

HTML5 with CSS3



BccFalna.com
097994-55505

Kuldeep Chand

In this EBook I have covered HTML5 and CSS3. Both are Latest Technology for Developing Website and Web Applications.

HTML5 is used not only for Webpages but also for various kinds of Gadgets and Applications for Mobile, Smart Phones, PCs, Windows7/Windows8, Tablets, etc... So, learning HTML5 is necessary for each and every Developer.

CSS3 is used for styling the Frontend of every Web Based Applications, not only for websites but also for various kinds of other Devices too. So, in this eBook you will get Good Details on HTML5 and CSS3 for Good Looking Frontend Development.

HTML5

with

CSS3

In Hindi



Kuldeep Chand

BetaLab Computer Center
Falna

HTML5 with CSS3 in Hindi

Copyright © 2012 by Kuldeep Chand

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editors: Kuldeep Chand

Distributed to the book trade worldwide by Betalab Computer Center, Behind of Vidhya Jyoti School, Falna Station Dist. Pali (Raj.) Pin 306116

e-mail bccfalna@gmail.com, or visit <http://www.bccfalna.com>.

For information on translations, please contact Betalab Computer Center, Behind of Vidhya Jyoti School, Falna Station Dist. Pali (Raj.) Pin 306116

Phone 91-97994-55505

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, the author shall not have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this book.

**This book is dedicated to those
who really wants to be
a
PROFESSIONAL DEVELOPER**

INDEX OF CONTENTS

Table of Contents

Web Development Fundamentals	15
Web Browsers	18
High Level Structure of Web Browser	18
The Rendering Engine	20
Rendering Engine Flow	20
Parsing	21
HTML DTD	22
DOM (Document Object Model)	22
Script and Stylesheet Parsing Order	23
Scripts	23
Stylesheets	23
Render Tree relation with DOM Tree	24
Internet Related General Terms	28
Network	28
LAN – Local Area Network or Intranet	28
MAN – Metropolitan Area Network	28
WAN – Wide Area Network or Internet	28
WWW – World Wide Web	28
Protocol	28
IP Address	29
Dynamic IP Address	29
Static IP Address	29
Host or Server	30
Hostname or Domain	30
Resources	31
URL – Uniform Resource Locator	31
HTML (Hyper Text Markup Language)	34
Elements	34
Attributes	35
Content Categories	38
Main Content Category	38
Metadata Content	38
Flow Content	38
Sectioning Content	39
Heading Content	39
Phrasing Content	39
Embedded Content	40
Interactive Content	40
Form-Associated Content	40
listed	40
labelable	41
submittable	41
resettable	41
Transparent Content Model	41
Architecture of HTML Document	43
<!DOCTYPE >	43
<html> Element	43
<!-- Comments -->	45
Elements Nesting	45
<head> Element - Document Header	45

<title> - Title of the HTML Document	46
<base> Element	46
<style> Element	47
<link> Element	49
<script> Element	51
<noscript> Element	52
<meta> Element	52
Document Body	58
Internet is for All	60
Hyperlinks	63
<a> - Anchor Element	63
Outline Hyperlink to Another Document	64
Inline Hyperlink to Bookmark inside current Document	65
Mixed use of Inline and Outline Hyperlink	66
Absolute URL	67
Relative URL	67
Absolute V/s Relative URL	70
target Attribute	71
Grouping Contents	74
<p> - Paragraph Element	74
<hr> - Horizontal Rule Element	76
<pre> - Preformatted Text Element	76
<blockquote> - Large Quote Element	77
List Elements	77
 - Ordered List	77
 - Un-Ordered List	77
 Element Attributes	79
<dl> - Definition List	80
<dfn> - Definition Element	82
<figure> and <figcaption> - Element	83
<div> - Division Element	84
Texts	87
Emphasizing the Content	87
 - Emphasis Element	87
 - Strong Element	88
<s> - Strike Element	88
<cite> - Cite Element	88
<small> - Small Element	89
<q> - Small Quote Element	89
<abbr> - Abbreviation Element	89
<code> Element	90
<var> Element	90
<samp> Element	91
<kbd> - Keyboard Element	91
<sub> - Sub-Script Element	91
<sup> - Super-Script Element	92
<i> - Italic Element	92
 - Bold Element	93
<mark> - Mark Element	93
<bdo> - Bi-Directional Override Element	93
dir Attribute	93
 Element	94

