of second paragraph).
2. Click **Insert >Horizontal Line** on the menu bar. Horizontal line will be inserted:

Changing Horizontal Line Properties



1. Double click on **Horizontal Line** to change its properties. Horizontal Line Properties dialog box will appear.
2. Under the **Size** title, select your desired line **Width** and **Height**.
3. Under the **Alignment** heading, click desired alignment.
4. Click **Color** drop down arrow to select your desired color.
5. Lines are by default shaded to give illusion of depth. To make the line of one color exactly, click the **Solid Line (no shading)** check box.
6. Click OK to apply new properties. The page will look like as follows:

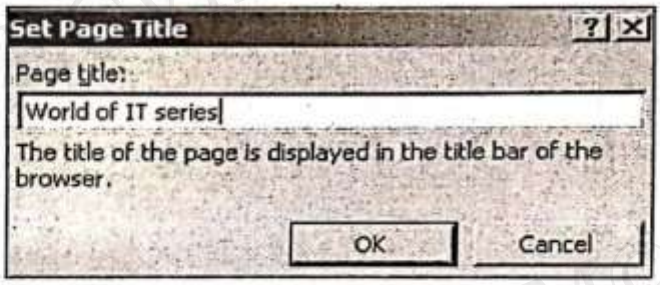


Saving the Page

1. Go to **File** menu and click **Save**. The **Save As** dialog box will appear.
2. Click **Save in** drop down arrow to select the location for saving the page.
3. Type a name for your page in **File Name** text box.
4. Press **Save** button to save the page.

Note: You can change your page title i.e. the text that appears on the title bar. By default, title is the based on the first line of text on the current page.

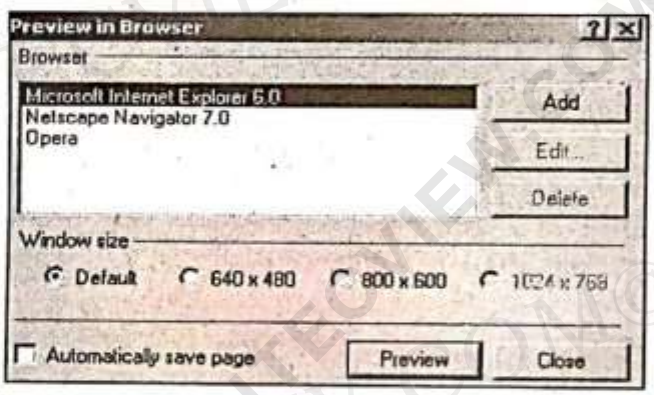
5. Click **Change** button next to **Page Title** heading in **Save As** dialog box.
6. Type a title for your page.



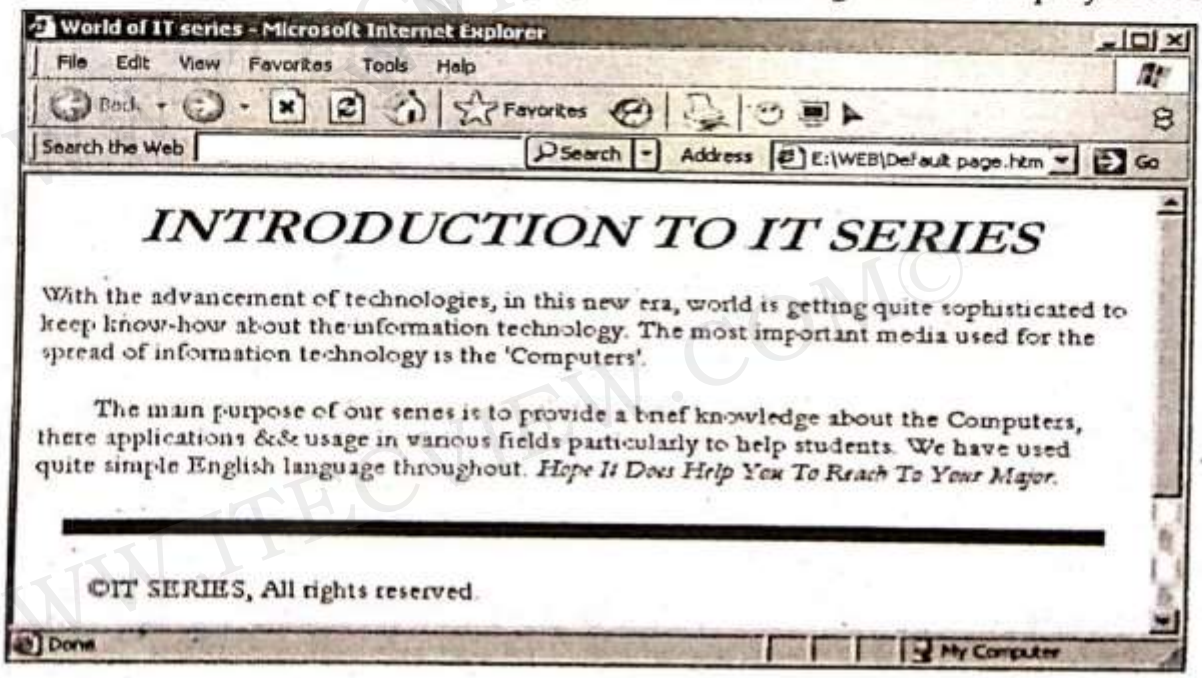
7. Click **OK** to change the title.

Viewing the Page in Browser

1. Click **File > Preview in Browser. Preview in Browser** dialog box will appear.



2. Select the browser in which you want to preview your page.
3. Press **Preview** button to see your page in browser. Page will be displayed in browser.



15.3 Hyperlinks

Hyperlinks are the text or graphic that are used to move the user to another location e.g. to another web page. Hyperlinks link various pages and websites. Hyperlinks can be very easily identified by following properties:

- They are normally underlined.
- Color of hyperlinks is normally different from normal text.
- When mouse pointer moves to a hyper link, pointer changes to a pointing hand.

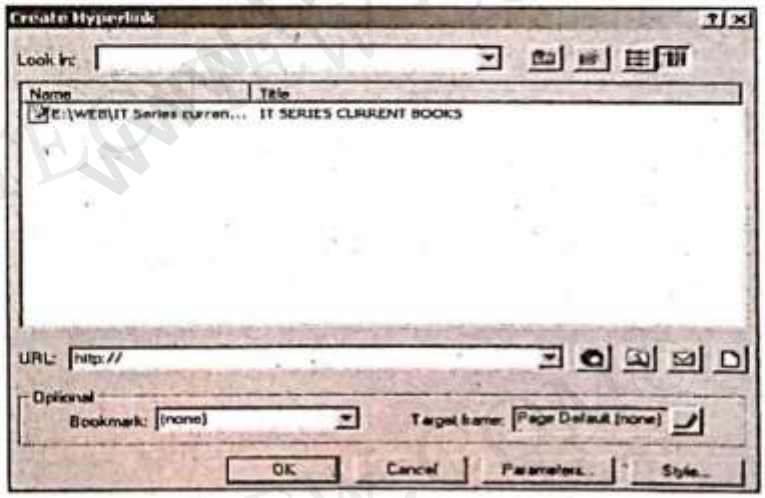
15.3.1 Types of Hyperlinks

Hyperlinks can be of three kinds:

- **Page-Jumps:** These are the hyperlinks that are linked with other locations within the same page. The location, to which this hyperlink is linked, is called **Bookmark.** These bookmarks help the user to easily navigate the site without too much scrolling.
- **Internal Links:** These are the hyperlinks that are linked with other web pages within the same website.
- **External Links:** These are the hyperlinks that are linked with other websites or other web pages of other websites.
- **Email Links:** These are the hyper links that are linked with emails. They start email application when the user clicks on them.

Creating Internal Links

1. Open the page in which you want to create an internal link.
2. Select the text to create a hyperlink.
3. Click **Insert >Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.



4. Click on icon to browse page on your computer.
5. Select your desired file from **Select File** dialog box.
6. Press **OK** to create a link to selected page.

Creating External Links

1. Open the page in which you want to create an External link.
2. Click **Insert >Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
3. Type a website address in **URL** text box starting with **http://**.

4. Press **OK** to create hyperlink to typed website.

Note: External URLs must begin with **http://** else they will not work.

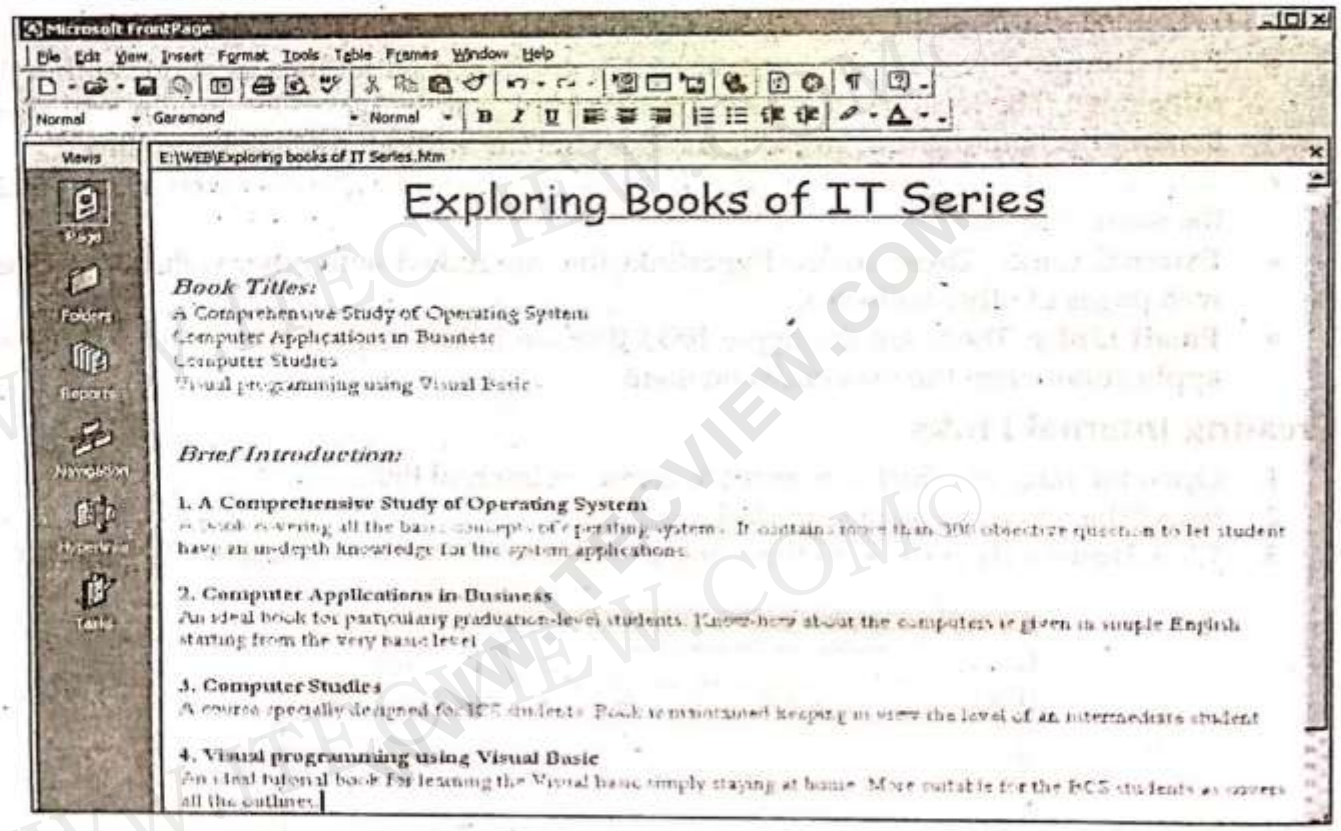
Creating Page Jumps

Procedure of creating jumps within the same page consists of two steps:

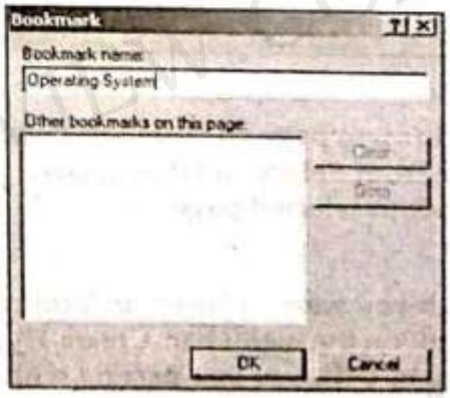
- Adding Bookmarks
- Creating Links to bookmarks

Adding Bookmarks

1. Open the page in which you want to create page jumps.



2. Place the cursor at start of text which you want to act as bookmark (I placed at start of first book's introduction).
3. Click **Insert >Bookmark** on the menu bar. **Bookmark** dialog box will appear.
4. Type a name for your bookmark in **Bookmark name** text box (I typed 'Operating System').



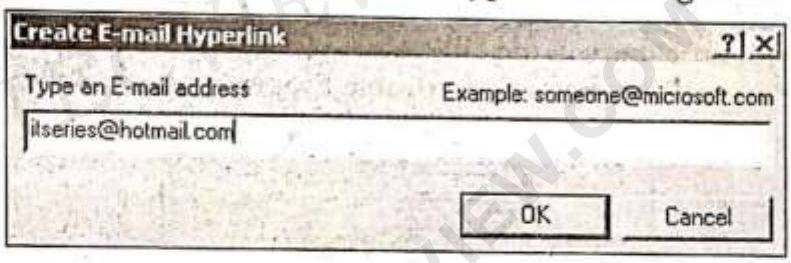
5. Press **OK** to add bookmark to page.
6. Repeat step 2 to 5 for adding more bookmarks. A flag will be added at each bookmark in FrontPage to indicate that text is book-marked.

Creating Links to Bookmarks

1. Select the text which will act as a link to bookmark (I selected first book's title).
2. Click **Insert >Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
3. Click **Bookmark** drop down list under optional heading to choose the bookmark for which you want a link (I selected 'Operating System').
4. Press **OK** to create link to selected bookmark.
5. Repeat step 1 to 4 for creating more links to other bookmarks.

Creating Email Links

1. Open the page in which you want to create an Email link.
2. Click **Insert >Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
3. Click the icon to open **Create E-mail Hyperlink** dialog box.



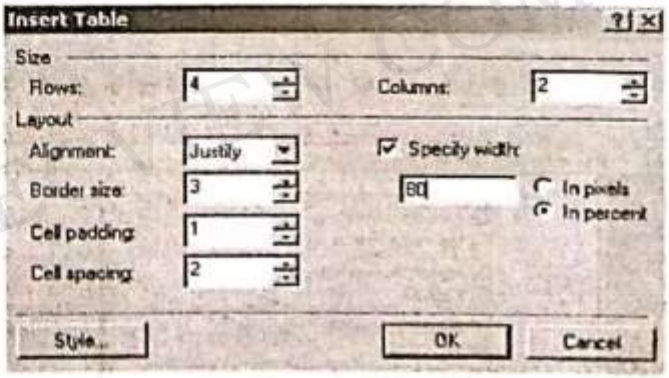
4. Type the email address, to which you want a link, in the **Type an E-mail address** text box (I typed itseries@hotmail.com).
5. Press **OK** button to close the dialog box. Your typed email address will appear in the **URL** text box (with "mailto: " prefix) of **Create Hyperlink** dialog box.
6. Press **OK** to create an email link.

15.4 Tables in FrontPage

Arrangement of data in the forms of rows and columns is called Table. Like other applications, in FrontPage, facility of tables is provided. Table can be created through many ways For example, explicitly through pencil, automatically through dialog box, etc.

Inserting Table

1. Open the page in which you want to create a table.
2. Click **Table > Insert > Table** on the menu bar. **Insert Table** dialog box will appear.

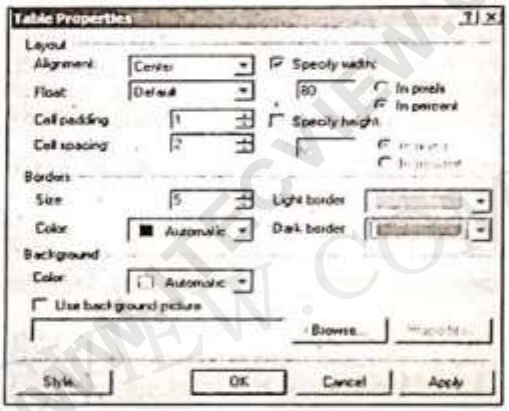


3. Type your required rows and columns in **Rows** and **Columns** text boxes.

4. Choose your desired alignment from the **Alignment** option list.
5. Type appropriate values in **Border size, Cell padding** and **Cell spacing** boxes.
6. If you want to specify the width of your table then check in the **Specify width** checkbox and type the desired value in the text box lying under it. You can set the width in pixels or percentage by selecting either **In pixels** or **In percent** radio button.
7. Press **OK** to insert a table with the selected properties as follows:

Changing Table Properties

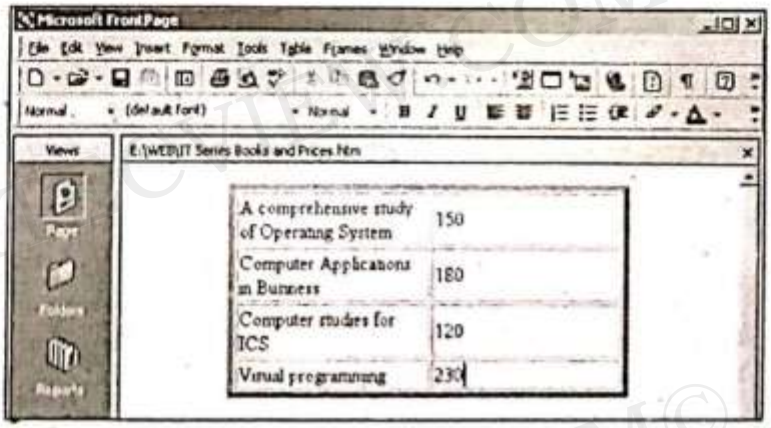
1. Place the cursor anywhere in the table and right click. A pop up menu will appear
2. Click Table Properties option to open Table Properties dialog box.



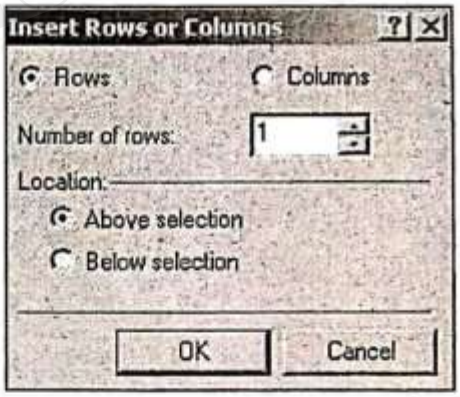
3. Change the properties as you desire.
4. Press **OK** to apply the new settings.

Adding Text to Table

1. Place the cursor in the cell where you want to add the text.
2. Type any text.
3. Press **Tab** to move to next cell. You can use **Shift + Tab** to move to previous cell.

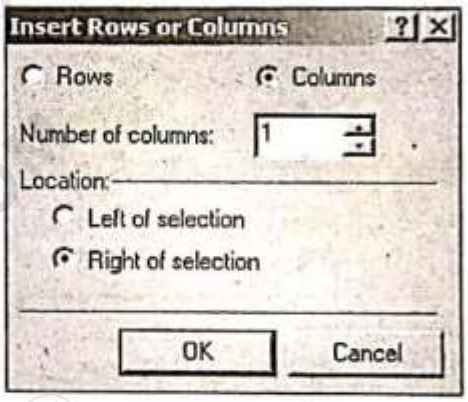


Inserting Rows



1. Place the cursor in the cell after or below which you want to insert the row.
2. Click **Table >Insert >Rows or Columns** on the menu bar. **Insert Rows or Columns** dialog box will appear.
3. Click **Rows** radio button.
4. Type any value in the **Number of rows** box.
5. Select any desired radio button under the **Location** heading.
6. Press **OK** to insert the row. New rows will be added in the table.

Inserting Columns

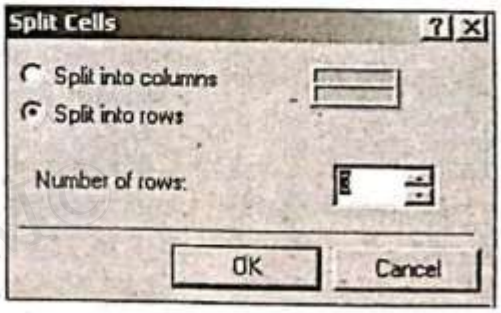


1. Place the cursor in any column at right or left of which you want to insert column.
2. Click **Table->Insert->Rows or Columns** on the menu bar. **Insert Rows or Columns** dialog box will appear.
3. Click **Columns** radio button.
4. Type required value in **Number of columns** box.
5. Select any desired radio button under the **Location** heading.
6. Press **OK** to insert the column.

Merging Cells

1. Select all the cells of table which you want to merge and right click. A popup menu will appear.
2. Click **Merge Cells** option from the pop up menu. The selected cells will merge into a single cell.

Splitting Cells



1. Place the cursor in the cell which you want to split and right click. A popup menu will appear.
2. Click **Split Cells** option from the pop up menu. **Split Cells** dialog box will appear.
3. Click your desired radio button.
4. Type number of rows or columns.
5. Press **OK** to split the cells.

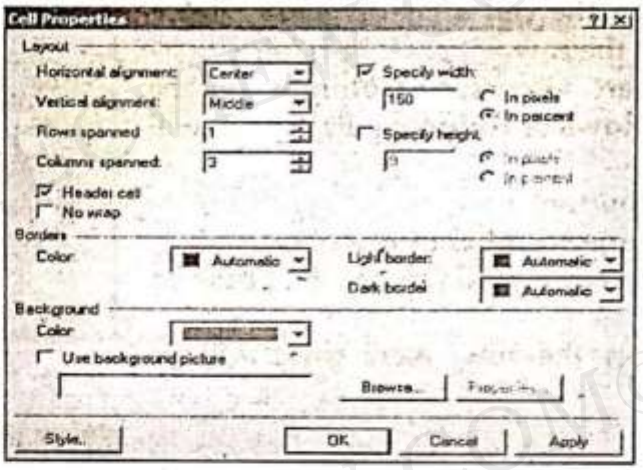
Deleting Cells

1. Select the cells which you want to delete and right click. A popup menu will appear.
2. Select **Delete Cells** option from the pop up menu.
3. All the selected cells will be deleted.

Changing Cell Properties

1. Right click in the cell whose properties you want to change. A popup menu will appear.

2. Click **Cell Properties** option from popup menu. **Cell Properties** dialog box will appear.



3. Change the properties as you desire.
4. Press **OK** to apply the new settings on the current cell.

Converting Text to Tables

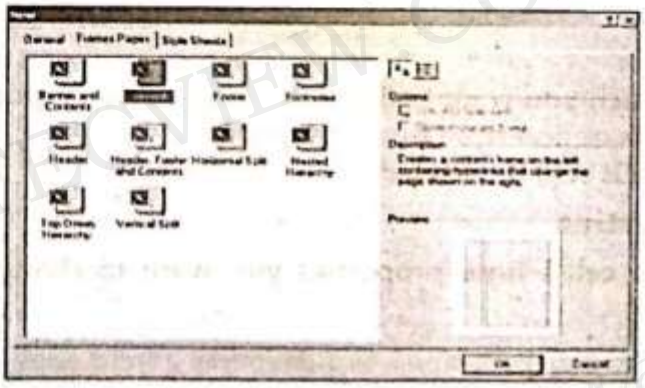
1. Select the text to convert into table.
2. Click **Table > Convert > Text To Table** on menu bar. **Convert Text To Table** dialog box will appear.
3. Select your desired radio button depending upon delimiter of your text under **Separate text at** heading.
4. Press **OK** to convert text into table.

15.5 Frames

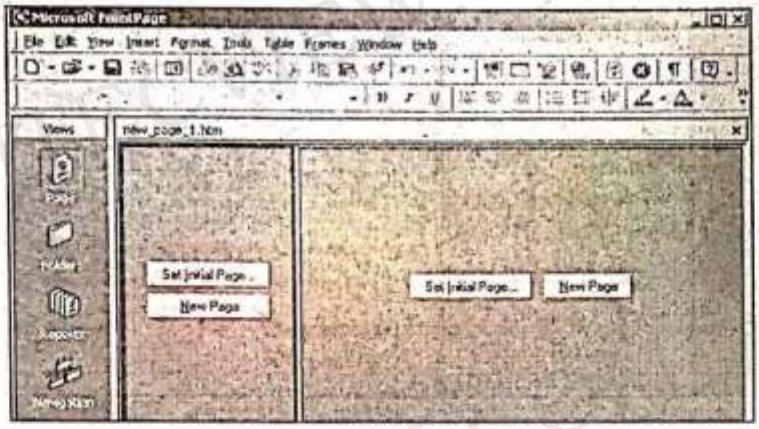
Frames are used to divide a web page into different sections. Many websites display contents list on one side and the introduction on the other side in a single page. This is the actual use of frames. Frames are helpful to use a portion of screen for static purposes and other for navigation. However, frames increase download time and may cause some troubles during linking and printing. Remember that frames are just containers. You have to design a web page separately that will be displayed in the frame.

Creating Frames Page

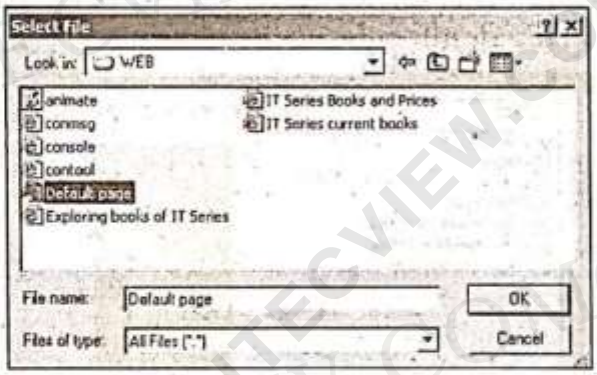
1. Open the **FrontPage.**
2. Select **File > New > Page** on the menu bar. **New** dialog box will appear.
3. Click Frames Pages tab.
4. Choose your desired frame format icon, by seeing the **Preview** window.



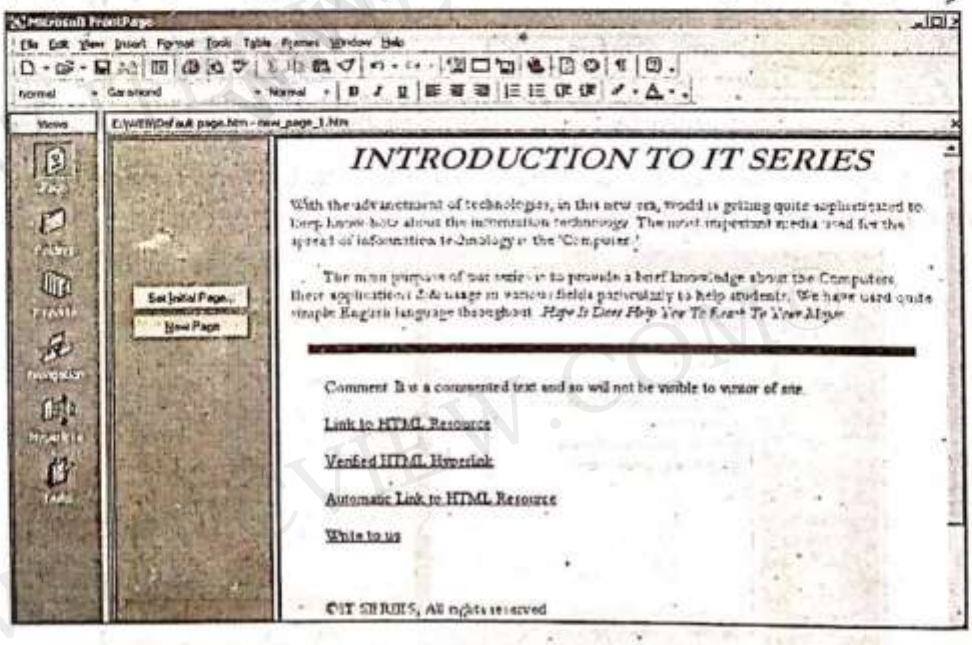
5. Press **OK**. The new frames page will open in the **Page view** and would have different frames according to the format you selected. Each frame will have two buttons: **Set Initial Page...** and **New Page**.



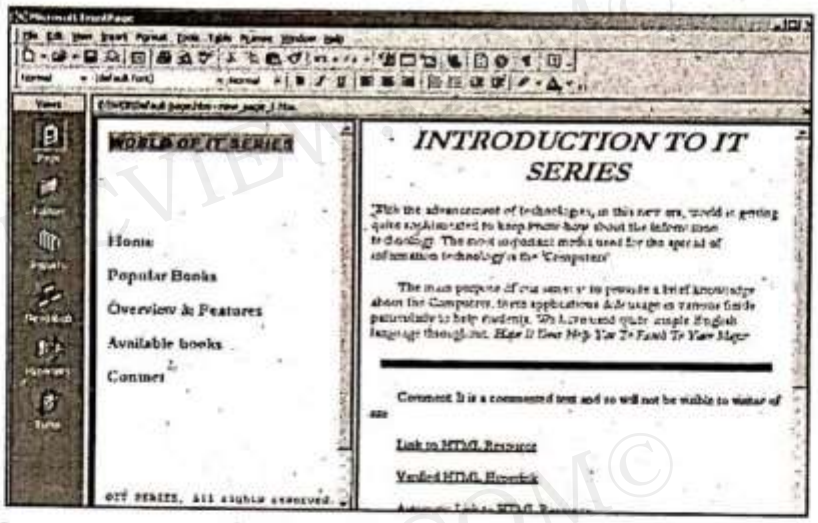
6. If source page for the frame already exists then click **Set Initial Page...** button. **Select File** dialog will appear.



7. Choose your required page and press **OK**. The selected page will open in the frame.

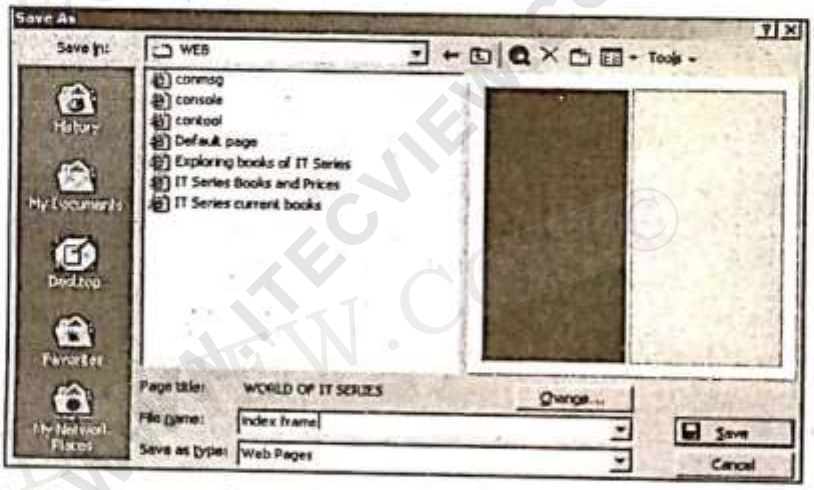


8. If source page for the frame does not exist then click **New Page** button. Frame for which you have clicked **New Page** button will change into white space.
9. Type and format any text in the page.

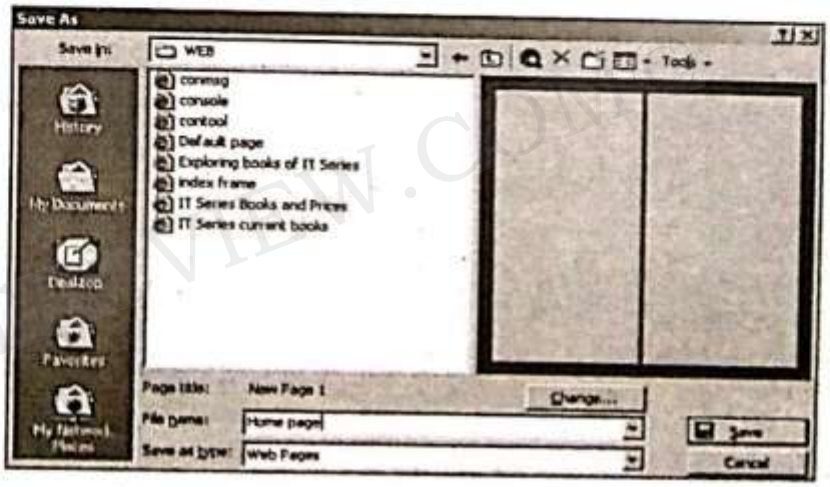


Saving Frames Page

1. Click **File > Save** on the menu bar. **Save As** dialog box will appear.
2. Choose a location and name for the current frame page that is being saved (see the preview window on top right).



3. Click **Save** button to save the selected frame.
4. Now, again choose a location and name for the whole frames page (preview window shows whole frames page) being saved.



5. Press **OK** to save the page.

15.5.1 Hyperlinks in Frames

Links are created in frames similar to the normal pages. However, there is a difference of changing the target frame. **Target Frame** points to the frame where the pointed page opens. In the FrontPage, target frame can be one of the following:

- **Page Default:** Loads the page in default target frame (indicated in the parentheses).
- **Same Frame:** Loads the page in the same frame where hyperlink is created.
- **Whole Page:** Opens the page in the same browser window.
- **New Window:** Opens the page in the new browser window.
- **Parent Frame:** Loads the page in the current window.

Creating Hyperlinks in Frames

1. Select the text which you want to act as a hyperlink.
2. Click **Insert > Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
3. Click Change Target Frame icon (next to Target Frame option). **Target Frame** dialog box will open.
4. Select your desired target under **Common targets** title (I selected **Page Default**).
5. Press **OK** to set the target.
6. Click on icon in **Create Hyperlink** dialog box to browse page on your computer.
7. Select your desired file from **Select File** dialog box.
8. Press **OK** to create a link to selected page.

Modifying Default Target Frame

Target frames are already set in the frames pages that are created using frames page templates. For example, in the Contents template, any hyperlinks created on the page in the Contents frame (left side frame) automatically display the page (pointed to by that hyperlink) in the Main frame (right side frame). It is possible to either change the default target frame for just a single hyperlink or for all the hyperlinks on the page.

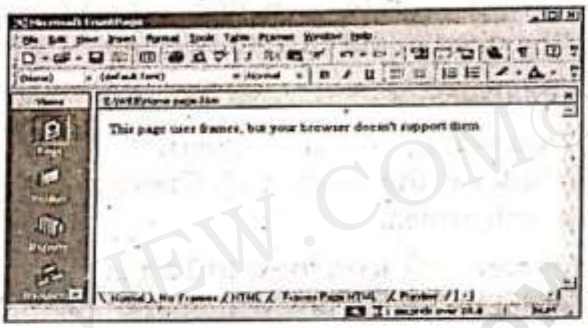
1. In the Frames page, Right-click the hyperlink you want to modify.
2. Click **Hyperlink Properties** from popup menu. **Edit Hyperlink** dialog box will open.
3. Click **Change Target Frame** icon. **Target Frame** dialog box will open.
4. Under the **Current frames page** area, click the frame in which you want to display the page pointed to by hyperlink.

Note: To set the selected frame as default target frame for all the hyperlinks on the page, clink in **Set as page default** check box. Now all the pages that are pointed to by some hyperlink, in the current frame, will automatically open in the specified target frame.

No Frames Page

No Frames tab shows a page that should be prepared for the users who have old versions of browsers. The old versions of browsers do not support frames. If any visitor is using an old browser, this page would be displayed instead of frames page to inform them.

By default, when you prepare a frames page in FrontPage 2000, an automatic text is appended in **No Frames** page to tell those old version visitors that frames are not supported by their browser. To see this page, open any frames page and then click **No Frames** tab, window will show text as shown below:



Note: In **No Frames** page, the user may include any additional text, links etc.

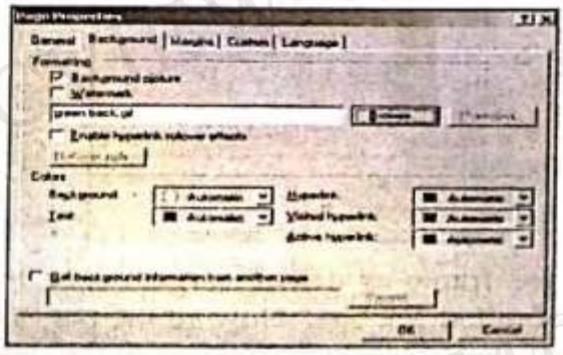
15.6 Graphics in FrontPage

Web pages can have many graphics as well. These images on the pages play a vital role for enhancing the page look. Generally, there are two basic formats that are recognized by HTML code. These graphic formats are as follows:

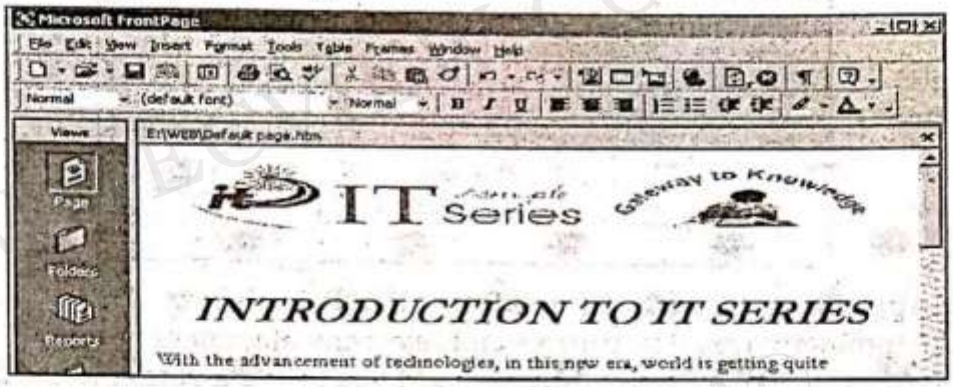
- **GIF (Graphic Interchange Format):** Files in this format are formed of 256 or less colors. Since number of colors is very small so they do not need compression. Mostly GIF images comprise of small photos, icons or bullets.
- **JPEG (Joint Photographic Exchange Group):** Files in this format are formed of 16 million colors. As number of colors is extensively large so they give excellent picture quality. Also they are stored in a compressed form as they are usually large enough and so take much more time for interpretation than GF images. Preference is always of JPEG images when Better quality is required.

Adding Background Images to the Page

1. Click **Format > Background** on menu bar. **Page Properties** dialog box will appear.
2. Click **Background** picture check box.
3. Click **Browse** button under **Formatting** heading. **Select file** dialog box will appear.
4. Choose your desired image and press **OK**.
5. Selected image's name and path appears in the text box under **Formatting** heading.



6. Press **OK** to apply the background.

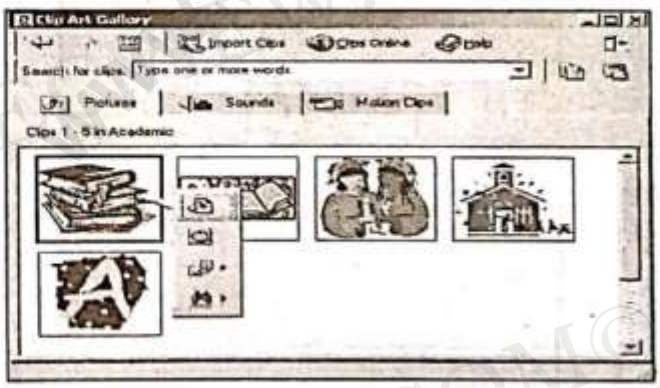


Adding Pictures from File

1. Open the page in which you want to insert the graphic.
2. Place the cursor at the desired position.
3. Click **Insert > Picture > From File** on the menu bar.
4. Choose your desired file from **Select file** dialog box.
5. Press **OK** to add the image.

Adding Clip Art Images

1. Open the page in which to insert clip art.
2. Place the cursor where to insert the image.
3. Click **Insert > Picture > Clip Art. Clip Art Gallery** dialog box will appear.
4. Click any category of images. All clips in the selected category will appear.

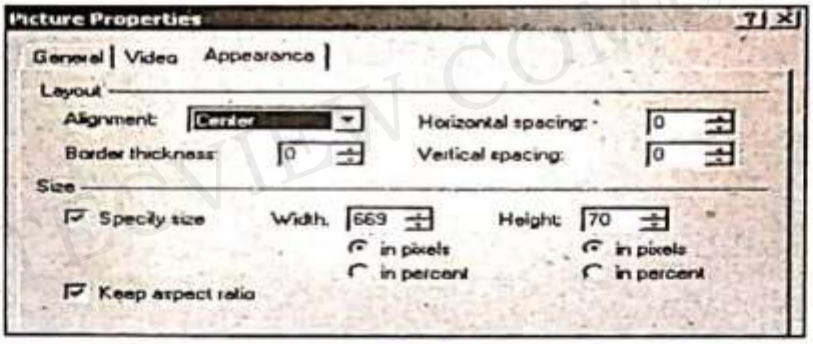


5. Click on your desired clip and choose **Insert Clip** button (first button on the drop down list). Clip will be inserted at the current pointer place, on the web page.

Changing Images Appearance

Image properties such as Alignment, Border size, picture size can be changed through **Picture Properties** dialog box. To set this image appearance, follow the given steps:

1. **Right click** the picture. A popup menu will appear.
2. Choose **Picture Properties** from the pop up menu appeared.

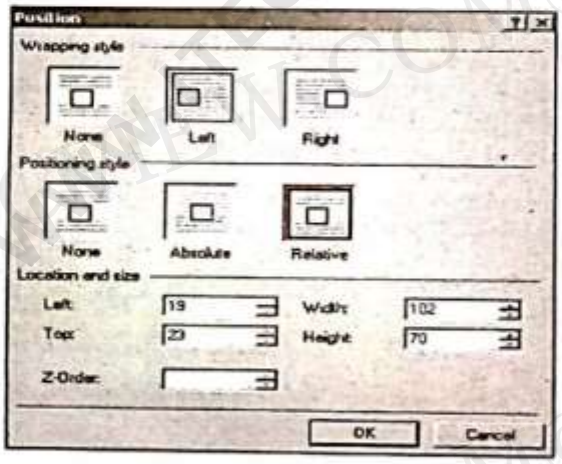


3. Click Appearance tab in the Picture Properties dialog box.
4. Click **Alignment** drop down arrow and select any alignment option.
5. Choose a value in the **Border thickness** text box to add a border to the picture.
6. Choose values in the **Horizontal spacing** and **Vertical spacing** text boxes to add space on left, right, top and below the picture.
7. Check **Specify size** text box and type new values in the **Width & Height** text boxes to change the size of the image.
8. Press **OK** to apply the settings.

Positioning Pictures in Relevance to Text

You can position text and pictures in relation to one another.

1. Select the image or text you want to position.
2. Go to **Format** menu and choose **Position. Position** dialog box will appear.
3. Under **Wrapping style** heading, select your desired style.
4. Under **Positioning style** heading, select your desired position.



5. Press **OK** to apply the settings.

Transparent Background of Images

If you have added a gif image in your web page then it is possible to make its background transparent as follows:

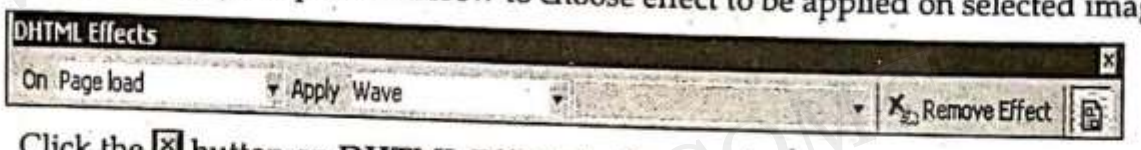
1. Click the image whose background you want to change.
2. Click the **Set Transparent Color** button on the **Image** toolbar. The cursor will turn into an arrowed pencil.
3. Place this pencil over any picture area which you want to make transparent and then click the color. The color becomes transparent.