 and <wbr> - Line Break Elements	95
Content Edit Elements	99
<ins> - Insert Element.....	99
 - Delete Element	99
Tabular Data	102
<table> Element.....	102
<caption> Element.....	102
<tr> - Table Row Element.....	102
<thead> - Table Header Element	103
<th> - Table Header Element	103
<tbody> - Table Body Element	104
<td> - Table Data Element.....	104
<tfoot> - Table Footer Element.....	105
Left or Right Column Heading	108
<colgroup> - Column Group Element	109
<col> - Column Element	111
Interactive Elements.....	116
<details> Element	116
<summary> Element.....	116
Embedded Contents	119
 - Image Element	119
alt – Alternative Text Attribute	119
height and width Attributes	119
src – Source Attribute.....	120
iframe – Inline Frame Element.....	120
height and width Attribute.....	121
src Attribute	121
srcdoc Attribute	121
Object Element	122
height and width Attribute.....	122
data Attribute	122
type Attribute	122
<param> - Parameter Element	122
data Attribute	123
name Attribute	123
value Attribute	123
<video> Element.....	123
autoplay Attribute	123
buffered Attribute.....	123
controls Attribute	123
height and width Attribute.....	123
loop Attribute	123
muted Attribute	124
poster Attribute	124
src Attribute	124
<audio> Element.....	125
<source> Element.....	126
<canvas> Element	126
<map> - Image Map Element	127
<area> Element	127
alt Attribute	127
href Attribute.....	127

target Attribute	127
media Attribute	127
shape Attribute	127
coords – Coordinate Attribute.....	128
Semantics – Page Organizational Elements	131
Outline	131
<body> Element.....	134
<section> Element	134
<header> Element	137
<h1> . . . <h6> Elements	139
<hgroup> Element	144
<nav> Element.....	148
<article> Element.....	149
<footer> Element	151
<address> Element.....	151
Article V/s Section.....	152
For Freely Available Contents	152
For Not Freely Available Contents	153
<aside> Element.....	154
Forms – User Interaction Interface.....	157
<form> Element	157
accept-charset – Accepted Characterset Attribute	157
enctype – Encoding Type Attribute	158
method Attribute	158
name Attribute	159
novalidate Attribute.....	160
action Attribute	160
autocomplete Attribute	160
target Attribute	161
<fieldset> Element	161
name Attribute	161
disabled Attribute.....	161
<legend> Element.....	162
<label> Element.....	163
for Attribute	163
form Attribute	163
<input> Element.....	164
type Attribute	165
accept Attribute	185
autocomplete Attribute	185
autofocus Attribute	186
checked Attribute.....	186
disabled Attribute.....	186
form Attribute	186
formaction Attribute	186
formenctype Attribute	186
formmethod Attribute	187
formnovalidate Attribute	187
formtarget Attribute.....	187
height and width Attribute.....	187
list Attribute.....	187
min and max Attribute	189

maxlength Attribute	189
multiple Attribute.....	190
name Attribute	191
pattern Attribute.....	191
placeholder Attribute	191
readonly Attribute	192
required Attribute.....	192
size Attribute.....	193
src Attribute	194
step Attribute	194
tabindex Attribute	194
value Attribute	195
width Attribute.....	195
<button> Element	195
<option> Element.....	197
selected Attribute.....	198
value Attribute	198
label Attribute	198
<select> Element	198
multiple Attribute.....	198
size Attribute.....	199
selectedIndex Attribute.....	201
<optgroup> - Option Group Element	201
<datalist> Element	204
<textarea> Element	206
cols Attribute.....	206
rows Attribute	206
maxlength Attribute	207
wrap Attribute	207
<progress> Element	207
max Attribute	208
value Attribute	208
<meter> Element	209
min Attribute	209
max Attribute	209
value Attribute	209
low Attribute.....	209
high Attribute	209
optimum Attribute	209
Global Attributes.....	212
accesskey Attribute.....	212
class Attribute	212
contenteditable Attribute	213
true Value	213
false Value.....	213
dir Attribute	214
ltr Value	214
rtl Value	214
auto Value	214
draggable Attribute	215
true Value	215
false Value.....	215

hidden Attribute.....	215
id Attribute.....	216
lang Attribute.....	216
spellcheck Attribute.....	216
true Value	216
false Value.....	217
style Attribute	217
tabindex Attribute.....	217
title Attribute.....	218
CSS – Cascading Style Sheet	221
Structure (HTML).....	221
Style (CSS).....	221
Behavior (JavaScript)	221
HTML DOM	224
Ancestors.....	225
Descendents.....	226
Parent	226
Children	226
Siblings	227
Inheritance	227
CSS – Basics	229
Selector.....	229
Declaration.....	229
Property and Value.....	229
CSS File Creation	233
Linking CSS File.....	233
CSS Fundamentals	237
Keywords	237
Strings.....	238
Length and Measurement Options	239
Absolute	239
Relative	242
Percentage	243
Colors – color Property	245
Color Keywords	245
RGB Values.....	246
RGB Percentage	246
RGBA – RGB with Alpha Channel Values	246
Hexadecimal Values.....	247
Hexadecimal Shorthand Values	247
Color Selection	248
CSS Selectors.....	254
Universal Selector.....	254
Type Selector.....	254
Group Selector.....	255
Child Selector	256
Descendent Selector	257
Direct Adjacent Sibling Selector	259
Class Selector.....	260
id Selector.....	262
CSS Attribute Selectors	265
Selection on the Presence of Attribute Selector	265

Selection on the Presence of Attribute Value	265
Selection based on Attribute Values Begins with a String (^=)	267
Selection based on Attribute Values Ends with a String (\$=)	268
Selection based on Attribute Values Contains a String (*=)	268
CSS Box Model	270
Box Model Fundamental	272
Padding	273
Border	273
Margin	273
Border Property	277
Color	278
Style	278
Width	278
Handling Box Border	279
Outline Property	284
Border Radius Property	286
CSS Box – Height and Width	289
auto Value	292
100% Percent Value	293
min-width, max-width, min-height, max-height	293
Overflow Property	295
auto Value	295
scroll Value	297
hidden Value	297
inherit Value	298
overflow-x and overflow-y Properties	298
display Value	298
hidden Value	298
scroll Value	299
auto Value	299
line-height Property	299
Margin – Something Special	301
Block and Inline - Types of Elements	303
CSS Positioning	306
Box Positioning - position Property	306
Normal Flow – static Value	306
Absolute Positioning – absolute Value	310
Offset Properties – top, right, bottom and left	315
Relative Positioning – relative Value	318
Fixed Positioning – fixed Value	336
Controlling Layering – z-index Property	337
Float Positioning – float Property	342
clear Property	355
Table Positioning – display Property	366
display Property	367
Table Related display Values	374
CSS Table	390
Table Background Basics	392
border Property	393
border-collapse Property	393
border-spacing Property	394
caption-side Property	396

empty-cells Property	397
Row Width Calculation.....	398
table-layout Property.....	400
auto Value	400
fixed Value.....	401
Table Height Algorithms	402
vertical-align property	402
baseline Value	403
middle Value.....	403
text-bottom Value	404
text-top Value	404
top and bottom Values	404
sub and sup Values.....	405
CSS Background.....	410
background-color Property	410
background-image Property	410
background-repeat Property	411
repeat Value	411
repeat-x Value	412
repeat-y Value	412
no-repeat Value.....	413
background-attachment Property	413
background-position Property	413
CSS Visibility	418
visible Value.....	418
hidden Value	418
collapse Value	418
visibility:hidden vs display:none	418
CSS Fonts	420
Font Controlling Properties.....	420
font Property.....	420
font-family Property	421
font-size Property	423
font-weight Property	424
font-style Property	425
font-variant Property.....	426
System Fonts	428
Text Formatting Properties	429
color Property	429
text-align Property	429
vertical-align Property.....	431
text-decoration Property	432
text-indent Property	433
text-shadow Property	434
text-transform Property.....	436
letter-spacing Property	436
word-spacing Property	437
white-space Property.....	438
direction Property	439
CSS Lists	441
list-style-type Property	441
none Value	441

disc Value.....	441
circle Value.....	441
square Value	441
decimal Value.....	441
decimal-leading-zero Value.....	441
lower-alpha Value.....	442
upper-alpha Value	442
lower-roman Value	442
upper-roman Value.....	442
list-style-image Property	442
list-style-position Property.....	443
marker-offset Property	443
list-style Property	444
CSS Miscellaneous	446
cursor Property	446
Inheritance	447
inherit keyword	448
CSS Cascading System	448
!important Rule	452
CSS Pseudo Elements and Pseudo Class Selectors	454
::first-letter and ::first-line Pseudo Element.....	454
::selection Element	455
::before and ::after Pseudo Elements	456
content Property	456
Pseudo Classes.....	464
Dynamic Psuedo Classes	464
Static Pseudo Classes	468
:first-of-type Pseudo Class	471
:last-of-type Pseudo Class	471
:only-of-type Pseudo Class	471
:not – Negation Pseudo Class.....	471
:target Pseudo Class	472
:enabled and :disabled Pseudo Class	474
:checked Pseudo Class.....	475
:nth Pseudo Classes	476