Note: You cannot make backgrounds of JPEG images transparent.

Creating Dynamic Images

You can animate the images on your page as follows:

1. Select the image you want to animate.
2. Click **Format > Dynamic HTML Effects** on the menu bar. **DHTML Effects** toolbar will appear.
3. Click **On** drop down arrow to choose the action at which DHTML effect will begin to appear.
4. Click on **Apply** drop down arrow to choose effect to be applied on selected image.



5. Click the ☒ button on **DHTML Effects** toolbar, to close it.

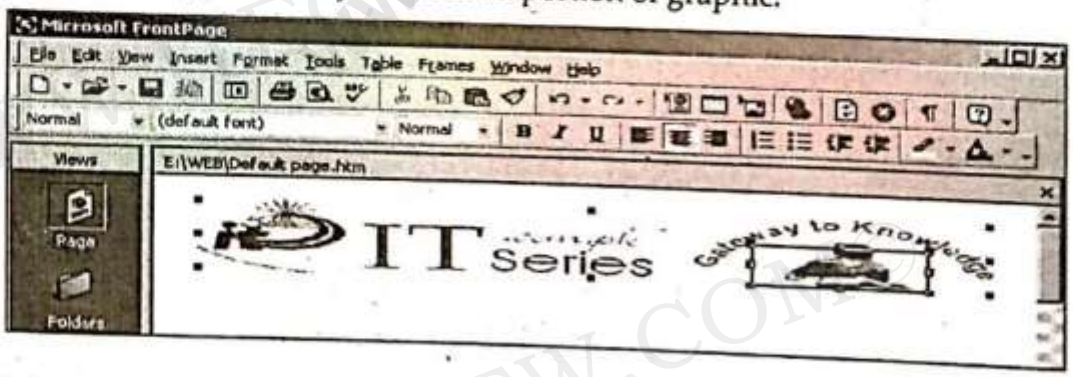
Hyperlinks of Images

1. Select the image to make it a hyperlink.
2. Click **Insert > Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
3. Click on icon to browse page on your computer.
4. Select your desired file from **Select File** dialog box.
5. Press **OK** to create selected image's link to desired page.

Hotspots

Normally, a single graphic links to just a single location. However, diferent parts of a single image can be linked to different locations. Such different portions of a single image that are used to link to different locations are called **hotspots**.

1. Select the picture on which you want to have hotspots.
2. Choose your desired hotspot shape from the **Drawing** toolbar.
3. Draw the shape around your desired portion of graphic.



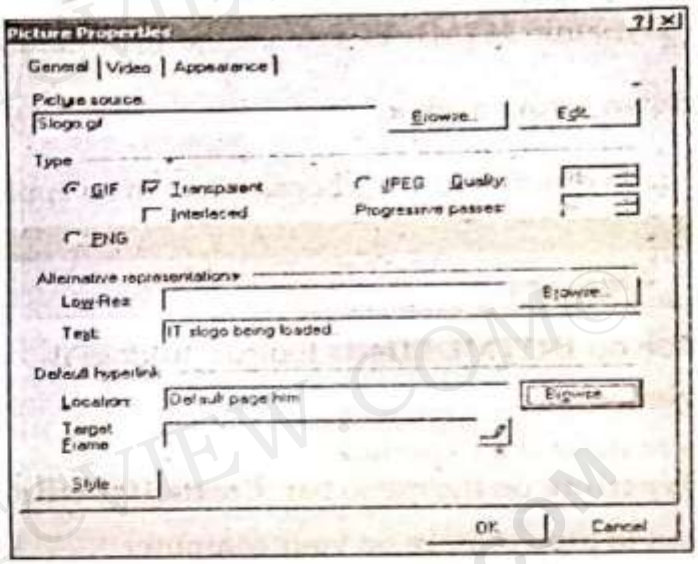
4. Release the mouse button. A hyperlink window will appear.
5. Enter the URL, e-mail address, or bookmark to link the hotspot.

Setting Default Hyperlink for Graphics

Suppose you have created different hotspots on the graphic. If you want to hyperlink all the remaining portion of the same graphic to a single location, then you need to set the default hyperlink for that graphic. The area not covered by hotspots will link to default link.

1. **Right click** on any area of the graphic that is not a hotspot.
2. Select **Picture Properties** from the pop up menu.
3. Under **Default hyperlink** heading, click **Browse** button.

4. Enter a URL, email address, or file for hyperlink in **Edit hyperlink** dialog box.
5. Press **OK** to close **Edit hyperlink** window. Your selected hyperlink will appear in the **Location** text box in **Picture Properties** window.



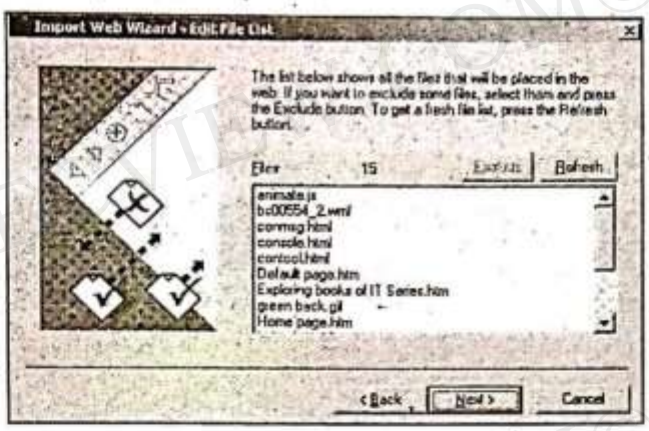
6. Press **OK** to set the default hyperlink for the current image.

15.7 Creating Web in FrontPage

Follow the given steps to create a web of already developed pages:

1. Select **File > New > Web** on the menu bar. **New** dialog box will appear.
2. Click **Import web wizard** icon.
3. Specify a location for your web in the **Specify the location of the new web** text box. You must create web in any folder not drive. **Import Web Wizard-Choose Source** dialog box will open.

4. Click your desired radio button.
5. Click **Browse** button to specify the location where your web pages exist.
6. Press **Next** to move ahead. The next windows will show the list of all the files that are going to be added in your web.
7. Select the file you want to delete (if any) and press **Exclude** button.



8. Press **Next** to move ahead. Last phase of wizard will open.

9. Press **Finish** to close the wizard. A new web, having your already developed pages, will be created.
10. Click on the **Folders** view to see the list of all the added files.

- Whenever a new web is created, FrontPage 2000 automatically includes a **private** folder to keep the information that you want to hide from your site visitors and an **image** folder to save all images in a single folder.
- You may create new folders and then arrange the files of your web.

Opening the Web

1. Click **File > Open Web** on the menu bar.
2. Select your desired web from the **Open Web** dialog box.
3. Press **OK** to open the selected web.

15.8 Web Site Publishing in FrontPage

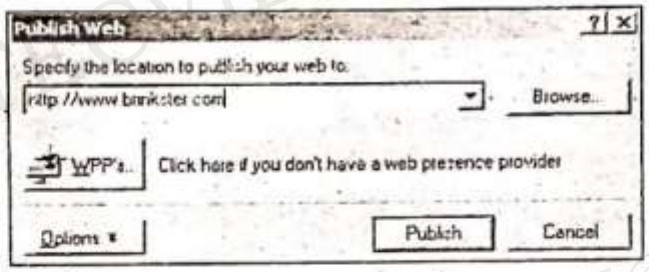
Web site publishing is a process of uploading the complete website on the Internet. When website is published on the Internet, any person can view it using Internet.

In web publishing, website is stored on a powerful computer that stays connected to the World Wide Web all the time. This computer is called **Web server** and is used to store website documents. All the process of transferring files to web server is called **website publishing**.

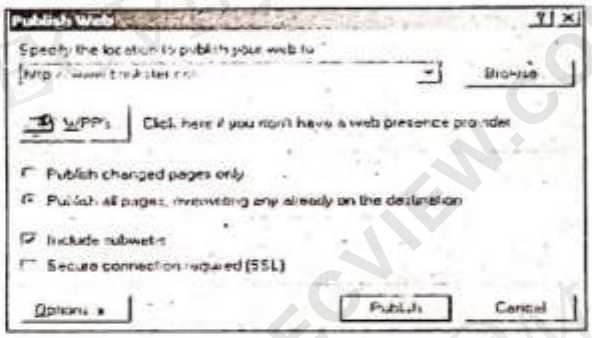
People visit websites by connecting to the web server through the World Wide Web. Many organizations use their own servers to connect to internet. Many people get access to internet through ISP's. **ISP (Internet Service Provider)** is a company that provides internet access and also web site documents space on its servers. You publish your site by renting space for your site on an ISP server. Some ISP's provide this space for free however mostly provide it by taking charges for the service.

Follow the given steps to publish your site using FrontPage 2000:

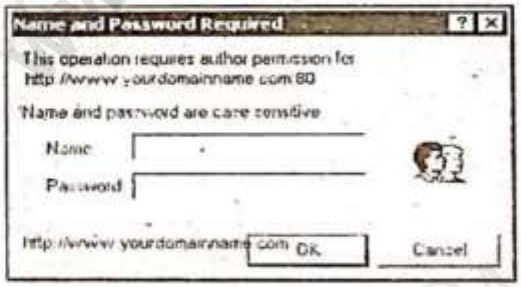
1. Get connected to Internet.
2. Open the web in FrontPage 2000 which you want to publish.
3. Click **File > Publish Web** on the menu bar. **Publish Web** dialog box will appear.



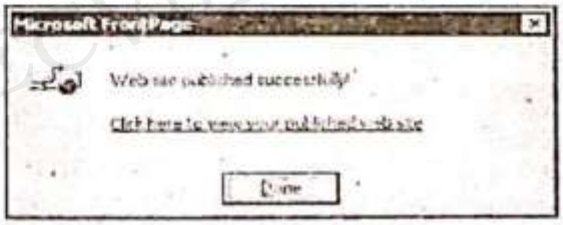
4. In **Specify the location to publish your web to** textbox, enter the full URL of your web server where space has been provided for your site.
5. Click **Options** button (at the bottom left side). **Publish Web** dialog box will extend to show you further options.



6. Click **Publish all pages...** radio button.
7. Click in **Include** subwebs check box.
8. Press **Publish** button. **Name and Password Required** dialog box will appear.



9. Enter your name in the **Name** text box and password in the **Password** text box.
10. Press **OK** to begin publishing. A dialog box will appear when site has been published as follows:



11. Press **Done** button to close the dialog box OR hit **Click here to view your published web site** link to see your published site on the internet.

Project

Start MS FrontPage 2000 and perform the following tasks:

1. Create an empty web named 'Business Web'.
2. In above web, create two pages named 'Products' and 'Orders'.
3. Create a Dynamic text heading in 'Products' page.
4. Add some text in both the pages.
5. Create a List of Products in 'Products' page.
6. Create a Table of Orders in 'Orders' page.
7. Set the Title of 'Products' page to 'Introducing our best products'.
8. Apply a single Theme to both the pages.
9. Create a graphic Email Link in 'Orders' page.
10. Create a text hyperlink in 'Products' page to connect to 'Orders' page.
11. Create a Frames page.

Solution

1. Creating an Empty Web

1. Click **Start > Programs > Microsoft FrontPage**.
2. Select **File > New > Web** on the menu bar. **New** dialog box will appear.
3. Click **Empty web** icon.
4. Specify a location for your web in the **Specify the location of the new web** text box. An empty web will be created.

2. Creating Products and Orders Page

1. By default, FrontPage opens with a page. Pressing **Ctrl+S** to save this page. **Save As** dialog box will open.
2. Type "Products" in **File name** text box.
3. Press **Save** to save the page.
4. Click **Close** button (on the top right) to close the page.
5. Click **File > New > Page** on the menu bar. **New** dialog box will open.
6. Click **Normal Page** icon.
7. Press **OK** to add the page.
8. Press **Ctrl + S**. **Save As** dialog box will open.
9. Type "Orders" in **File name** text box.
10. Press **Save** to save the page.
11. Click **Close** button (on the top right) to close the page.

3. Creating Dynamic Text Heading in Products Page

1. Click **File > Open** on the menu bar. **Open File** dialog box will appear.
2. Select "Products" file from the available list.
3. Press **Open** to open the selected page.
4. Type and Format the heading text as shown below:

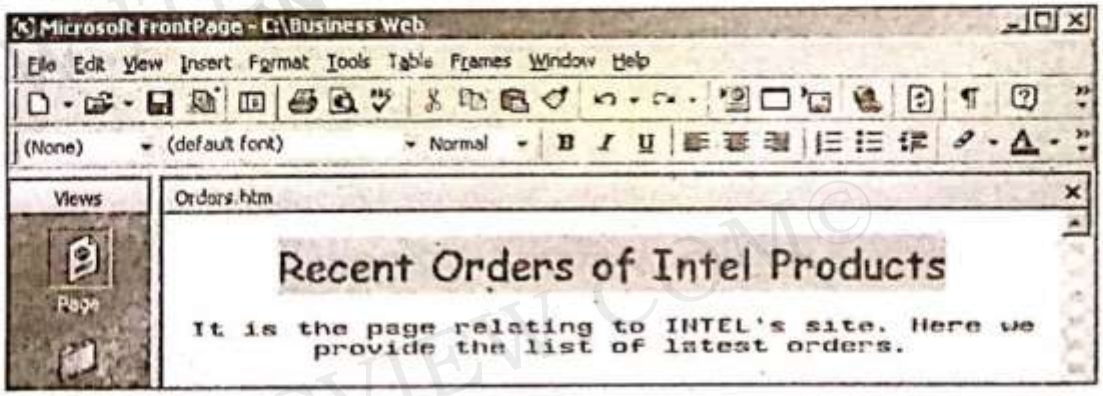
5. Select the heading to apply some dynamic effect.
6. Click **Format > Dynamic HTML Effects** on the menu bar. **DHTML Effects** toolbar will appear.
7. Click **On** drop down arrow and select **Page load**.
8. Now click on **Apply** drop down arrow & select **Wave**.

9. Click on **Apply** drop down arrow and select **Wave**. Dynamic effect will be applied.
10. Close the page.

4. Adding Text

1. Open **Products** page.
2. Add & Format text as shown below:

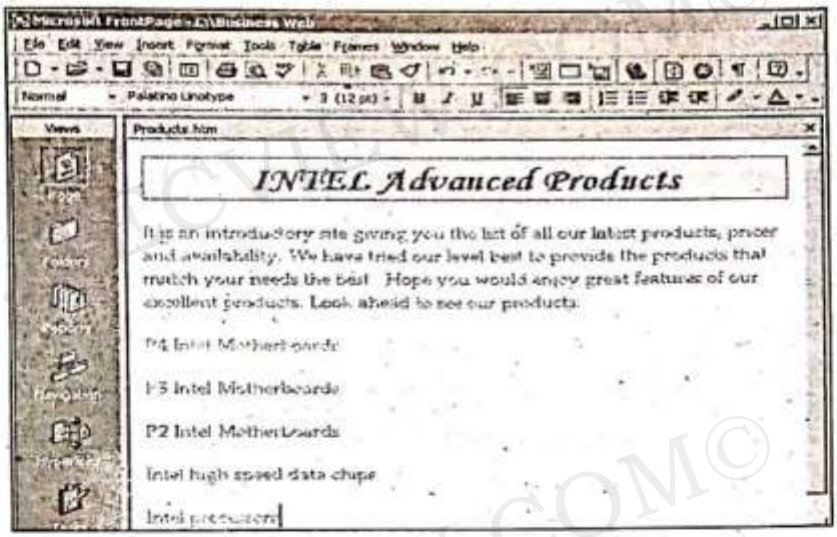
3. Save and close the page.
4. Now open **Orders** page.
5. Add and Format text as shown below:



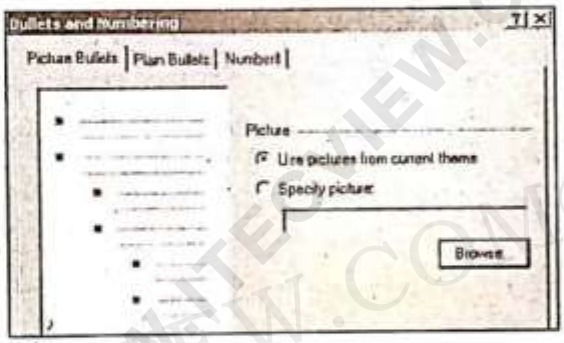
6. Save and close the page.

5. Creating Products List in "Products" Page

1. Open the **Products** page.
2. Type the text as shown below (press **Enter** after each product):



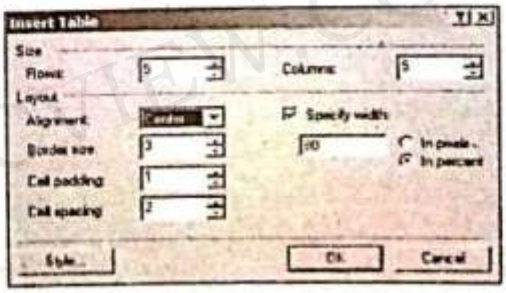
3. Select text from 'P4 Intel Motherboards' to 'Intel Processors'.
4. Click Format > Bullets and Numbering on the menu bar. Bullets and Numbering dialog box will appear.
5. Click **Picture Bullets** tab and then click **Use Pictures from current theme** option.



6. Press **OK** to apply the list on selected text.
7. Save and close the page.

6. Creating Orders table in 'Orders' Page

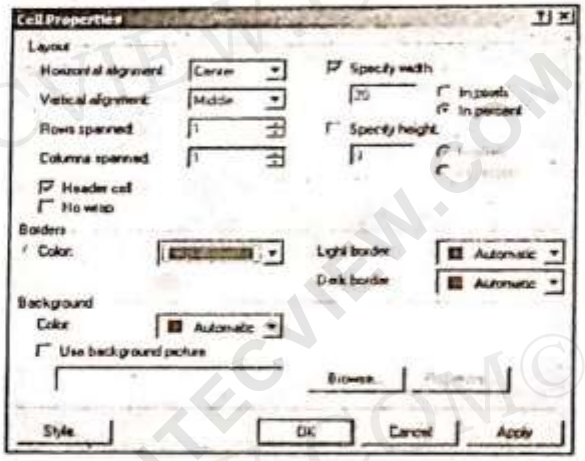
1. Open the 'Orders' page.
2. Place the cursor on the new line (next to last paragraph).
3. Click **Table > Insert > Table** on the menu bar. **Insert Table** dialog box will appear.
4. Type 5 in **Rows** box and 5 in **Columns** box under **Size** heading.
5. Choose **Center** from **Alignment** option list.
6. Type 3 in Border size, 1 in Cell padding and 2 in Cell spacing.



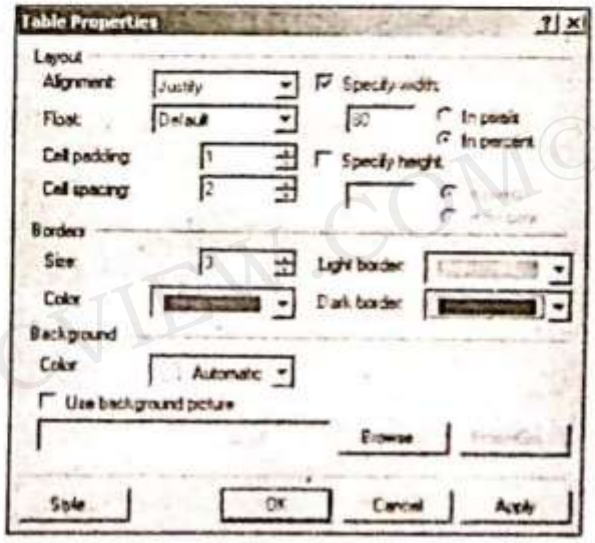
7. Press **OK** to insert the table
8. Now type the text in the table as shown below:

Product name	Number of orders	Order Date	Delivery Date	Customer
P3 CPU	13	12/3/05	15/3/05	E-Links FSD
Motherboard	10	15/4/05	30/4/05	IT-World LHR
P4 CPU	30	3/5/05	1/6/05	IT-World LHR
Motherboard	13	3/5/05	15/5/05	Yahoo-Net KHR

9. Select the first row and right click.
10. Click **Cell Properties** option from the pop up menu. A dialog box will open.
11. Set the attributes as shown in the picture below:



12. Click **OK**.
13. Right click anywhere in the table.
14. Select **Table Properties** option from pop up menu. **Table Properties** dialog box will open.
15. Change the properties as shown in the picture below:



16. Press **OK** to apply the new table settings.
17. Now select the whole table and right click.
18. Choose **Font** option form the pop up menu. **Font** dialog box will open.

19. Set the attributes as shown below:

20. Press **OK** to apply the formatting.
21. Save and close the page.

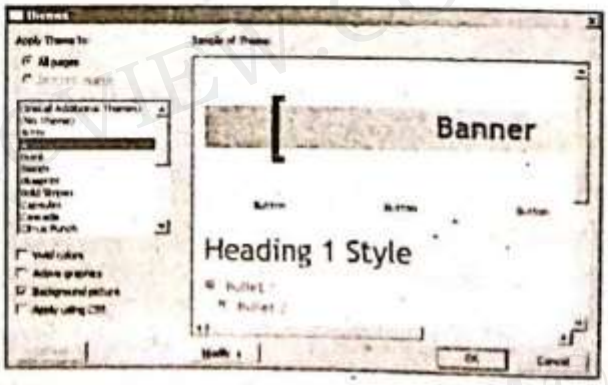
8. Changing Title of 'Products' Page

1. Open the 'Products' page.
2. Click **File > Properties** on the menu bar. **Page Properties** dialog box will open.
3. Type 'Introducing our best products' in **Title** text box.

4. Press **OK** the change the title.
5. Save and close the page.

9. Applying Theme to Both the Pages

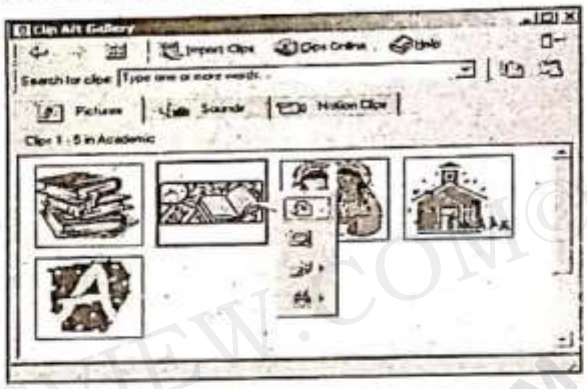
1. Click **Format > Theme** on the menu bar. **Themes** window will open.
2. Select the attributes as shown below:



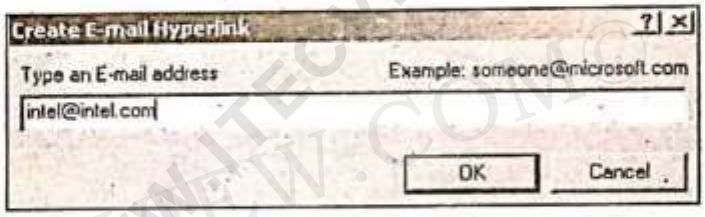
Theme will be applied to whole web.

10. Creating Graphic Email Link in 'Orders' Page

1. Open the 'Orders' page.
2. Place the cursor at the end of page.
3. Click **Insert > Picture > Clip Art. Clip Art Gallery** dialog box will appear.
4. Click **Academic** icon.
5. Click second clip and choose **Insert Clip** button. Clip will be inserted.



6. Now resize as you desire.
7. Select the above inserted picture.
8. Click **Insert > Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
9. Click the icon to open **Create E-mail Hyperlink** dialog box.
10. Type the email address (as shown below) in the **Type an E-mail address** text box.



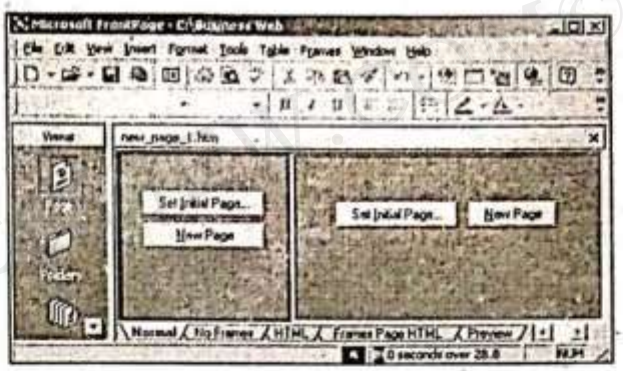
11. Press **OK** button to close the dialog box.
12. The email address will appear in **URL** text box of **Create Hyperlink** dialog box.
13. Press **OK** to create a graphic email link.
14. Save and close the page.

11. Creating Text Hyperlink in 'Products' Page

1. Open the 'Products' page.
2. Type the text 'Orders view' at the end of page.
3. Click **Insert > Hyperlink** on the menu bar. **Create Hyperlink** dialog box will appear.
4. Click icon to browse page on your computer. **Select File** dialog box will open.
5. Click **Look in** drop down arrow to choose location of your web.
6. Now select 'Orders' page from the files & folders list.
7. Press **OK** to create a link to selected page.
8. Save and close the page.

12. Creating Frames Page

1. Select **File > New > Page** on the menu bar. **New** dialog box will appear.
2. Click Frames Pages tab.
3. Choose **Contents** frame page icon.
4. Press **OK**. The new frames page will open in the **Page view**.



5. Click **Set Initial Page...** button for right side frame. **Select File** dialog will appear.
6. Choose 'Products.htm' and press **OK**. The selected page will open in the frame.
7. Now click **New Page** button for left side frame.
8. Type and Format the text in this frame as shown below:

9. Save and close the page.

15.9 Forms

A form is a set of data-entry fields on a page that are processed on a server. The data is sent to the server when a user submits the form by Clicking on a button or an image. Fields are the building blocks of forms. To insert a form into your web page, locate the "Insert" menu and select "Form." To insert a form field select one of six available field types.

<FORM>

The Form element contains properties that are invisible to the user, but are essential in specifying how the data will be processed, transported and encoded.

ACTION

Is the URL of the CGI (Common Gateway Interface) program that is going to accept the data from the form, process it, and send a response back to the browser.

METHOD

GET (default) or POST specifies which method will be used to send form's contents to the web server. The CGI application should be written to accept the data from either method.

NAME

Is a form name used by VB script or JavaScript.

TARGET

Is the target frame where the response page will be displayed to the user.

Form Properties

Some important form properties are as follows:

Send to

If the form is not to be processed by a CGI script, this option can be used to generate a text file. To save the file in a particular directory select browse and choose the directory. You must enter a filename with a .txt extension. To E-mail the results of the form enter an appropriate E-mail address. To control the format of the results select "Options."

Send to other

Select this option if the results of your form are being processed by a script, a discussion group handler or a registration handler. Select "Options" to configure the required parameters for forms processed by scripts and handlers.

Form Name

Enter the name you have chosen for the form here.

Target Frame

When form is submitted, a response is shown to the user to inform that the form has been submitted successfully. The response is given by a separate page. The response page can be shown in a specific frame using this property. The possible options are Page Default, Same Frame, Whole Page, New Window, or Parent Frame.

Options

It is used to configure parameters for either "Send to" or "Send to other."

Advanced

Select "Advanced" to add Hidden fields to your form. Select "Add" to create new hidden fields. "Modify" to modify existing hidden fields and "Remove" to delete existing hidden fields.

Form Elements

Form elements also have properties called "form fields." A form field is where the user fills in or selects information to be submitted. FrontPage provides you with the following fields: One-Line Text Box, Scrolling Text Box, Check Box, Radio Button, Drop-Down Menu, and Push Button. You can insert form fields by Clicking on "Insert" menu, selecting "Form" and selecting the desired field type.

<INPUT> Elements Properties	
TYPE =	Type of INPUT entry field
NAME =	Variable name passed to CGI application
VALUE =	The data associated with the variable name to be passed to the CGI application
CHECKED =	Button/box checked by default
SIZE =	Number of characters in text field
MAXLENGTH =	Maximum number of characters accepted

Field Types

Different types of fields are as follows:

Text Boxes

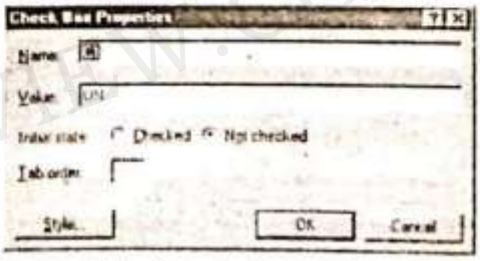
To create a textbox, Click on the Insert Text box button . You can edit the properties of the text box by Right Clicking on the text box and selecting "Form Field Properties."

In the Text Box Properties dialog box, you can modify the following attributes:

- **Name:** The name of the variable to be sent to the CGI application.
- **Initial Value:** Usually left blank, but can have a default value.
- **Width in Characters:** The number of characters the user can enter in the text box. You can also change the size of the text box by dragging the sizing handles.
- **Tab Order:** Controls the order in which the user can move from field to field.
- **Password field:** A text box can be used as a password field for user validation.
- **Validate:** This button displays the Form Field Validation dialog box. The Validation dialog box controls what the user can enter in the text box.

Check Box

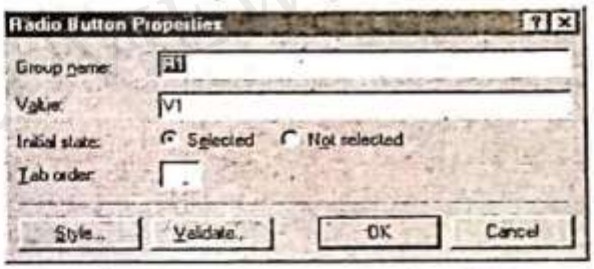
Checkbox is used to select multiple options. symbol is used to insert checkbox.



- **Name:** The name of the variable to be sent to the CGI application.
- **Value:** The default value is on but can be changed to OFF.
- **Initial Status:** The status of whether it will appear checked or unchecked when displayed in the web browser.
- **Tab Order:** Controls the order in which the user can move from field to field.

Radio Button

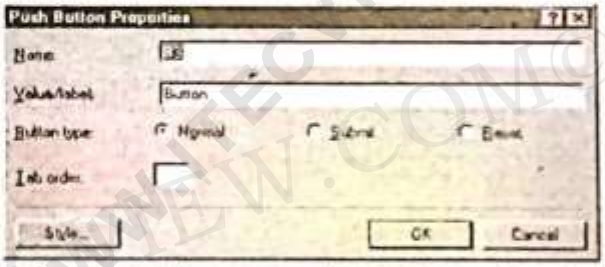
Radio buttons are used to select only one option. To insert a radio button Click on .



- **Name:** The name of the variable to be sent to the CGI application.
- **Value:** The value associated with button. Each radio button must have its own value.
- **Initial Status:** The status of whether it will be selected or not selected when displayed in the web browser.
- **Tab Order:** Controls the order in which the user can move from field to field.
- **Validate:** This button displays the Form Field Validation dialog box. Use it to define the group a set of radio buttons belong to, the value of each button, the initial selection state, and the tap order.

Push Button

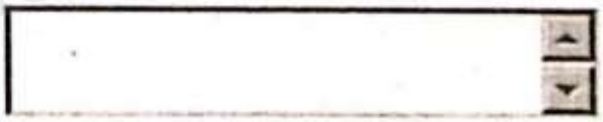
Push Buttons are elements used with either JavaScript or VBScript to cause an action to take place.



- **Name:** The name of the variable sent to the CGI application.
- **Value/Label:** The label of the push button. If you supply a label, it replaces the default label on the push button.
- **Button type:** Select Submit to use the button to submit the form; when you select Submit, the default label is Submit. Select Reset to use the button to reset the form to its initial state; when you select Reset, the default label is Reset. Select Normal if you want to insert a generic button to which you can assign a script; when you select Normal, the default label is Button.
- **Tab Order:** Controls the order in which the user can move from field to field.

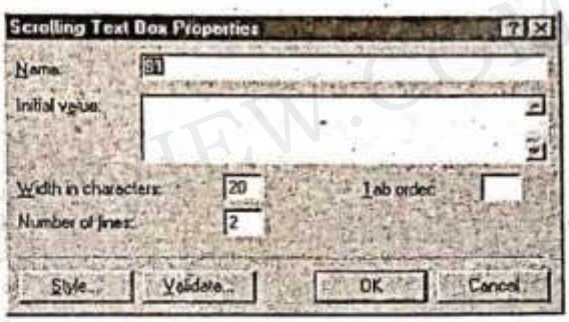
Scrolling Text Box

Allows users to input multiple lines of text. To insert a scrolling text box Click on .



You can edit the properties of the scrolling text box by Right Clicking on the scrolling text box and selecting "Form Field Properties."

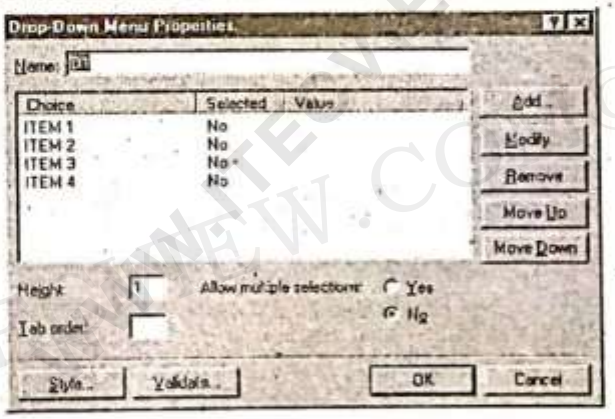
WWW.ITECHVIEW.COM©



- **Name:** The name of the variable to be sent to the CGI application.
- **Initial Value:** Usually left blank, but can have a default value.
- **Width in Characters:** The number of characters the user can enter in the scrolling text box. You can also change the size of the scrolling text box by dragging the sizing handles.
- **Number of Lines:** The number of lines or height of the scrolling text box.
- **Tab Order:** Controls the order in which the user can move from field to field.
- **Validate:** This button displays the Form Field Validation dialog box. The Validation dialog box controls what the user can enter in the scrolling text box.

Drop-Down Menu

The following two are types of <SELECT> </SELECT> elements, where the elements are set differently. Drop-Down Menus contain a list of selections users choose from.



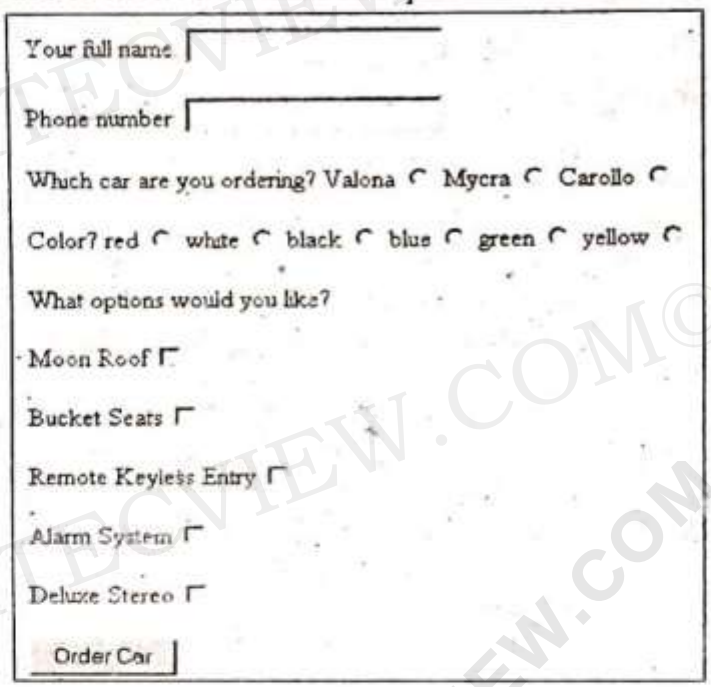
- **Name:** The name of the variable to be sent to the CGI application.
- **Choice:** lists the names of the menu choices associated with this drop-down menu. You can add menu choices by {Clicking} on the "Add" button.

Each choice is designated by <OPTION> </OPTION> tags in HTML.

- **Selected:** Choose whether you want the menu choice on the current row to be selected or not selected when it is first displayed in a Web browser.
- **Value:** Sets the value of the menu choice on the current row.
- **Move Up:** Click Move Up to move the currently selected choice up in the menu.
- **Move Down:** Click Move Down to move currently selected choice down in menu.
- **Height:** Click Height to set the height of the menu. The height of a drop-down menu field controls how the menu appears:
- **Allow multiple selections:** Select Yes to allow more than one menu item to be selected at a time. Select No to permit only one selection at a time.
- **Tab order:** Controls the order in which the user can move from field to field.

Order Form Project

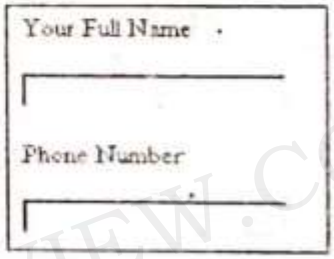
You will create the order form for a car. You will need to provide a choice of options and colors, as well as space to fill in the name and phone number.



Procedure

1. In FrontPage create a new file.
2. Click on Text Input button on Forms Toolbar **OR** choose **Insert > Form > One-Line Text Box**. It will create a blank, 20 character text input box.
3. Type "Your First Name:" as a label for your text input field.
4. Right Click on the Input box and select 'Form Field Properties".
5. In Name field type "NAME" and click OK.
6. Type "Phone number: " in next line and insert a text box with Name "PHONE."
7. Click OK.

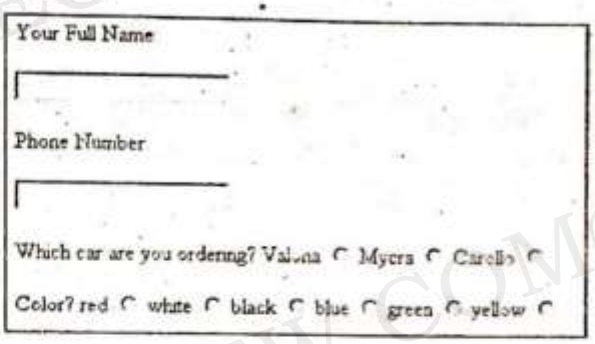
Switch from Normal View to Preview. Your document should look similar to following:



Inserting Radio Buttons

1. Place cursor to the right of last text box field and press ENTER.
2. Type "Which car are you ordering? Valona ".
3. Click on Radio button from **Insert > Form** options.
4. Follow the same procedure for Myrca and Carollo.
5. Right click on each radio button and select **Form Field Properties**.
6. Type "CAR" in **Name** field.
7. In **Value** field, enter "Valona" and click **OK**.

8. Follow the same procedure for radio buttons for other two cars, typing "CAR" in the **Name** field and entering type of car (Valona, Myrca or Carollo) in **Value Box**. Provide same name for all radio buttons so that the user may select only one button.
9. Press Enter and type " Color? ".
10. Create radio buttons for red, white, black, blue, green and yellow. Use "Color" as name of all radio buttons.



Adding Checkboxes

1. Type "What options would you like?" and press ENTER key.
2. Type "Moon Roof."
3. Click on Checkbox button on FORMS menu.
4. Right click on checkbox and select "Form Field Properties."
5. Enter "Moon Roof" in Name Box.
6. Repeat the process to add other options: Bucket Seats, Remote Keyless Entry, Alarm System and Deluxe Stereo.
7. Insert Push Button and right click on it to change its properties so that it is a Submit Button. Enter the VALUE of the Submit Button as "Order Car."
8. Click **OK**.

9. Right click "Order Car" and select "Form Properties"
10. Select **Send to** option and type "mailto:info@itseries.com.pk" in **Email address** box.
11. Save your document and preview it in your browser.

15.10 Cascading Style Sheets in FrontPage

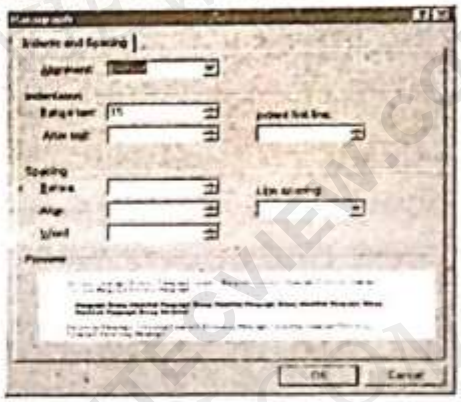
There are three ways you can add Style Information to your web page:

- Inline styles
- Embedded style sheets
- Linked style sheets

15.10.1 Inline Styles

The style sheet properties are applied to content within <BODY> </BODY> tags. For example, a certain paragraph in document may use custom indentation settings by applying inline style sheet.

When you select certain elements or formatting effects, FrontPage applies an inline style. Right Click in a paragraph and choose **Paragraph...** from popup menu. Any settings specified in Indentation or Spacing fields will be written as CSS style commands and specified within the paragraph element as the value of the **Style** attribute.



HTML View

```
<body>

<p>This is a normal Paragraph</p>
<p style="margin-left: 15">This paragraph has a style command applied as an
Inline CSS Style</p>
<p>This is a normal Paragraph</p>

</body>
```

Normal View

This is a normal Paragraph

This paragraph has a style command applied as an Inline CSS Style

This is a normal Paragraph

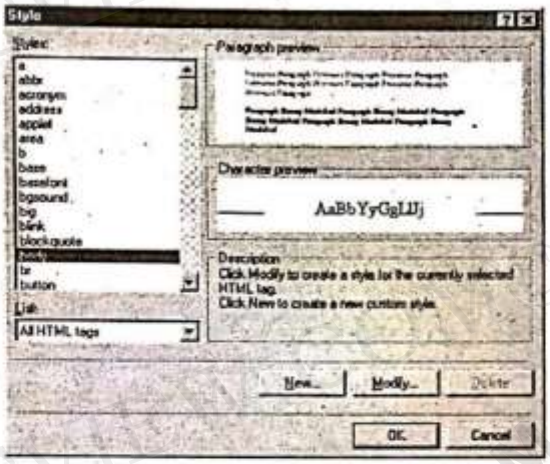
CSS can be used to specify style commands for paragraphs other than indentation and spacing but FrontPage does not provide dialog boxes to insert those commands. To apply additional inline styles to elements, you will have to add them using HTML editor view.An inline style is only valid for the one instance of the element in which it is applied.

15.10.2 Embedded Style Sheets

Embedded Style Sheets use <STYLE> </STYLE> element in the Head of the document. A required attribute of Style element is TYPE attribute. It must be set to "text/css". Embedded style sheet only applies to one document.

Creating an Embedded Style Sheet

1. Switch to **Normal View** in FrontPage Editor.
2. Select **Style...** from **Format** menu.



3. The style dialog box is used to choose a page element from list in "styles:" window.
4. Click on the modify button to launch Modify Style dialog box.
5. To apply a style, Click on **Format** button. A menu will appear with different options for **Font, Paragraph, Border, Numbering** and **Position.**
6. You can apply multiple style commands to as many elements as required. When you Click **OK** button on Style dialog box, FrontPage creates an embedded style sheet.
7. You will need to add the type attribute to the code generated by FrontPage so that the HTML looks as follows:

```
<STYLE TYPE="text/css">
```

HTML View:

```
<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>This is a normal Paragraph</title>

<style>
<!--
body        { background-color: #FFFFFF }
h1          { color: #000000 }
p           { color: #FF0000; font-size: 12pt; font-family: Verdana, Arial }
-->
</style>

</head>

<body>

<p>This is a normal Paragraph</p>
<p style="margin-left: 15">This paragraph has a style
command applied as an Inline CSS Style.</p>

<p>This is a normal Paragraph</p>

</body>

</html>
```

Normal View

This is a normal Paragraph

This paragraph has a style command applied as an Inline CSS Style.