WEB DEVELOPMENT FUNDAMENTALS

Web Development Fundamentals

Web Development = 50% Science +50% Art

हम Web को दो हिस्सों में Divide कर सकते हैं।

- **Web Site**
- **Web Application**

Website सामान्यतया Advertisement के लिए उपयोगी होती है जबकि Web Application, Data को Manage करने के लिए उपयोगी है।

यदि हम Web को अन्य तरीके से Categories करें, तो हम फिर से Web को दो हिस्सों में बांट सकते हैं:

- **Front End**
- **Back End**

हम एक Website को भी दो हिस्सों में बांट सकते हैं :

- **Static Web Site**
- **Dynamic Web Site**

Dynamic Website को भी दो हिस्सों में बांट सकते हैं :

- **Client Side Dynamic**
- **Server Side Dynamic**

Client Side Dynamic Websites को Interactive Website भी कहते हैं। सामान्यतया इस प्रकार की Websites में Client Side में JavaScript या इसके किसी Framework का प्रयोग करके Front End को Interactive बनाया जाता है। जबकि Server Side Dynamic Website में Webpage पर दिखाई देने वाले Contents User की जरूरत के आधार पर Server से बनकर Client Web Browser में Display होते हैं।

Client Side के Web Browser में User के सामने दिखाई देने वाला Web Page, Front End कहलाता है, जबकि उस Front End को Generate करने वाला अथवा उसमें Modification करने वाला Owner Website के जिस हिस्से को Access करता है, उसे Back End कहा जा सकता है।

Website का दिखाई देने वाला हिस्सा भी तीन भागों में बांटा जा सकता है :

- **Structure of Web Page**
- **Style of Web Page**
- **Behavior of Web Page**

Webpage के Structure को तय करने का काम **HTML** का होता है।

Webpage के Appearance को Define करने का काम **CSS** का होता है।

Webpage को Interactivity व Dynamic बनाने का काम **JavaScript** का होता है।

Website के Back End को दो हिस्सों में बांटा जा सकता है:

- **Server Side Scripting Language**
- **Server Side Database**

Server Side की Scripting Language के रूप में सामान्यतया ASP, PHP, JSP आदि का प्रयोग किया जाता है, जबकि Website से संबंधित Data को जिस Software में Store किया जाता है, उसे Database Software कहा जाता है, जो कि सामान्यतया SQL या MySql होता है।

Client Side से आने वाले Data को किस प्रकार से Process करना है, इस बात का निर्णय Server Side Scripting Language लेता है और Data को Process करने के बाद उसे जहां Store किया जाता है, वह DBMS Software होता है।

इनके अलावा Web Browser सामान्यतया Client Software होता है जबकि Web Server, Server Software होता है। Client किसी Resource की Request करता है और Web Server उस Request को पूरा करके Client को उसका वांछित Resource उपलब्ध करवाता है।

WEB BROWSERS

Web Browsers

Web Browsers की Functionality को समझे बिना हम Web Development को ठीक से नहीं समझ सकते। इसलिए हम सबसे पहले Web Browsers व उनकी Fundamental Functionality को ठीक से समझेंगे। वर्तमान समय में पांच Web Browsers मुख्य हैं जो सबसे ज्यादा उपयोग में लिए जाते हैं:

- **Internet Explorer**
- **Mozilla FireFox**
- **Apple Safari**
- **Google Chrome**
- **Opera**

Web Browser का मुख्य काम HTML Files को HTML व CSS Specifications के आधार पर Display करना होता है। HTML Files को सामान्यतया Web Pages कहा जाता है तथा Web Pages को Web Browser में Display होने की प्रक्रिया को Web Page का Render होना कहते हैं।

कोई Web Page किसी Web Browser में किस तरह से Render होना चाहिए, ताकि वह Web Page विभिन्न Devices जैसे कि Mobile Phones, Desktop Computers, Laptops, Notebooks, iPad, Tablet आदि में एक समान दिखाई दे, इस बात को तय करने के लिए W3C (World Wide Web Consortium) नाम की एक Organization Web के लिए विभिन्न प्रकार के Specifications तैयार करती है और विभिन्न Web Browsers बनाने वाली Companies, W3C द्वारा तय किए गए इन Standards के आधार पर अपने Web Browsers बनाती है।

W3C Organization वास्तव में बहुत सारी बड़ी कम्पनियों जैसे कि MicroSoft, Google, Apple, Dell, AT&T आदि का एक समूह है, जो आपस में मिलकर ये तय करते हैं कि Web किस तरह से Develop हो, ताकि इन सभी Companies के Software व Hardware आपस में एक दूसरे के साथ ज्यादा से ज्यादा Compatible रहें और Web Developers को विभिन्न प्रकार की Devices व Softwares के लिए अलग-अलग तरह के Codes न लिखने पड़ें।

सभी Web Browsers का User Interface लगभग एक समान ही होता है जिसमें :

- 1 URL Insert करने के लिए एक Address Bar होता है।
- 2 अगले व पिछले पेज पर जाने के लिए एक Back व Forward Button होता है।
- 3 किसी पेज को Bookmark करने के लिए Bookmark Option होता है।
- 4 एक Refresh व Stop Button होता है, जो Web Page को Web Browser में फिर से Load होने के लिए Refresh व Web Page को Web Browser में Load होने से रोकने के लिए Stop करता है।
- 5 Home Button, जो कि Web Site के Home Page पर पहुंचने के लिए होता है।

High Level Structure of Web Browser

किसी भी Web Browser के निम्न Components होते हैं :

- 1 User Interface

HTML5 WITH CSS3 IN HINDI

Web Browser का वह हिस्सा जहां पर Page दिखाई देता है, के अलावा Home Button, Back Button, Forward Button, Address Bar, Bookmarking Option आदि User Interface के अन्तर्गत आते हैं।

2 Browser Engine

ये हिस्सा Rendering Engine व User Interface Layer के बीच होता है और दोनों के बीच आपस में Communication स्थापित करवाता है।

3 Rendering Engine

User द्वारा Request किए गए Web Page को Web Browser में Render करने के लिए ये हिस्सा जिम्मेदार होता है। इसी हिस्से को पता होता है कि विभिन्न HTML Markup Elements व CSS Rules का क्या मतलब है। किसी <h1> Heading या <p> Paragraph को किस तरह से Web Browser के Document Area में Render करना है और विभिन्न Elements पर किसी CSS Style को किस तरह से Apply करना है।

4 Networking

ये हिस्सा Network Calls जैसे कि HTTP Request से संबंधित होता है। ये Platform Independent Interface होता है।

5 User Interface Backend

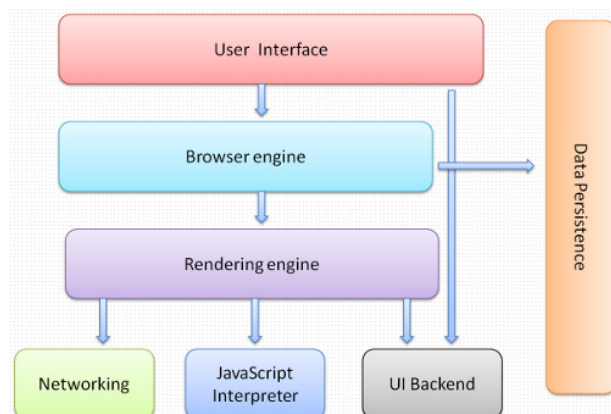
HTML Page पर दिखाई देने वाले विभिन्न User Interface Elements जैसे कि Text Box, List Box, Window आदि को Web Browser का ये हिस्सा Handle करता है। विभिन्न Form Level Elements को Handle करने के लिए ये हिस्सा Operating System के User Interface Methods को उपयोग में लेता है।

6 JavaScript Interpreter

Web Browser का ये हिस्सा JavaScript Codes को Parse व Execute करने का काम करता है।

7 Data Storage

ये हिस्सा Web Page के विभिन्न जरूरी Data जैसे कि Cookies, Browser History आदि को Device की Storage जैसे कि Hard Disk या Memory Card में Store करता है। HTML5 Specification में Local Database के Concept को Specify किया गया है, जिसमें Web Browser के साथ Local Device (Computer/Mobile आदि) पर Database Storage होना चाहिए, ताकि HTML5 का प्रयोग Local Applications Create करने के लिए भी किया जा सके।



Google Chrome एक ऐसा Web Browser है, जिसमें हर Tab के लिए एक अलग Rendering Engine होता है जबकि अन्य Web Browsers में सभी Tabs के लिए केवल एक ही Rendering Engine होता है।