This is a normal Paragraph

The STYLE defined in HEAD section has made all text of same font face and color. In addition, there is an Inline STYLE applied to the second paragraph.

15.10.3 Linked Style Sheets

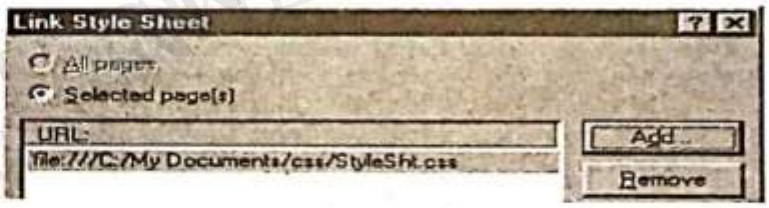
The preferred method of adding styles to a web page is to link a separate file that contains all of the formatting information. A style sheet file is a text file with a file extension of .css. This file can be linked to multiple HTML pages. Multiple style sheets can be linked to one document.

To create a linked style sheet:

1. Switch to Normal View in FrontPage and Click on **File** menu.
2. Point to **New** and select **Page**
3. Select Style Sheets Tab and select the appropriate style.
4. You can now create new style by Clicking on **Format** menu and then **Style** button.
5. In Style Dialogue Box, you can create your own style.
6. To create this file into an external Cascading Style Sheet, you have to save it with an extension of .CSS

To link a Style Sheet

1. Click on the **Format** menu and select **Style Sheet Links.**
2. In **Link Style Sheet** dialog box Click on **Add** button
3. Navigate to the style sheet that you wish to link to or enter the URL and Click **OK.**
4. The style sheet should now be listed in the dialog box; Click **OK.**



Classes and IDs

CSS can be used to associate a CLASS or an ID attribute with a set of style properties and values. It is helpful to format a particular element instance in certain way. In a style sheet, a CLASS will start with a dot and an ID will start with a number symbol:

.classname = CLASS
#idname = ID

Style commands can be attached to a CLASS or ID incidence in the document the same way they are attached to elements.

```
<style>
<!--
.intro { background-color: rgb(128,128,0); color: rgb(255,255,0) }
.sales { background-color: rgb(0,255,0); color: rgb(0,0,0) }
-->
</style>
```

```
<body>
<h2 class="intro">Section 1</h2>
<p class="intro">kjhkj kjh kjh kjh kjhk jhkjhkj hkjh kjh kjh</p>
<h2 class="sales">Section 2</h2>
<p class="sales">kjhkj kjh kjh kjh kjhk jhkjhkj hkjh kjh kjh</p>
</body>
```

Classes

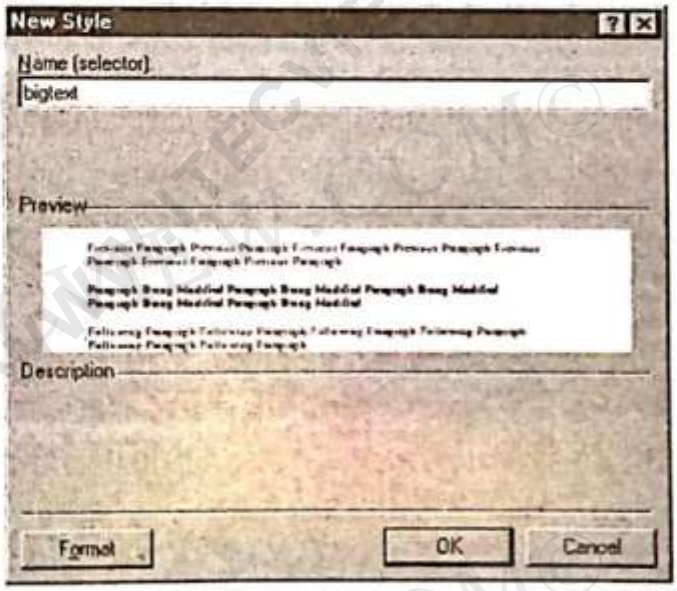
A class is a group of elements that have the same value for their CLASS attribute. To utilize a class, you set CLASS attribute of as many elements as you wish to the same value.

IDs

While many elements can have the same CLASS attribute value, only one element in any document can have a particular ID attribute value. IDs are intended to be unique identifiers for elements in a document.

Defining Classes

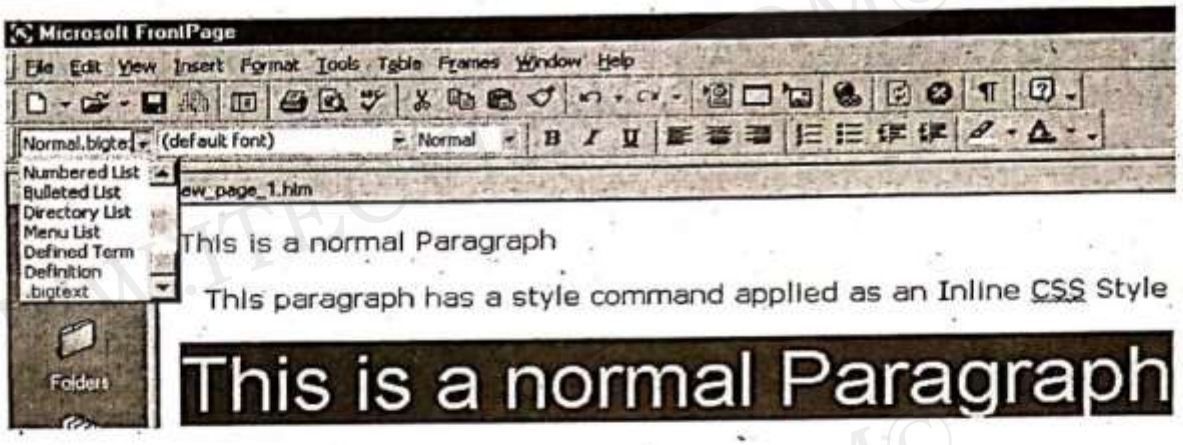
1. Switch to **Normal View** in FrontPage Editor.
2. Select **Style...** from **Format** menu.
3. Select **New**. The **New Style** dialog box will appear.
4. Enter name for new style.



5. To define a style, Click on **Format** button. A menu will appear for **Font, Paragraph, Border, Numbering** and **Position** options.
6. Select desired options and define new settings for the style.
7. To finish this process click **OK**.

Applying New Styles

1. Select the content you want to apply the style to.
2. Click the **Style** dropdown menu.
3. Select the new style from the list.



Practice

Create an inline style sheet that defines 35 indentation for a paragraph.

1. Open a new page in FrontPage.
2. Type a paragraph.
3. Right click in the paragraph and select **Paragraph.**
4. In the indentation section enter a setting of 35 in the "Before text" field
5. Click **OK**. The style will be applied to the paragraph.
6. Save the file and preview in the browser.

Practice

Write an embedded style sheet that defines green color for <H2> tag and red color, 12pt size and Garamond, serif font style for paragraphs.

1. Open a new file in FrontPage.
2. Go to **Format** menu and select **Style** from list of options.
3. Select H2 element from styles list.
4. Click Modify.
5. Select **Format** then Choose **Font.**
6. Click on color dropdown.
7. Choose the green color.
8. Click **OK.**
9. Click **OK.**
10. In the **List** dropdown select **All HTML Tags.**
11. Select P element from the styles list.
12. Click **Modify** then **Format** then Choose **Font.**
13. Set the color to red.
14. Set the size to 12pt.
15. In **Font** field type "Garamond, serif".
16. Click OK on all dialog boxes.
17. Save and preview the page in browser. Your code should look similar to this:

```
<html>

<head>
<title>FrontPage 98 - Cascading Style Sheets</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<style>
<!--
h2          { color: #008000 }
p           { color: #FF0000; font-size: 12pt; font-family: Garamond, serif }
-->
</style>
</head>

<body>

<h2>This H2 is green</h2>

<p>The text in this paragraph is red, size 12 points and belongs to the Garamond, serif
font family.</p>

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

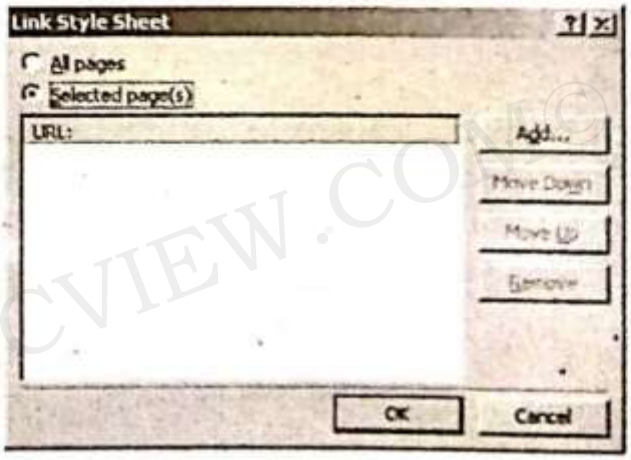
Practice

Write a CSS file to define green color for paragraphs and link it to a web page.

1. Open Microsoft FrontPage.
2. Click File > New > Page or Web.
3. Select a **Normal Style Sheet** template. A blank CSS file will open.
4. Type the following code:

```
P
{
  color: green;
}
```

5. Save the file as "Test.css".
6. Open a new web page.
7. Click **Format > Style Sheet Links...** The following dialog box will appear:



8. Click **Add** button. The **Select Style Sheet** dialog box will appear.
9. Select "Test.css" file and click **OK** button. The file will be linked to the web page.
10. Write any paragraph. The paragraph text will appear in green color.

Part - 7

Active Server Pages

Includes:

Chapter 16	Web Applications with ASP
Chapter 17	Programming Constructs in ASP
Chapter 18	ASP Objects
Chapter 19	Microsoft Access and SQL
Chapter 20	Database Connectivity in ASP
Chapter 21	ASP Projects

WWW.ITECVIEW.COM©

Web Applications with ASP

16.1 Web Applications

A Web application is different from a traditional client-server application. In client-server application, the client maintains a connection with the server while the application is being executed. The client sends and receives data from the server. It also checks to see if the connection is active at regular intervals. If the server goes down, the client sends an error message to the user before exiting.

On the other hand, a Web-enabled application uses Internet which is a connectionless network. Secondly, Web pages are stateless. When a client makes a request for a page, the Web server sends the page to the client. The server does not keep track of the requests even if the request is for the same page.

In a typical two-tier client-server application, the application connects to a data source on the server. In a three-tier system, data source can be accessed indirectly. The application accesses a server program on the Intranet or Internet. The server program connects to the data source and processes the required data. The data is then returned to the client application. The client does not connect directly to the data source.

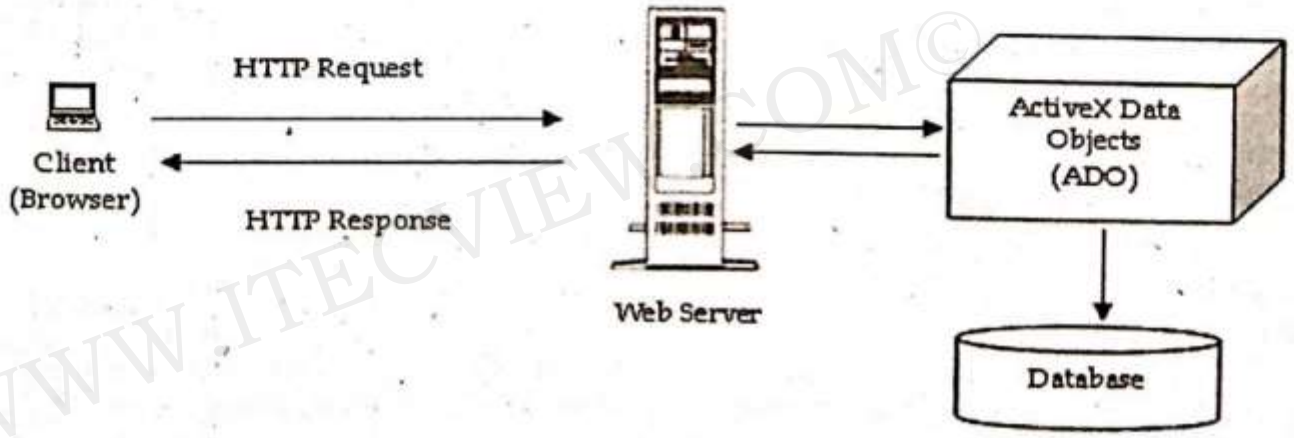


Figure: Accessing a database

WWW.ITECVIEW.COM©

16.1.1 Web Application Requirements

The architecture and requirements to develop a Web-enabled application is also different from traditional client-server applications. A Web-enabled application cannot work as isolated application. It requires several network and application components to deliver the desired results to the user.

A client machine is connected to a Proxy Server within an organisation. The Proxy Server contains firewalls required to filter information that is being sent to and received from the Internet or corporate Intranet. A Web server provides access to other resources, a database server and other data stores.

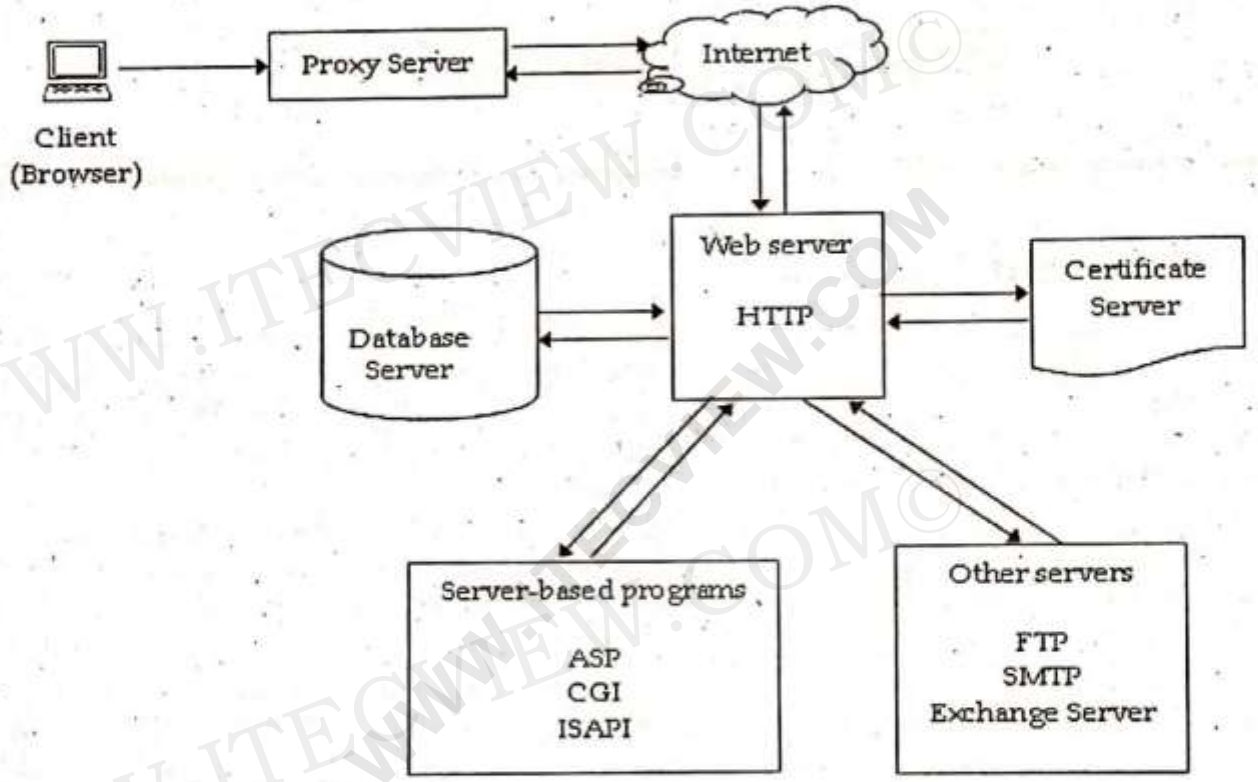


Figure: The typical infrastructure for a Web-enabled application

Proxy Server

Proxy Server is a software element used to pass network traffic between networks within an organization. Proxy Server contains firewalls required to filter information that is being sent to and received from the Internet or Intranet. Typically, the client browser sends a request for a page to proxy server. Proxy server forwards the request to Web server. The Web server processes the request and sends the response to proxy server. The proxy server then sends the response to client browser.

Proxy server is typically used for security purposes. It can be configured to monitor requests sent to and data received from the Internet or intranet.

Database Servers

Most of the Web applications use data. The application is required to store, retrieve and maintain data in a database. Typically, the user can search through the database of products and services being offered.

A **database server** is used to store data in database. Users can access and manipulate data. A Web application can provide the user with the interface to the database. For example, users can browse through the catalogue of an online bookstore before placing the order for a book. Once an order has been placed, the stocks are updated to show current availability.

Certificate Servers

The Internet is a network of networks that is open to all users. Many organizations use the Internet as an extension of their private networks. They provide access to internal information through the Internet. The users may be their own employees or even external users searching for information about an organization.

This access can create security problems. Organizations have to ensure that only authorised users can access internal information. Similarly, it is very important for the users to be sure that the information they provide is secure. For example, no user can provide his credit card to an unauthorised person. Certificate servers are used to ensure data integrity of the Web site or the application. Data integrity involves two phases:

1. **Client and Server Authentication:** It can be maintained through the use of digital certificates. These certificates are maintained by **certificate server.** It ensures that only authorized users are allowed to access a Web site or an application.
2. **Data Integrity in Transit:** Data that is being transferred over the Internet must be secured to prevent unauthorized access. The certificate server uses digital certificates with client and server encryption techniques. For example, **Secure Socket Layers (SSL)** can be used to ensure data integrity.

16.2 Web Application Development Cycle

The phases of web application development are as follows:

1. Planning the Web Application

In this stage, the functionality of the web application is planned. You should identify the audience of the application and where it will be deployed. Users may access the application through Intranet, the Internet or extranet. It helps in specifying the purpose of the Web application. Once the purpose is identified, the contents of application can be defined. Finally, the resources for developing the application is also identified.

2. Designing the Web Application

A Web application may consist of many projects. It may contain ActiveX components that to be developed and used in the project. In this phase, you create the prototype of web application. The basic theme and layout of the application defined. The HTML or ASP pages for the web application are identified. It is also determined which type of scripts should be used. **Visual InterDev Site Diagram** can be used to plan the structure of web application and define the navigational scheme for the project.

3. Constructing and Testing the Components of Web Application

In this phase, application components are created such as HTML pages, ASP documents, ActiveX components etc. **Visual InterDev** has an integrated debugger to debug errors that may occur as the application is being developed and tested.

4. Web Application Deployment

In this phase, various components of the application are deployed on the Web server. Users now have access to the web application.

5. Production and Maintenance

In this phase, the contents of web application is updated at regular intervals. The web application can also be upgraded if required.

16.3 Client Server Model

The **client/server system** consists of two components called **clients** and **servers**. The clients are network nodes that request for services from the servers. Servers act as service providers to fulfill the client's request. The client/server model has been more popular due to its simple architecture.

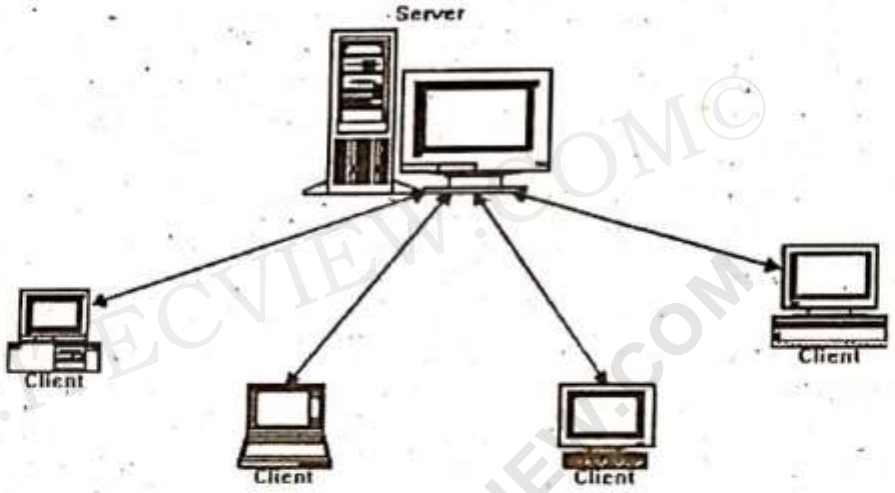


Figure: Client/server Model

16.3.1 Components of Client/Server Model

The client server model consists of three groups of components. These components are broadly classified as Presentation, Logic and Data Access.

- **Presentation** components are used to handle I/O devices, data validation and data presentation.
- **Logic** components are used to implement business policies, logics and specific procedures.
- **Data access** components are used to store and retrieve information from the storage.

Different architectural models of a client/server-based network as follows:

16.3.2 Single-tier

A mainframe or Unix server with its dumb terminals is an example of one-tier client sever architecture. The clients can accept request and display the required results. They have no processing power and the server performs all calculations and processing. In this model, all the three components of a client/server model are placed on the server.

16.3.3 Two-tier

In two-tier model, the client shares some processing load with the server. The clients can request services directly from the database servers. The data resides completely on the database server and the business logic resides on both the server and the client. It exists in the form of simple calculations, computations and presentation. The business logic is built in the server in the form of stored procedures.

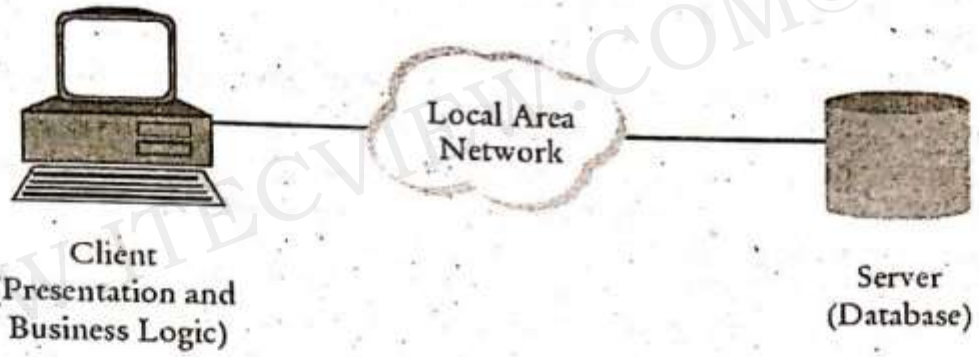


Figure: Two-Tier Client/Server Model

The processing logic in the two-tier architecture is distributed in two tiers or layers. Suppose the values for **Quantity** and **Rate** columns are retrieved from database and total amount is required. This value is not stored in the database. The client can compute the value of **Amount** before displaying result.

16.3.4 Three-tier

Three-tier architecture consists of three tiers or layers called **Client, Database Server** and **Application Server**. The following figure shows a three-tier client/server model. In this model, the Client requests for service from the application server through the standard protocols such as HTTP. The application server interacts with database server using standard data access protocols like JDBC and ODBC.

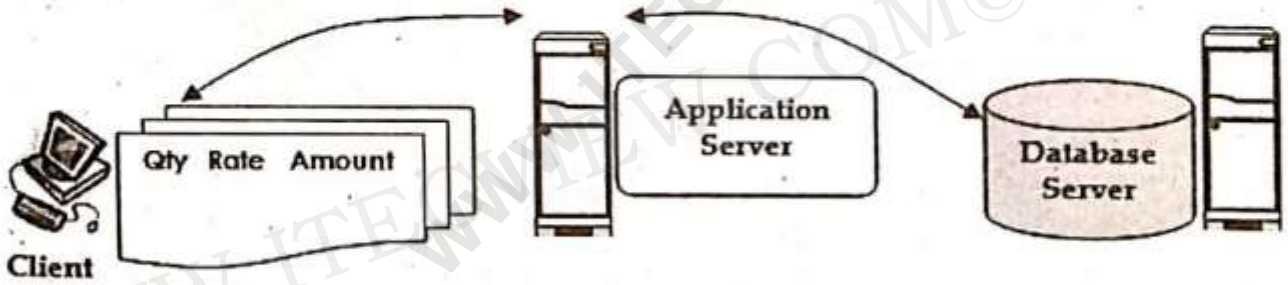


Figure: Three-tier Client/Server Model

The advantages of the three-tiered architecture are as follows:

- The client is independent of the data and the network. The location of data and the number of servers in the network is not known to the client.
- In this model, only the required amount of data is transmitted to client by application layer. The client receives processed data in the required format. The client performs very little presentation formatting.
- The user interface can be changed without affecting the application methods or data. Similarly, the business logic in the application can be changed without affecting data or user interface.
- The database queries are written in standard SQL that is platform independent.
- The database connections can be shared by many users because the individual users are not directly connected to databases. The connections to database server always pass through the application server.
- The procedures that contain the business logic are stored in the application server and not in the database. It simplifies the process of modifying the business logic.

16.3.5 N-tier

The current configuration client/server model can have any number of tiers. The application is divided into as many components as required.

Suppose an invoice printing application is developed. The total invoice amount has to be written in words in Invoice. A component that converts numbers to words can be created separately. This component can be used by other applications also. One application server in the network can store similar utilities to be used by other applications. When a user requests for invoice printing, the user enters the details in the front-end application provided by the client. The information will be stored in a database server. The amount conversion from numbers to words is performed by the component stored in the application server. The fully formatted invoice can be passed on to a print server for printing. The following figure shows a network with n-tier architecture.

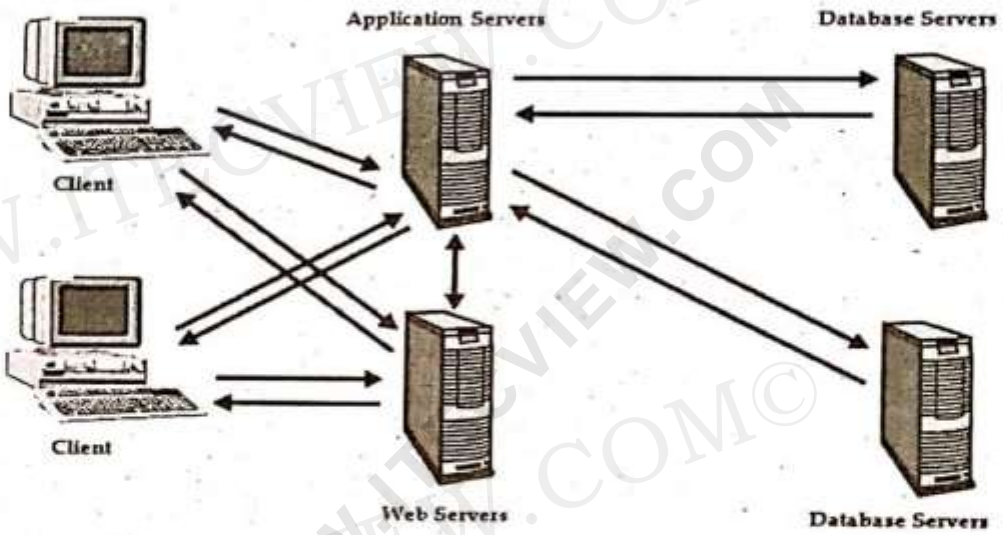


Figure: N-tier Client/Server Model

A server can perform any number of roles in n-tier architecture. It can be a database server as well as as an application server. It can also be a print server. A server can request for a service from another server. In this situation, the requesting server acts as client.

The n-tier architecture is completely flexible. It also allows the use of two-tier or three-tier applications in the network. These applications can work in the network with other n-tier applications.

16.4 Static and Dynamic Pages

A web application may contain two types of web pages. These are as follows:

Static Web Pages

A **static web page** is an HTML document that stored in a file and does not change in response to user input. HTML is the language that is used to created states web pages for a web application. Static web pages are processed as follows:

- A web browser requests a page from a web server by sending the server a message known as **HTTP request**. For a static web page, the request includes the name of HTML file being requested.
- A web server replies to the request by sending a message known as **HTTP response** back to the browser. For a static web page, the response includes HTML document.

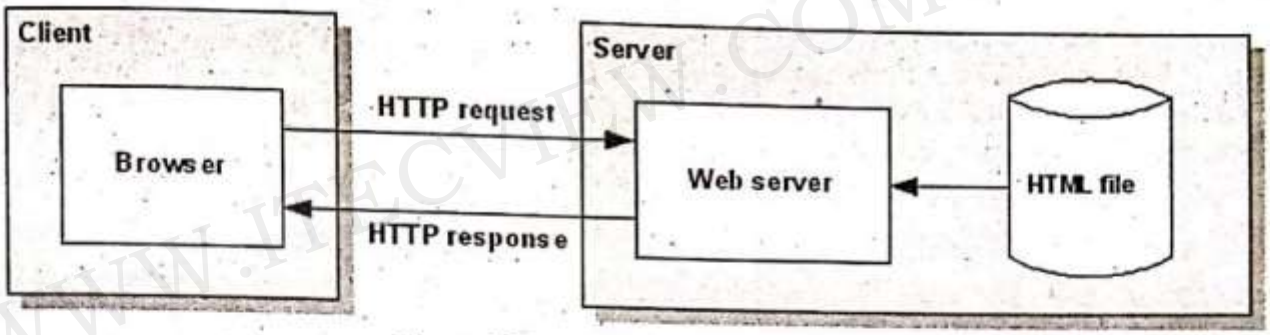


Figure: Processing of Static Web Pages

Dynamic Web Pages

A **dynamic web page** is an HTML document that is generated by a web application. Mostly, the web page changes according to the parameters are sent to the web application by the web browser. Dynamic web pages are processed as follows:

- When a web server receives a request for a dynamic web page, the server passes the request to the web application.
- The application generates a response, which is usually an HTML document, and returns it to the web server.
- The web server wraps the generated HTML document in an HTTP response and sends it back to the browser.
- The browser does not know whether the HTML was retrieved from a static HTML file or was dynamically generated by the web application. The browser displays the HTML document that is returned.

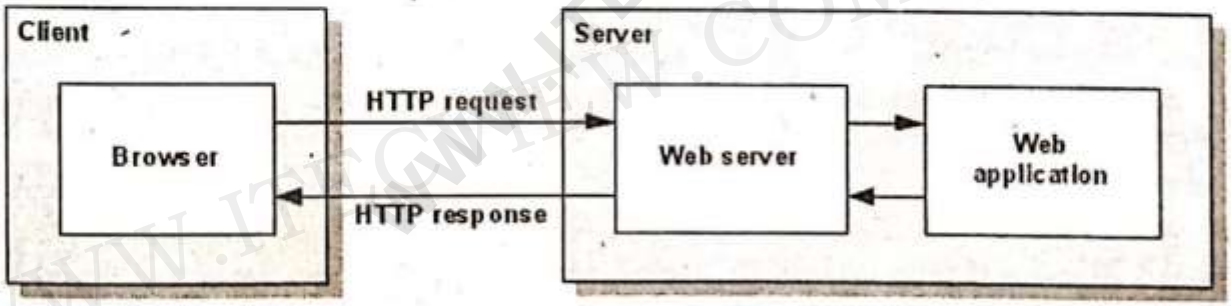


Figure: Processing of Dynamic Web Pages

16.5 Active Server Pages

Active Server Pages (ASP) is a technology that is used to create **dynamic** web pages. It was developed by Micrsofot Corporation. It is an open and compile-free application environment. It is used to combine HTML, scripts, and reusable ActiveX server components to create powerful **Web-based applications.**

An ASP document has an **.asp** filename extension. It may contain HTML, client-side script, server-side script, ASP objects, applets and ActiveX components etc. The server-based code is identified by <% opening tags and %> closing tags.

When a user types an address in the Address box of browser or clicks a link, a request is communicated to a web server to send a file to the web browser. If the file is a normal HTML file, the web server sends the file to the client computer. The browser on client computer interprets the HTML code and displays it.

The following steps are involved:

1. The user types an address in Address bar or clicks a link.
2. The request for a file is sent from client to the server computer.
3. The server finds the file.
4. The server sends the HTML code to the client computer.
5. The browser on client computer interprets the HTML code and displays it.

The HTML file may also contain server-side scripts. **Server-side scripts** are special commands in Web pages that are processed on the server before they are sent to the web browser. The browser on the client computer then displays the contents of the HTML file. The following steps are involved:

1. The user types an address in Address bar or clicks a link.
2. The request for a file is sent from client to the server computer.
3. The server finds the file.
4. The server processes the server-side scripts and generates HTML code.
5. The server sends the HTML code to the client computer.
6. The browser on client computer interprets the HTML code and displays it.

16.5.1 Advantages of ASP

Some important advantages of ASP are as follows:

1. Dynamic Contents

ASP is used to create a Web page with dynamic content. The Web server processes the code and generates HTML depending on the request made by the user. For example, an Active Server Page can display a welcome message based on the time of day.

2. Platform Independent

An ASP sends only ASCII text to a client browser by default. Any browser running on any operating system can access the applications embedded in the ASP script.

3. Security

ASP script is stored an processed on the server. The server processes ASP script, generates HTML and then send HTML code to the browser. The user can view the results of ASP script but cannot see the source code. The user cannot make any changes to the script.

As a developer, you can make changes to the .asp file on the server and save the changes to the file. The next time the page is loaded, the script will be automatically compiled. The reason being, ASP technology is built into all Microsoft Web servers.

16.5.2 Requirements for ASP

The following things are required to run ASP:

1. Internet Explorer

Internet Explorer 4.01 or later is required to run ASP pages.

2. Web Server

A web server is required to run ASP pages on the computer. The web server provides an environment that is similar to the Internet. It provides an easier way to test the web pages without connecting to the Internet. Different types of web servers are available. Most important web servers are Personal Web Server (PWS), Inetnet Information Service (IIS) and Apache Web Server etc.

16.6 Personal Web Server (PWS)

Personal Web Server is a free ASP web server for Win95/98 or NT Workstation. PWS is available as a free download from Microsoft. There are differences between PWS for NT Workstation and PWS for Win95/98. FTP service is not available in Win95/98 version.

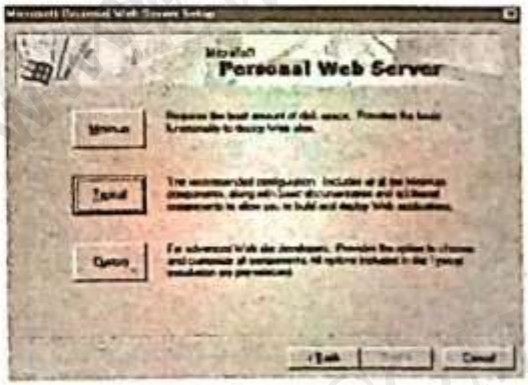
16.6.1 Installing PWS from Windows 98 CD

The Windows 98 CD includes PWS. Microsoft recommends at least a 90 MHz Pentium processor with 20-32 MB RAM, 40MB free disk space and a Super VGA monitor.

The following steps are performed to start the installation:

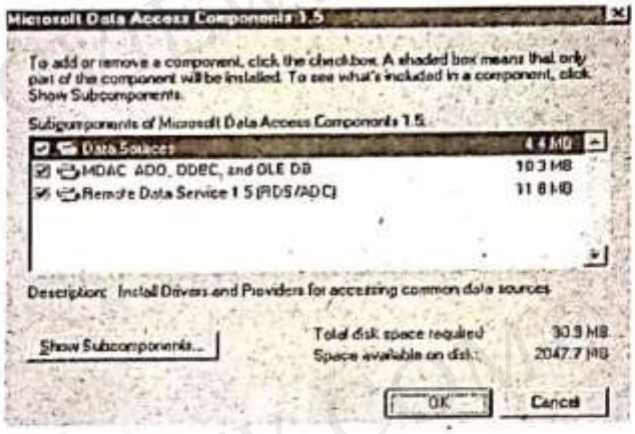
1. Insert the Windows 98 CD in your CD-ROM drive.
2. Click the **Start** button and then click **Run**.
3. In the dialogue box type X:\add-ons\pws\setup.exe (substitute X with appropriate letter) then click **OK**. The PWS setup window will appear as follows:

4. Click **Next** button. The next screen will appear as follows:



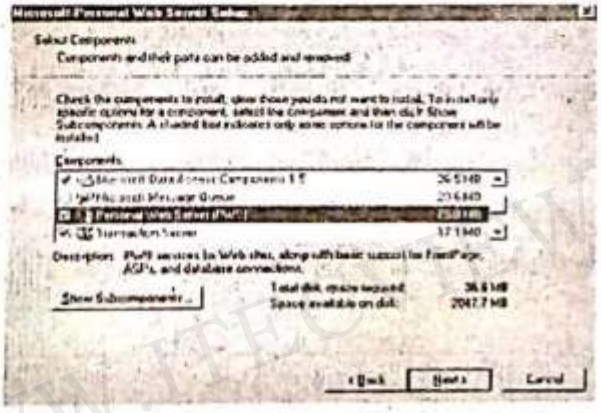
5. Click **Custom** button. The following screen will appear:

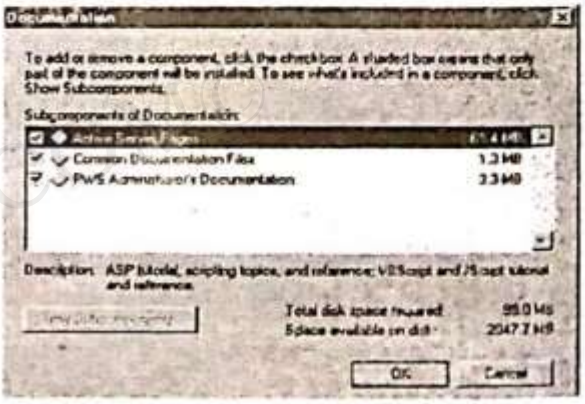
6. Double Click on **Microsoft Data Access Components**..The following window will appear:



7. Double click on **MDAC: ADO, ODBC, and OLE DB**. The following window will appear:

8. Check **ADO documentation** to be installed.
9. Click **OK** to close the window and **OK** in the second window to close it as well.
10. Double click **PWS** and then double click Documentation and checkmark **Active Server Pages** in the window that appears.





11. Close the two windows by clicking **OK**.
12. Click **NEXT** in the PWS Setup Window.
13. Notice the default Web publishing home directory, x:\Inetpub\wwwroot where"X" is a drive letter. Click **Next** to accept the default.

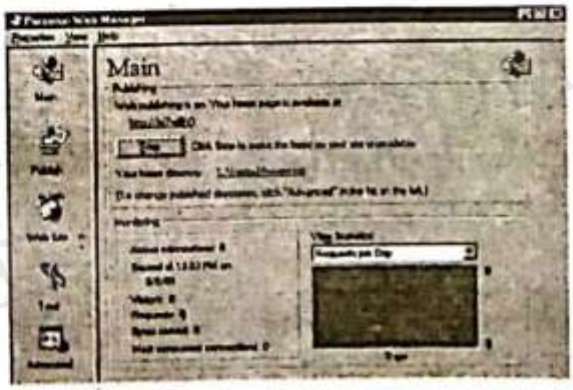
14. Leave **MTS (Microsoft Transaction Server)** install folder to default value and click the **Next** button.

15. PWS set up will complete the installation. When it is finished, click **Finish** button.
16. Restart your computer.

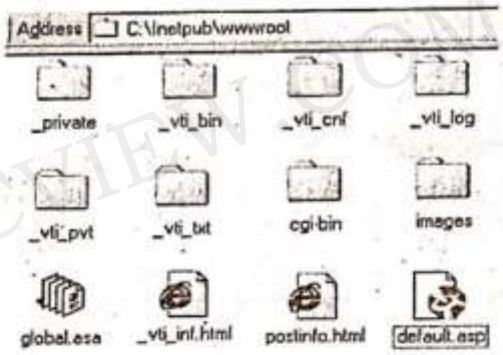
Using Personal Web Server

When the personal web server is installed, its icon appear in the system as follows:

Double clicking on the **PWS** icon will bring up the "Personal Web Manager" window:



Clicking on the **Your home directory** link (typically C:\Inetpub\wwwroot) opens your home directory and displays the files stored there by default.



All **.ASP** files must be stored in this directory. PWS creates an ASP file in the directory named "**default.asp**". It is the **PWS** default web page. If you click "**Start > Run**" type in **http://localhost/default.asp** and click **OK**, this default page will open in your web browser.

localhost

Any web page available on the Internet is actually stored on a computer. A user connects to a remote computer using URL. For example,

`http://www.google.com/`

The above address refers to a computer called **host**. www.google.com is called the hostname of that computer.

localhost is a special hostname. It always refers to the personal computer on which web server is installed. It refers to the directory C:\Inetpub\wwwroot.

16.7 IIS

IIS stands for **Internet Information Services.** It was formerly called **Internet Information Server.** It is a web server application and set of modules created by Microsoft to be used with Microsoft Windows. It is the second most popular web.server in the world after Apache HTTP Server. IIS 7 supports different protocols such as FTP, FTPS, SMTP, NNTP, and HTTP/HTTPS.

16.7.1 IIS Version

- IIS 1.0, Windows NT 3.51 available as a free add-on

- IIS 2.0, Windows NT 4.0
- IIS 3.0, Windows NT 4.0 Service Pack 3
- IIS 4.0, Windows NT 4.0 Option Pack
- IIS 5.0, Windows 2000
- IIS 5.1, Windows XP Professional, Windows XP Media Center Edition
- IIS 6.0, Windows Server 2003 and Windows XP Professional x64 Edition
- IIS 7.0, Windows Server 2008 and Windows Vista (Home Premium, Business, Enterprise, Ultimate Editions)
- IIS 7.5, Windows Server 2008 R2 and Windows 7

IIS is not automatically installed with the installation of Windows. It can be selected from the list of optional features. It is available in all editions of Windows Vista and Windows 7. However, some features are not supported on client versions of Windows.

16.7.2 IIS Features

IIS 7 is developed on a modular architecture. Modules or extensions can be installed or uninstalled individually according to the required functionality. IIS 7 includes native modules as part of the full installation. These modules are individual features used by the server to process requests. These modules include the following:

- **HTTP modules:** It is used to perform the tasks related to HTTP in the request processing pipeline such as responding to information and inquiries sent in client headers, returning HTTP errors and redirecting requests.
- **Security modules:** It is used to perform the tasks related to security in the request-processing pipeline such as specifying authentication schemes, performing URL authorization and filtering requests.
- **Content modules:** It is used to perform the tasks related to contents in the request-processing pipeline such as processing requests for static files, returning a default page when a client does not specify a resource in a request and listing the contents of a directory.
- **Compression modules:** It is used to perform the tasks related to compression in the request-processing pipeline such as compressing responses and performing pre-compression of static content.
- **Caching modules:** It is used to perform the tasks related to caching in the request-processing pipeline such as storing processed information in memory on the server and using cached content in subsequent requests for the same resource.
- **Logging and Diagnostics modules:** It is used to perform the tasks related to logging and diagnostics in the request-processing pipeline such as passing information and processing status to HTTP.sys for logging, reporting events and tracking requests currently executing.

16.7.3 IIS Architecture

Internet Information Services (IIS) 7 provides a new request-processing architecture. It includes the following:

- A new service called **Windows Process Activation Service (WAS)** that enables sites to use protocols other than HTTP and HTTPS.
- A Web server engine that can be customized by adding or removing modules.
- A new approach to processing requests, integrating the request-processing pipelines from IIS and ASP.NET.

IIS 7 contains several components that perform important functions for the application and Web server roles in Windows Server 2008 (IIS 7.0) and Windows Server 2008 R2 (IIS 7.5). Each component has responsibilities, such as listening for requests made to the server, managing processes and reading configuration files. These components include protocol listeners such as HTTP.sys and services such as **World Wide Web Publishing Service (WWW service)** and **Windows Process Activation Service (WAS)** etc.

16.7.4 Protocol Listeners

Protocol listeners receive protocol-specific requests, send them to IIS for processing and return responses to clients. For example, HTTP listener **HTTP.sys** picks up the request and sends it to IIS for processing when a client browser requests a Web page from the Internet. **HTTP.sys** returns a response to the client browser when IIS processes the request.

By default, IIS 7 provides HTTP.sys as the protocol listener that listens HTTP and HTTPS requests. HTTP.sys was introduced in IIS 6.0 as HTTP-specific protocol listener for HTTP requests. It is an HTTP listener in IIS 7 and also includes support for **Secure Sockets Layer (SSL)**.

Hypertext Transfer Protocol Stack (HTTP.sys)

HTTP listener is part of the networking subsystem of Windows operating systems. It is implemented as a kernel-mode device driver called **HTTP protocol stack (HTTP.sys)**. HTTP.sys listens HTTP requests from the network, passes the requests to IIS for processing and returns processed responses to client browsers. HTTP.sys provides following benefits:

- **Kernel-mode caching:** Requests for cached responses are served without switching to user mode.
- **Kernel-mode request queuing:** Requests cause less overhead in context switching because the kernel forwards requests directly to correct worker process. If no worker process is available to accept a request, it is held in the kernel-mode request queue holds until a worker process picks it up.

16.7.5 World Wide Web Publishing Service (WWW service)

- The functionality previously handled by World Wide Web Publishing Service is divided between two services in IIS 7. These services are **WWW Service** and a new service **Windows Process Activation Service (WAS)**.
- WWW service no longer manages worker processes in IIS 7. The WWW Service is the listener adapter for HTTP listener and HTTP.sys. As the listener adapter, the WWW Service is primarily responsible for configuring HTTP.sys, updating HTTP.sys when configuration changes and notifying WAS when a request enters the request queue.
- WWW Service continues to collect counters for Web sites. The performance counters remain part of WWW Service. They are HTTP specific and do not apply to WAS.

16.7.6 Windows Process Activation Service (WAS)

- In IIS 7, Windows Process Activation Service (WAS) manages application pool configuration and worker processes instead of WWW Service. It enables the user to use the same configuration and process model for HTTP and non-HTTP sites.
- Additionally, the user can run WAS without WWW Service if HTTP functionality is not required.

16.7.7 HTTP Request Processing in IIS 7

IIS 7 has a similar HTTP request-processing flow as IIS 6.0. The diagrams in this section provide an overview of an HTTP request in process.

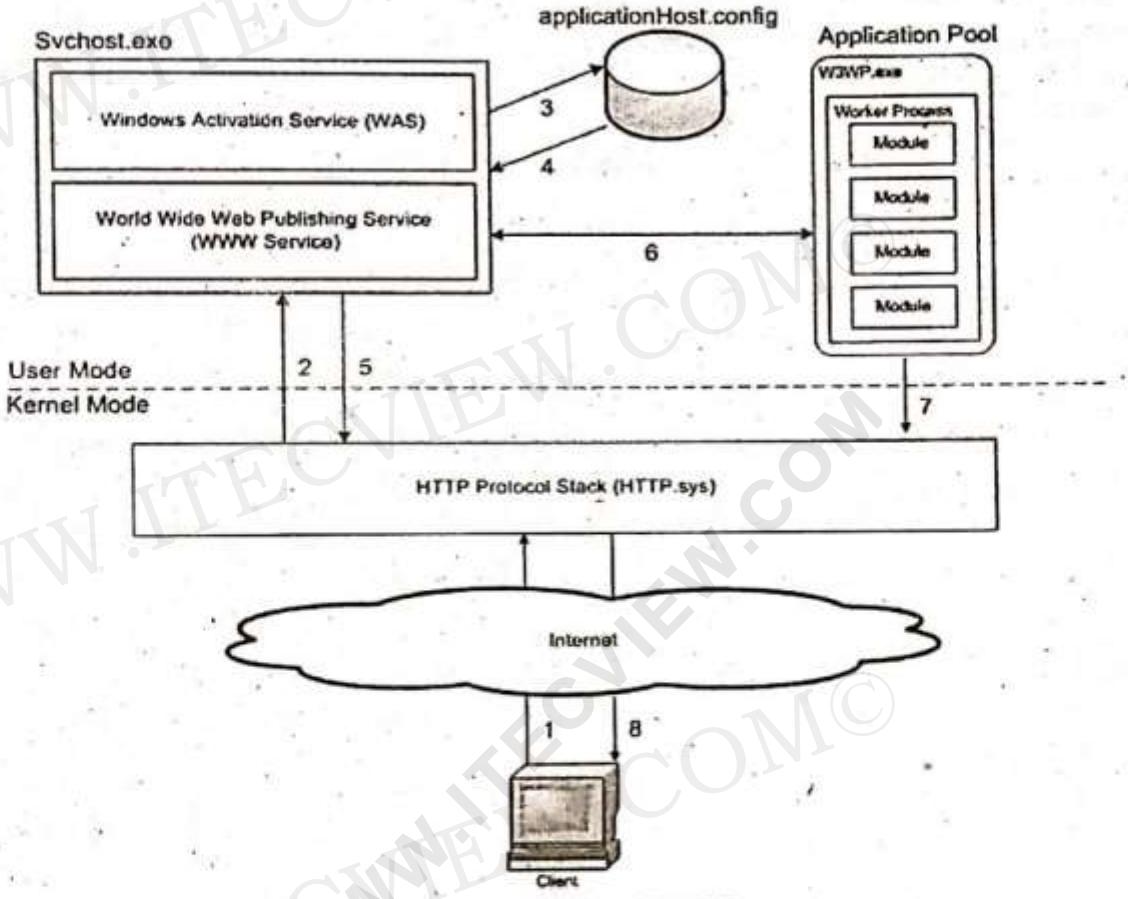


Figure: Overview of an HTTP Request

The following list describes the request-processing flow shown in the above figure:

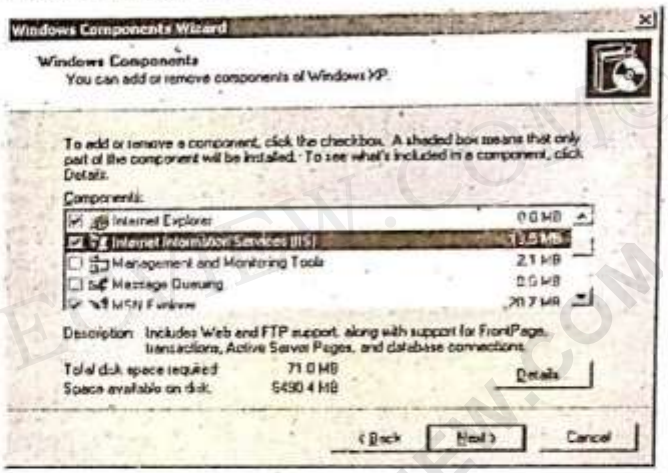
1. **HTTP.sys** intercepts the request when a client browser initiates an HTTP request for a resource on Web server.
2. **HTTP.sys** contacts WAS to obtain information from configuration store.
3. WAS requests configuration information from configuration store **applicationHost.config**.
4. WWW Service receives configuration information such as application pool and site configuration.
5. WWW Service uses configuration information to configure **HTTP.sys**.
6. WAS starts a worker process for application pool to which the request was made.
7. The worker process processes the request and returns a response to **HTTP.sys**.
8. The client receives a response.

In a worker process, an HTTP request passes through several ordered steps or events in the Web Server Core. At each event, a native module processes part of the request such as authenticating the user or adding information to the event log. If a request requires a managed module, the native **ManagedEngine** module creates an **AppDomain**. The managed module can perform the necessary processing such as authenticating a user with Forms authentication in **AppDomain**. The response is returned to HTTP.sys when the request passes through all of the events in the Web Server Core.

16.7.8 Installing IIS on Windows XP

Windows XP Professional uses **Internet Information Server (IIS).** It is available for free on the Windows XP Pro installation CD.

1. Insert the Windows XP Professional CD-ROM into CD Drive.
2. Double click on Add/Remove Program icon in control panel.
3. Click on **Open Add/Remove Windows Components.** Check Internet Information Services (IIS) checkbox as follows:



4. Click **Next** button. The installation process will start as follows:
5. Click **Finish** button when the installation process completes.
6. Type *http://localhost* in the address bar of web browser and press **Enter.** The Home page of localhost will open as follows:

The default web directory to store ASP files in is *'C:\Inetpub\wwwroot'*. You can create subdirectories in this folder.

16.7.9 Installing IIS on Windows Vista

The following steps are performed to install IIS on Windows Vista:

1. Open the Control Panel from the Start menu
2. Double-click **Programs and Features.**
3. Click **Turn Windows features on or off** on the left side.
4. Select the check box for **Internet Information Services (IIS)** and click **OK.**

Note: All patches for bugs and security problems should be installed after installing IIS by running **Windows Update.**

16.7.10 Installing IIS on Windows 7

The following steps are performed to install IIS on Windows 7:

1. Click **Start > Control Panel.**
2. Click **Programs** and then click **Turn Windows features on or off.**
3. Click **Internet Information Services** in **Windows Features** dialog box.
4. Click **OK.**

16.7.11 Installing IIS on Windows Server 2003

The following steps are performed to install IIS on Windows Server 2003:

1. The **Manage Your Server wizard** appears when Windows Server 2003 is started.
2. Go to **Administrative Tools** and select **Manage Your Server** if the wizard is not displayed
3. Click **Add or Remove a Role** in the wizard and click **Next.**
4. Select **Custom Configuration** and click **Next.**
5. Select **Application Server role** and click **Next.**
6. Select **Enable ASP.NET** and click **Next.** The wizard may ask for **Server 2003 CD.**
7. Insert the CD and let it run until it is finished.
8. Click **Finish** button. The wizard will show the **Application Server** role installed.
9. Click on **Manage This Application Server** to bring up the **Application Server Management Console (MMC)**
10. Expand **Internet Information Services (IIS) Manager.**
11. Expand your server and then the **Web Sites** folder. The **Default Web Site** will appear and it should be (Stopped).
12. Click on **Web Service Extensions** folder in **Internet Information Services (IIS) Manager.**
13. **Active Server Pages are Prohibited** will appear. It is default configuration of IIS 6.
14. Highlight **Active Server Pages** and click **Allow** button. ASP will be now active.

16.8 Embedding ASP in HTML

ASP script is embedded in HTML code using special tags. The tag <% indicates the beginning of ASP code and the tag %> indicates the end of ASP code. The code written between these tags are processed by the server.

Practice 16.1

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>My First ASP Page</TITLE>
</HEAD>
<BODY>
<%
        Response.Write "Hello Active Server Pages!"
%>
</BODY>
</HTML>
```

2. Save the file as "First.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory. (Create **ASPFiles** directory in **wwwroot** folder. It is not created available by default).
3. Type **http://localhost/ASPFiles/First.asp** in web browser and press **Enter** key. The page will appear as follows:

In the above example, a single line of ASP code is enclosed in <% and %>. The line uese a method Response.Write to display a message Hello Active Server Pages! in the web page. The line can also be written as follows:

```
<% = "Hello Active Server Pages!" %>
```

The above line also displays the same message on the web page. The use of = sign after <% is similar to Response.Write method.

16.9 Variables

The variables in ASP are declared using **Dim** keyword. Dim stands for **dimension**. The syntax for declaring variable is as follows:

```
Dim VarName
```

Here, **VarName** is the name of the variable. In ASP, all variables are **variants**. It means that the type of variable is determined automatically by the runtime interpreter.

By default, variable declaration is not necessary in ASP. A variable can be used directly without declaring it. But it may create some problems in the script. It is better to make it compulsory to declare all variables. A simple directive **Option Explicit** is used to make variable declaration compulsory.

Example

```
<%
Option Explicit
Dim myVar
%>
```

Option Explicit must be the first statement of ASP page, otherwise an error is generated.

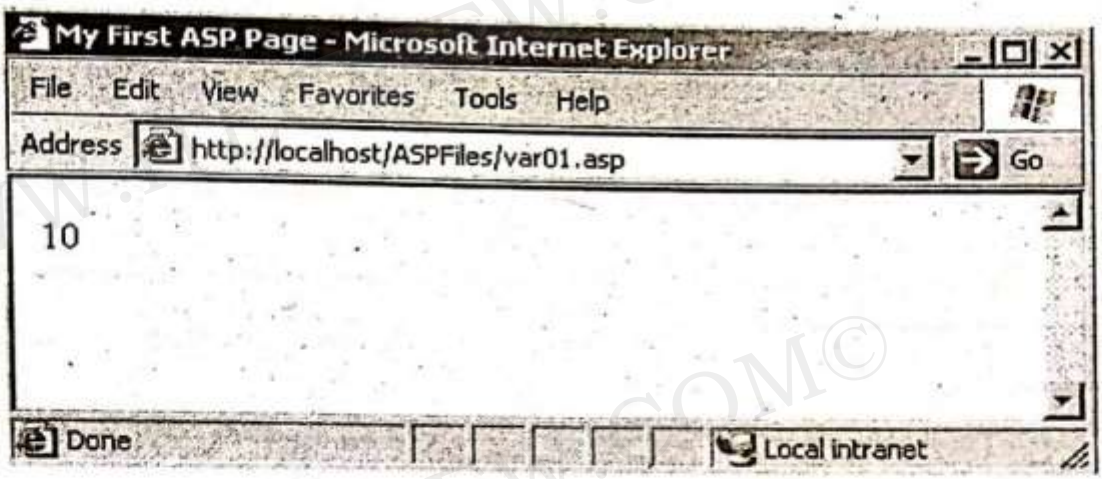
Practice 16.2

1. Open Notepad and type the following code:

```
<%
    n = 10
    Response.Write (n)
%>
```

2. Save the file as "var01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.

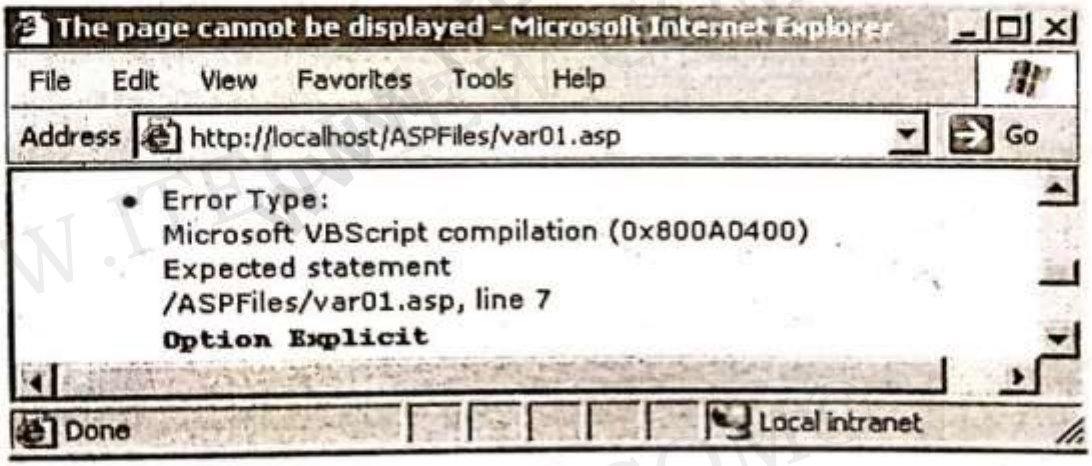
3. Type **http://localhost/ASPFiles/var01.asp** in web browser and press **Enter** key. The page will appear as follows:



4. Change the ASP code as follows:

```
<%
    Option Explicit
    n = 10
    Response.Write (n)
%>
```

5. Save the changing made in the file and refresh the page in the browser. The following error will appear:



6. Again change the ASP code as follows:

```
<%
    Option Explicit
    Dim n
    n = 10
    Response.Write (n)
%>
```

7. Save the changing and open the file again in the web browser. The error will be removed and the result will appear again correctly.

16.10 Mathematical Operators

Operators are mathematical symbol that is used to perform some operation on data. Mathematical operators are the symbols that are used to perform arithmetic operations on data. ASP provides different mathematical operators to manipulate data and perform calculations that are as follows:

Operation	Symbol	Description
Addition	+	Adds two values
Subtraction	-	Subtracts one value from another value
Multiplication	*	Multiplies two values
Division	/	Divides one value by another value
Integer Division	\	Divides one value by other giving integer result
Exponentiation	^	Raises a value to a power
Modulus	Mod	Gives the remainder

Table: Mathematical operators

Following are some examples that explain the use of mathematical operators:

Operation	Example	Result
Addition	7 + 2	9
Subtraction	6 – 4	2
Multiplication	8 * 4	32
Division	17 / 5	3.4
Integer division	17 \ 5	3
Exponentiation	3 ^ 2	9
Modulus	7 mod 4	3

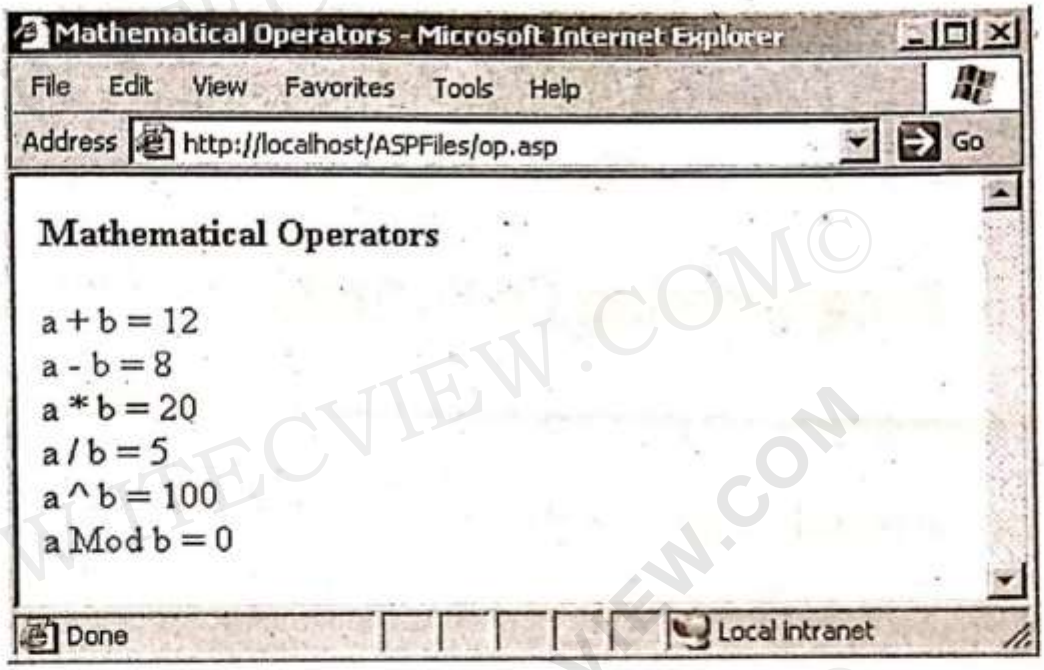
Table: Examples of Mathematical operators

Practice 16.3

1. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Mathematical Operators</TITLE>
</HEAD>
<BODY>
<%
        Dim a, b
        a = 10
        b = 2
        Response.Write ("<B>Mathematical Operators</B><P>")
        Response.Write ("a + b = " & a + b & "<BR>")
        Response.Write ("a - b = " & a - b & "<BR>")
        Response.Write ("a * b = " & a * b & "<BR>")
        Response.Write ("a / b = " & a / b & "<BR>")
        Response.Write ("a ^ b = " & a ^ b & "<BR>")
        Response.Write ("a Mod b = " & a Mod b )
%>
</BODY>
</HTML>
```

2. Save the file as "op.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/op.asp** in web browser and press **Enter** key. The page will appear as follows:



The symbol & is used as concatenation operator in the above example.

16.11 Comments

Comments are the part of script that is not executed. Comments are actually notes about different lines of code that explain the purpose of the code. This explanation is very useful in future.

Comments can be added anywhere in the code. In ASP, comments starts with an apostrophe ('). Any thing written on the right side of apostrophe is considered as comments and will be ignored during execution. Following line is an example of comments:

```
' Hello! Comments are important to maintain things easily...
```

WWW.ITECVIEW.COM©

Programming Constructs in ASP

17.1 Conditional Statements

ASP provides us **branching** or **conditional structures** for making decisions. These structures check the given **condition** before executing a particular statement. The conditional structures in ASP are as follows:

- If...Then
- If...Then...Else
- Select...Case

17.1.1 If...Then Structure

In this structure, the given statement is executed only if the given condition is **true**. If the condition is **false** then the statement is not executed.

Syntax

```
If condition Then
        Statement(s)
End If
```

If, Then, and **End If** are keywords. You can write any number of statements between **Then** and **End If**. This area is called the **body** of **If** statement.

Example

```
<%
        If result = 1 Then
         Response.Write "Good"
%>
```

If only a single statement is to be executed in **Then** block, the complete statement can be written in one line as follows:

```
If result = 1 Then Response.Write "Good"
```

If there are multiple statements to execute in **Then** block, the first statement should begin on the next line after **Then**.

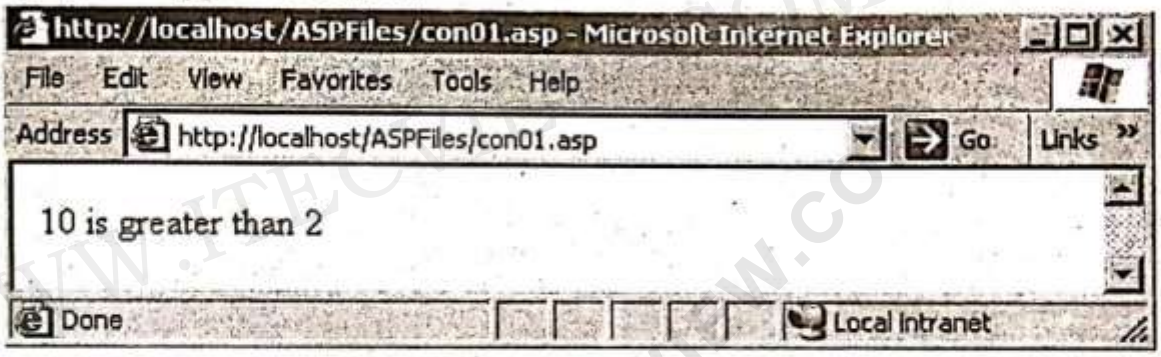
WWW.ITECVIEW.COM©

Practice 17.1

1. Open Notepad and type the following code:

```
<%
    Dim a, b
    a = 10
    b = 2
    if a > b then
      Response.Write (a & " is greater than " & b)
    end if
%>
```

2. Save the file as "con01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/con01.asp** in web browser and press **Enter** key. The page will appear as follows:



17.1.2 Relational Operators

The relational operators are used for specifying different types of conditions in programming. Relational operators compare value at left side to the value at the right side and produce result as true or false. The relational operators are sometimes called the conditional operators because they test conditions that are either **true** or **false**. ASP provides the following relational operators:

Operator	Example	Description
>	Marks > 40	The **greater than** operator. Returns **true** if the value on left side of > is greater than the value on right. Otherwise returns **false**.
<	Pay < 2000	The **less than** operator. Returns **true** if the value on the left side of < is less than the value on the right. Otherwise returns **false**.
=	X = Y	The **equal to** operator. Returns **true** if the values on both sides of = are equal to each other. Otherwise returns **false**.
>=	Marks >= 80	The **greater than or equal to** operator. Returns **true** if the value on the left side of >= is greater than or equal to the value on the right. Otherwise returns **false**.
<=	A <= B	The **less than or equal to** operator. Returns **true** if the value on the left side of <= is less than or equal to the value on the right. Otherwise returns **false**.
<>	A <> B	The **not equal to** operator. Returns **true** if the value on the left side of <> is not equal to the value on the right. Otherwise returns **false**.

Table: Relational Operators

Following are some examples that explain the use of different relational operators.

Relation Expression	Result
100 > 15	True
25 < 5	False
30 <= 12	False
40 >= 20	True
"Visual Basic" = "ASP"	False
0 >= 0	True
0 <= 0	True
1 <> 2	True
5 <> 5	False

17.1.3 If...Then...Else Structure

If...Then...Else structure is used when you want to execute one block of statement(s) when the **condition** is **true** and the other when it is **false**. In any situation, one block is executed and the other is skipped.

In this structure:

- Both blocks of statement can never be executed.
- Both blocks of statements can never be skipped.

Its syntax is as follows: -

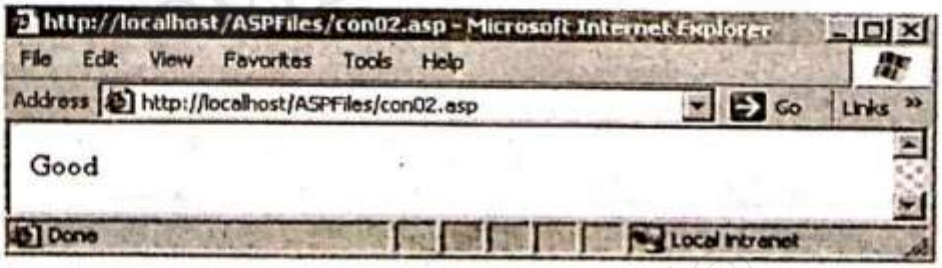
```
If condition Then
        Statement(s)
Else
        Statement(s)
End If
```

Practice 17.2

1. Open Notepad and type the following code:

```
<%
        Dim n
        n = 52
        if n < 50 then
          Response.Write ("Bad")
        else
          Response.Write ("Good")
        end if
%>
```

2. Save the file as "con02.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/con02.asp** in web browser and press **Enter** key. The page will appear as follows:



17.1.4 Compound Comparison

A type of comparison in which more than one conditions are evaluated is called compound comparison. You may want to execute a statement if many conditions are true at the same time. For example, you may want to congratulate a student if he got more than 80 marks both in Math and English etc. This is possible by using logical operators.

17.1.5 Logical Operators

Logical operators are used to evaluate more than one condition in single conditional structure. There are three logical operators:

Operator	Usage	Description
And	If (A > 50) And (B > 50)	It produces True if both conditions are true. It produces False result if any one condition is false.
Or	If (A > 50) Or (B < 50)	It produces True if either condition true. It produces False result if both conditions are false.
Not	If Not (N = 100)	It produces the opposite true or false result.

Suppose, we have two variables A = 100 and B = 50. The use of logical operators is explained with the following examples:

Example	Result
If A > 50 And B > 50	False
If A > 50 Or B > 50	True
If A > 100 And B < 100	False
If A > 100 Or B < 100	True
If Not (A = 100)	False
If Not (B < 50)	True

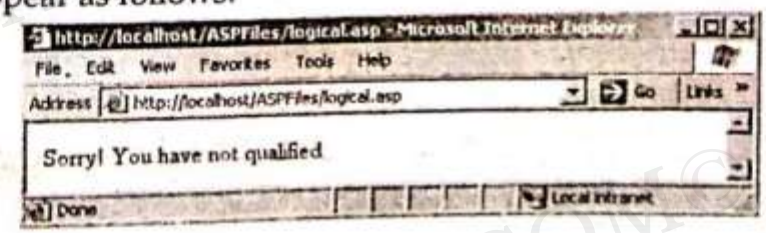
Table: Logical operators examples

Practice 17.3

1. Open Notepad and type the following code:

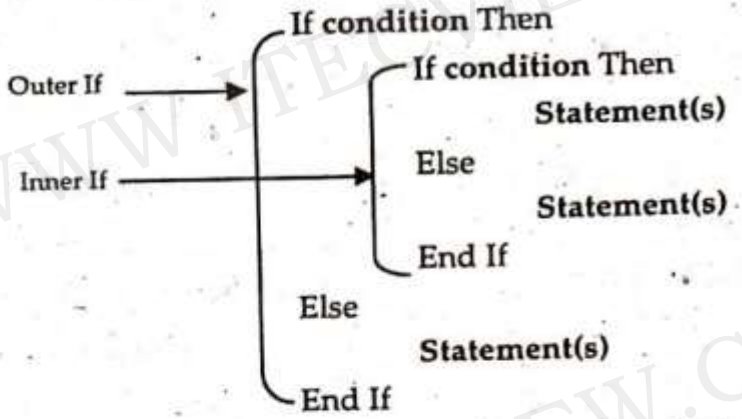
```
<%
    Dim English, Math
    English = 90
    Math = 55
    if English > 80 and Math > 80 then
      Response.Write ("You have qualified.")
    else
      Response.Write ("Sorry! You have not qualified.")
    end if
%>
```

2. Save the file as "logical.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/logical.asp** in web browser and press **Enter** key. The page will appear as follows:



17.1.6 Nested If...Then...Else Structure

An If structure within an If structure is called nested If structure. The syntax of nested If is as follows:



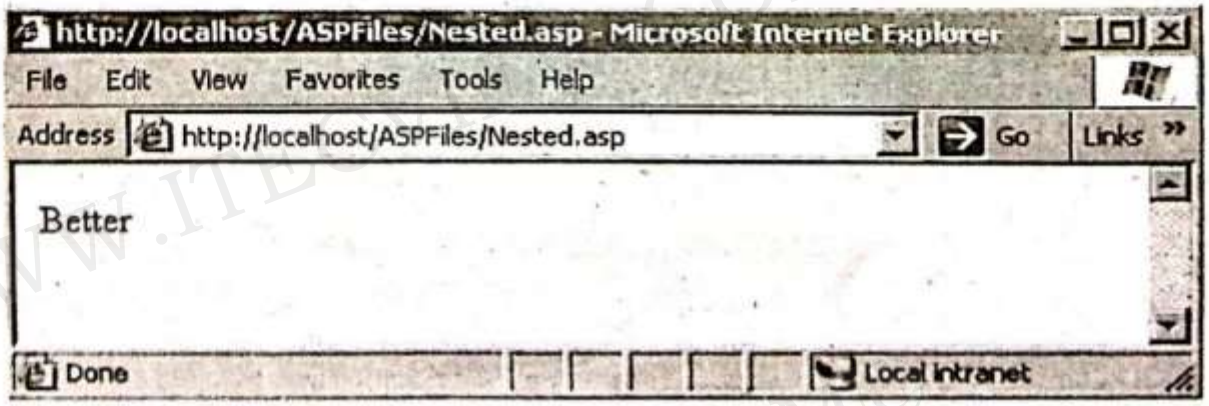
In nested structure, the control enters into the inner If block only if the outer condition is true. Only one block of statements can be executed and the remaining blocks are skipped automatically.

Practice 17.4

1. Open Notepad and type the following code:

```
<%
    Dim English, Math
    English = 70
    Math = 85
    if Math > 80 then
     if English > 80 then
      Response.Write ("Excellent")
     else
      Response.Write ("Better")
     end if
    else
     Response.Write ("Bad")
    end if
%>
```

2. Save the file as "Nested.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Nested.asp** in web browser and press **Enter** key. The page will appear as follows:



17.1.7 Select...Case Structure

Select...Case statement is another conditional structure. It is used for checking the result of a single expression with multiple values. The expression is evaluated at the top of the structure and its result is compared with different values. If the result matches with any value, the corresponding block of statements are executed. The syntax for writing this structure is as follows:

```
Select Case Expression
  Case value
    One or more ASP statements
  Case value
    One or more ASP statements


  Case Else
    One or more ASP statements
End Select
```

Here, Select Case, Case Else and End Select are keywords. Expression can be any valid expression that results in integer or string value. You can use any number of "Case value" statements. Each "Case value" represents a choice. This structure is a good alternative of nested If structure. It can easily be used when there are many choices available. It is very simple and easy to understand.

Practice 17.5

1. Open Notepad and type the following code:

```
<%
    Dim n
    n = 5
    select case n
    case 1
      Response.Write ("One")
    case 2
      Response.Write ("Two")
    case 3
      Response.Write ("Three")
    case 4
      Response.Write ("Four")
    case 5
      Response.Write ("Five")
    case else
      Response.Write ("Out of range")
    end select
%>
```

2. Save the file as "sel01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/sel01.asp** in web browser and press **Enter** key. The following output will appear in the browser:

Five

17.2 Loops

ASP provides a control structure for repeating the same statement again and again. This is known as **iterative, repetitive** or **looping** structure. There are different types of loops available in ASP. Loops are basically used for two purposes:

- To execute a statement or number of statements for a specified number of times. For example, you may want to display a message for 100 times etc.
- To use a sequence of values. For example, you may want to display first 50 natural numbers etc.

17.2.1 While...Wend Loop

While loop is the simplest iterative structure of ASP. This loop executes one or more than one statements while the given condition remains **True**.

Syntax

The syntax of this structure is as follows:

```
While condition
        Statement(s)
Wend
```

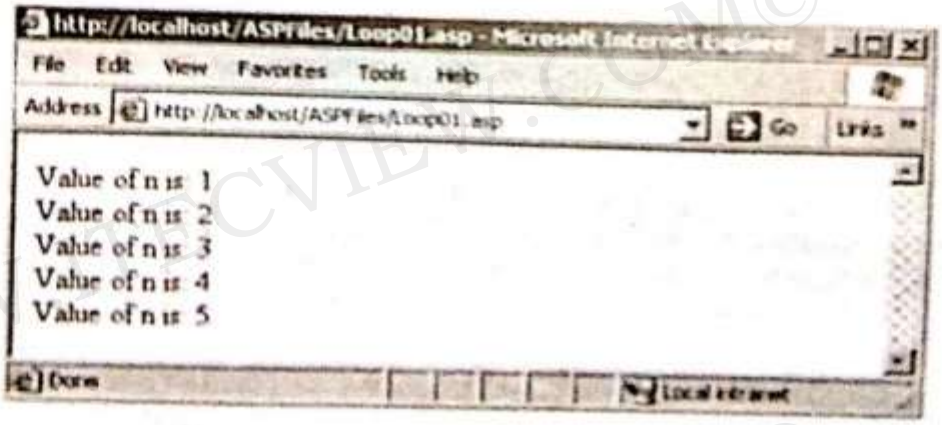
First of all, the condition is evaluated. If it is **true**, the control enters the body of the loop and executes all statements in the body. After executing the body, it again moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains true. The keyword **Wend** indicates the end of loop body.

Practice 17.6

1. Open Notepad and type the following code:

```
<%
        Dim n
        n = 1
        while(n<=5)
          Response.Write ("Value of n is: " & n & "<BR>")
          n = n + 1
        wend
%>
```

2. Save the file as "Loop01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop01.asp** in web browser and press **Enter** key. The page will appear as follows:



17.2.2 Do...While...Loop

This loop is similar to **while...wend** loop. It executes one or more statements while the given condition is true.

Syntax

The syntax of this loop is as follows:

```
Do While condition
    Statement(s)
Loop
```

First of all, the condition is evaluated. If it is true, the control enters the body of the loop and executes all statements in the body. After executing the body, it again moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains **true**. The keyword Loop indicates the end of loop body.

Practice 17.7

Write ASP code that displays back counting from 10 to 1.

1. Open Notepad and type the following code:

```
<%
    Dim n
    n = 10
    do while(n>=1)
      Response.Write (n & "  ")
      n = n – 1
    loop
%>
```

2. Save the file as "Loop02.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop02.asp** in web browser and press **Enter** key. The following output will appear in the browser:

```
10  9  8  7  6  5  4  3  2  1
```

17.2.3 Do...Until...Loop

This loop executes one or more statements while the given condition remains false.

Syntax

The syntax of this loop is as follows:

```
Do Until condition
    Statement(s)
Loop
```

Working of Do...Until...Loop

First of all, the condition is evaluated. If it is false, the control enters the body of the loop and executes all statements. After executing the loop body, it moves to the start of the loop and evaluates the condition again. This process continues as long as the condition remains false. The keyword Loop indicates the end of loop body.

Practice 17.7

Write ASP code that displays the sum of all odd numbers from 1 to 25.

WWW.TECVIEW.COM©

1. Open Notepad and type the following code:

```
<%
    Dim n, sum
    n = 1
    sum = 0
    do until (n > 25)
      if n mod 2 <> 0 then
        sum = sum + n
      end if
      n = n + 1
    loop
      Response.Write ("Sum = " & sum)
%>
```

2. Save the file as "Loop03.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop03.asp** in web browser and press **Enter** key. The following output will appear in the browser:

```
Sum = 169
```

17.2.4 For...Next Loop

This loop executes one or more statements for a specified number of times.

Syntax

The syntax of this loop is as follows:

```
For <counter variable> = <start value> To <end value> Step <increment>
    Statement(s)
Next
```

The number of iterations is determined by the start value and end value of counter variable. By default, the counter variable is incremented by 1 after each iteration. But you can change this value by using **Step** clause, which is optional. The keyword **Next** indicates the end of loop body.

Practice 17.8

Write ASP code that displays the sum of first five number.

1. Open Notepad and type the following code:

```
<%
    Dim s
    s = 0
    for i = 1 to 5
      s = s + i
    next
    Response.Write ("Sum of first five numbers = " & s )
%>
```

2. Save the file as "Loop04.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop04.asp** in web browser and press **Enter** key. The following output will appear in the browser:

```
Sum of first five numbers = 15
```

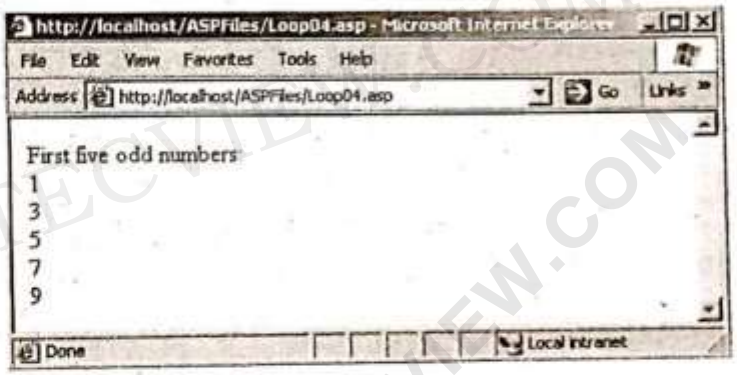
Practice 17.9

Write ASP code that displays the first five odd numbers.

1. Open Notepad and type the following code:

```
<%
    Response.Write ("First five odd numbers:<BR>")
    for i = 1 to 10 step 2
      Response.Write (i & "<BR>" )
    next
%>
```

2. Save the file as "Loop04.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop04.asp** in web browser and press **Enter** key. The page will appear as follows:



17.2.5 Early Termination of Loops

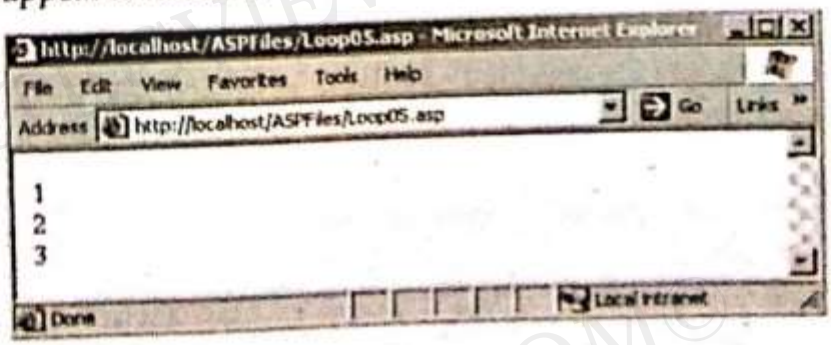
Normally, the termination of the loop depends on the given condition. But you can terminate a loop earlier, without looking at the condition. Exit For statement is used in the body of **For...Next** loop to terminate it abnormally. Exit Do statement is used with Do loops for the same purpose.

Practice 17.10

1. Open Notepad and type the following code:

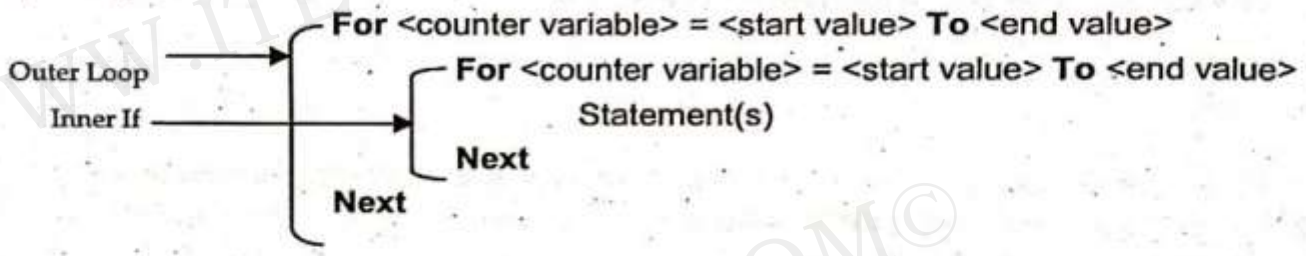
```
<%
    for i = 1 to 10
      Response.Write (i & "<BR>" )
      if I >= 3 then exit for
    next
%>
```

2. Save the file as "Loop05.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop05.asp** in web browser and press **Enter** key. The page will appear as follows:



17.2.6 Nested Loops

A loop within a loop is called nested loop. In nested loops, the inner loop is executed completely with each change in the value of counter variable of outer loop. The syntax of nested loop is as follows:



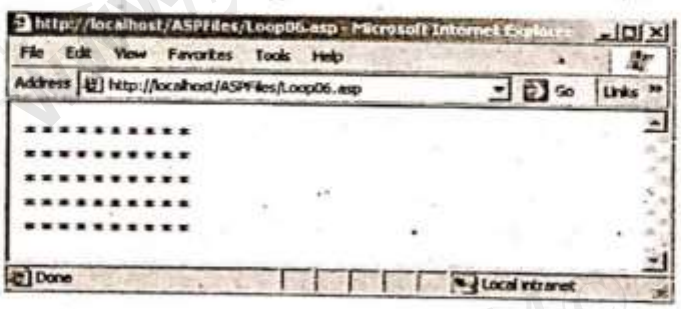
Practice 17.11

Write ASP code to display five rows of asterisks with each row containing ten asterisks.

1. Open Notepad and type the following code:

```
<%
    for i = 1 to 5
      for j = 1 to 10
        Response.Write ("* ")
      next
      Response.Write ("<BR>")
    next
%>
```

2. Save the file as "Loop06.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Loop06.asp** in web browser and press **Enter** key. The page will appear as follows:



17.3 Procedures

A procedure is a block of code that performs some operation. The code written is executed only when it is **called**. A simple procedure can be called any number of times by just writing its name.

When a user-defined procedure is called, the control moves in that procedure. All statements of the procedure are executed and then the control again moves back to the point where the procedure was called.

Syntax

The basic syntax for a procedure is as follows:

```
Sub procedure_name
        Statement(s)
End Sub
```

Sub and **End Sub** are the keywords that indicate the start and end of procedure body. procedure_name is the name of procedure. The rules for procedure name are similar to the rules for declaring a variable.

Example

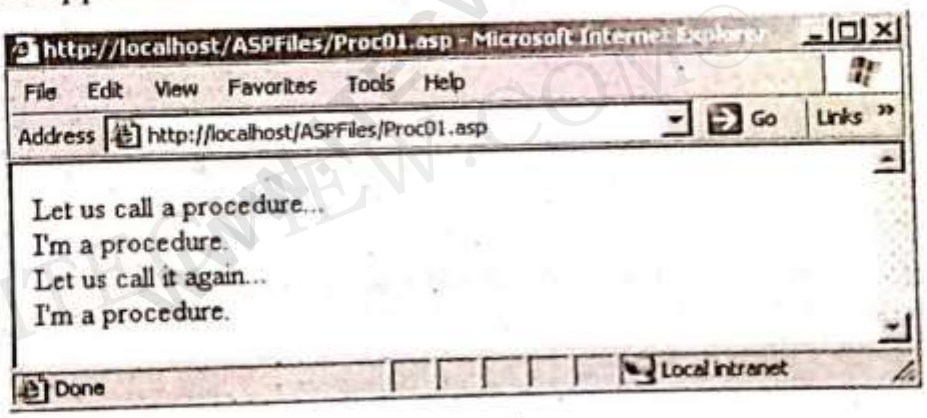
```
Sub Message()
Reponse.Write ("Hello World")
End Sub
```

Practice 17.12

1. Open Notepad and type the following code:

```
<%
    sub show()
      Response.write("I'm a procedure.<br>")
    end sub
    Response.Write ("Let us call a procedure...<br>")
    call show
    Response.Write ("Let us call it again...<br>")
    call show
%>
```

2. Save the file as "Proc01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Proc01.asp** in web browser and press **Enter** key. The page will appear as follows:



17.3.1 Passing Parameters to Procedures

Parameters are the values, which are passed to a procedure when the procedure is called. Parameters are given in the parentheses. The number of parameters and their type is decided during procedure coding.

Practice 17.13

Write ASP code that passes two numbers to a procedure. The procedure adds these numbers and displays the sum.

1. Open Notepad and type the following code:

```
<%
    Dim a, b
    a = 10
    b = 20
    call sum(a, b)
```

```
    sub sum(x, y)
      Response.write("Sum = " & x+y)
    end sub
%>
```

2. Save the file as "Proc02.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Proc02.asp** in web browser and press **Enter** key. The page will appear as follows:

17.4 Functions

A function is a block of code that performs some action. It is similar to the procedure. The difference is that it can return a value back to the position where it is called. The statements written in a function are executed only if it is called by its name.

When a user-defined function is called, the control moves in that function. All statements of the procedure are executed and then the control again moves back to the point where the function was called along with the returned value.

Syntax

The basic syntax for a function is as follows:

```
Function function_name
  Statement(s)
End Function
```

Function and **End Function** are the keywords that indicate the start and end of function body. function_name is the name of function. The rules for function name are similar to the rules for declaring a variable.

Example

```
Function Msg()
  Response.Write("Hello Function!")
End Function
```

17.4.1 Returning a Value from Function

A function can return a value back to the calling point. You can return a value by simply assigning that value to the name of the function. Assume that the name of the function is a variable. Any value assigned to the name of the function is actually returned. In the above figure, 10 is returned to the point from where the function was called.

Practice 17.14

Write ASP code that passes two numbers to a function. The function adds these numbers and returns the sum.

1. Open Notepad and type the following code:

```
<%
        Dim a, b, c
        a = 100
        b = 200
        c = sum(a, b)
        Response.Write("Sum = " & c)

        function sum(x, y)
          Dim s
          s = x+y
          sum = s
        end function
%>
```

2. Save the file as "Fun01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Fun01.asp** in web browser and press **Enter** key. The following output will appear in the browser:

```
Sum = 300
```

17.4.2 Exit Sub & Exit Function

Exit Sub and **Exit Function** statements are similar to **Exit For** statement. Exit Sub is used inside a procedure to exit that procedure abnormally and Exit Function is used inside a function for the same purpose.

17.5 Arrays

An array is a group of consecutive memory locations with same name and type. Simple variable is a single memory location with a unique name and a type. But an array is a collection of different adjacent memory locations. All these memory locations have one collective name and type.

Arrays are used to store a large amount of similar kind of data. For example, if we want to store the marks of 100 students, we have to declare 100 variables. It is time-consuming process to use these variables individually. We can save this time by using arrays. Instead of using 100 variables, we can declare only one array of 100 elements.

Declaring One-Dimensional Arrays

One-dimensional array is used to store different values of similar type as a single row or column. The syntax of declaring one-dimensional array is as follows:

Dim <array name> (length)

The following example declares an array to store the marks of five students:

```
Dim Marks(4)
```

This statement will create an array in the memory that consists of five memory locations numbered from 0 to 4. We can store five integers in this array.

Each block in the array is called **cell** or **element**. Each element is assigned a unique number in the array. This number is called the **index** or **subscript** of that element. The index is used to access an individual element of the array. Normally, index starts from 0. So, **Marks(0)** refers to the first element of the array, **Marks(1)** refers to the second element.

Practice 17.15

1. Open Notepad and type the following code:

```
<%
    Dim n(3)
    n(0) = 10
    n(1) = 20
    n(2) = 30
    n(3) = 40
    for i = 0 to 3
      Response.Write (n(i) & "<BR>")
    next
%>
```

2. Save the file as "arr01.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/arr01.asp** in web browser and press **Enter** key. The page will appear as follows:

Initializing Array

The array can be initialized with the values as follows:

Dim variable_name = Array (list_of_values)

Dim and **Array** are keywords. variable_name is the name of array and list_of_values indicates the values stored in array.

Example

```
Dim a = Array(1,2,3)
```

Practice 17.16

Write ASP code that stores five numbers in an array. It then calculates and displays the sum of these numbers.

1. Open Notepad and type the following code:

```
<%
    Dim n, i, s
    s = 0
    n = Array(10,20,30,40)
    for i = 0 to 3
      s = s + n(i)
    next
    Response.Write ("Sum of array elements = " & s)
%>
```

2. Save the file as "arr02.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/arr02.asp** in web browser and press **Enter** key. The following output will appear in the browser:

Sum of array elements = 100

17.5.1 Searching in Arrays

Searching is a process of finding the required data in the array. Searching becomes more important when the length of the array is very large. A simple way of search an array for the desired value is as follows:

1. Visit the first element of the array and compare its value with the required value.
2. If the value of array matches with the desired value, the search is complete.
3. If the value of array does not math, move to the next element and repeat the same process.

Loops are frequently used to visit the elements of array for searching a value. You can start the counter variable of loop from 0 and move it to the last index of the array. For example if the array to be searched has 10 elements, you will start the counter variable from 0 and move to 9.

Practice 17.17

Write ASP code that stores five numbers in an array. It then searches the number 30 in the array and displays the result.

1. Open Notepad and type the following code:

```
<%
    Dim arr, i, n, loc
    n = 30
    loc = -1
    arr = Array(10,20,30,40)
    for i = 0 to 3
      if n = arr(i) then
            loc = i
            exit for
      end if
    next
    if loc = -1 then
      Response.Write ("Number not found.")
    else
      Response.Write("Number found at element: " & loc)
    end if
%>
```

2. Save the file as "arr03.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/arr03.asp** in web browser and press **Enter** key. The following output will appear in the browser:
   Number found at element: 2

Exercise Questions

1. Write ASP code to input a year in textbox and display whether it is a leap year or not.
2. Write ASP code to input sides of triangle in textboxes and display type of triangle.
3. Write ASP code to input a century and display all leap years in that century.
4. Write ASP code to display back counting from 10 to 1.
5. Write ASP code to display first five numbers with their squares and cubes.

ASP Objects

18.1 Client-Server Communication

The client and server communicate with each other using HTTP protocol. When the browser on client computer opens a web page, it actually establishes a connection with the server on which the web page is stored. The client sends a request to the server. The server processes the request and sends the required web page to the client. This communication consists of different **requests** and **responses**.

A request from the client consists of information about the client and some other parameters. The parameters contain the data that is processed at server. A request can be divided into two parts:

1. Header

The header contains the following information about the client:

- Data and time of the request
- Details about the browser that made the request
- URL of the requested page
- IP address of the client computer

2. Body

The message body contains the contents of HTML form. The **method** attribute of the form specifies how the form data is sent to the server for processing. The body of message may also contain some other data such as word document etc.

Active Server Pages uses two built-in objects called **Request** and **Response** that refer to the request of the client and response of the server.

18.2 ASP Objects

Some important objects in ASP are as follows:

18.2.1 Request Object

The **Request** object is used to retrieve the information sent in a request from client to the server. It reads the information when **Submit** button is clicked.

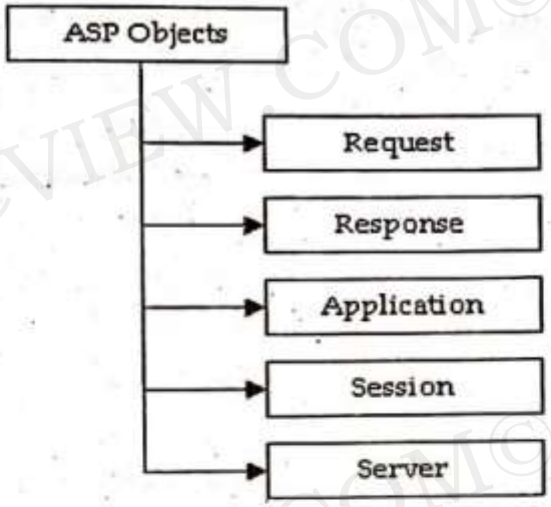


Figure: ASP Objects

Collections

The syntax of using a collection is as follows:

Request.Collections("Variable") OR Request("Variable")

Some important collections of Request object are as follows:

1. ClientCertificate It refers to the values of fields stored in the client certificate that is sent in the HTTP request. The server should be configured to request client certificates before this collection is used.

2. Cookies It refers to the values of cookies sent in the HTTP request. It can be used to access the values of different cookies.

Example: Suppose a cookie **userid** is created. The following example will access and display its value:

```
<%
    var = Request.Cookies("userid")
    Response.Write("User ID = " & var)
%>
```

3. Form It refers to the values of form elements in HTTP request body. It can be used to access the values of different form elements.

Example: Suppose a form contains the following textbox:

```
<INPUT TYPE="Text" Name="FName" Value="Usman">
```

Form collection can be used to display the value of above textbox:

```
<% Response.Write(Request.Form("FName")) %>    OR
<% Response.Write(Request("FName")) %>
```

4. QueryString It refers to the values of variables in the HTTP query string. It can be used to access the values of query string variables.

Example: Suppose the file **about.asp** is loaded as follows:

http://localhost/about.asp?name=Usman&age=16

QueryString collection can be used to access the above values as follows:

```
<%
    Response.Write("Name = " & Request.QueryString("name"))
    Response.Write("Age = " & Request.QueryString("age"))
%>
```

5. ServerVariables It refers to the values of predefined environment variables.

Example: The following example will display server's domain name:

```
<%
    Response.Write("Domain Name: ")
    Response.Write(Request.ServerVariables("server_name"))
%>
```

Properties

An important property of Request object is as follows:

TotalBytes It is a ready-only property. It specifies the total number of bytes sent by the client in the body of request.

Example: The following example will display total bytes sent by client:

```
<% Response.Write("Total Bytes: " & Request.TotalBytes) %>
```

Methods

An important method of Request object is as follows:

BinaryRead It retrieves data sent to the server from client as part of a POST request.

Example: The following example will retrieve data send to the server:

```
<%
    dim x, y
    x = Request.TotalBytes
    y = Request.BinaryRead(x)
%>
```

18.2.2 Response Object

The **Response** object is used to send data from the server to the client.

Collections

An important collection of Response object is as follows:

Cookies It specifies cookie values. It can be used to set the values of cookies.

Example: The following example will create two cookies and set their values:

```
<%
    Response.Cookies("FName") = "Usman"
    Response.Cookies("Age") = "30"
%>
```

Properties

Some important properties of Response object are as follows:

1. Buffer It indicates whether the page output is buffered or not.

Example: The following example will set the page output to buffer:

```
<% Response.Buffer = true %>
```

2. CacheControl It determines whether proxy servers are able to cache the output generated by ASP or not. Its value can be private or public. The **private** value is default and indicates that proxy servers will not cache the pages. The **public** value indicates proxy servers will cache the pages.

Example: The following example will set cache control to public:

```
<% Response.CacheControl = "Public" %>
```

3. Charset It appends the name of the character set to the content-type header.

Example: Suppose the following example is used:

```
<% Response.Charset = "ISO-8859-1" %>
```

The content-type header will be as follows:

```
content-type:text/html; charset=ISO-8859-1
```

4. ContentType It specifies HTTP content type for response. The default content-type header is as follows:

```
content-type:text/html
```

Example: The following examples specify different content types:

```
<% Response.ContentType="image/GIF"%>
<% Response.ContentType="image/JPEG"%>
<% Response.ContentType="image/JPEG"%
```

5. Expires It specifies the length of time before a page cached on a browser expires.

Example: The following examples specify that the page will expire after 50 minutes:

```
<% Response.Expires = 50 %>
```

6. ExpiresAbsolute It specifies the date and time on which a page cached on a browser expires.

Example: The following examples specify that the page will expire at 8:00 PM on March 31, 2011:

```
<% Response.ExpiresAbsolute = #March 31, 2011 20:00:00# %>
```

7. IsClientConnected It indicates whether the client has disconnected from the server.

Example: The following examples specify that the page will expire at 8:00 PM on March 31, 2011:

```
<%
    If Response.IsClientConnected = true then
      Response.write ("Connected!")
    else
      Response.write("Not connected!")
    end if
%>
```

8. Pics It adds the value of a PICS label to the pics-label field of the response header. **PICS** stands for **Platform for Internet Content Selection.** It is a rating system that is used to rate the content in a web site. It looks like the following:

```
PICS-1.1 "http://www.rsac.org/ratingsv01.html"
by "user@website.com" for "http://www.website.com"
on "2002.10.05T02:15-0800"
r (n 0 s 0 v 0 l 0)
```

9. Status It is the value of the status line returned by the server. It is also used to modify the status line returned by server. The status values are defined in HTTP specification. It consists of three-digit code and its description such as 404 Not Found.

Methods

Some important methods of Response object are as follows:

1. AddHeader It sets the HTML header name to value. The header cannot be removed once it has been added.

Example: The following example will set a header name to a value:

```
<% Response.AddHeader "WARNING", "This action is not allowed!!" %>
```

2. AppendToLog It adds a string to the end of Web server log entry for this request.

Example: The following example adds a string to the end of log entry:

```
<% Response.AppendToLog "End of log entry" %>
```

3. BinaryWrite It writes the given information to the current HTTP output without any character-set conversion. This method is used to write image data from a database to a browser.

4. Clear It erases any buffered HTML output. This method does not clear response headers. It only clears response body. It causes an error if Response.Buffer is set to false.

Example: The following example will display nothing in the browser because of using Response.Clear method:

```
<%
    Response.Buffer = true
%>
<HTML>
<BODY>
<P>Hello! Web Development with ASP is really a fun!!.</P>
<%
    Response.Clear
%>
</BODY>
</HTML>
```

5. End It stops processing the .asp file and returns the current result. It will flush the buffer if Response.Buffer is set to true.

Example: The following example will display **Hello** only. The remaining text will not appear because Response.End is used before it:

```
<HTML>
<BODY>
<P>Hello
<%
   Response.End
%>
World! </P>
</BODY>
</HTML>
```

6. Flush It sends the buffered output immediately. It will cause an error if Response.Buffer is set to false.

Example: The following example displays output using Response.Flush:

```
<%
    Response.Buffer = true
%>
```

WWW.ITECVIEW.COM©

```
<HTML>
<BODY>
<P>First Paragraph</P>
<P>Second Paragraph</P>
<%
    Response.Flush
%>
</BODY>
</HTML>
```

7. Redirect It sends a redirect message to the browser to attempt to connect to a different URL.

Example: The following example redirects browser to hotmail website:

```
<%
    Response.Redirect("http://www.hotmail.com")
%>
```

8. Write It writes a variable to the current HTTP output as a string. It can also be used to display string constant and HTML tags.

Example: The following example will display the message using HTML tags and a variable.

```
<%
    Dim name
    name = "Usman"
    Response.Write("<I>Hello</I> " & name & "<BR>")
    Response.Write("<B>You have got admission!! </B>")
%>
```

Practice 18.1

Create a web page that inputs name in textbox and province in a list box. The page is submitted with POST method to ASP file that displays the name and province given by user.

1. Open Notepad and type the following code:

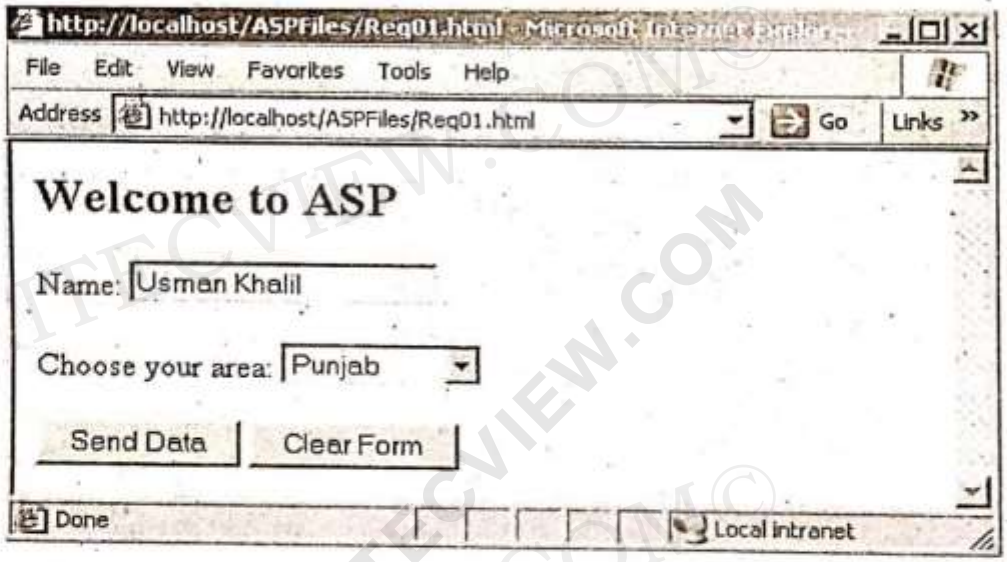
```
<HTML>
<BODY>
<H2>Welcome to ASP</H2>
<FORM NAME="form1" METHOD="post" ACTION="Res01.asp">
Name: <INPUT TYPE="text" NAME="name"><P>
Choose your area:
<SELECT NAME="area">
        <OPTION VALUE="Punjab">Punjab
        <OPTION VALUE="Sindh">Sindh
        <OPTION VALUE="Khyber ">Khayber PakhtoonKhwah
        <OPTION VALUE="Balochistan">Balochistan
        <OPTION VALUE="AJK">Azad Jammu Kashmir
</SELECT>
<P>
<INPUT TYPE="submit" VALUE="Send Data" NAME="submit">
<INPUT TYPE="reset" VALUE="Clear Form" NAME="reset">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Req01.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.

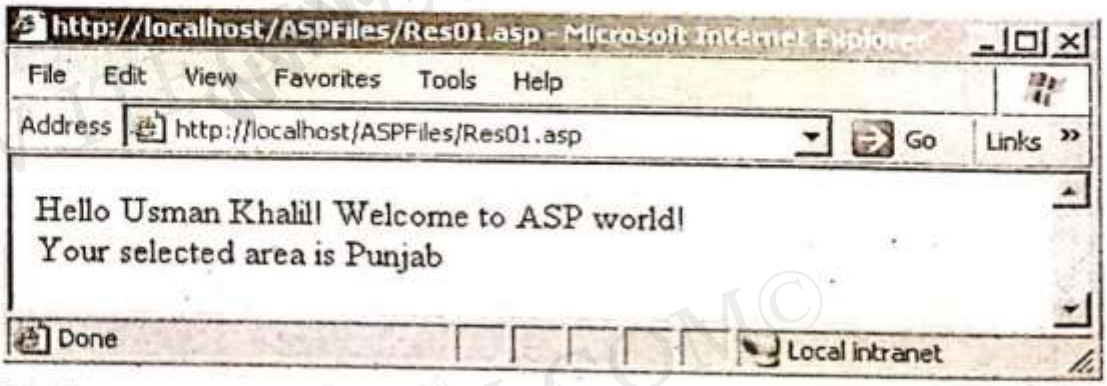
3. Open a new file in Notepad and type the following code:

```
<%
    Dim n, a
    n = Request.Form("name")
    a = Request.Form("area")
    Response.Write("Hello " & n & "! Welcome to ASP world!<BR>")
    Response.Write("Your selected area is " & a)
%>
```

4. Save the file as "Res01.asp" C:\Inetpub\wwwroot\ASPFiles directory.
5. Type **http://localhost/ASPFiles/Req01.html** in web browser and press **Enter** key. The page will appear in the browser. Enter data in the form as follows:



6. Click on **Send Data** button. The form data will be sent to Res01.asp that will appear as follows:



How it Works

The above practice uses two files. The file **Req01.html** is used as request and **Res01.asp** is used as response. The request file uses a form to get input from the user. The **ACTION** property of **FORM** tag defines how the form will be processed. The above practice uses Res01.asp in **ACTION** property. It means that form data will be submitted to **Res01.asp** file when the user clicks submit button. The **Form** collection of **Request** object contains the values of all form fields. The statement *n = Request.Form("name")* stores the value of **name** text field in the variable **n**. The statement *a = Request.Form("area")* stores the value of **area** text field in the variable **a**. Finally, the values stored in these variables are displayed in the browser using *Response.Write* method.

Using QueryString Collection

The **QueryString** collection of **Request** object also refers to the values of form elements. It is used when the fom data is sent by using **get** value in **METHOD** property. In this situation, the values of form elements are displayed in the address bar as part of URL.

Practice 18.2

Create a web page that inputs name and age in textboxes. The page is submitted with GET method to ASP file that displays the name and age given by user.

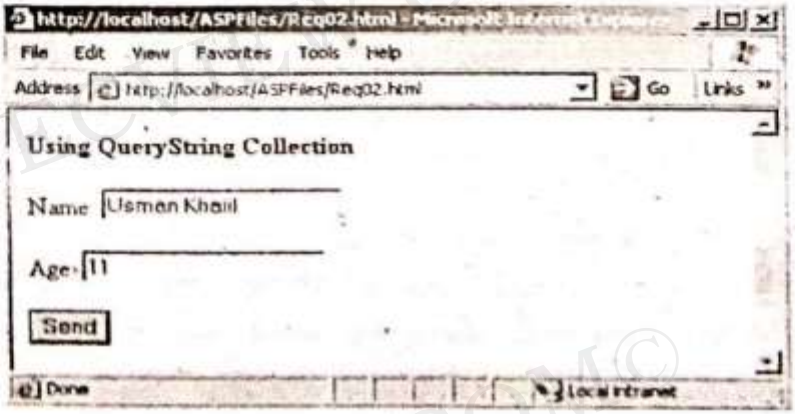
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Using QueryString Collection</H4>
<FORM NAME="form1" METHOD="get" ACTION="Res02.asp">
Name: <INPUT TYPE="text" NAME="name"><P>
Age: <INPUT TYPE="text" NAME="age"><P>
<P>
<INPUT TYPE="submit" VALUE="Send" NAME="submit">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Req02.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
    Dim n, a
    n = Request.QueryString("name")
    a = Request.QueryString("age")
    if n = "" then
      Response.Write("Name must be entered!")
    elseif a = "" then
      Response.Write("Age must be entered!")
    else
      Response.Write("Hello " & n & "!<BR>")
      Response.Write("Your age is " & a)
    end if
%>
```

4. Save the file as "Res02.asp" C:\Inetpub\wwwroot\ASPFiles directory.
5. Type **http://localhost/ASPFiles/Req02.html** in web browser and press **Enter** key. The page will appear in the browser. Enter data in the form as follows:



6. Click on **Send** button. The form data will be sent to Res02.asp that will appear as follows:

How it Works

The above practice uses *Res02.asp* in **ACTION** property and **get** in **METHOD** property. The **QueryString** collection of **Request** object contains the values of all form fields. The form elements and their values are also displayed in the address bar when the user clicks **Submit** button. The statement *n = Request.QueryString("name")* stores the value of **name** text field in the variable **n**. The statement *a = Request.QueryString("age")* stores the value of **age** text field in the variable **a**. Finally, the values stored in these variables are displayed in the browser using *Response.Write* method.

Redirecting

Redirection is a very useful process for moving the browser from one location to another. The *Response.Redirect* method is used for redirection.

Practice 18.3

Create a web page that inputs name in textbox and interest in a list box. The page is submitted with POST method to an ASP file that loads a website according to the interest selected by the user.

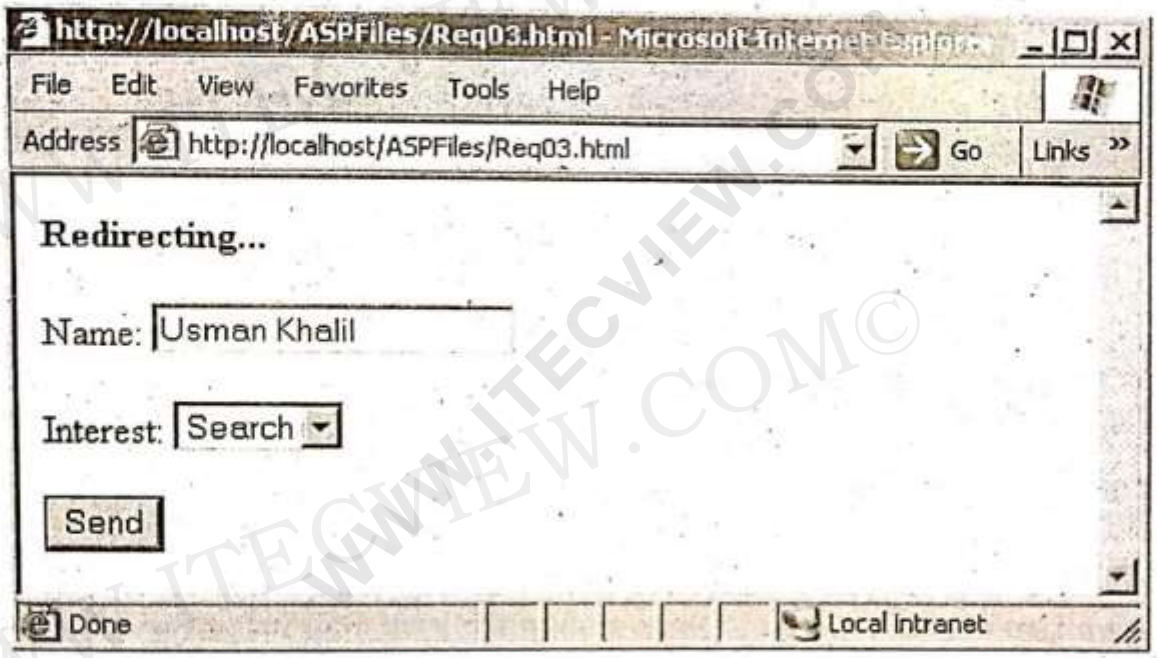
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H4>Redirecting...</H4>
<FORM NAME="form1" METHOD="post" ACTION="Res03.asp">
Name: <INPUT TYPE="text" NAME="name"><P>
Interest:
<SELECT NAME="interest">
<OPTION VALUE="Books">Books
<OPTION VALUE="News">News
<OPTION VALUE="Search">Search
</SELECT>
<P>
<INPUT TYPE="submit" VALUE="Send" NAME="submit">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Req03.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
    Dim n, i
    n = Request.Form("name")
    i = Request.Form("interest")
    if i = "Books" then
      Response.Redirect "http://www.itseries.com.pk"
    elseif i = "News" then
      Response.Redirect "http://www.bbc.com"
    elseif i = "Search" then
      Response.Redirect "http://www.google.com"
    end if
%>
```

4. Save the file as "Res03.asp" C:\Inetpub\wwwroot\ASPFiles directory.
5. Type **http://localhost/ASPFiles/Req03.html** in web browser and press **Enter** key. The page will appear in the browser. Enter data in the form as follows:



6. Click on **Send** button. If you are connect to the Internet, the Res03.asp will redirect the browser to http://www.google.com as follows:

18.2.3 The "global.asa" File

The **global.asa** file provides the **Application** and **Session** objects with the script. It also contains the scripts for **OnStart** and **OnEnd** events for Application and Session objects. These scripts uses the property **RUNAT** with the value **Server.**

Structure of "global.asa"

The basic structure of **global.asa** file is as follows:

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Application_OnStart

END SUB
</SCRIPT>

<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Application_OnEnd

END SUB
</SCRIPT>

<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Session_OnStart

END SUB
</SCRIPT>

<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Session_OnEnd

END SUB
</SCRIPT>
```

18.2.4 Application Object

Application object is used to share application-level information. The development of an **Application** requires establishing a directory that is served by an Internet Information Server. This directory contains the source code for the application. Each application can have a single **global.asa** file.

The **Application** object can be used to maintain the application. For example, it can be used to count the number of user who have visited the webiste. An application variable can be used to store the count of visitors. The application variables are accessible in all web pages of the website.

Collections

Some important collections of Application object are as follows:

Contents It contains all items added to the Application through script commands.

Example: The following example will display all items in this collection:

```
<%
    For c = 1 to Application.Contents.Count
        Response.Write(c & " = " & Application.Contents(c) & "<BR>")
    Next
%>
```

StaticObjects It contains all objects added to the session with the <OBJECT> tag.

Example: The following example will display all items in this collection:

```
<%
    For each x in Application.StaticObjects
     Response.Write(x & "<BR>")
    Next
%>
```

Methods

Some important methods of Application object are as follows:

1. Contents.Remove It is used to delete an item from the Contents collection.

Example: The following example will delete **Marks** item:

```
<%
  Application.Contents.Remove("Marks")
%>
```

2. Contents.RemoveAll It is used to delete all items from the Contents collection.

Example: The following example will delete all items in collection:

```
<%
  Application.Contents.RemoveAll( )
%>
```

3. Lock It prevents other users from modifying Application object properties. It ensures that one user can modify Application variables at a time.

Example: The following example will lock the application:

```
<%
  Application.Lock
%>
```

4. Unlock It allows other clients to modify Application object properties.

Example: The following example will unlock the application:

```
<%
  Application.Unlock
%>
```

Events

Some important events of Application object are as follows:

1. Application_OnStart It occurs when the first user accesses any part of the application.

Example: The following example will create an application variable **visitors** and sets its value to 0:

```
Sub Application_OnStart
    Application("visitors") = 0
End Sub
```

2. Application_OnEnd It occurs when the application ends.

Example: The following example will store the value of application variable **visitors** to another application variable **total:**

```
Sub Application_OnStart
    Application("total") = Application("visitors")
End Sub
```

Practice 18.4

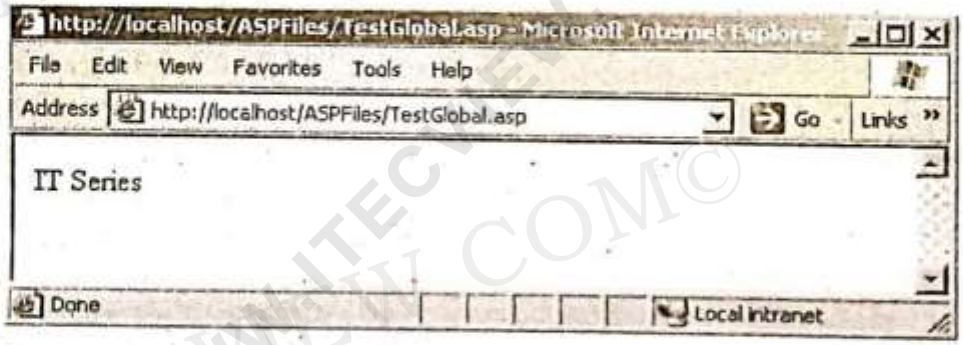
1. Open Notepad and type the following code:

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Application_OnStart
    application("info") = "IT Series"
END SUB
</SCRIPT>
```

2. Save the file as "global.asa" in **C:\Inetpub\wwwroot** directory.
3. Open a new file in Notepad and type the following code:

```
<HTML>
<BODY>
<%
  Response.Write (application("info"))
%>
</BODY>
</HTML>
```

4. Save the file as "TestGlobal.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Type **http://localhost/ASPFiles/TestGlobal.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



How it Works

The above practice uses **Application_OnStart** event of **global.asa** file. The statement *application("info") = "IT Series"* creates an application variable **info**. The file TestGlobal.asp displays the values of this application variable in the browser.

18.2.5 Session Object

The **Session** object is used to store information about particular user. It is used to manage user sessions. A session starts as the user accesses any web page of the website. The session ends when the user leaves the website. The properties of Session object can be referenced on any page during the session.

Collections

Some important collections of Session object are as follows:

1. Contents It contains the items added to the session with script commands.

Example: The following example will display all items in this collection:

```
<%
    For c = 1 to Session.Contents.Count
        Response.Write(c & " = " & Session.Contents(c) & "<BR>")
    Next
%>
```

2. StaticObjects It contains objects created with <OBJECT> tag and with session scope.

Example: The following example will display all items in this collection:

```
<%
    For each x in Session.StaticObjects
     Response.Write(x & "<BR>")
    Next
%>
```

Properties

Some important properties of Session object are as follows:

1. CodePage It specifies the character set that will be used while displaying dynamic content. For example, 1252 code is used for American English and most European languages.

Example: The following example will display the code page:

```
<%
    Response.Write(Session.CodePage)
%>
```

2. LCID It is used to set or get an integer that specifies a location or region. The date, time and currency is displayed according to that location or region.

Example: The following example will display the default LCID. It will also display the data and currency format according to the LCID:

```
<%
    Response.Write("LCID: " & Session.LCID & "<BR>")
    Response.Write("Date format: " & date() & "<BR>")
    Response.Write("Currency format: " & FormatCurrency(350))
%>
```

3. SessionID It returns the session identification for each user. The unique ID is generated by the server.

Example: The following example will display the session ID:

```
<%
    Response.Write(Session.SessionID)
%>
```

4. Timeout It is the timeout period for the session state for this application in minutes. The session ends if the user does not refresh or request a page within the timeout period.

Example: The following example will display the default timeout and set the new timeout to 60 minutes:

```
<%
    Response.Write("Default Timeout: " & Session.Timeout)
    Session.Timeout = 60
    Response.Write("New Timeout: " & Session.Timeout)
%>
```

Methods

Some important methods of Session object are as follows:

1. Abandon It destroys a Session object and releases its resources. When it is called, the current Session object is deleted after processing all scripts on current page.

Example: The following example will abandon the current session:

```
<%
    Session.Abandon
%>
```

2. Contents.Remove It is used to delete an item from the Contents collection.

Example: The following example will delete **Marks** item:

```
<%
  Session.Contents.Remove("Marks")
%>
```

3. Contents.RemoveAll It is used to delete all items from the Contents collection.

Example: The following example will delete all items in collection:

```
<%
  Session.Contents.RemoveAll( )
%>
```

Events

Scripts for the following events are declared in the **global.asa** file.

1. Session_OnStart It occurs when the server creates a new session.

Example: The following example will increment value of application variable **visitors** by 1:

```
Sub Session_OnStart
    Application.Lock
    Application("visitors") = Application("visitors") + 1
    Application.Unlock
End Sub
```

2. Session_OnEnd It occurs when a session is abandoned or times out.

Example: The following example will decrement the value of application variable **visitors** by 1:

```
Sub Session_OnStart
    Application.Lock
    Application("visitors") = Application("visitors") – 1
    Application.Unlock
End Sub
```

Practice 18.5

1. Open Notepad and edit **global.asa** file as follows:

```
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Application_OnStart
   application("visitors") = 0
END SUB
</SCRIPT>

<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
SUB Session_OnStart
   application("visitors") = application("visitors") + 1
END SUB
</SCRIPT>
```

2. Open a new file in Notepad and type the following code:

```
<HTML>
```

```
<BODY>
<H4>Using sessions...</H4>
<%
  Response.Write ("You are visitor number " & application("visitors"))
%>
</BODY>
</HTML>
```

3. Save the file as "TestSession.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
4. Type **http://localhost/ASPFiles/TestSession.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:

5. Close the browser.
6. Start the browser and open TestSession.asp again. The page will appear as follows:

How it Works

The above practice uses **Application_OnStart** & **Session_OnStart** events of **global.asa** file. The statement *application("visitors") = 0* in **Application_OnStart** event creates an application variable **visitors** and initialzes it to 0 value. When a user accesses any page of the website, a new session starts. The statement *application("visitors") = application("visitors") + 1* increases the values of **visitors** variable at the start of each session. The file **TestSession.asp** displays the values of this application variable in the browser.

Locking and Unlocking

The application variables defined in **global.asa** can be accessed and updated by any page in the whole website. In some situations, many pages may try to update the value of an application variable. It may create some consistency problems. The **Lock** and **Unlock** methods of application object are used to avoid such problem.

The **Lock** method locks the application before updating the value of application variable. While the value is being updated, no other page can update its value. After the value is updated successfully, the **Unlock** method is used. The Unlock method again allows some other page to update the value of the application variable. The code of TestSession.asp can be writted as follows:

```
<%
  Application.Lock
  Response.Write ("You are visitor number " & application("visitors"))
  Application.Unlock
%>
```

Session Time Out

The **Timeout** property of **Session** object is used to specify the time period after which the session will expire. It is given in seconds. The script in **Session_OnEnd** is executed when the session expires. The script can be used to manage sessions effectively.

18.2.6 Server Object

The **Server** object is used to access the resources on the server.

Properties

An important property of Server object is as follows:

ScriptTimeout It is the amount of time that a script can run before the server terminates it. It is given in seconds. The default value is 90 seconds.

Example: The following example will display the default script time out and then set it to 150 seconds:

```
<%
    Response.Write(Server.ScriptTimeout)
    Server.ScriptTimeout = 150
%>
```

Methods

Some important methods of Server object are as follows:

1. CreateObject It creates an instance of a server component. An object created with this method has page scope. It is destroyed when the server finishes the processing of current ASP page.

Example: The following example will create a connection object:

```
<%
    Set con = Server.CreateObject("ADODB.Connection)
%>
```

2. Execute It executes an ASP file within another ASP file. The control returns to the original file after executing the ASP file.

Example: The following example will execute another ASP file within the current ASP file:

```
<%
    Response.Write("Original file.<BR>")
    Server.Execute("file2.asp")
    Response.Write("Original file again.")
%>
```

3. HTMLEncode It applies HTML encoding to the specified string.

Example: The following example will apply HTML encoding to a string:

```
<%
    Response.Write(Server.HTMLEncode("Bold tag: <B>"))
%>
```

4. MapPath It maps the specified virtual path, either the absolute path on the current server or the path relative to the current page, into a physical path. It cannot be used in Session.OnEnd and Application.OnEnd.

Example: The following example will maps the path into physical path:

```
<%
    Response.Write(Server.MapPath("about.asp"))
%>
```

5. Transfer It is used to send all the state information created in one ASP file to a second ASP file such as application or session variables and all items in the request collections. The control does not return to original ASP page after the second ASP page completes.

Example: The following example will transfer the state information from one file to another:

```
<%
    Response.Write("Original file.<BR>")
    Server.Transfer("info.asp")
    Response.Write("Original file again<BR>")
%>
```

The third line in the above example will not be executed because the control does not return to original file from **info.asp** after transfer.

6. URLEncode It applies URL encoding rules including escape characters to the string.

Example: The following example will apply URL encoding to given URL:

```
<%
    Response.Write(Server.URLEncode("http://www.hotmail.com"))
%>
```

Creating Objects

There are five Server Components in Active Server Pages:

- AdRotator
- Browser Capabilities
- Content Linking
- TextStream
- Database Access Component

The **Server.CreateObject** method is used to create instances of these components. Each component produces an object. The object contains predefined properties and methods.

HTMLEncode

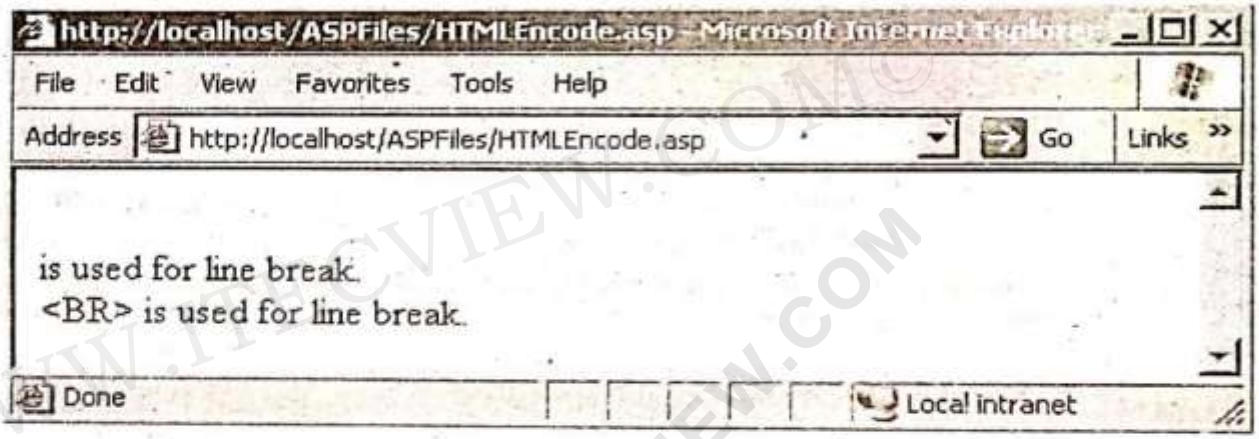
The **Server.HTMLEncode** is used to display HTML tag in browser. It dispalys actual HTML source code in the browser. For example if you want to display *<H1> is heading 1* in the browser, it may not appear correctly using simple *Reponse.Write* method.

Practice 18.6

1. Open Notepad and type the following code:

```
<%
  Response.Write ("<BR> is used for line break. <BR>")
  Response.Write (Server.HTMLEncode("<BR> is used for line break."))
%>
```

2. Save the file as "HTMLEncode.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/HTMLEncode.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



How it Works

In the above practice, first statement *Response.Write ("
 is used for line break.
")* does not display <BR> tag in the output. Instead, it inserts a line break. In the second statement *Response.Write (Server.HTMLEncode("
 is used for line break."))* the <BR> tag is displayed in the browser. The *Server.HTMLEncode* method is used for this purpose.