The Rendering Engine

Rendering Engine का मुख्य काम है Web Page को Render करना यानी Web Page को Browser Screen (Document Windows) में Display करना। मूल रूप से Rendering Engine HTML, XML व Images को Render कर सकता है। जबकि अन्य प्रकार के Content को Display करने के लिए Browser Extension या Plugins का प्रयोग करना पड़ता है। उदाहरण के लिए PDF Document को Display करने के लिए हमें सामान्यतया PDF Viewer Plug-in (Adobe Reader) का प्रयोग करना पड़ता है।

FireFox अपने Web Browser में Gecko Rendering Engine का प्रयोग करता है जो कि FireFox का Home Made Rendering Engine है जबकि Chrome व Safari दोनों Webkit Rendering Engine का प्रयोग करते हैं। Webkit एक Open Source Rendering Engine है जो Linux Platform के लिए Develop किया गया था। बाद में Apple ने Mac OS व Windows को Support करने के लिए इसे Modify किया। फलस्वरूप Chrome व Safari Windows व Mac OS के लिए बन सके।

Rendering Engine Flow

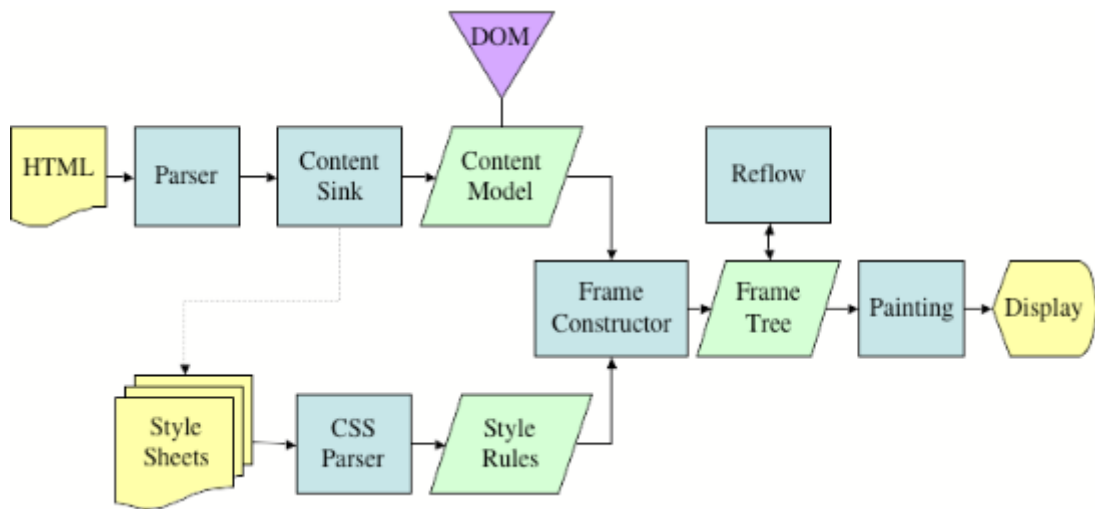
Rendering Engine Requested Document का Content Networking Layer से प्राप्त करता है। सामान्यतया ये 8KB Memory Chunk के रूप में Rendering Engine को प्राप्त होता है। Network Layer से Document Content प्राप्त करने के बाद Rendering Engine निम्न Flow में काम करता है :



- 1 Rendering Engine HTML Document की Parsing करना शुरू करता है और विभिन्न HTML Elements को DOM Nodes के रूप में Convert करता है, जिसे **Content Tree** कहते हैं। साथ ही ये Style Data (Inline Style, External Stylesheet) को Parse करता है और Structural Information (HTML Elements) तथा Styling Information (Stylesheet) दोनों को आपस में Combine करके एक नई Tree Create करता है, जिसे **"Render Tree"** कहते हैं।
- 2 सभी Style Attributes के साथ Render Tree में हर Node का एक Rectangle बनता है, जिसमें Color, Dimensions आदि की Information होती है। सभी Rectangles उसी क्रम में होते हैं, जिस क्रम में उन्हें HTML Document में Specify किया गया है।
- 3 Render Tree बनने के बाद ये Layout Process पर जाता है यानी Rendering Engine Tree के हर Node के Exact Coordinates Specify करके ये तय करता है कि हर Node, Web Browser की Screen (Document Window) में Exactly कहां पर दिखाई देगा।

- 4 अन्त में User Interface Backend Layer द्वारा Render Tree के हर Node की Traversing होती है और हर Node उसकी Exact Location पर Web Browser की Screen में Paint होता है।

Rendering Engine पूरे Document के Web Browser में Download होने का इन्तजार नहीं करता है, बल्कि जैसे-जैसे Document का Content Rendering Engine में Download होता जाता है, Rendering Engine उसे Web Browser में Render करता रहता है।



सामान्यतया सभी Web Browsers इसी क्रम को Follow करते हैं। लेकिन Webkit की Working Strategy थोड़ी अलग है। Gecko “**Frame Tree**” नाम के एक Visually Formatted Elements की Tree को Call करता है, जिसमें हर Element एक Frame होता है। जबकि Webkit “**Frame Tree**” के स्थान पर “**Render Tree**” शब्द का प्रयोग करता है जिसके सभी Elements को “**Render Objects**” कहा जाता है।

Elements को Place करने की प्रक्रिया को Webkit में “**Layout**” कहा जाता है जबकि Gecko में इसे “**Reflow**” कहते हैं। DOM Nodes व Visual Information को Connect करने की प्रक्रिया को Webkit में “**Attachment**” शब्द द्वारा Denote किया जाता है।

Webkit व Gecko दोनों में एक मुख्य अन्तर यही है कि Gecko में HTML व DOM Tree के बीच एक Extra Layer होती है, जिसे “**Content Sink**” कहा जाता है, जो कि DOM Elements Create करने का काम करता है।

Parsing

Document की Parsing एक ऐसी प्रक्रिया है, जिसका मतलब “*Document को एक ऐसे Structure में Organize करना है, जिसका कोई Meaning हो और जिसे Code द्वारा समझा व Access किया जा सके*” Parsing के Result के रूप में सामान्यतया Nodes की एक Tree प्राप्त होती है, जो Document के Structure को Represent करती है। इसे **Parse Tree** या **Syntax Tree** कहते हैं।

HTML DTD

HTML DTD एक प्रकार का Format होता है, जो HTML के सभी Allowed Elements की Definitions, Attributes व Hierarchy को Hold करता है और इस बात को Specify करता है कि कौनसे HTML Elements किस प्रकार से Organize होने चाहिए।

DOM (Document Object Model)

Parse Tree या Render Tree, DOM Elements व Attribute Nodes की एक Tree होती है। Tree के हर Node को **Object** कहा जाता है। DOM का हर Mode किसी HTML Document के विभिन्न HTML व Attribute Node तथा उनके Interfaces के समूह को Represent करता है।

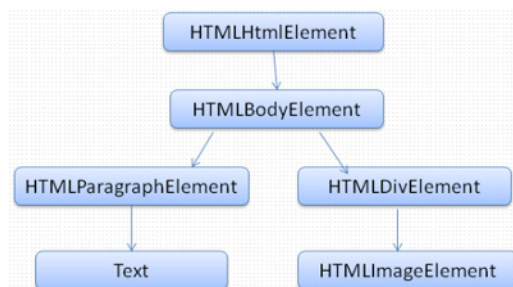
हर Object वास्तव में Properties व Methods का एक Collection होता है। Properties किसी Object की Appearance व States से संबंधित जानकारियों को Hold करता है जबकि Methods उन जानकारियों को Access करने का काम करते हैं।

Render Tree में हर HTML Element व Attribute एक Object की तरह होता है, जिसके विभिन्न Attributes जैसे कि Styles आदि उन HTML Elements की Properties के रूप में होते हैं और इन Properties को JavaScript जैसी किसी Programming Language द्वारा विभिन्न Interface Methods के Through Access किया जाता है।

DOM में Render Tree के हर Element के बीच One to One की Relationship होती है। जैसे:

```
<html>
  <body>
    <p>
      Hello World
    </p>
    <div> </div>
  </body>
</html>
```

जब ये HTML Document Parse होगा, तब बनने वाला Render Tree निम्नानुसार एक DOM Tree के रूप में Represent होगा :



HTML की तरह ही DOM भी W3C Organization द्वारा Specified है। ये Document को Manipulate करने के लिए बनाया गया एक Generic Specification है, जो तय करता है कि किसी HTML Document के विभिन्न Elements व Attributes तक किस तरह से पहुंचा जाए और किस तरह से उन्हें Access किया जाए।

DOM Tree किसी Document का In-Memory Representation होता है।

Script and Stylesheet Parsing Order

Scripts

जब HTML Document की Parsing होती है और Parser `<script>` Element पर पहुंचता है, तो Document की Parsing तब तक के लिए रुक जाती है, जब तक कि Script पूरी तरह से Web Browser में न आ जाए। यदि Script External हो, तो Parsing तब होना शुरू होता है, जब वह Script पूरी तरह से Web Browser में Download हो जाती है।