Using MapPath

In an ASP application, all files are stored in a virtual directory. Any file stored in the virtual directory can be accessed by using **MapPath** method relative to the virtual directory. For example the physicall directory **C:\Inetpub\wwwroot** is accessed by **http://localhost/** virtual path. The syntax of using this method is as follows:

Server.MapPath(path)

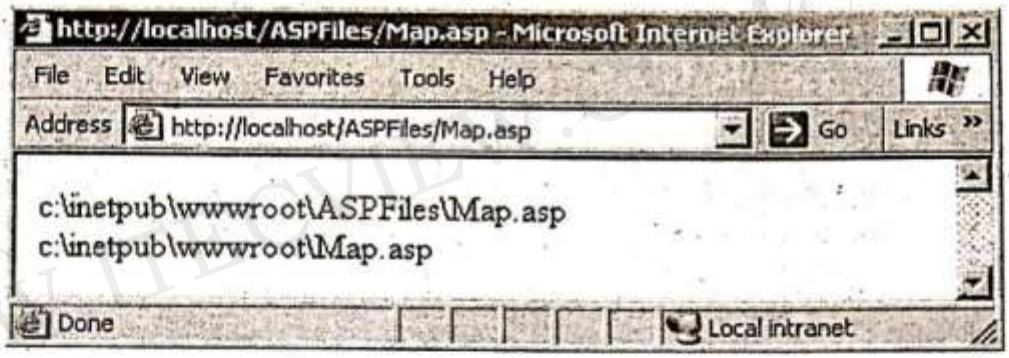
The **path** is a string that specifies the virtual or physical directory. If the path starts with slash or backslash, the MapPath method will return the full virtual path. If slash is not used, MapPath will return a parth relative to the directory of the ASP file being used.

Practice 18.7

1. Open Notepad and type the following code:

```
<%
    Response.Write (Server.MapPath("Map.asp") & "<BR>")
    Response.Write (Server.MapPath("/Map.asp"))
%>
```

2. Save the file as "Map.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Map.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



18.3 Server Side Includes (SSI)

Server Side Includes or **SSI** is a very simple programming language. It has a very limited number of instructions. An important instruction is used to include existing files in web pages. The syntax of including a file is as f ollows:

```
<!-- #INCLUDE FILE="filename" -->
```

The purpose of #include directive is to include the an existing file at the given location. Any script that is present within that file is automatically executed.

Include files are normally used to store application-level functions and procedures etc. For example, a function **ValidateEmail ()** for checking valid email address can be stored in a file. The file can be inlcuded in different webpages for validating email addresses.

Another advantage of including file is that you can insert the same contents on every page. For example, you can create a file that displays a poll. The file can be included in many pages without any need to copy and paste it everywhere. Updates are also easier as one file is modified and the change is automatically reflected everywhere.

Another form of the INCLUDE directive uses the keyword **virtual**:

```
<!-- #INCLUDE VIRTUAL="/directory/filename" -->
```

The above line will locate the file considering it as the virtual path. The file is expected to be found at *www.domain.com/directory/filename*

Practice 18.8

1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<H2>Welcome to SSI</H2>
<FORM NAME="form1" METHOD="post" ACTION="Enter.asp">
Name: <INPUT TYPE="text" NAME="name"><P>
Who are you?
<SELECT NAME="status">
<OPTION VALUE="Teacher">Teacher
<OPTION VALUE="Student">Student
</SELECT>
<P>
<INPUT TYPE="submit" VALUE="Send Data" NAME="submit">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Enter.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.

3. Open another file in Notepad and type the following code:

```
<%
        Dim n, s
        n = Request.Form("name")
        s = Request.Form("status")
        Response.Write("Hello " & n)
        if s = "Teacher" then
%>
          <!--#include virtual="/ASPFiles/Teachers.html" -->
        <% else %>
          <!--#include file="Students.html" -->
        <% end if %>
```

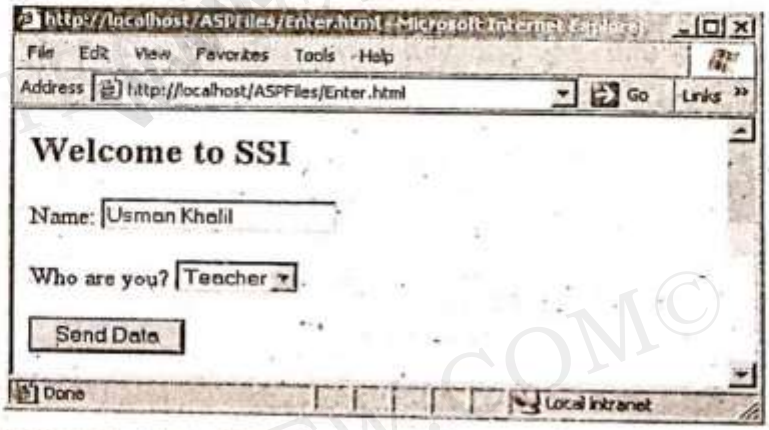
4. Save the file as "Enter.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Open another file in Notepad and type the following code:

```
<HTML>
<BODY>
        Welcome to Teacher's Page...
</BODY>
</HTML>
```

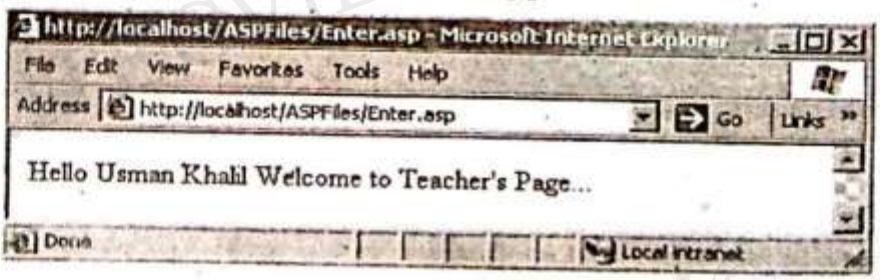
6. Save the file as "Teachers.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
7. Open another file in Notepad and type the following code:

```
<HTML>
<BODY>
        Welcome to Student's Page...
</BODY>
</HTML>
```

8. Save the file as "Students.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
9. Type **http://localhost/ASPFiles/Enter.html** in web browser and press **Enter** key. The page will appear in the browser. Enter the data as follows:



10. Click on **Send Data** button. The page will appear as follows:



Microsoft Access & SQL

19.1 Database

Database is an **organized** collection of **related** data that is stored in an **efficient** and **compact** manner. The word **organized** means that data is stored in such a way that the user can use this data easily. The word **related** means that a database is normally created to store the data about a particular topic. The word **efficient** means that the user can search the required data quickly. The word **compact** means that the stored data occupies as little space as possible in computer.

19.1.1 Database Management System

A **database management system** (DBMS) is a collection of computer programs that are used to create, maintain and access databases in a convenient and efficient manner.

19.1.2 Relational Databases

A database in which data is stored in relations is called **relational database.** Relation is another term used for table. A table in a database has unique name that identifies its contents.

Characteristics

Two important characteristics of relational databases are as follows:

- All data in a relational database is stored in tables.
- All data from a relational database is retrieved as tables.

19.2 Primary Key

A **primary key** is field or group of fields that is used to identify each row uniquely within a relation. Primary key can be one of the following:

- Simple primary key
- Composite primary key

If primary key consists of single field, it is called **simple primary key.** If primary key consists of two or more fields, it is called **composite primary key.** Most important points about a primary key are:

- A table can have only one primary key.

- Each value in primary key field must be unique.
- Primary key cannot contain null value.

Example

Suppose a relation Student contains different attributes such as RegNo, Name and Class. The attribute RegNo uniquely identifies each student in the table. It can be used as primary key for this table. The attribute Name cannot uniquely identify each row because two students can have same name. It cannot be used as primary key.

Primary Key

RegNo	Name	Class
10	Nadeem Khalil	MSc
20	Muhammad Usman	BSc
30	Noman Qadir	FSc

Figure: Primary Key

19.3 Foreign key

A **foreign key** is an attribute or set of attributes in a relation whose values match a primary key in another relation. The relation in which foreign key is created is known as **dependent table** or **child table**. The relation to which the foreign key refers is known as **parent table**. The key connects to another relation when a relationship is established between two relations. A relation may contain many foreign keys.

The following figure shows two relations. The RollNo attribute in Parent relation is used as primary key. The RollNo attribute in Child relation is used as foreign key. It refers to RollNo attribute in Parent relation.

Primary Key

RollNo	Name	Address
1	Ali	Faisalabad
2	Ahmad	Lahore
3	Nadeem	Peshawar

Parent Table

Composite Primary Key

RollNo	Subject	Marks
1	English	52
1	Math	77
1	Computer	64
2	English	58
2	Math	69
2	Computer	49
3	English	82
3	Math	98
3	Computer	86

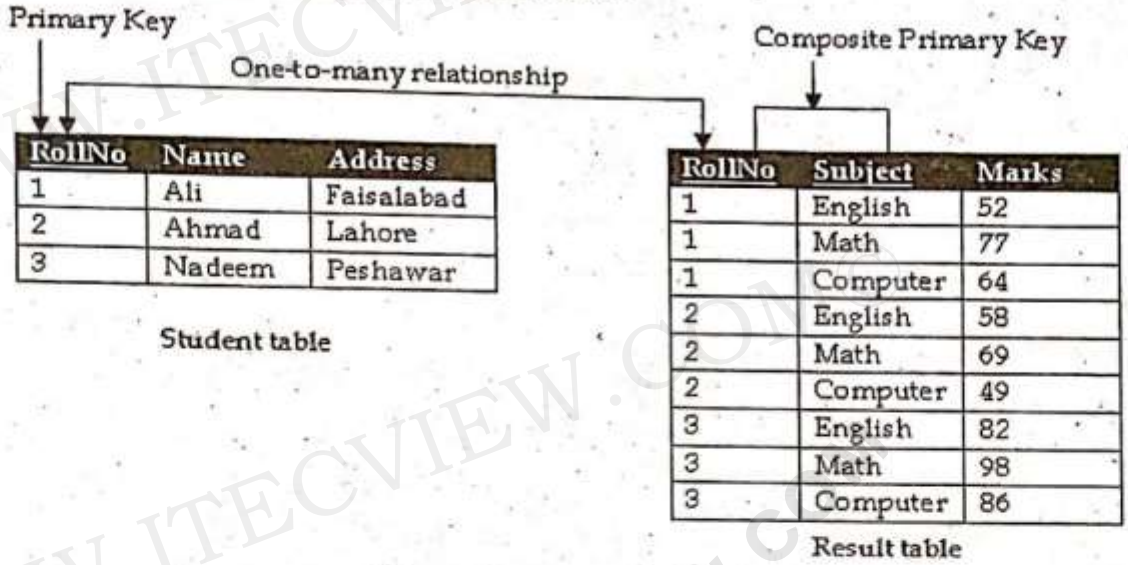
Child Table

19.4 Referential Integrity

Referential Integrity means that the user cannot enter a value in a foreign key field that does not have a corresponding value in the primary key field of the related table.

Example

The following figure shows two tables. The master table **Student** contains **RollNo, Name** and **Address** of students. The child table **Result** contains **RollNo, Subject** and **Marks** of students. Both tables are connected with RollNo.



RollNo	Name	Address
1	Ali	Faisalabad
2	Ahmad	Lahore
3	Nadeem	Peshawar

RollNo	Subject	Marks
1	English	52
1	Math	77
1	Computer	64
2	English	58
2	Math	69
2	Computer	49
3	English	82
3	Math	98
3	Computer	86

Figure: Two rows of a Relation

The user cannot enter record in **Result** table without first entering the data of the same record in **Student** table. Suppose the result of a student with RollNo 1 is to be entered. The user will first enter his record in Student table and then he can enter the detail of his marks in Result table.

Similarly, the user cannot delete record in Child table without first deleting the data of the same record in the Parent table. For example, the user cannot delete record in Parent table if its related record exists in the Child table.

19.5 Database Creation in MS Access

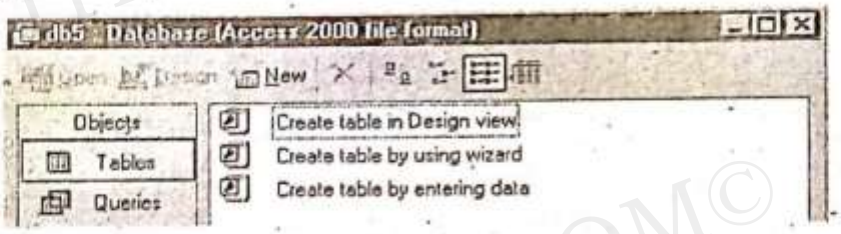
A new database in MS Access can be created in two ways:

- Using dialog box
- Using main menu

Creating New Database using Dialog Box

1. Start **Microsoft Access**. It will display a dialog box as follows:

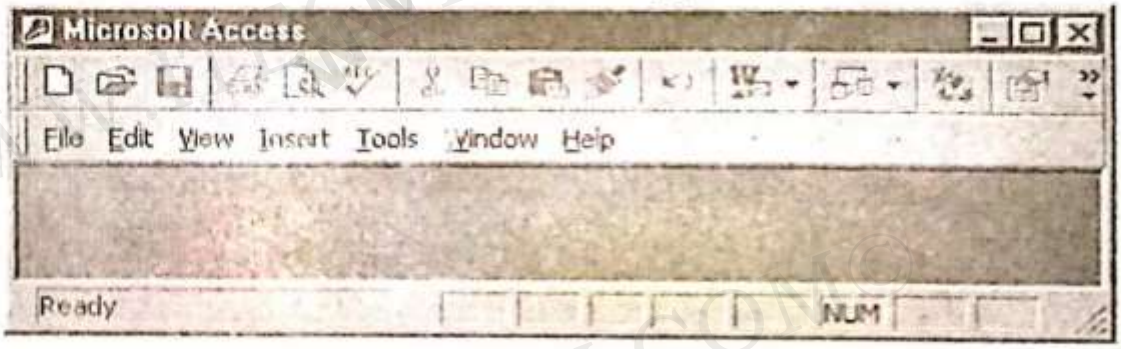
2. Select the **Blank Access database** option.
3. Click on the **Ok** button. A new dialog box will appear as follows:
4. Type a file name in **File name** box.
5. Select any location from **Save in** option.
6. Click on the **Create** button. The database will be created and following **Design view** will appear.



New Database using Main Menu

1. Start Microsoft Access. A dialog box will appear as follows:

2. Click **Cancel** button. The main window will appear follows:



3. From the **File** Menu, choose **New**. The **New** dialog box will appear.

4. On the **General Tab**, Choose **database**.

5. Click **ok**. The **File New Database** dialog box will appear.
6. Type a file name in **File name** box.
7. Select any location from **Save in** option.
8. Click on **Create** button. The database will be created and **Design view** will appear.

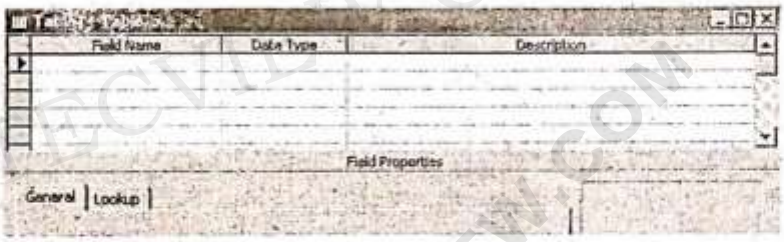
19.6 Table Creation in MS Access

A table is created in a database. The option **Create table in Design view** is the most common way of creating a table.

Creating Table in Design View

Design View is used to define fields of a table. The window is divided into two parts:

1. Create new blank database.
2. Double click **Create table in Design view.** The Design View will be displayed as follows:



3. Type the name of the first field in **Field Name** column.
4. Press tab key to move to **Data Type** column. Select required data type for the field.
5. Press **Tab** key to move to **Description** column. It is used to enter comments about the field. This is optional.
6. Press **Tab** key to move to the **Field Name** for the next field.
7. Repeat the above steps for entering any number of fields in the table.

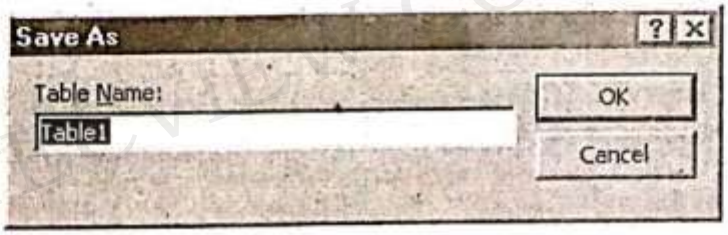
Setting Primary key

1. Select the field for primary key.
2. On the **Table Design** toolbar, click **Primary Key** OR

- From the **Edit** menu, select **Primary Key**. The row that is chosen as the primary key is marked with a small symbol in the selector button.

3. On the **Table Design** toolbar, click **Save** OR

- From **File** menu, select **Save**. The **Save As** dialogue box will appear.



- Enter a table name and click **OK**. The table will be saved. The new table will appear in the main database as follows:

Defining Composite Primary Key

A primary key that consists of two or more fields is called **composite primary key**. Composite primary key can be defined on a table as follows:

1. Open a table in **Design view**.
2. Click on the field selector button of the first field that will be used as primary key.
3. Press **CTRL** key and click on the field selector button of the second field that will be used as primary key. Both fields will be selected.
4. On the **Table Design** toolbar, click **Primary Key** OR

- From the **Edit** menu, select **Primary Key**. Both fields that are selected as primary key will contain a small symbol in the selector button.

19.7 Data types

MS Access provides the following data types:

- **Text:** It is used to store alphabetic, numbers and special characters. It can store up to 255 characters. The default setting is 50 characters.
- **Memo:** Memo data type is used to store lengthy text. It is normally used to store comments etc. Memo fields can hold up to 64,000 characters.
- **Number:** It is used to store numeric data. The fields with Number data type can be used in mathematical calculations. Different types of numeric data types are as follows:

Data Type	Values	Size in bytes
Byte	0 to 255	1
Integer	-32,768 to 32,767	2
Long Integer	-2,147,483,648 to 2,147,483,647	4
Single		4
Double		8

The **Byte, Integer** and **Long Integer** data types cannot store decimal values. If you need to store values with up to four decimal places only, use the **Currency** data type in preference to **Single** or **Double** number fields.

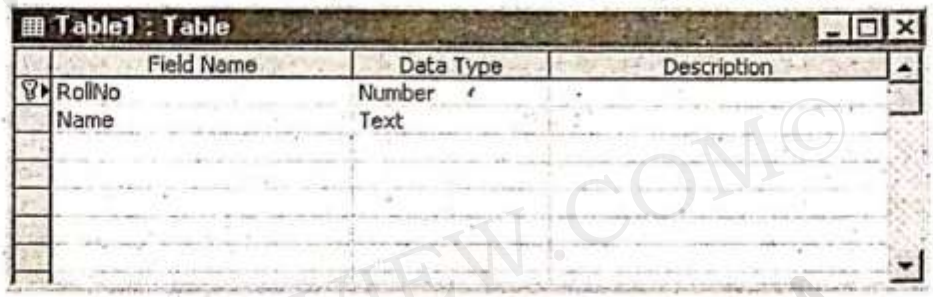
- **Date/ Time:** Date/Time data type is used to store date and time. MS Access stores date in the standard mm/dd/yy format.
- **Currency:** It is used to store numbers representing currency. The value in Currency field is rounded to two decimal places. Negative currency is displayed in brackets.
- **AutoNumber:** It is used to generate the next number automatically when a new record is added. It creates a unique number for each record. The value starts from 1 and is incremented by 1 in each record.
- **Yes/No:** It is used to store Boolean value. The possible values in this field are **True** and **False**.

19.8 Views in MS Access

Table view is a way of looking at the table. MS Access provides two table views that are as follows:

1. Design View

The table view that is used to design the structure of a table is called **design view**. It is used to specify name, data types and description of fields. Primary key is also specified in this view. The structure of an existing table can also be changed in design view.



2. Datasheet View

The table view that is used to enter, delete or modify data in a table is called **datasheet view**. The table in this view is displayed in rows and columns. The name of each field is displayed at the top of the column.

Following procedure is used to switch between table views:

1. Click on **View** menu.
2. Select **Datasheet View** or **Design View** from the menu. The table will be displayed according to the selected view option.

19.9 Opening a Table

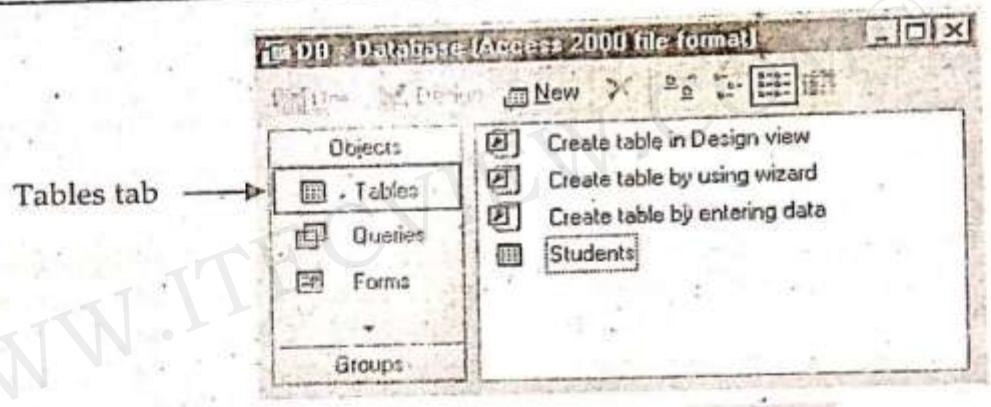
A table that has been created in a database can be opened in two views:

1. Design View
2. Datasheet View

1. Opening Table in Design View

The **design view** is used to create or modify the structure of a table. An existing table can be opened in design view as follows:

- Open the database that contains the table.
- In **database window**, click on **Tables** tab. All tables in the database will appear in the **database window**.



3. Select the table and click on **Design** button Design to open it in design view.

2. Opening Table in Datasheet View

The **datasheet view** is used to add, delete or modify data in a table. An existing table can be open in datasheet view as follows:

1. Open the database that contains the table.
2. In **database window**, click on **Tables** tab. All tables in the database will appear in the window.
3. Double click the table to open it in datasheet view. OR

- Select the table and click on **Open** button Open.

19.10 Adding Data to a Table

The data in a table can be entered by performing the following steps:

1. Open the table in datasheet view.
2. Click on the first field of the first record.
3. Type in the data.
4. Press **Enter** key. The cursor will move to the next field of the current record.
5. Enter data in all fields press **Enter** key. The cursor will move to the first field of the second record.

If a table is opened that already contains a large number of records, a new record can be added in it as follows:

1. Open the table in datasheet view.
2. Right click on record selector button. A popup menu will appear.
3. Select **New Record** option. A new record will be added at the end of the table and the cursor will move in the first field of new record.
4. Enter data in all fields and press **Enter** key.

19.11 Deleting a Record from Table

The deletion of record from a table is very simple. The following procedure is used to delete record from table:

1. Open the table in **datasheet view**.
2. Right click on the record selector at the left of the record to be deleted. A popup menu will appear.
3. Choose **Delete Record** option. A message will appear to confirm the deletion of the record.
4. Click on **Yes** button to remove the record. The record will be deleted permanently.

19.12 SQL

SQL stands for **Structured Query Language**. SQL is a standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, etc.

You can execute SQL statements called queries to retrieve, update, insert, and delete records from tables in database.

- **SELECT** - extracts data from a database table
- **UPDATE** - updates data in a database table
- **DELETE** - deletes data from a database table
- **INSERT INTO** - inserts new data into a database table

19.12.1 The SELECT Statement

The SELECT statement is used to select data from a table. The tabular result is stored in a result table.

Syntax

```
SELECT column_name(s)FROM table_name;
```

Select Some Columns

To select the columns named "LastName" and "FirstName", use a SELECT statement like this:

```
SELECT LastName, FirstName FROM Authors;
```

"Authors" Table

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer

Result

LastName	FirstName
Mahmood	Tariq
Saeed	Imran
Mustafa	Tasleem
Raza	Ahsan

Select All Columns

To select all columns from the "Authors" table, use a * symbol instead of column names, like this:

```
SELECT * FROM Authors;
```

WWW.ITECVIEW.COM©

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer

19.12.2 The WHERE Clause

WHERE clause is used to select data from a table conditionally.

Syntax

```
SELECT column FROM table WHERE condition;
```

The following operators can be used with the WHERE clause:

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between a range
LIKE	Search for a pattern

Example 19.1

To select only those authors whose designation is "lecturer", we add a WHERE clause to the SELECT statement:

```
SELECT * FROM Authors WHERE Designation='Lecturer';
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer

Using Quotes

SQL uses single quotes around text values. Most database systems will also accept double quotes. Numeric values should not be enclosed in quotes.

The LIKE Condition

The LIKE condition is used to specify a search for a pattern in a column.

Syntax

```
SELECT column FROM table WHERE column LIKE pattern;
```

A "%" sign can be used to define wildcards (missing letters in the pattern) both before and after the pattern.

Example 19.2

The following SQL statement will return authors with first names that start with an 'O':

SELECT * FROM Authors WHERE FirstName LIKE 'O%';

The following SQL statement will return authors with first names that end with an 'a':

SELECT * FROM Authors WHERE FirstName LIKE '%a'

The following SQL statement will return authors with first names that contain the pattern 'ri':

SELECT * FROM Authors WHERE FirstName LIKE '%ri%'

19.12.3 The INSERT INTO Statement

The INSERT INTO statement is used to insert new rows into a table.

Syntax

INSERT INTO table_name VALUES (value1, value2,....)

You can also specify the columns for which you want to insert data:

INSERT INTO table_name (column1, column2,...) VALUES (value1, value2,....)

Example 19.3

This SQL statement:

INSERT INTO Authors VALUES ('Bayross', 'Ivan', 'G-11', 'London', 'Professor')

will add a new record in the table:

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer
Bayross	Ivan	G-11	London	Professor

Insert Data in Specified Columns

You can also insert values only in the specified fields. For example, if you want add only FirstName and Designation in a new record then you have to mention the column names in the Insert statement.

Example 19.4

This SQL statement:

INSERT INTO Authors (LastName, Designation) VALUES ('Deitel', 'Professor')

will give this result:

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer
Bayross	Ivan	G-11	London	Professor
	Deitel			Professor

19.12.4 The Update Statement

The UPDATE statement is used to modify the data in a table.

Syntax

```
UPDATE table_name SET column_name = new_value
WHERE column_name = some_value;
```

Update one Column in a Row

We want to modify the last name of the person with first name of "Ahsan":

```
UPDATE Authors SET LastName = 'Sattar'
WHERE FirstName = 'Ahsan';
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Sattar	Ahsan	Peoples Colony	Faisalabad	Lecturer
Bayross	Ivan	G-11	London	Professor
	Deitel			Professor

Update Several Columns in a Row

We want to change the address and add the name of the city:

```
UPDATE Authors SET Address = 'Zero Point', City = 'New York'
WHERE LastName = 'Bayross'
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer
Bayross	Ivan	Zero Point	New York	Professor
	Deitel			Professor

19.12.5 The DELETE Statement

The DELETE statement is used to delete rows in a table.

Syntax

```
DELETE FROM table_name WHERE column_name = some_value
```

Example 19.5

"Deitel" will be deleted with the following statement:

```
DELETE FROM Authors WHERE FirstName = 'Deitel';
```

Delete All Rows

It is possible to delete all rows in a table without deleting the table. This means that all data in the table will be deleted but the table structure, attributes, and indexes will be intact:

DELETE FROM table_name; or DELETE * FROM table_name;

Sort the Rows

The ORDER BY clause is used to sort the rows.

Example 19.6

The following statement will display the authors in alphabetical order with respect to first name:

```
SELECT * FROM Authors ORDER BY FirstName;
```

Result

LastName	FirstName	Address	City	Designation
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Bayross	Ivan	Zero Point	New York	Professor
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor

Example 19.7

You can give more than one column for sorting. The preference is given to first column. If values in the first column are similar then the records will be sorted with respect to the second column. The following statement will display the table in descending order with respect to the Designation. If the designation is similar then the records will be sorted by LastName.

```
SELECT * FROM Authors ORDER BY Designation, LastName DESC;
```

Result

LastName	FirstName	Address	City	Designation
Bayross	Ivan	Zero Point	New York	Professor
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer
Raza	Ahsan	Peoples Colony	Faisalabad	Lecturer
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Mustafa	Tasleem	Gulistan Colony	Faisalabad	Assistant Professor

AND & OR

AND and OR are used to give two or more conditions in a WHERE clause. The AND operator displays a row if ALL conditions listed are true. The OR operator displays a row if ANY of the conditions listed are true.

Example 19.8

The following query uses AND to display the author with the first name equal to "Tariq", and the last name equal to "Mahmood":

```
SELECT * FROM Authors
WHERE FirstName='Tariq' AND LastName='Mahmood';
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer

Example 19.9

The following example uses OR to display the author with the first name equal to "Tariq", or the last name equal to "Saeed":

```
SELECT * FROM Authors WHERE FirstName='Tariq' OR LastName='Saeed';
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer

Example 19.10

AND and OR can be combined in one statement and parentheses are used in complex expressions:

```
SELECT * FROM Authors WHERE (FirstName='Tariq' OR FirstName='Imran')
AND Designation = 'Lecturer';
```

Result

LastName	FirstName	Address	City	Designation
Mahmood	Tariq	Madina Town	Faisalabad	Lecturer
Saeed	Imran	Gulistan Colony	Faisalabad	Lecturer

BETWEEN...AND

The BETWEEN...AND operator selects a range of data between two values. These values can be numbers, text, or dates.

```
SELECT column_name FROM table_name
WHERE column_name BETWEEN value1 AND value2
```

Students Table

Roll No	Name	Marks	Grade
1	Usman	100	A
2	Ali	55	C
3	Nadeem	72	B
4	Adnan	61	B
5	Waqar	90	A

Example 19.11

The following query will display the students with marks between 60 and 80:

```
SELECT * FROM Students WHERE Marks BETWEEN 50 AND 80;
```

Result

Roll No	Name	Marks	Grade
2	Ali	55	C
3	Nadeem	72	B
4	Adnan	61	B

Example 19.12

The following example will display the students outside the range used in the previous example by using the NOT operator:

SELECT * FROM Students WHERE Marks NOT BETWEEN 50 AND 80;

Result

Roll No	Name	Marks	Grade
1	Usman	100	A
5	Waqar	90	A

19.13 Functions

There are different built-in functions that can be used to perform different tasks easily. The syntax for built-in SQL functions is:

SELECT function(column) FROM table;

19.13.1 Types of Functions

The basic types of functions are as follows:

1. Aggregate Functions

Aggregate functions operate against a collection of values, but return a single value.

Function	Description
AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
LAST(column)	Returns the value of the last record in the specified field
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
SUM(column)	Returns the total sum of a column

Scalar Functions

Scalar functions operate against a single value, and return a single value based on the input value.

Function	Description
UCASE(c)	Converts a field to upper case
LCASE(c)	Converts a field to lower case
MID(c,start[,end])	Extract characters from a text field
LEN(c)	Returns the length of a text field
INSTR(c)	Returns the numeric position of a named character within a text field
LEFT(c,number_of_char)	Return the left part of a text field requested
RIGHT(c,number_of_char)	Return the right part of a text field requested
ROUND(c,decimals)	Rounds a numeric field to the number of decimals specified
MOD(x,y)	Returns the remainder of a division operation

Example 19.13

The following example will get the sum of all students:

SELECT SUM(Marks) FROM Students;

The following example will get the marks of the student with maximum marks:

SELECT MAX(Marks) FROM Students;

Database Connectivity in ASP

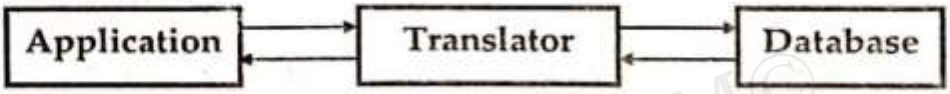
20.1 Database Connectivity in ASP

Database is the source of data for any application. Web applications also use databases for this purpose. The way of accessing data in Web application may be different from a traditional client-server application.

Each DBMS stores data in a particular format. For example, Microsoft Access stores data in the 'mdb' file while FoxPro uses the 'dbf' format etc. Traditionally, a database application was developed for a particular type of database. If you change the DBMS, you also need to rewrite the application to handle data in a new format.

A method was required for accessing databases such that the application remains unchanged even the databases is changed. This is known as **open database connectivity.**

It is possible to build applications that are able to communicate with different databases. This is possible if the application gives a command in a particular manner. This command is then translated such that it is understood by the DBMS. By using different translators between the application and the DBMS, we can communicate with databases created by different DBMS.



20.2 Open Database Connectivity

ODBC uses a standard method for communication between the application and the DBMS. This is accomplished by defining an **Application Programming Interface (API).** An API in programming terms means a set of functions that perform a particular task.

The application will call an API function to specify a particular task. The API call is passed on to the translator. The translator issues command in the language of DBMS to specify the requested action.

The translator is referred to as the **ODBC Driver.** It is the driver that implements the various API such that they are understood by the DBMS. Thus, there is a driver for each DBMS to be accessed.

The **ODBC Driver** accepts the command from the application and converts it to a format understood by the target DBMS. In addition, the **ODBC Driver** also receives the result of the command execution from the DBMS and passes it back to the application.

20.3 OLE DB

ODBC defined API that could be used to access any data but it did not support the object orientation approach. **OLE DB** is a database architecture that enables applications to have uniform access to data stored in diverse information sources, regardless of the location or type. **OLE DB** defines a set of **COM** (Component Object Model) interfaces that enable a uniform access to data stored in diverse DBMS.

20.4 ActiveX Data Object

ActiveX Data Object (ADO) is a Microsoft technology that is used to access and manipulate databases. It works with OLE DB for providing interfaces between programs and data sources. This is the latest technology for connecting to different types of data sources like Microsoft Access, Oracle, SQL Server and many other data sources.

ADO is ActiveX-based and it can work in different platforms and different programming languages. It can access many different kinds of data such as data displayed in the Internet browsers, email text and even graphics other than the usual relational and non-relational database information.

20.5 The ADO Object Model

ADO is the most complex components of ASP. ADO provides three primary objects to the developer. These objects are Connection Object, Command Object and RecordSet Object.

1. Connection Object

This object represents an open connection. It is used to establish a connection between data source and Web application. You can access and manipulate database with this object.

2. Command Object

This object defines a specific command you wish to execute against data.

3. Recordset Object

This object represents a set of records from a table, command object, or SQL query. It can be created independently, without any underlying Connection object.

20.6 Connection Object

The ADO Connection object is used to create an open connection to a data source to access and manipulate a database. This object is not essential for manipulating a database. You can also connect to a database by using Command or Recordset object. However, such a connection is only good for a particular single query. If you desire to access a data source multiple times, it is far more efficient to establish a connection using the Connection object.

20.6.1 Properties of Connection Object

Some important properties of Connection object are as follows:

Attributes

Sets or returns a long value defining the characteristics of a **Connection** object.

Syntax: ConnectionObject.Attributes

CommandTimeout Property

It sets the number of seconds to wait while attempting an Execute method. If the attempt is not successful within specified time, an error message is generated.

Syntax: ConnectionObject.CommandTimeout

ConnectionString Property

It sets or returns a string value that contains the details used to create a connection to a data source.

Syntax: ConnectionObject.ConnectionString

Property	Description
Provider	Specifies the name of the provider used for the connection.
Data Source	Specifies the name of a data source for the connection.
User ID	Specifies the user name to open a connection.
Password	Specifies the password to open a connection.

ConnectionTimeout Property

It sets the number of seconds to wait while attempting to create a connection. If connection is not established within specified time, an error message is generated.

Syntax: ConnectionObject.ConnectionTimeout

DefaultDatabase Property

It sets or returns a string value that is the default name of the database available from the provider for a Connection object.

Syntax: ConnectionObject.DefaultDatabase

Provider Property

It sets or returns the string value that is the provider name.

Syntax: ConnectionObject.Provider

State Property

It returns a value describing if the connection is open or closed.

Syntax: ConnectionObject.State

Version Property

It returns a string value that is the ADO version number.

Syntax: ConnectionObject.Version

20.6.2 Collections of Connection Object

Some important collections of Connection object are as follows:

Errors Collection

This is a collection of Error objects that were generated by the failure of a connection operation.

Properties Collection

This is a collection of Property objects.

20.6.3 Methods of Connection Object

Some important methods of Connection object are as follows:

BeginTrans Method

It begins a new transaction and returns a long value indicating the number of nested transactions.

Syntax: ConnectionObject.BeginTrans

Close Method

It closes a connection.

Syntax: ConnectionObject.Close

CommitTrans Method

It saves any changes and ends the current transaction. It can also be set to automatically start a new transaction.

Syntax: ConnectionObject.CommitTrans

Execute Method

It executes the query, SQL statement, stored procedure, or provider-specific text.

Syntax: ConnectionObject.Execute (CommandText, RecordsAffected, Options)

Open Method

It opens a physical connection to a data source.

Syntax: ConnectionObject.Open (ConnectionString, UserID, Password, Options)

Parameters	Description
ConnectionString	Optional. A string containing the Connection information.
UserID	Optional. User name required to establish the connection.
Password	Optional. Password required when establishing the connection.

RollbackTrans Method

Cancels any changes that have occurred during the current transaction and then ends the transaction. It can also be set to automatically start a new transaction.

Syntax: ConnectionObject.RollbackTrans

Practice 20.1

1. Create an Access database **Sample.mdb** in **C:\Inetpub\wwwroot\ASPFiles** folder.
2. Create table **Test** in the database with the following fields.

Field	Data Type
RollNo	Number
Name	Text
Age	Number

3. Insert the following data in the table.

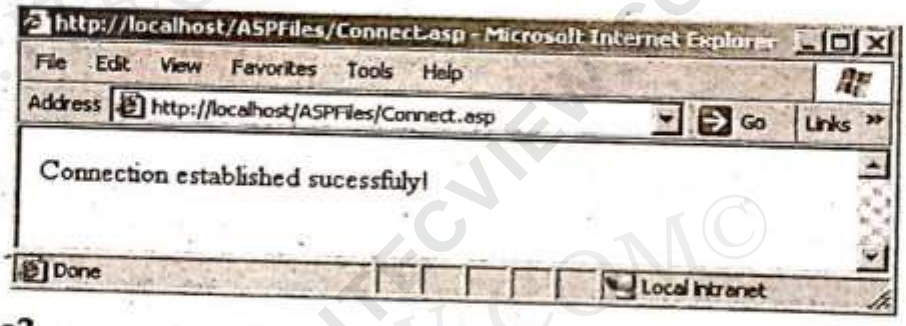
RollNo	Name	Age
1	Usman Khalil	11
2	Nadeem	23
3	Adnan	22

4	Waqar	24
5	Abdullah	9

4. Open Notepad and type the following code:

```
<%
  Dim conn
  Set conn = Server.CreateObject("ADODB.Connection")
  conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" _
  & "Data Source=" & Server.MapPath("Sample.mdb")
  if conn.State then
    Response.Write ("Connection established sucessfuly!")
  else
    Response.Write ("No connection established!")
  end if
  conn.close
%>
```

5. Save the file as "Connect.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
6. Type **http://localhost/ASPFiles/Connect.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



How It Works?

The first line in the above practice declares a connection variable **conn**. The second statement uses built-in method to open the connection. It includes the following:

- **Provider** – Name of the provider. We are using Microsoft.Jet.OLEDB.4.0.
- **Data Source** – Name of and path of the data source for establishing connection.

If the connection establishes successfully, **State** property becomes **true**. So a message appears in the browser indicating that the connection has been established.

20.7 Recordset Object

The ADO Recordset object is used to store the data retrieved from a database. This object consists of records and fields. The Recordset object is considered to be the heart of ADO. We can retrieve and modify data by using this object.

The Recordset object contains many properties, collections, methods, and events that are used to manipulate the retrieved data.

20.7.1 Properties of Recordset Object

Some important properties of Recordset object are as follows:

ActiveCommand Property

It returns a variant that is the Command object associated with the Recordset object.

Syntax: RecordsetObject.ActiveCommand

ActiveConnection Property

It sets or returns a variant defining the Connection object to which the Recordset belongs, or returns a string value if there is no connection.

Syntax: RecordsetObject.ActiveConnection

BOF Property

It returns a Boolean value indicating if the current record position is before the first record or not.

Syntax: RecordsetObject.BOF

CursorType Property

It sets or returns the type of cursor being used.

Syntax: RecordsetObject.CursorType

Cursor Type	ADO Constant	Description
Forward- Only	adOpenForwardOnly	It behaves like dynamic cursor but allows scrolling only forward through records.
Keyset	adOpenKeyset	It prevents from seeing records other users add, and prevents access to records other users delete.
Dynamic	adOpenDynamic	It allows additions, changes, deletions and allows all types of navigation of Recordset.
Static	adOpenStatic	It allows all types of movement with Recordset.

EOF Property

It returns a Boolean value indicating if the current record position is after the last record or not.

Syntax: RecordsetObject.EOF

Filter Property

It sets or returns a value that is used to filter data.

Syntax: RecordsetObject.Filter

LockType Property

It sets or returns the type of locks that are in effect while editing records.

Syntax: RecordsetObject.LockType

Lock Type	Description
AdLockReadOnly	(Default) Read-only. Data cannot be altered.
AdLockPessimistic	The page containing record is locked immediately on editing.
AdLockOptimistic	The page containing the record is locked only when the records are updated.
AdLockBatchOptimistic	It uses BatchUpdate method to write the entire batch of rows to the Recordset.

MaxRecords Property

It sets or returns the maximum number of records that can be returned to a Recordset object as the result of a query.

Syntax: RecordsetObject.MaxRecords

RecordCount Property

It returns the number of records in a Recordset object.

Syntax: RecordsetObject.RecordCount

Sort Property

It sets or returns a string value that is the names of fields in the Recordset to sort.

Syntax: RecordsetObject.Sort

State Property

It returns a value describing if the Recordset object is open, closed, or in the process of connecting, executing, or retrieving.

Syntax: RecordsetObject.State

20.7.2 Collections of Recordset Object

Some important collections of Recordset object are as follows:

Fields Collection

A collection containing all of the Field objects associated with a Recordset object.

Properties Collection

A collection of Property objects. Each Property object contains a provider-specific property.

20.7.3 Methods of Recordset Object

Some important methods of Recordset object are as follows:

AddNew Method

It is used to create a new record.

Syntax: RecordsetObject.AddNew (FieldList, Values)

Close Method

It closes a Recordset object.

Syntax: RecordsetObject.Close

Delete Method

It deletes the current record, a group of records, or all records.

Syntax: RecordsetObject.Delete (AffectRecords)

Find Method

It searches for a row in a Recordset that matches the given criteria.

Syntax: RecordsetObject.Find (Criteria, SkipRows, SearchDirection, Start)

Move Method

It moves the position of the current record pointer.

Syntax: RecordsetObject.Move (NumRecords, Start)

MoveFirst Method

It moves the position of the current record pointer to the first record.

Syntax: RecordsetObject.MoveFirst

MoveLast Method

It moves the position of the current record pointer to the last record.

Syntax: RecordsetObject.MoveLast

MoveNext Method

It moves the position of the current record pointer forward to the next record.

Syntax: RecordsetObject.MoveNext

MovePrevious Method

It moves the position of the current record pointer back to the previous record.

Syntax: RecordsetObject.MovePrevious

Open Method

When used on a Recordset object, this opens a cursor that is used to navigate through records.

Syntax: RecordsetObject.Open (Source, ActiveConnection, CursorType, LockType, Options)

Requery Method

It is used to update (refresh) the data in a Recordset object. This is essentially equivalent to a Close followed by an Open.

Syntax: RecordsetObject.Requery Options

Seek Method

It uses the index of a Recordset to locate a specified row.

Syntax: RecordsetObject.Seek KeyValues, SeekOption

Update Method

It is used to save any changes made to the current row of a Recordset object.

Syntax: RecordsetObject.Update (Fields, Values)

Practice 20.2

1. Open Notepad and type the following code:

```
<%
    Dim conn, rec
    Set conn = Server.CreateObject("ADODB.Connection")
    Set rec = Server.CreateObject("ADODB.Recordset")
    conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
    & "Data Source=" & Server.MapPath("Sample.mdb")
    rec.Open "Test", conn, adOpenDynamic
    for i = 1 to 5
      Response.Write (rec.Fields(1) & "<BR>")
      rec.MoveNext
    next
    conn.close
%>
```

2. Save the file as "Rec.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Rec.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:

How It Works?

The above practice uses a Recordset to retrieve data from **Sample** database. The first parameter of **Open** method of Recordset **rec** is the name of the table **Test** from which records will be retrieved. The second parameter specifies the Connection object **conn** that refers to the database in which **Test** table is created. The third parameter is the type of Recordset. The **For...Next** loop displays the second column of all records in the browser. The statement *rec.Fields(1)* refers to the index of the Recordset.

Practice 20.3

Create an ASP page that receives Roll No, Name and Age from user in an HTML form and stores it in Sample database. It should ensure that the given Roll No is not already stored in database.

1. Open Notepad and type the following code:

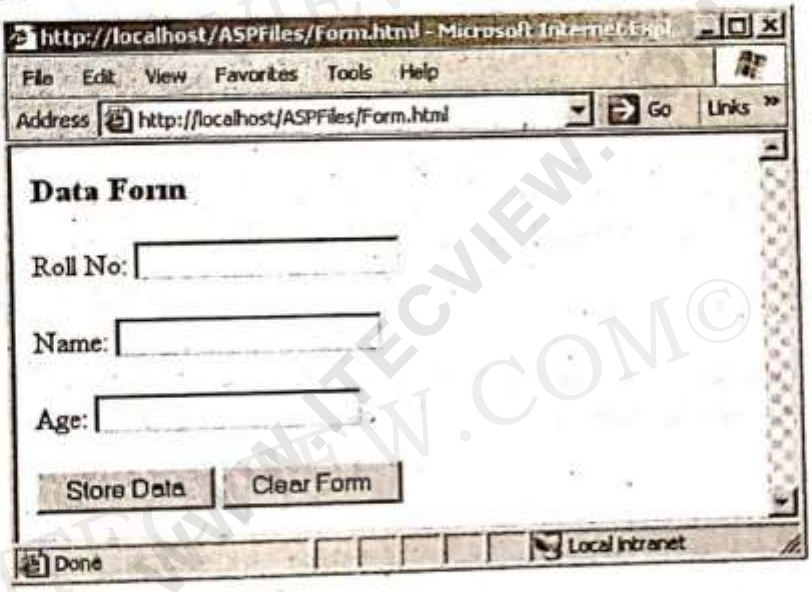
```
<HTML>
<BODY>
<H3>Data Form</H3>
<FORM NAME="form1" METHOD="post" ACTION="Store.asp">
Roll No: <INPUT TYPE="text" NAME="rollno"><P>
Name: <INPUT TYPE="text" NAME="name"><P>
Age: <INPUT TYPE="text" NAME="age"><P>
<INPUT TYPE="submit" VALUE="Store Data">
<INPUT TYPE="reset" VALUE="Clear Form">
</FORM>
</BODY>
</HTML>
```

2. Save the file as "Form.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
 Dim conn, rec, r, n, a
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Sample.mdb")
 r = Request.Form("rollno")
 n = Request.Form("name")
 a = Request.Form("age")
 if Not IsNumeric(r) Then
  Response.Write("Invalid Roll No")
```

```
elseif Not IsNumeric(a) Then
  Response.Write("Invalid age")
else
  Set rec = conn.Execute("Select * from Test where RollNo=" & r)
  if Not rec.EOF then
    Response.Write("Roll No already exists")
  else
    conn.Execute("Insert into Test values(" & r & ",'" & n & "'," & a & ")")
    Response.Write("Record Saved.")
  end if
end if
conn.close
%>
```

4. Save the file as "Store.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Type **http://localhost/ASPFiles/Form.html** in web browser and press **Enter** key. The page will appear in the browser as follows:



6. Enter values in the fields and press **Store Data** button. If the values are correct, the following message will appear:

How It Works?

The above practice uses two page. The "Form.html" file displays a form to get input. The "Store.asp" page stores data in the database. The ASP code checks Roll No and Age using IsNumeric function to ensure that these are numeric values. It uses the recordset to check that the Roll No is not already exists in the database.

Practice 20.4

Create an ASP page that receives Roll No from user in an HTML form and deletes the record of that Roll No from Sample database.

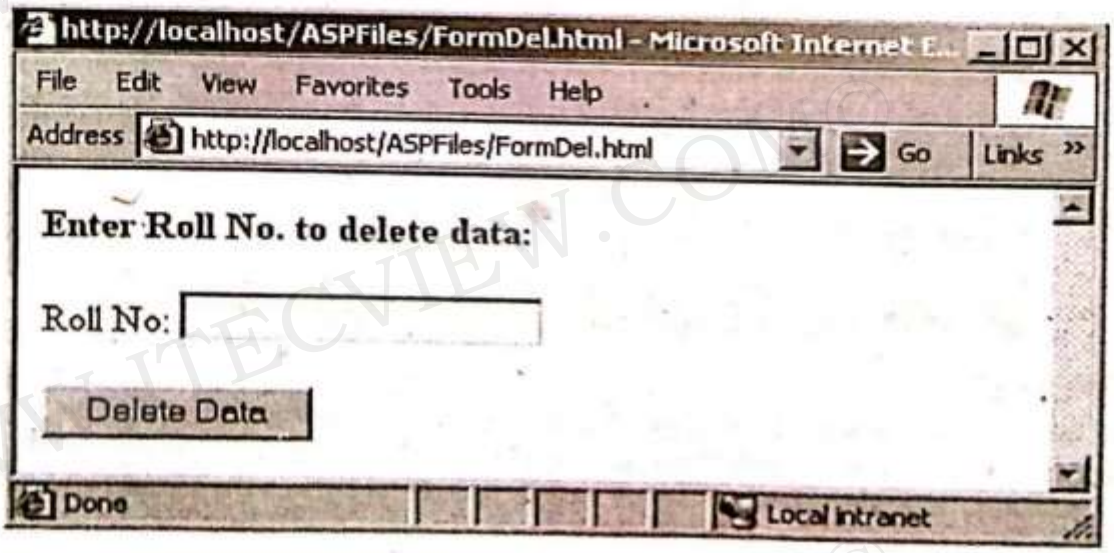
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<B>Enter Roll No. to delete data:</B>
<FORM NAME="form1" METHOD="post" ACTION="Delete.asp">
Roll No: <INPUT TYPE="text" NAME="rollno"><P>
<INPUT TYPE="submit" VALUE="Delete Data">
</FORM>
</BODY>
</HTML>
```

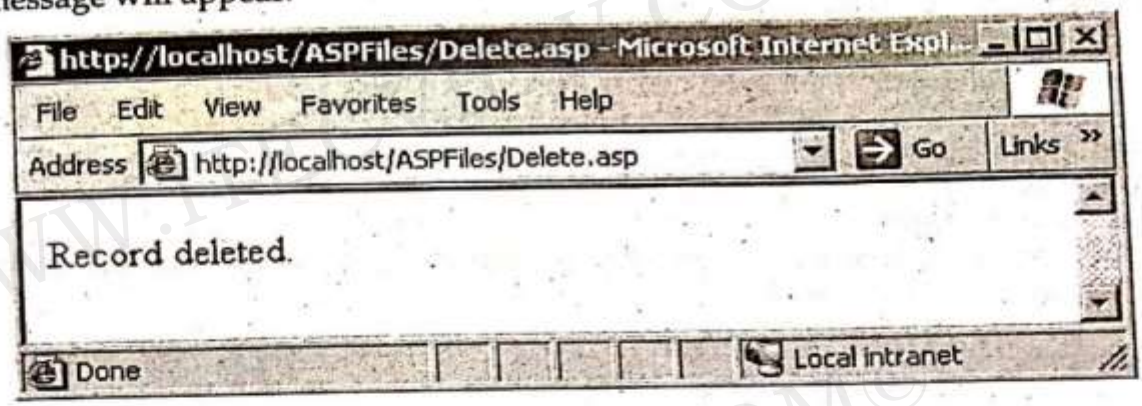
2. Save the file as "FormDel.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
 Dim conn, rec, r
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Sample.mdb")
 r = Request.Form("rollno")
 Set rec = conn.Execute("Select * from Test where RollNo=" & r)
 if rec.EOF then
  Response.Write("Roll No not found.")
 else
  conn.Execute("Delete from Test where RollNo=" & r)
  Response.Write("Record deleted.")
 end if
 conn.close
%>
```

4. Save the file as "Delete.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Type **http://localhost/ASPFiles/FormDel.html** in web browser and press **Enter** key. The page will appear in the browser as follows:



6. Enter Roll No and press **Delete Data** button. If Roll No is correct, the following message will appear:



How It Works?

The above practice uses two page. The "FormDel.html" file displays a form to get input. The "Delete.asp" page deletes data from the database. The ASP code checks Roll No to ensure that it exists in the database before attempting to delete it.

Practice 20.5

Create an ASP page that displays a form for updating records in Sample database. It displays all existing Roll numbers in a list box and two textboxes for name and age.

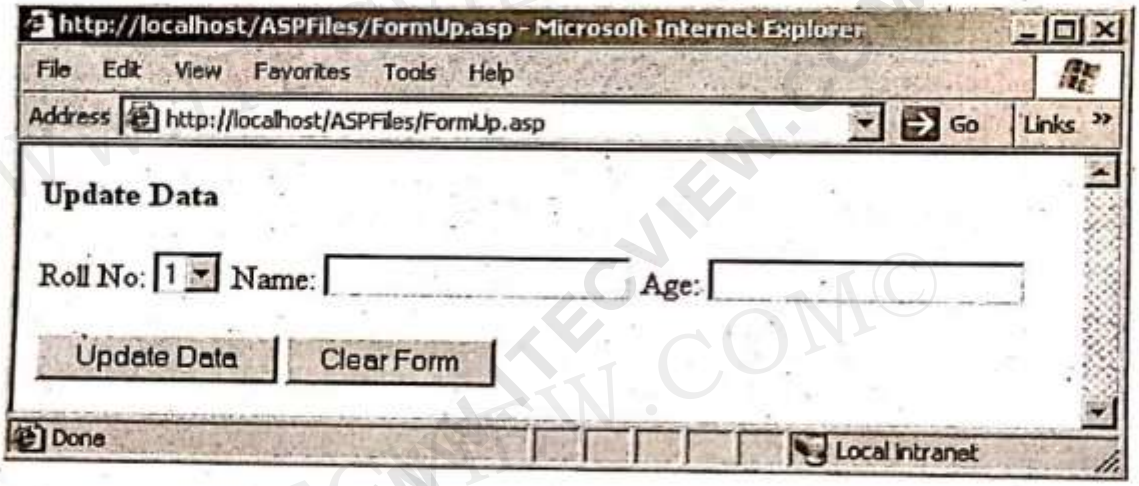
1. Open Notepad and type the following code:

```
<HTML>
<BODY>
<B>Update Data</B>
<FORM NAME="form1" METHOD="post" ACTION="Update.asp">
Roll No: <SELECT NAME="rollno">
<%
 Dim conn, rec
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Sample.mdb")
 Set rec = conn.Execute("Select RollNo from Test")
 while Not rec.EOF
   Response.Write("<OPTION>" & rec(0))
 rec.MoveNext
 wend
%>
</SELECT>
Name: <INPUT TYPE="text" NAME="name">
Age: <INPUT TYPE="text" NAME="age"><P>
<INPUT TYPE="submit" VALUE="Update Data">
<INPUT TYPE="reset" VALUE="Clear Form">
</FORM>
</BODY>
</HTML>
```

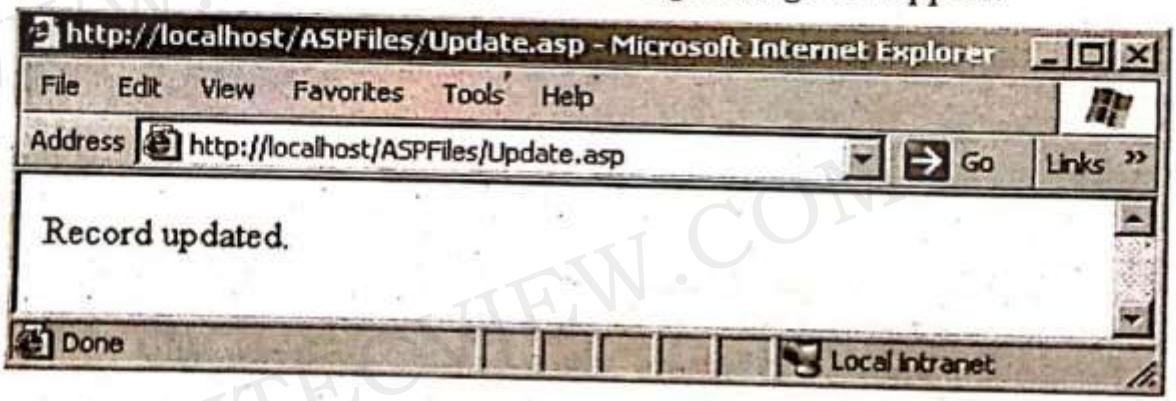
2. Save the file as "FormUp.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
 Dim conn, rec, r, n, a
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Sample.mdb")
 r = Request.Form("rollno")
 n = Request.Form("name")
 a = Request.Form("age")
 conn.Execute("Update Test set Name='" & n & "',Age=" & a & " where RollNo=" & r)
 Response.Write("Record updated.")
 conn.close
%>
```

4. Save the file as "Update.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Type **http://localhost/ASPFiles/FormUp.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



6. Select Roll No from list, enter new values to update database and press **Update Data** button. If the values are correct, the following message will appear:



How It Works?

The above practice uses two page. The "FormUp.asp" uses ASP code to retrieve all existing roll numbers from the database and displays it in a list. The "Update.asp" page updates the database.

Practice 20.6

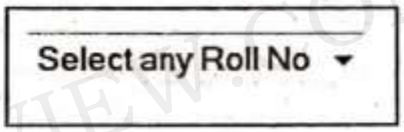
Design a form with two textboxes and one select control. As any option is selected, the relevant record should show in the textboxes. For example, if the user selects Roll No 1 from select control, all records of Roll No 1 should show in the textboxes.

Note: The following example uses database **Test.mdb** created in MS Access. The database contains a table **Sample** with three fields **RollNo, Name** and **Email.**

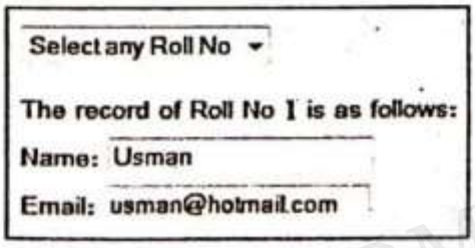
1. Open Notepad and type the following code:

```
<%
 connectionstring="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath
("Test.mdb") & ";"
 set rs1=Server.CreateObject("ADODB.Recordset")
 set rs2=Server.CreateObject("ADODB.Recordset")
 r = Request.Form("rn")
 rs1.open "select RollNo from Sample order by RollNo",connectionstring,1,2
 if not rs1.EOF then
        Response.Write("<form name=r action='Process.asp' method='post'>")
        Response.Write("<select name='rn' onChange='form.submit();'>")
        Response.Write("<option>Select any Roll No")
        while Not rs1.EOF
                Response.Write("<option>" & rs1("RollNo"))
                rs1.MoveNext
        wend
        Response.Write("</select>")
        Response.Write("</form>")
 end if
 rs1.close
 Response.Write("<br>")
 if r <> "" then
        rs2.open "select * from Sample where RollNo=" & r ,connectionstring,1,2
        Response.Write("The record of Roll No " & rs2("RollNo") & " is as follows: <BR>")
        Response.Write("Name: <input type='text' value=" & rs2("Name") & "><BR>")
        Response.Write("Email: <input type='text' value=" & rs2("Email") & "><BR>")
 rs2.close
 end if
%>
```

2. Save the file as "Process.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Process.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



4. Select Roll No 1 from the list box. The record of Roll No 1 will appear as follows:



5. Select Roll No 3 from the list box. The record of Roll No 3 will appear as follows:

Select any Roll No

The record of Roll No 3 is as follows:

Name: Faiq

Email: faiq@hotmail.com

Practice 20.7

Create an ASP form with five textobxes (Roll No, Name, Father Name, Class and Subject) and three buttons Delete, Insert and Update. The database is connected with ADO control. As any button clicked, it should perform the required operation.

1. Create an Access database **Test.mdb** in **C:\Inetpub\wwwroot\ASPFiles** folder.
2. Create table **Students** in the database with the following fields.

Field	Data Type
RollNo	Number
SName	Text
FName	Text
SClass	Text
Subject	Text

3. Insert the following data in the table.

RollNo	SName	FName	SClass	Subject
1	Usman Khalil	M. Khalil	FSc	Engineering
2	Faiq	Imran Saeed	Matric	Science
3	Ali	Ahmed	FSc	Medical
4	Waqar	Kaleem	Matric	Arts
5	Abdullah	Ashfaq	Matric	Science

4. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Students Data</TITLE>
</HEAD>
<BODY>
<FORM NAME="Students" ACTION="Action.asp" METHOD="post">
        Roll No: <INPUT TYPE="text" NAME="rno"><BR>
        Name: <INPUT TYPE="text" NAME="nam"><BR
        Father Name: <INPUT TYPE="text" NAME="fnr  ><BR>
        Class: <INPUT TYPE="text" NAME="cla"><BR>
        Subject: <INPUT TYPE="text" NAME="sub"><BR>
        <INPUT TYPE="submit" NAME="submit" VALUE="Delete">
        <INPUT TYPE="submit" NAME="submit" VALUE="Insert">
        <INPUT TYPE="submit" NAME="submit" VALUE="Update">
</FORM>
</BODY>
</HTML>
```

5. Save the file as "MyData.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.

6. Open a new file in Notepad and type the following code:

```
<%
        a = Request.Form("submit")
        if a = "Delete" then
                Server.Transfer("Delete.asp")
        elseif a = "Insert" then
                Server.Transfer("Insert.asp")
        else
                Server.Transfer("Update.asp")
        end if
%>
```

7. Save the file as "Action.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
8. Open a new file in Notepad and type the following code:

```
<%
        Dim conn, rec, r
        Set conn = Server.CreateObject("ADODB.Connection")
        Set rec = Server.CreateObject("ADODB.Recordset")
        conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
        & "Data Source=" & Server.MapPath("Test.mdb")
        r = Request.Form("rno")
        Set rec = conn.Execute("Select * from Students where RollNo=" & r)
        if rec.EOF then
                Response.Write("Roll No not found.")
        else
                conn.Execute("Delete from Students where RollNo=" & r)
                Response.Write("Record deleted.")
        end if
        conn.close
%>
```

9. Save the file as "Delete.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
10. Open a new file in Notepad and type the following code:

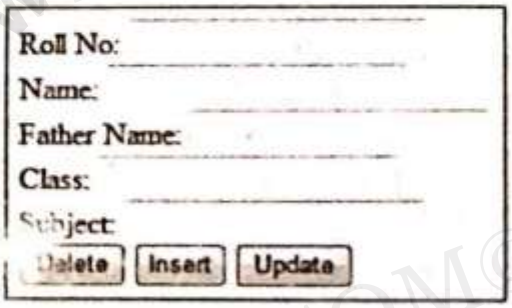
```
<%
 Dim conn, rec, r, n, f, c, s
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Test.mdb")
 r = Request.Form("rno")
 n = Request.Form("nam")
 f = Request.Form("fnm")
 c = Request.Form("cla")
 s = Request.Form("sub")
 if Not IsNumeric(r) Then
   Response.Write("Invalid Roll No")
 else
   Set rec = conn.Execute("Select * from Students where RollNo=" & r)
   if Not rec.EOF then
     Response.Write("Roll No already exists")
   else
     conn.Execute("Insert into Students values(" & r & ",'" & n & "','" & f & "','" & c & "','" & s &
"')")
```

```
    Response.Write("Record Saved.")
   end if
  end if
  conn.close
%>
```

11. Save the file as "Insert.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
12. Open a new file in Notepad and type the following code:

```
<%
 Dim conn, rec, r, n, f, c, s
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Test.mdb")
 r = Request.Form("rno")
 n = Request.Form("nam")
 f = Request.Form("fnm")
 c = Request.Form("cla")
 s = Request.Form("sub")
 Set rec = conn.Execute("Select * from Students where RollNo=" & r)
 if rec.EOF then
    Response.Write("Roll No not found.")
 else
    conn.Execute("Update Students set SName='" & n & "',FName='" & f & "',SClass='" & c &
"',Subject='" & s & "' where RollNo=" & r)
    Response.Write("Record updated.")
 end if
 conn.close
%>
```

13. Save the file as "Update.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
14. Type **http://localhost/ASPFiles/MyData.html** in web browser and press **Enter key**. The page will appear in the browser as follows:



Practice 20.8

Create an ASP page that displays a search form with three search options. The user can search by Roll No, Name or Age. A textbox get the value to be searched and matching records are displayed.

1. Open Notepad and type the following code:

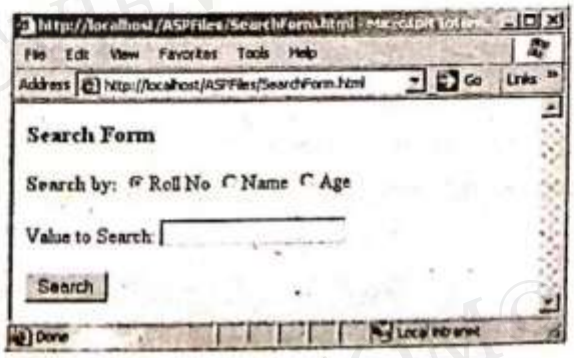
```
<HTML>
<BODY>
<H3>Search Form</H3>
<FORM NAME="form1" METHOD="post" ACTION="Search.asp">
<B>Search by:</B>
```

```
<INPUT TYPE="radio" VALUE="RollNo" NAME="s" CHECKED>Roll No
<INPUT TYPE="radio" VALUE="Name" NAME="s">Name
<INPUT TYPE="radio" VALUE="Age" NAME="s">Age <P>
Value to Search: <INPUT TYPE="text" NAME="svalue"><P>
<INPUT TYPE="submit" VALUE="Search">
</FORM>
</BODY>
</HTML>
```

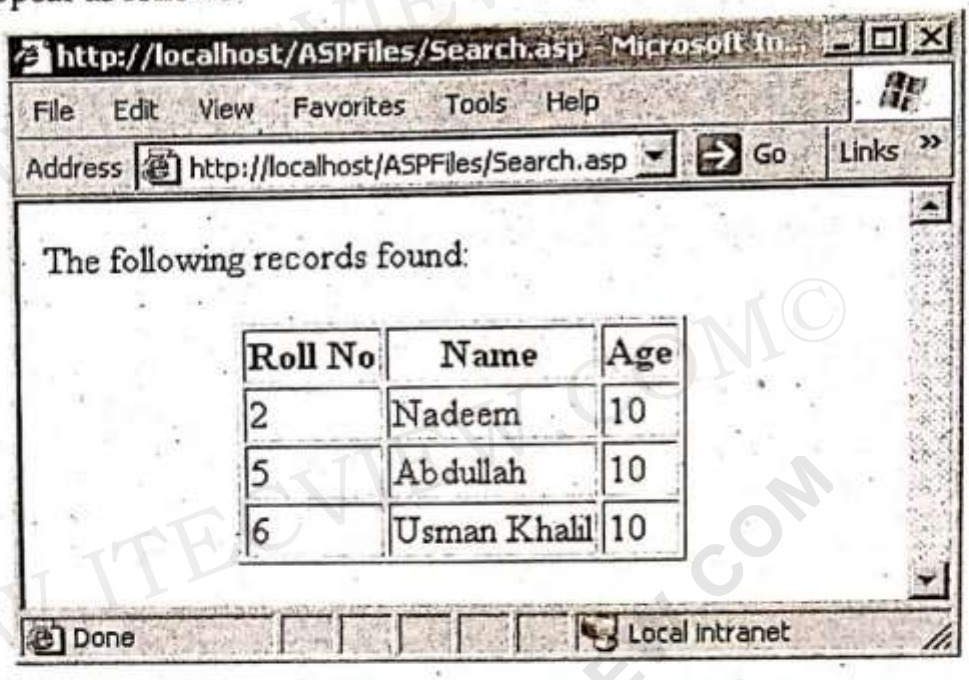
2. Save the file as "SearchForm.html" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Open a new file in Notepad and type the following code:

```
<%
 Dim conn, rec, c, v
 Set conn = Server.CreateObject("ADODB.Connection")
 Set rec = Server.CreateObject("ADODB.Recordset")
 conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
 & "Data Source=" & Server.MapPath("Sample.mdb")
 c = Request.Form("s")
 v = Request.Form("svalue")
 if c = "RollNo" or c = "Age" then
  Set rec = conn.Execute("Select * from Test where " & c & "=" & v)
 else
  Set rec = conn.Execute("Select * from Test where " & c & "='" & v & "'")
 end if
 if rec.EOF Then
  Response.Write("No record found.")
 else
  Response.Write("The following records found:<P>")
  Response.Write("<TABLE BORDER=1 ALIGN=center>")
  Response.Write("<TR><TH>Roll No</TH><TH>Name</TH><TH>Age</TH>")
  while Not rec.EOF
   Response.Write("<TR><TD>" & rec(0) & "</TD><TD>" & rec(1) & "</TD><TD>" & rec(2)
& "</TD></TR>")
   rec.MoveNext
  wend
  Response.Write("</TABLE>")
 end if
 conn.close
%>
```

4. Save the file as "Search.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
5. Type **http://localhost/ASPFiles/SearchForm.html** in web browser and press **Enter** key. The page will appear in the browser as follows:



6. Select **Age** radio option.
7. Enter 10 in **Value to Search** textbox and press **Search** button. All records with 10 age will appear as follows:



Practice 20.9

1. Create a new table **Users** in **Sample** database with the following fields.

Field	Data Type
FName	Text
LName	Text
UName	Text
Password	Text

2. Open Notepad and type the following code:

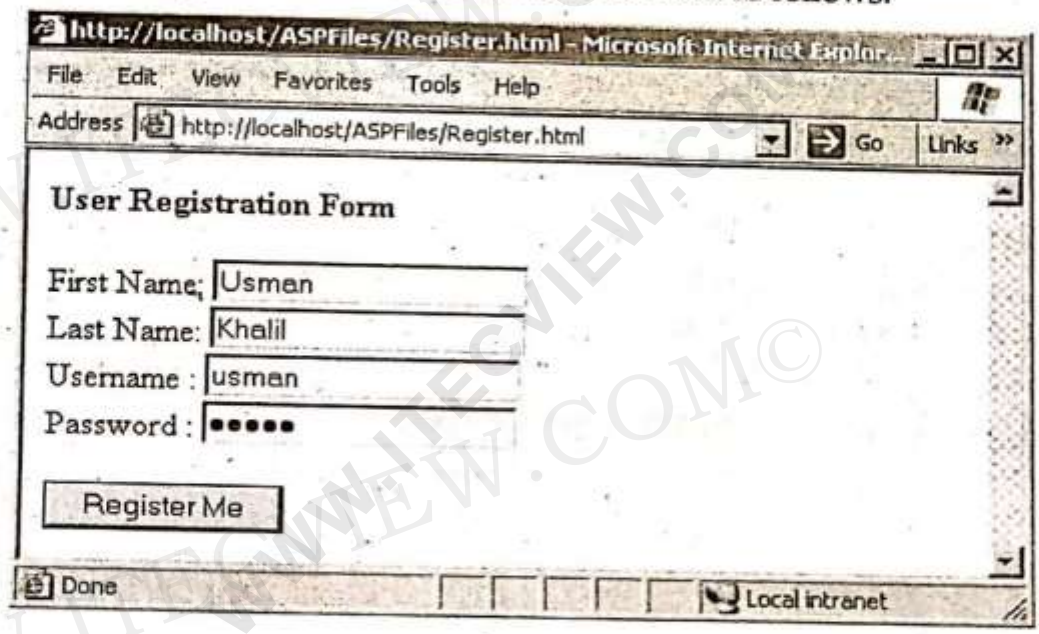
```
<HTML>
<BODY>
<H4>User Registration Form</H4>
<FORM NAME="form1" METHOD="post" ACTION="Register.asp">
First Name: <INPUT TYPE="text" NAME="fname"><BR>
Last Name: <INPUT TYPE="text" NAME="lname"><BR>
Username : <INPUT TYPE="text" NAME="uname"><BR>
Password : <INPUT TYPE="password" NAME="pass"><BR>
<P>
<INPUT TYPE="submit" VALUE="Register Me" NAME="submit">
</FORM>
</BODY>
</HTML>
```

3. Save the file as "Register.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
4. Open a new file in Notepad and type the following code:

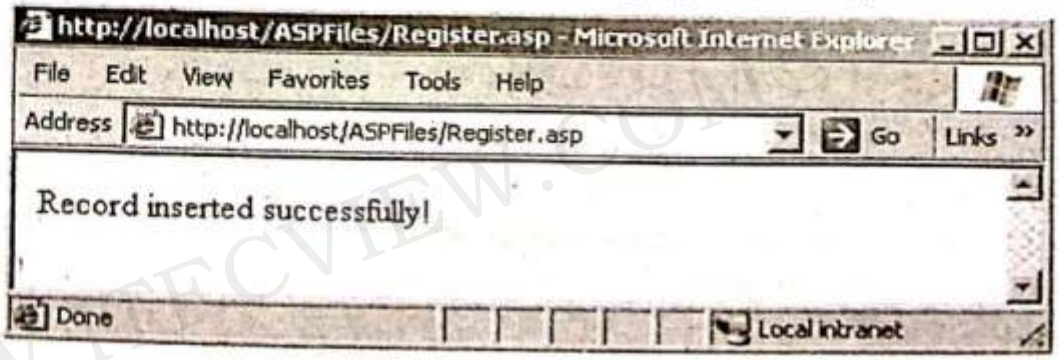
```
<%
  Dim con, rec, fn, ln, un, ps
  Set con = Server.CreateObject("ADODB.Connection")
  Set rec = Server.CreateObject("ADODB.Recordset")
```

```
fn = Request.Form("fname")
ln = Request.Form("lname")
un = Request.Form("uname")
ps = Request.Form("pass")
con.Open "Provider=Microsoft.Jet.OLEDB.4.0;"_
& "Data Source=" & Server.MapPath("Sample.mdb")
if fn = "" or ln = "" or un = "" or ps = "" then
  Response.Write("Fill complete form!")
else
  con.Execute("insert into Users values('" & fn & "','" & ln & "','" & un & "','" & ps & "');")
  Response.Write("Record inserted successfully!")
end if
%>
```

5. Save the file as "Register.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
6. Type **http://localhost/ASPFiles/Register.html** in web browser and press **Enter** key. The page will appear in the browser. Enter the data as follows:



7. Click on **Register Me** button. The form data will be submitted to **Register.asp** that will store the data in **Users** table and will display the message as follows:



20.8 Command Object

The ADO Command object is used to submit and execute a specific query against a database. The query can perform different actions such as adding, creating, deleting, retrieving, or updating databases.

If the query is used to retrieve data, then the data will be returned as a Recordset object. The retrieved data can be stored in a Recordset object.

One of the major features of the Command object is the ability to use stored queries and procedures that accept parameters. This requires access to the Parameter collection, which is a collection that is unique to the Command object.

20.8.1 Properties of Command Object

Some important properties of Command object are as follows:

ActiveConnection Property

Sets or returns a variant value defining the Connection object to which the Command object belongs, or returns a string value if there is no connection.

Syntax: CommandObject.ActiveConnection

CommandText Property

Sets or returns the text of a command statement to be applied against a data provider.

Syntax: CommandObject.CommandText

Name Property

Sets or returns the string name of the Command object.

Syntax: CommandObject.Name

State Property

Returns a long value describing if the Command object is open, closed or in the process of connecting, executing or retrieving.

Syntax: CommandObject.State

20.8.2 Collections of Command Object

Some important collections of Command object are as follows:

Parameters Collection

A collection of Parameter objects that allow the passing of parameters into stored queries and stored procedures.

Properties Collection

A collection of Property objects. Each Property object contains a provider-specific property.

20.8.3 Methods of Command Object

Some important methods of Command object are as follows:

Cancel Method

Cancels the execution of a pending Execute call.

Syntax: CommandObject.Cancel

Execute Method

Executes the query, SQL statement or stored procedure.

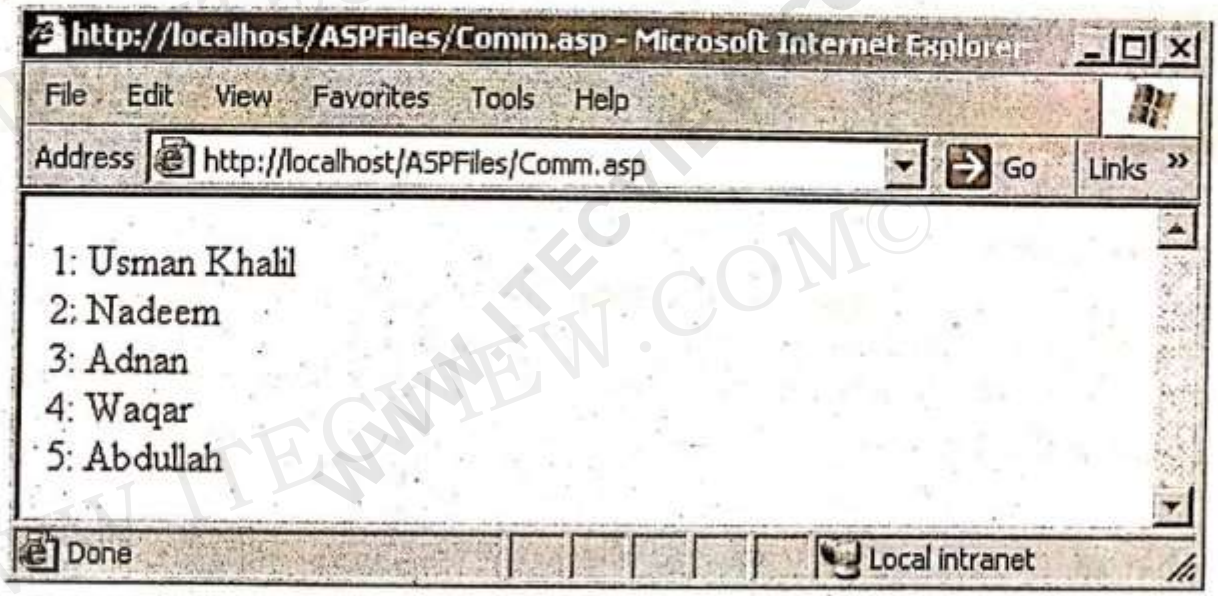
Syntax: CommandObject.Execute (NumberRecords, Parameters, Options)

Practice 20.10

1. Open Notepad and type the following code:

```
<%
    Dim com, rec
    Set com = Server.CreateObject("ADODB.Command")
    Set rec = Server.CreateObject("ADODB.Recordset")
    com.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;"_
    & "Data Source=" & Server.MapPath("Sample.mdb")
    com.CommandText = "Select * from Test;"
    Set rec = com.Execute
    for i = 1 to 5
      Response.Write (rec(0) & ": " & rec(1) & "<BR>")
      rec.MoveNext
    next
%>
```

2. Save the file as "Comm.asp" in **C:\Inetpub\wwwroot\ASPFiles** directory.
3. Type **http://localhost/ASPFiles/Comm.asp** in web browser and press **Enter** key. The page will appear in the browser as follows:



How It Works?

The above practice does not use Connection object for establishing connection with database. It uses Command object and Recordset objects. The information of connection is stored in ConnectionString property of the Command object. The **CommandText** property contains SQL statement for retrieving data from the **Student** table. The **Execute** method executes the SQL statement and returns the result from the table. The result is stored in the Recordset **rec**. The **For...Next** loop displays the first two fields of the table in the browser.

ASP Projects

Project 1: User Registration

Project Description

The project uses a databases to store information about registered users. It displays a page for getting information from user and then stores the information in database. It uses JavaScript code to validate form contents. It uses ASP file to store information in database.

Files Used in the Project

The project uses the following files:

- **News.html:** It is an HTML file that displays a form to get information from the user.
- **Join.asp:** It is an ASP file that manipulates form data and then stores it in database.
- **Newsletter.mdb:** It the Access database that is used to store user data.

Procedure

1. Open MS Access and create a database file "Newsletter.mdb".
2. Create a table "Members" with the following fields:

Email	Name
abc@hotmail.com	
itseries@msn.com	IT Series

3. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>IT Series Newsletter</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function validate_form ( )
{
  valid = true;
```

```
  if ( document.newsletter.name.value == "" )
  {
     alert ( "Please enter your Name." );
     valid = false;
  }
  else if (document.newsletter.email.value.indexOf("@")<1 ||
document.newsletter.email.value.indexOf(".")<1)
  {
   alert("Invalid Email address!");
   valid = false;
  }
  else
  {

window.open('','PopUp','toolbar=0,location=0,directories=0,status=0,menubar=0,scrollbars=0,
resizable=0,scrollbars=0,width=400,height=200,maximize=no');
   document.newsletter.submit();
   valid = false;
  }
  return valid;
}
</SCRIPT>
</HEAD>
<BODY TEXT="yellow">
<TABLE ALIGN="center" BORDER=0 BGCOLOR="green">
<TR>
        <TD COLSPAN=2 ALIGN="center"><B><I>Join Newsletter</I></B></TD>
</TR>
<FORM NAME="newsletter" ACTION="Join.asp" METHOD="post" onSubmit="return
validate_form ( );" TARGET="PopUp">
<TR>
        <TD>Enter name:</TD><TD><INPUT TYPE="text" NAME="name"></TD>
</TR>
<TR>
        <TD>Enter email:</TD><TD><INPUT TYPE="text" NAME="email"></TD>
</TR>
<TR>
        <TD COLSPAN=2 ALIGN="right"><INPUT TYPE="submit" VALUE="Join Me"></TD>
</TR>
</FORM>
</TABLE>
</BODY>
</HTML>
```

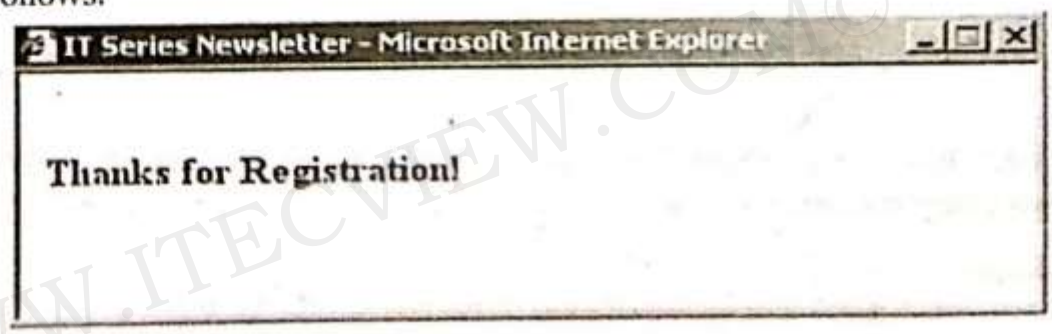
4. Save the file as "News.html" in **C:\Inetpub\wwwroot\Projects\Newsletter.**
5. Open a new file and type the following code:

```
<HTML>
        <HEAD>
                <TITLE>IT Series Newsletter</TITLE>
        </HEAD>
<%
  dim cmd, rs, em, str
  set rs = Server.CreateObject("ADODB.Recordset")
```

```
Set cmd = Server.CreateObject("ADODB.Command")
em = Request.Form("email")
str = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" &
Server.MapPath("Newsletter.mdb") & ";"
        cmd.ActiveConnection = str
        cmd.CommandText = "select * from Members where Email='" & em & "';"
        Set rs = cmd.Execute
        If Not rs.EOF Then
                Response.Write("<p><b>Email already Registered!</b>")
        Else
                cmd.CommandText = "Insert into Members(Email) values('" & em & "');"
                cmd.Execute
                Response.Write("<br><b>Thanks for Registration!</b>")
        End If
        Set cmd = Nothing
        Set rs = Nothing
%>
</HTML>
```

6. Type **http://localhost/Projects/Newsletter/News.html** and press Enter. The page will appear as follows:

7. Enter your name and email and click **Join Me** button. A popup window will appear as follows:



Note: If the user does not enter Name or Email, JavaScript will display error message and the form will not be submitted. Similarly, if the email address is already registred in the database, the popup window will display a message and will not duplicate it.

Project 2: Online Survey Poll

Project Description

The project uses a databases to store information about online survey poll. It displays a question with three options from which one can be selected by the user. When the user submits the form, the result is stored in a database and current suvery status is displayed in a popup window.

Files Used in the Project

The project uses the following files:

- **Poll.htm:** It is HTML file that displays a question, three answers and submit button.
- **Poll.asp:** It is an ASP file that manipulates form data and stores it in database.
- **IT.mdb:** It is the Access database that is used to store the results of survey poll.
- **blue.gif:** It is an image file that is used to show survey result.

Procedure

1. Open MS Access and create a database file "IT.mdb".
2. Create a table "Members" with the following fields:

PK	AU	SA
0	0	0

3. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>IT Series Poll</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function submit_form ( )
{
window.open('','PopUp','toolbar=0,location=0,directories=0,status=0,menubar=0,scrollbars=0,
resizable=1,scrollbars=0,width=400,height=200,maximize=no');
  document.poll.submit();
}
</SCRIPT>
</HEAD>
<BODY  TEXT="yellow">
<TABLE ALIGN="center" BORDER=0 BGCOLOR="green">
<TR>   <TD ALIGN="center"><B><I>IT Series Poll</I></B></TD> </TR>
<FORM NAME="poll" ACTION="Poll.asp" METHOD="post" TARGET="PopUp">
<TR>
        <TD>Who will win the next cricket World Cup?</TD>
</TR>
<TR>
        <TD><INPUT TYPE="radio" NAME="cup" VALUE="pk" CHECKED> Pakistan </TD>
</TR>
<TR>
        <TD><INPUT TYPE="radio" NAME="cup" VALUE="au"> Australia </TD>
</TR>
```

```
<TR>
        <TD><INPUT TYPE="radio" NAME="cup" VALUE="sa"> South Africa </TD>
</TR>
<TR>
        <TD        ALIGN="right"><INPUT        TYPE="button"        VALUE="Submit"
onClick="submit_form();"></TD>
</TR>
</FORM>
</TABLE>
</BODY>
</HTML>
```

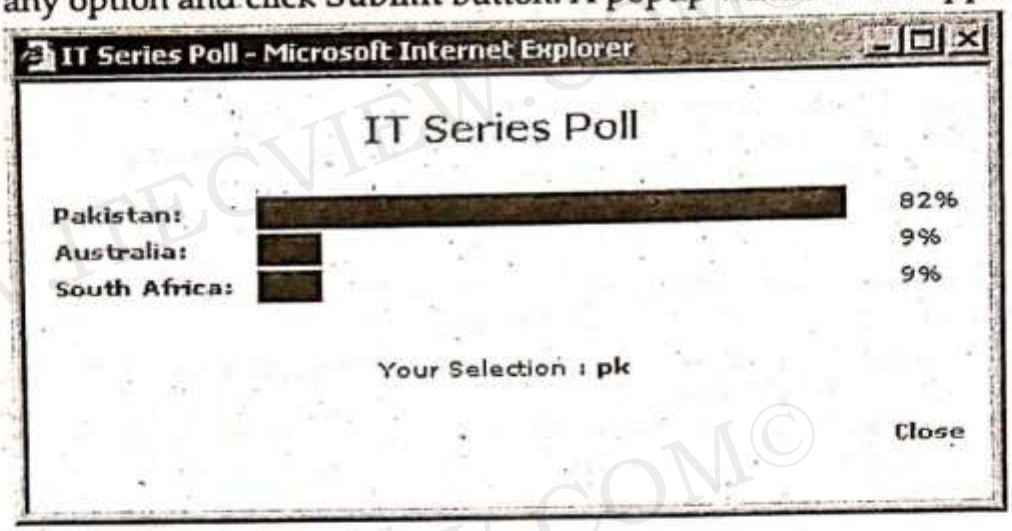
4. Save the file as "Poll.htm" in **C:\Inetpub\wwwroot\Projects\Poll**.
5. Open a new file and type the following code:

```
<HTML>
<HEAD>
<TITLE>IT Series Poll</TITLE>
</HEAD>
<DIV ALIGN="center">
 <P><FONT  COLOR="#FF9900"  SIZE=3  FACE="Verdana,  Arial,  Helvetica,  sans-
serif"><STRONG>IT Series Poll</STRONG></FONT></P>
 <P>
  <%
        Dim Con, DBPath, StrCon, StrQ, Rec, PollVal(2), Ch, Total, PollPer(2)
        Set Con = Server.CreateObject("ADODB.Connection")
        Set Rec = Server.CreateObject("ADODB.Recordset")
        DBPath = Server.MapPath("IT.mdb")
        StrCon = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & DBPath
        Con.Open(StrCon)
        StrQ = "Select * from Poll"
        Set Rec = Con.Execute(StrQ)
        PollVal(0) = Rec.Fields(0)
        PollVal(1) = Rec.Fields(1)
        PollVal(2) = Rec.Fields(2)
        Ch = Request.Form("cup")
        If Ch = "pk" Then
                Con.Execute ("Update Poll Set PK=" & (PollVal(0) + 1))
        ElseIf Ch = "au" Then
                Con.Execute ("Update Poll Set AU=" & (PollVal(1) + 1))
        ElseIf Ch = "sa" Then
                Con.Execute ("Update Poll Set sa=" & (PollVal(2) + 1))
        End If
        Rec.Requery
        PollVal(0) = Rec.Fields(0)
        PollVal(1) = Rec.Fields(1)
        PollVal(2) = Rec.Fields(2)
        Total = PollVal(0) + PollVal(1) + PollVal(2)
        PollPer(0) = (PollVal(0) / Total) * 100
        PollPer(1) = (PollVal(1) / Total) * 100
        PollPer(2) = (PollVal(2) / Total) * 100
        Response.write "<table border=0 cellpadding='0' cellspacing='2'>"
        Response.write "<tr>"
```

```
        Response.write "<td width=150><b><font color=#FF9900 size=1 face=Verdana, Arial, Helvetica, sans-serif>Excellent:</font></b></td>"
        Response.write "<td width=310><IMG SRC='blue.gif' width=" & PollPer(0)*3 & " height=15 ALT='blue' BORDER=0></td>"
        Response.write "<td><font color=#666699 size=1 face=Verdana, Arial, Helvetica, sans-serif>" & FormatNumber(PollPer(0),0) & "%</font></td>"
        Response.write "</tr>"
        Response.write "<tr>"
        Response.write "<td width=150><b><font color=#FF9900 size=1 face=Verdana, Arial, Helvetica, sans-serif>Good:</font></b></td>"
        Response.write "<td width=310><IMG SRC='blue.gif' width=" & PollPer(1)*3 & " height=15 ALT='blue' BORDER=0></td>"
        Response.write "<td><font color=#666699 size=1 face=Verdana, Arial, Helvetica, sans-serif>" & FormatNumber(PollPer(1),0) & "%</font></td>"
        Response.write "</tr>"
        Response.write "<tr>"
        Response.write "<td width=150><b><font color=#FF9900 size=1 face=Verdana, Arial, Helvetica, sans-serif>Normal:</font></b></td>"
        Response.write "<td width=310><IMG SRC='blue.gif' width=" & PollPer(2)*3 & " height=15 ALT='blue' BORDER=0></td>"
        Response.write "<td><font color=#666699 size=1 face=Verdana, Arial, Helvetica, sans-serif>" & FormatNumber(PollPer(2),0) & "%</font></td>"
        Response.write "</tr>"
        Response.write "</table>"
        Response.write "<br><font color=#666699 size=1 face=Verdana, Arial, Helvetica, sans-serif>Your Selection : <b>" & Request.Form ("cup") & "</b></font>"
        Rec.Close
        Set Rec = Nothing
        Con.Close
        Set Con = Nothing
%>
 </p></div>
<p align="right" onClick="window.close()"><font color="#FF9900" size="1" face="Verdana, Arial, Helvetica, sans-serif" style="cursor:hand"><strong>Close</strong></font></p>
</html>
```

6. Save the file as "Poll.asp" in **C:\Inetpub\wwwroot\Projects\Poll**.
7. Type **http://localhost/Projects/Poll/Poll.htm** and press Enter. The page will appear as follows:

8. Select any option and click **Submit** button. A popup window will appear as follows:



Project 3: Online Result

Project Description

The project uses a databases with two tables to store the records of students. It displays a textbox to get Roll No from the user and then displays the result.

Files Used in the Project

The project uses the following files:

- **Result.html:** It is HTML file that displays a textbox to get Roll No and submit button.
- **Result.asp:** It is an ASP file that displays result from the database.
- **IT.mdb:** It is the Access database that is used to store the results of students.

Procedure

1. Open MS Access and create a database file "IT.mdb".
2. Create a table "Students" with the following fields:

RollNo	Name	FName
1	Usman Khalil	Muhammad Khalil
2	Aysha Ashfaq	Ashfaq Shahid
3	Abdullah	Ejaz Saeed
4	Ehsan Qadir	Muhammad khalil
5	Waqar Ahmed	Ali Ahmed

3. Create another table "Marks" with the following fields:

RollNo	Subject	Marks
1	English	92
1	Math	100
1	Urdu	80
2	English	98
2	Math	100
2	Urdu	97

Note: Store the marks of other students also in the above table.

4. Create a one-to-many relationship between two tables using RollNo field.
5. Open Notepad and type the following code:

```
<HTML>
<HEAD>
<TITLE>Online Result</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function vform ( )
{
  if ( document.result.rollno.value == "" )
     alert ( "Enter Roll No." );
  else
    document.result.submit();
}
</SCRIPT>
</HEAD>
<BODY TEXT="yellow">
<TABLE ALIGN="center" BORDER=0 BGCOLOR="green">
<TR>
        <TD COLSPAN=2 ALIGN="center"><B><I>Check your Result</I></B></TD>
</TR>
<FORM NAME="result" ACTION="Result.asp" METHOD="post">
<TR>
        <TD>Enter Roll No:</TD><TD><INPUT TYPE="text" NAME="rollno"></TD>
</TR>
<TR>
        <TD COLSPAN=2 ALIGN="right"><INPUT TYPE="Button" VALUE="Show Result"
onClick="vform();"></TD>
</TR>
</FORM>
</TABLE>
</BODY>
</HTML>
```

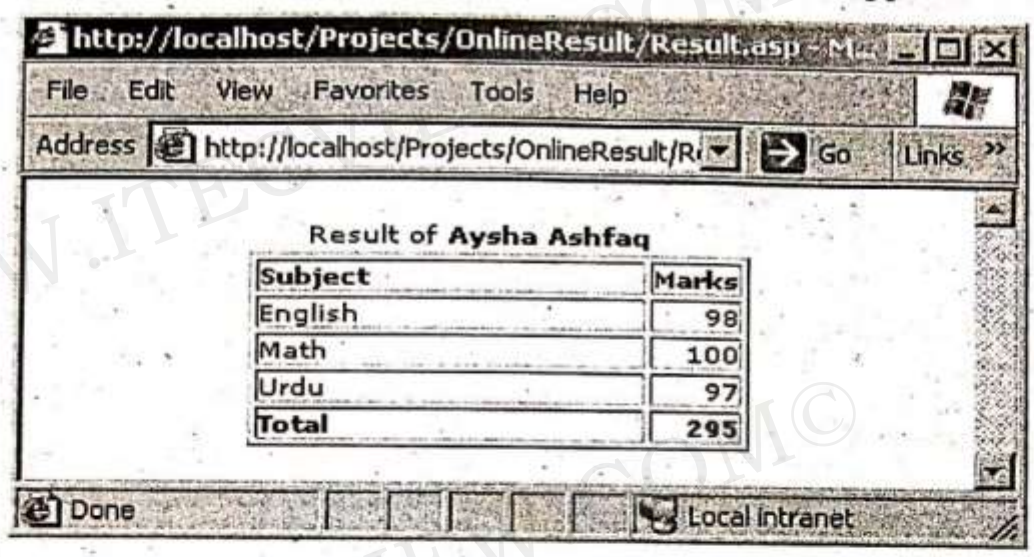
6. Save the file as "Result.htm" in **C:\Inetpub\wwwroot\Projects\OnlineResult.**
7. Open a new file and type the following code:

```
<%
        Dim Con, RecM, RecD, DBPath, StrCon, r, Total
        Set Con = Server.CreateObject("ADODB.Connection")
        Set RecM = Server.CreateObject("ADODB.Recordset")
        Set RecD = Server.CreateObject("ADODB.Recordset")
        DBPath = Server.MapPath("IT.mdb")
        StrCon = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & DBPath
        r = Request.Form("rollno")
        Total = 0
        Con.Open(StrCon)
        Set RecM = Con.Execute("Select * from Students where RollNo=" & r)
        If RecM.EOF Then
                Response.Write("Invalid Roll No.")
        Else
                Set RecD = Con.Execute("Select * from Marks where RollNo=" & r)
        End If
```

```
        Response.write "<table align='center' border=0 cellpadding='0' cellspacing='2'>"
        Response.write "<tr>"
        Response.write "<td width=150><font size=1 face=Verdana, Arial, Helvetica, sans-
serif>Result of <b> " & RecM.Fields("Name") & "</b></font></td>"
        Response.write "</tr>"
        Response.write "</table>"
        Response.write "<table align='center' border=1 cellpadding='0' cellspacing='2'>"
        Response.write "<tr>"
        Response.write "<td width=150><b><font size=1 face=Verdana, Arial, Helvetica,
'sans-serif>Subject</font></b></td>"
        Response.write "<td><b><font size=1 face=Verdana, Arial, Helvetica, sans-
serif>Marks</font></b></td>"
        Response.write "</tr>"
        While Not RecD.EOF
        Response.write "<tr>"
        Response.write "<td width=150><font size=1 face=Verdana, Arial, Helvetica, sans-
serif>" & RecD.Fields(1) & "</font></td>"
        Response.write "<td align='right'><font size=1 face=Verdana, Arial, Helvetica, sans-
serif>" & RecD.Fields(2) & "</font></td>"
        Response.write "</tr>"
        Total = Total + RecD.Fields("Marks")
        RecD.MoveNext
        Wend
        Response.write "<tr>"
        Response.write "<td width=150><b><font size=1 face=Verdana, Arial, Helvetica,
sans-serif>Total</font></b></td>"
        Response.write "<td align='right'><b><font size=1 face=Verdana, Arial, Helvetica,
sans-serif>" & Total & "</font></b></td>"
        Response.write "</tr>"
        Response.write "</table>"
        RecM.Close
        RecD.Close
        Set RecM = Nothing
        Set RecD = Nothing
        Con.Close
        Set Con = Nothing
%>
```

8. Save the file as "Result.asp" in **C:\Inetpub\wwwroot\Projects\OnlineResult**.
9. Type **http://localhost/Projects/OnlineResult/Result.html** and press Enter. The page will appear as follows:

10. Enter Roll No and click **Show Result** button. The result will appear as follows:



Project 4: Login Window

Project Description

The project uses a databases to store information about registered users. It displays a login window to input user ID and password. It also provides the facility to register and change password.

Files Used in the Project

The project uses the following files:

- **MyDB.mdb:** It the Access database that is used to store user data.
- **index.html:** It is an HTML file that displays different in two horizontal frames.
- **Menu.html:** It is an HTML file that displays a menu.
- **Login.html:** It is an HTML file that displays a form to login.
- **Login.asp:** It is an ASP file that processes "Login.html" file.
- **Register.html:** It is an HTML file that displays a form for registration.
- **Register.asp:** It is an ASP file that processes "Register.html" file.
- **ChangePass.html:** It is an HTML file that displays a form to change password.
- **ChangePass.asp:** It is an ASP file that processes "ChangePass.html" file.

Procedure

1. Open MS Access and create a database file "MyDB.mdb".
2. Create a table "Members" with the following fields:

UID	UPASS	FNAME	LNAME	EMAIL
admin	Admin	Usman	Khalil	usman@itseries.com.pk

3. Open Notepad and type the following code:

```
<HTML>
<HEAD>
</HEAD>
```

```
<FRAMESET BORDER=0 ROWS="15%,85%">
<FRAME SRC ="Menu.html" SCROLLING="NO" NORESIZE>
<FRAME SRC ="Login.html" NORESIZE="NORESIZE" NAME="main">
</FRAMESET>
</HTML>
```

4. Save the file as "index.html" in **C:\Inetpub\wwwroot\Projects\Login.**
5. Open a new file and type the following code:

```
<HTML>
<HEAD>
<STYLE>
a{Font-Size:10pt;Text-Decoration:None;color="pink";}
a:link{color="blue";}
a:hover{color: "red";}
a:Visited{color="green";}
</style>
</HEAD>
<BODY BGCOLOR="white">
<CENTER>
<A HREF="Login.html" TARGET="main">Login</A> |
<A HREF="Register.html" TARGET="main">Register</A> |
<A HREF="ChangePass.html" TARGET="main">Change Password</A>
</BODY></CENTER>
</HTML>
```

6. Save the file as "Menu.html" in **C:\Inetpub\wwwroot\Projects\Login.**
7. Open a new file and type the following code:

```
<HTML>
<HEAD>
<SCRIPT>
function doValid(){
        var v = 1;
        var frm = document.FR1;
        var cuid,cpas ;
        cuid = frm.ID.value;
        cpas = frm.PASS.value;
        if(frm.ID.value==="")
        {
                alert("User ID is Missing");
                var v=0;
        }
        else if(frm.PASS.value==="")
        {
                alert("Password is Missing");
                var v=0;
        }
        if(v==1)
        {
        frm.submit(FR1);
        }
}
</SCRIPT>
</HEAD><CENTER>
```

```
<BODY BGCOLOR="white">
<FORM NAME="FR1" ACTION="login.asp" TARGET="main" METHOD="post">
<TABLE>
<TR>
 <TD ALIGN="left">User ID: <TD>
 <TD><INPUT TYPE="text" NAME="ID" Maxlength="10"><TD>
</TR>
<TR>
 <TD ALIGN="left">Password<TD>
 <TD><INPUT TYPE="password" NAME="PASS" Maxlength="10"><TD>
</TR>
<TR>
 <TD ALIGN="left"> <TD>
 <TD ALIGN="right"><INPUT TYPE="BUTTON" VALUE="Login" MAXLENGTH="10"
onClick="doValid()"><TD>
</TR>
</TABLE>
</FORM></CENTER>
</BODY>
</HTML>
```

8. Save the file as "Login.html" in **C:\Inetpub\wwwroot\Projects\Login.**
9. Open a new file and type the following code:

```
<HTML>
<BODY>
<%
        dim conn, rs, fuid, fpass, sqlst
        set conn=server.createobject("ADODB.Connection")
        set rs=server.createobject("ADODB.Recordset")
        conn.open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
        Server.MapPath("MyDB.mdb")
        fuid = request.form("id")
        fpass = trim(request.form("pass"))

        rs.open "select * from Members where uid like '" & fuid & "' ",conn
        if rs.eof then
          Response.Write("Invalid User ID.")
        elseif fpass<>rs("upass") then
          Response.Write("Invalid password.")
        else
          Response.Write("Welcome " & rs("fname") & " You have logged in.")
        end if
        rs.close
        conn.close
%>
</BODY>
</HTML>
```

10. **Save the file as "Login.asp" in C:\Inetpub\wwwroot\Projects\Login.**
11. **Open a new file and type the following code:**

```
<HTML>
<BODY>
<CENTER>
```

```
<FORM NAME="frm1" ACTION="Register.asp" METHOD="post">
</CENTER>
<TABLE  ALIGN=CENTER BORDER="0">
 <TR>
  <TH COLSPAN="2">Registration Form</TH>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">User ID :</TD>
  <TD><INPUT TYPE="TEXT" SIZE=20 MAXLENGTH=25 NAME="fuid"></TD>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">Password :</TD>
  <TD><INPUT TYPE="password" SIZE=15 MAXLENGTH=15 NAME="fpass" ></TD>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">Repeat Password :</TD>
  <TD><INPUT TYPE="password" SIZE=15 MAXLENGTH=15 NAME="fpass2" ></TD>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">First Name :</TD>
  <TD><INPUT TYPE="TEXT" SIZE=20 MAXLENGTH=20 NAME="ffname"></TD>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">Last Name :</TD>
  <TD><INPUT TYPE="TEXT" SIZE=20 MAXLENGTH=20 NAME="flname"></TD>
 </TR>
 <TR>
  <TD ALIGN="RIGHT">Email Address :</TD>
  <TD><INPUT TYPE="TEXT" SIZE=20 MAXLENGTH=20 NAME="femail"></TD>
 </TR>
 <TR>
  <TD ALIGN="CENTER" COLSPAN=2><INPUT TYPE="submit" SIZE=8 VALUE="Register
Me"></TD>
 </TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>
```

12. Save the file as "Register.html" in **C:\Inetpub\wwwroot\Projects\Login.**
13. Open a new file and type the following code:

```
<%
        dim conn,rs,fuid,fpass,fpass2,ffname,flname,femail,sqlst

        set conn=server.createobject("ADODB.Connection")
        set rs=server.createobject("ADODB.Recordset")
        conn.open        "Provider=Microsoft.Jet.OLEDB.4.0;Data        Source="        &
Server.MapPath("MyDB.mdb")

        fuid    = request.form("fuid")
        fpass   = trim(request.form("fpass"))
        fpass2  = trim(request.form("fpass2"))
```

```
        ffname  = trim(request.form("ffname"))
        flname  = trim(request.form("flname"))
        femail  = trim(request.form("femail"))

        if fuid = "" then
          Response.Write("You Must Enter ID")
        elseif fpass = "" then
          Response.Write("You Must Enter Password")
        elseif instr(fpass," ")>0  then
          Response.Write("Password can not contain spaces")
        elseif fpass2 = "" then
          Response.Write("You Must Repeat Password")
        elseif fpass<>fpass2 then
          Response.Write("Repeated Password does not match with password")
        elseif ffname = "" then
          Response.Write("You Must Enter First Name")
        elseif femail="" then
          Response.Write("Please Provide Your Email Account")
        elseif instr(femail,"@")<=2 then
          Response.Write("Invalid Email Account")
        else
          rs.open "select uid from Members where uid like '" & fuid & "'",conn
          if not rs.eof then
            Response.Write("User ID already exists. Try another ID Name")
            rs.close
          else
            sqlst ="Insert into Members(UID,UPASS,FNAME,LNAME,EMAIL) VALUES ('" &
fuid & "','" & fpass & "','" & ffname & "','" & flname & "','" & femail & "')"
            Response.Write("Congratulation! You are registred now.")
            conn.execute(sqlst)
            rs.close
            conn.close
          end if
        end if
%>
```

14. Save the file as "Register.asp" in **C:\Inetpub\wwwroot\Projects\Login.**
15. Open a new file and type the following code:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function ValidValue(frm)
{
        var v = 1;
        var frm1=document.frm;
        if(frm1.ID.value=="")
        {
                alert("User ID is Missing");
                var v=0;
        }
        else if(frm1.PASS.value=="")
        {
                alert("Password is Missing");
```

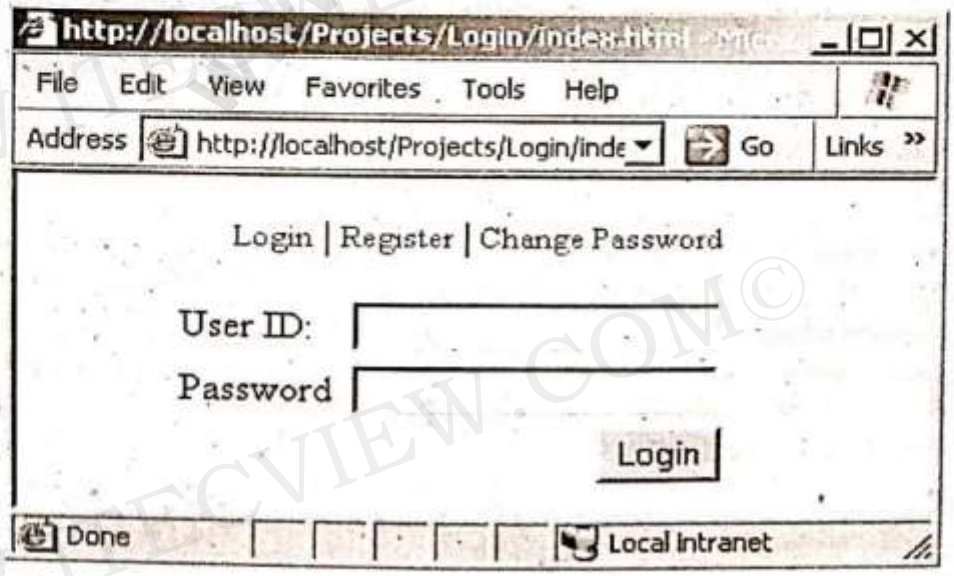
```
                var v=0;
        }
        else if(frm1.NPASS.value=="")
        {
                alert("New password is Missing");
                var v=0;
        }
        if(frm1.NPASS.value!=frm1.NPASS2.value){
                alert("Repeated New Password does not Match");
                var v=0;
        }
        if(v==1)
        {

          frm1.submit(frm);
        }
}
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<FORM NAME="frm" ACTION="ChangePass.asp" TARGET="main" METHOD="post">
<H3>Change Password</H3>
<TABLE BORDER="0">
<TR>
        <TD>User ID  :</TD>
        <TD><INPUT TYPE="text" NAME="ID" SIZE=15 MAXLENGTH=25></TD>
</TR>
<TR>
        <TD>Old Password  :</TD>
        <TD><INPUT TYPE="password" NAME="PASS" SIZE=15 MAXLENGTH=15></TD>
</TR>
<TR>
        <TD>New Password  :</TD>
        <TD><INPUT TYPE="password" NAME="NPASS" SIZE=15
MAXLENGTH=15></TD>
</TR>
<TR>
        <TD>Repeat New Password  :</TD>
        <TD><INPUT TYPE="password" NAME="NPASS2" SIZE=15
MAXLENGTH=15></TD>
</TR>
<TR>
        <TD COLSPAN=2 ALIGN="right"><INPUT TYPE="button" VALUE="Change"
onClick="ValidValue()"></TD>
</TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>
```

16. Save the file as "ChangePass.html" in C:\Inetpub\wwwroot\Projects\Login.

17. Open a new file and type the following code:

```
<%
    dim conn,rs,fuid,pass,npass,npass2
    fuid = request.form("id")
    pass = request.form("pass")
    npass = request.form("npass")
    npass2 = request.form("npass2")

    set conn=server.createobject("ADODB.Connection")
    set rs=server.createobject("ADODB.Recordset")
    conn.open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
Server.MapPath("MyDB.mdb")

    rs.open "Select * from Members where uid like '" & fuid & "'", conn
    if rs.eof then
      Response.Write("Invalid User ID")
    elseif rs("upass")<>pass then
      Response.Write("Invalid User Password")
    else
      sqlst = "update Members set upass='" & npass & "' where uid = '" & fuid & "'"
      conn.execute(sqlst)
      Response.Write("Password changed successfully.")
    end if
    rs.close
    conn.close
%>
```

18. Save the file as "ChangePass.asp" in **C:\Inetpub\wwwroot\Projects\Login.**
19. Type **http://localhost/Projects/Login/index.html** and press Enter. The page will appear as follows:



You can use different options to login, register or change password.

Short Questions

Q.1. Differentiate between traditional client-server application and Web application.

In client-server application, the client maintains a connection with server while the application is being executed. If the server goes down, the client sends an error message to the user. On the other hand, a Web-enabled application uses Internet, which is a connectionless network. Secondly, when a client makes a request for a page, the Web server sends the page to the client. The server does not keep track of the requests even if the request is for the same page.

Q.2. What is a proxy server?

Proxy Server is a software element used to pass network traffic between networks within an organization. Proxy Server contains firewalls required to filter information that is being sent to and received from the Internet or Intranet.

Q.3. What are database servers?

A database server is used to store data in database. Users can access and manipulate data. A Web application can provide the user with the interface to the database.

Q.4. What is certificate server?

Certificate servers are used to ensure data integrity of the Web site or the application.

Q.5. List different phases of Web application development cycle.

The phases of web application development are as follows:

1. Planning the Web Application
2. Designing the Web Application
3. Constructing and Testing the Components of Web Application
4. Web Application Deployment
5. Production and Maintenance

Q.6. Explain the relationship of ODBC, OLE DB and ADO.

ODBC was created to solve the problems of relational databases and data sources like spreadsheets. It is an interface between Web server and database server. OLE DB provides an object-oriented interface to data of almost any type. It can also be used as an interface to ODBC data sources. It was developed to support the processing of nonrelational data as well. ADO was designed as a cover over OLE DB objects. It enables programmers to be able to access OLE DB functionality.

Q.7. What is request object?

The Request object is used to retrieve the information sent in a request from client to the server. It reads the information when Submit button is clicked.

Q.8. What is response object?

The Response object is used to send data from the server to the client.

Q.9. What is application object?

The application object is used to share application-level information. The development of an application requires establishing a directory that is served by an Internet Information Server. This directory contains the source code for the application. Each application can have a single global.asa file.

Q.10. What is session object?

The Session object is used to store information for a particular user. A session starts as the user accesses any web page of the website. The session ends when the user leaves the website. The properties of Session object can be referenced on any page during the session.

Q.11. What is server object?

The Server object is used to access the resources on the server.

Q.12. What is "global.asa" file?

The global.asa file provides the Application and Session objects with the script. It also contains the scripts for OnStart and OnEnd events for Application and Session objects. These scripts uses the property RUNAT with the value Server.

Q.13. What is the use of "MapPath" function?

In an ASP application, all files are stored in a virtual directory. Any file stored in the virtual directory can be accessed by using MapPath method relative to the virtual directory. For example the physicall directory C:\Inetpub\wwwroot is accessed by http://localhost/ virtual path.

Q.14 What is server side includes?

Server Side Includes or SSI is a very simple programming language. It has a very limited number of instructions. An important instruction is used to include existing files in web pages.

Q.15. List the objects in ADO object model and explain their relationships.

Connection object is the first ADO object to be created. It is the basis for other objects. One or many Recordset objects or Command objects can be created for one Connection object. The errors are stored in Errors collection. Each RecordSet object has a Fields collection; each Field element corresponds to a column in the recordset.

Q.16. What is the function of Connection object?

The connection object establishes a connection with an ODBC data source or with an OLE DB data source. It is the basis of the other objects in ADO.

Q.17. What is the purpose of the Command object?

Its purpose is to execute queries and stored procedures that are stored with the database.

Q.18. Identify and correct the errors in each of the following statements:

a) Dim False as Integer

False is a keyword and may not be used as identifier. Use a non-keyword as the variable name.

b) Dim variable, inputvalue as Integers

Integers should be Integer.

c) x + y = z

The variable storing the result of the assignment (z) must be the left operand of the assignment operator. The statement should be rewritten as z=x+y

Q.19. State the order of evaluation of orators in each of the following statements, and show the value of z after each statement is performed. Assume z to be an integer variable.

a) z=7+2*6\2-1
b) z=2 Mod 2+2*4-2/2
c) z=8+10\2*5-20\2
d) z=-5-8 Mod 4+7*(2^4+3)
e) z= 10 mod 3 ^1^2-8

a)

Z=7+12\2-1

Z=7+6-1
Z=13-1
Z=12

b)

z= 2 mod 2+8-2/2
z=2 mod 2+8-1
z=0+8-1
z=7

c)

z=8+10\10-20\2
z=8+1-20\2
z=8+1-10
z=9-10
z=-1

d)

Z=-5-8 Mod 4+7*(2^4+3)
Z= -5-8 mod 4+7 *(8+3)
Z= -5-8 mod 4+7 *11
Z= -5-8 mod 4+77
Z= -5-0+77
Z=-5 +77
Z= 72

e)

Z= 10 mod 3^1^2-12
Z=10 mod 3 ^2-12
Z= 10 mod 9 –12
Z= 1-12
Z= -11

Q.20. Determine the true or false result of the following relational tests:

A. "zz" < "ZZ"
B. 161 <= 161
C. 3.4 < 4
D. (3 <> 4) And (5 = 5)
E. (3 <> 4) Or (5 = 5)
F. (3 = 3) And (10 < 9)

Answer.

A. False
B. True
C. True
D. True
E. True
F. False

Q.21. How many times does the following loop execute?

I = 10
Do While I > 1
I = I - 1
Loop
Answer. 9

WWW.ITECVIEW.COM©

Q.22. How many times does the following loop execute?

```
I = 10
Do While I >= 1
I = I - 1
Loop
```

Answer. 10

Q.23. How many times does the following loop execute?

```
I = 10
Do Until I > 1
I = I - 1
Loop
```

Answer. 0

Q.24. How does FindFirst and MoveFirst methods differ?

The MoveFirst method positions the record pointer on the first record whereas the FindFirst method positions the record pointer on the first record that meets stated criteria.

Q.25. Explain how BOF and EOF properties are set used in a project?

BOF property is automatically set to True when the record pointer is before the first record in Recordset. It happens when the first record is current and the user chooses MovePrevious. BOF property is also true if the Recordset is empty. EOF property is True when the record pointer moves beyond the last record in the Recordset, and when the Recordset is empty.

Q.26. Specify the methods used to navigate through the recordset.

Methods used for navigating through the recordset are:

A. MoveFirst
B. MoveNext
C. MovePrevious
D. MoveLast

Multiple Choice

1. **Web servers are also known as:**
   a. HTML servers b. HTTP servers c. Browser servers. d. None
2. **Web servers provide the _____ of the World Wide Web.**
   a. Internet b. Browsers c. Infrastructure d. All
3. **A Web server functions as a service center for:**
   a. Information sharing. b. Data exchange. c. Security. d. All
4. **Server hardware can be a:**
   a. Personal computer. b. Minicomputer. c. Mainframe. d. All
5. **The application server is located between the:**
   A. Database server and the Web server. b. Client Web browser and Web server.
   C. Proxy server and the database server. d. Firewall and the security server.
6. **Middleware is also known as the:**
   a. Intranet. b. Application server. c. Exchange server. d.proxy server.

7. **SQL server is a popular:**
   a. Database server. b. Proxy server. c. Application server. d. None
8. **A proxy server sits between the:**
   a. Database server and the Web server. b. Client Web browser and Web server.
   c. Proxy server and the database server. d. Firewall and the security server.
9. **The architecture of a dynamic, interactive, data-driven Web site includes:**
   a. Web server. b. Server-side script file. c. Query generator. d. All
10. **Which method of coding ensures that source code cannot be viewed from client browsers?**
    a. JavaScript b. CGI script c. Client-side scripting d. Server-side scripting
11. **ASP pages can be viewed in:**
    a. Netscape Navigator b. Internet Explorer c. Opera d. All browsers
12. **How many primary ASP objects are there?**
    a. One b. Three c. Five d. Seven
13. **Which of the following is NOT one of the primary ASP objects?**
    a. Request object b. Response object c. Session object d. Security object
14. **Which of the following is NOT one of the ADO objects?**
    a. Connection object b. Server object c. Recordset object d. Command object
15. **Which object is used in HTML files to get information from client browsers for?**
    a. Request b. Response c. Connection d. Command
16. **These delimiters are used to open and close script blocks of an ASP file.**
    a. <? . . . ?> b. <asp> </asp> c. <% . . . %> d. / . . . \
17. **Delimiters tell the ASP engine that everything between the delimiters:**
    a. is for executing data communication b. is for browsing.
    c. is to be saved. d. is to be viewed as source code.
18. **The ASP Request object is concerned with information that is coming:**
    a. From the client browser to the server b. From the server to the client browser.
    c. From the server to external components. d. From client browser to external components.
19. **The ASP Response object is concerned with information flowing:**
    a. From the client browser to the server b. From the server to the client browser.
    c. From the server to external components d. From client browser to external components.
20. **Which ASP object can change value of a cookie that already exists on client?**
    a. Response b. Request c. Connection d. Command
21. **Redirect is a useful feature of the ASP _____ object.**
    a. Response b. Request c. Connection d. Command
22. **Which object can redirect users to login page when a Website requires a valid user name and password for login.**
    a. Response b. Request c. Server d. Command
23. **Which object is used in almost every ASP file to create instances or connect to external components?**
    a. Response b. Request c. Server d. d. Command

24. The process of a Web user's visit through the pages is call a(n):

a. Episode. b. Event. c. Sequence. d. Session.

25. Which object is created when new visitor arrives at an application page?

a. Server b. Session c. Event d. Episode

26. Which ASP object stores and shares information that exists over the entire lifetime of an ASP application?

a. Application b. Request c. Command d. Server

27. Which ASP object is created at the Web server when you create a virtual directory for the application?

a. Application b. Request c. Command d. Server

28. The object type that enables a Web-based client/server application to be connected actively with a database through ADO is a(n):

a. Linked object. b. Embedded object.

c. Connection object. d. Application object.

29. Which object enables a Web-based client/server application to obtain the results of an executed SQL query?

a. Command object b. Request object c. Recordset object d. Connection object

30. A Command object is used to:

a. Get records from a database. b. Execute SQL queries.

c. Directly manipulate data. d. All

31. Which of the following type(s) of files are created with a combination of HTML tags, text, script, and SQL?

a. JAVA b. ASP c. both A and B d. neither A nor B

32. Which programming language is required for ASP files?

a. VisualBasic b. Java c. Perl d. ASP is language-independent

33. Which of the following is the default scripting language of ASP?

a. VBScript b. Java c. Perl d. There is no default scripting language for ASP

34. To assign a constant with a numeric literal value, you would enclose it in:

a. Number signs (# . . . #). b. Quotation marks (" . . . ").

c. Percent signs (% . . . %). d. Double brackets << . . . >>.

35. To assign a constant with a string literal value, you would enclose it in:

a. Number signs (# . . . #). b. Quotation marks (" . . . ").

c. Percent signs (% . . . %). d. Double brackets << . . . >>.

36. Which operators include exponentiation, multiplication, division, addition and subtraction?

a. Arithmetic operators b. comparison operators.

c. Logical operators. d. Concatenation operators.

37. Which operators involve equality, inequality, less than, greater than?

a. Arithmetic operators. b. Comparison operators.

c. Logical operators. d. Concatenation operators.

38. Operators that contain conjunction, negation, disjunction, exclusion, equivalence and implication are called:

a. Arithmetic operators.
b. Comparison operators.
c. Logical operators.
d. Concatenation operators.

39. Which operator(s) has (have) the highest order of precedence in arithmetic expressions?

a. Multiplication and division
b. Addition and subtraction
c. Exponentiation
d. Operators are not significant

40. What does the <> symbol mean when used in a decision structure?

a. Greater than but not equal to
b. Not equal to
c. Less than or equal to
d. Greater than or equal to

41. A statement block can consist of how many lines?

a. 0
b. 2
c. An unlimited amount
d. at most 255

42. When can the Case structure be used?

a. Any time an Else could be used.
b. When a decision is testing many variables for a single value
c. When a decision is testing many variables for many values
d. When a decision is testing the same variable for several different values.

43. When does the 'Case Else' branch of a Case structure execute?

a. When two or more previous cases are true
b. When none of the previous cases are false
c. When all previous cases are true
d. When all previous cases are false

44. In a Case structure, if two or more cases happen to be true, which case is executed?

a. The first matching case encountered
b. All matching cases encountered.
c. An error is generated
d. The last matching case encountered.

45. When can the 'ElseIf' statement be used?

a. When another If-Then is needed within the Else branch of a decision
b. When all of the cases of a Case structure are true.
c. When none of the cases of a Case structure are true
d. When another If-Then is needed in the True branch of a decision.

46. The following is an example of:

```
Do While Counter <= 500
Loop
```

a. A pre-test loop that continues as long as the condition is false
b. A post-test loop that continues as long as the condition is true.
c. A pre-test loop that continues as long as the condition is true.
d. A post-test loop that continues as long as the condition is false.

47. The following is an example of:

```
Do
    <Steps to be repeated>
    Loop While Counter <= 500
```

a. A pre-test loop that continues as long as the condition is false.

b. A post-test loop that continues as long as the condition is false.
c. A pre-test loop that continues as long as the condition is true
d. A post-test loop that continues as long as the condition is true

48. Which of the following statements is not true about variables?
a. They can be set only once during your program's execution
b. Their names have a limit of 255 characters
c. The first character of their name must be a letter
d. They may contain either numeric or string data

49. Expressions are comprised of which of the following:
a. Constant b. variables c. functions d. all of these

50. Which of the following is not a typical type of expression:
a. Numeric b. string c. assignment d. logical

51. What will be the value of *MyValue* after the following line of code executes:

MyValue = 5 Mod 25 + 4 / 3 * 10 Mod 30

a. 15 b. 18 c. 5 d. None

52. A variable's lifetime refers to:
a. How long it remains in memory
b. Its data type
c. How widely it is recognized to procedures in the project
d. Whether it is reinitialized every time a procedure is called

53. The conditional portion of the If statement can contain:
a. Any valid constant
b. Any expression that can be evaluated as True or False
c. Any valid variable
d. Any expression that can be evaluated as True or False

54. A Select Case block stops evaluating as soon as:
a. All conditions have been evaluated b. A condition has been found to be False
c. A criterion has been found to be True d. It reaches the End Select statement

55. The keyword that marks the end of a Do structure is:
a. END DO b. EXIT DO c. END WHILE d. LOOP

56. If the number of times a group of statements must be repeated is not known in advance, the most suitable programming structure is:
a. For...Next b. DO....LOOP UNTIL
c. FOR...EACH NEXT d. FOR....WHILE

57. Variable that holds a large group of homogeneous data is a(n):
a. Constant b. scalar value c. array d. multiple variable

58. Given an array declared by the statement Dim Employees(71), the number of elements in the array is:
a. 71 b. 72 c. 0 d. None
e. it is impossible to determine from the information given

59. All of the following are valid array declarations except:

a. Dim Name(5000) as String
b. Dim Salary(15) as Double
c. Dim Students() as String
d. Dim Students (0 to 15) as String

Answers

1.	b	2.	c	3.	d	4.	d
5.	a	6.	b	7.	a	8.	b
9.	d	10.	d	11.	d	12.	c
13.	d	14.	b	15.	c	16.	c
17.	a	18.	a	19.	b	20.	a
21.	a	22.	a	23.	c	24.	d
25.	b	26.	a	27.	a	28.	c
29.	c	30.	d	31.	b	32.	d
33.	a	34.	a	35.	c	36.	a
37.	b	38.	c	39.	c	40.	b
41.	c	42.	d	43.	d	44.	a
45.	a	46.	b	47.	d	48.	a
49.	d	50.	c	51.	a	52.	a
53.	b	54.	c	55.	d	56.	b
57.	c	58.	b	59.	d		

Fill in the Blanks

1. _____ is the program that runs the server.
2. _____ server manages collaboration and messaging applications.
3. _____ server filters and caches Web content and improves network performance.
4. _____ server restricts unauthorized access.
5. ASP has _____ primary ASP objects.
6. Five primary ASP objects are _____, _____, _____, _____.
7. ASP objects relate to _____ objects whe connecting interactive Web pages with Web-enabled database applications.
8. ADOs are designed to interface with the relationa databases through the _____.
9. _____ object is used in HTML files to get information from client browsers.
10. _____ object is used in HTML files to send information from the server to the client browser.
11. The result of each Web user's visit to a Web site is called a(n) _____.
12. The _____ object stores the information variables about a visitor's Web-.

WWW.ITECVIEW.COM©

13. The _____ object enables a Web-based client/server application to be connected actively with a database through ADO.
14. ____ files are created with a combination of HTML tags, text, script and SQL statements.
15. ____ script is unique because its source code cannot be viewed from a client browser.
16. Even though ASP files start and end with HTML tags, they must be saved with _____ file name extension.
17. The default scripting language of ASP is _____
18. ASP stands for_____
19. PWS stands for_____
20. IIS stands for_____.
21. SSI stands for_____.

Answers

1.	Server software			2.	Exchange
3.	Proxy	4.	Proxy	5.	Five
6.	Request, Response, Server, Session and Application			7.	ADO (ActiveX Data Objects)
8.	Open Database Connectivity (OBDC)			9.	Request
10.	Response	11.	Session	12.	Server session
13.	Connection	14.	ASP	15.	Server-side
16.	Asp	17.	VBScript	18.	Active Server pages
19.	Personal Web Server	20.	Internet Information Server	21.	Server side includes

True / False

1. The scripting approach uses HTTP and script languages to develop the application.
2. Web servers provide the infrastructure for the World Wide Web.
3. A personal computer can act as a Web server.
4. A Web server must be at least a minicomputer.
5. All Web servers run on the UNIX operating system.
6. A database server is often called middleware.
7. Oracle is an example of a database server.
8. The use of proxy servers can improve network performance.
9. A server can be either a mainframe computer or a minicomputer.
10. Middleware is another term for application server.
11. Client requests are sent to the Web server in the form of SQL.
12. Microsoft Access and Oracle are both Web-enabled database software applications.
13. ASP or JSP files are used to develop Web pages on the browser presentation layer.
14. The creation of DSNs on the server is required for the database file to be accessible through the application layer.
15. ASP is a computer programming language.

WWW.ITECVIEW.COM©

16. ASP pages can be viewed with Internet Explorer but not with Netscape Navigator.
17. The Request object is used in HTML files to get information from client browsers and submit it to the Web server.
18. The Response object handles the opposite function of the Request object.
19. Redirect is a function of the Request object.
20. The Session object provides the ability to redirect users to a new page, such as a login page.
21. ADOs provide a mechanism for the Web-based client/server application to access various databases.
22. The results of a SQL query are collected by the Recordset object.
23. ASP files must be written in VBScript.
24. HTML is a component of an ASP file.
25. VBScript is a server-side scripting language.
26. A computer must be connected to the internet before running a web server on it.

Answers

1. F	2. T	3. T	4. F
5. F	6. F	7. T	8. T
9. T	10. T	11. T	12. T
13. F	14. T	15. F	16. F
17. T	18. T	19. F	20. T
21. T	22. T	23. F	24. T
25. T	26. F		

WWW.ITECVIEW.COM©

Excellent Books of IT Series

URL: http://www.itseries.com.pk
Email: info@itseries.com.pk
Mob: 0321 661 56 56

Available at: **Kitab Markaz Faislabad, Pakistan**
Imtiaz Book Depot Lahore, Pakistan

ISBN 978-969-9157-16-5