HTML5 में हम किसी Script को Asynchronous Mark करके Web Browser को ये बता सकते हैं कि Document Normal तरीके से बिना रुके हुए Download हो तथा Script एक अलग Thread में Download हो, ताकि Script के Load होने तक, Document की Parsing न रुके यानी Render Engine Script के Load होने का Wait न करे।

Webkit व Gecko, दोनों ही Engines को इस सुविधा के लिए Optimize किया गया है। जब एक Thread Script को Execute कर रहा होता है, उसी समय दूसरा Thread Document की Parsing कर रहा होता है और Document के लिए जरूरी सभी Resources को Network से प्राप्त कर रहा होता है। इस तरह से HTML5 Supported Web Browsers में Parallel Form में Document की Rendering होती है।

ध्यान रखने वाली बात ये है कि ये प्रक्रिया केवल External Scripts, Style Sheets व Images पर Apply होती है। Internal Styles, Scripts व Images Normal तरीके से ही Parse होते हैं तथा Document के आधार पर बनने वाली DOM Tree में भी किसी तरह का कोई परिवर्तन नहीं होता।

Stylesheets

Stylesheets सामान्यतया DOM Tree को किसी प्रकार से Modify नहीं करते इसलिए Stylesheet का Script से पहले Load होना जरूरी नहीं होता, लेकिन यदि कोई Script DOM Tree के किसी Style को Access करता है, तो उस स्थिति में Stylesheet का Script से पहले Load होना जरूरी होता है। इस Issue को Webkit व Gecko Engines दोनों अपने तरीके से Handle करते हैं।

यदि किसी Script को किसी Style Sheet के Content की जरूरत हो, तो FireFox Script को तब तक के लिए Block कर देता है, जब तक कि Stylesheet पूरी तरह से Load व Parse नहीं हो जाता जबकि Webkit केवल उसी स्थिति में किसी Script को Block करता है, यदि Script किसी Unloaded Style Sheet की Properties को Modify करने की कोशिश करता है।

जब DOM Tree Construct हो रहा होता है, तब Browser एक दूसरी Tree Create करता है, जिसे Render Tree कहते हैं। ये Tree Visual Elements के उस क्रम में होती है, जिस क्रम में Visual Elements Web Browser में Display होने वाले होते हैं। ये Document का Visual Representation होता है। इस Tree का मूल उद्देश्य Web Browser में Contents को सही क्रम में Paint करना होता है।

FireFox Render Tree के Elements को “**Frames**” कहते है जबकि Webkit में इसे Renderer या Render Object कहा जाता है। Renderer या Render Object ये जानता है कि उसे व उसके Childers को Web Browser में किस प्रकार से Layout व Paint होना है।

Render Tree relation with DOM Tree

हर Render Object, DOM Element से संबंधित होता है, लेकिन इनका Relation One to One का नहीं होता। Non-Visual DOM Elements (जैसे Head Element या जिन Elements की Display Property को None किया गया है।) Render Tree में Insert नहीं होते, हालांकि Hidden Visibility वाले Elements Render Tree में Appear होते हैं।

कुछ ऐसे DOM Elements भी होते हैं जो कई Visual Objects से संबंधित होते हैं। जैसे “select” Element के तीन Render Object होते हैं। एक Display Area के लिए, दूसरा Drop Down List Box के लिए व तीसरा Click किए जाने वाले Dropdown Button के लिए।

इसी तरह से किसी Multi Line Text Box में जब Text एक ही Line में पूरी तरह से नहीं समा सकता और दूसरी Link में जाता है, तब हर Line के लिए एक नया Render Object Create होता है।

कुछ Render Objects एक DOM Mode से संबंधित होते हैं लेकिन Tree में समान Location पर नहीं। जैसे Floats व Absolutely Positioned Elements Document के Normal Flow से बाहर चले जाते हैं इसलिए ये एक Render Objects की Tree में समान Place पर नहीं बल्कि किसी Different Place पर Place होते हैं।

यानी यदि सरल शब्दों में कहें, तो किसी HTML Document का हर Element व Attribute, Web Browser की Memory में एक Object के रूप में Modal हो जाता है, ताकि Web Browser, W3C Organization द्वारा Specified Specification के अनुसार उस Modal को Display कर सके। Document के इस Memory Representation को सरल शब्दों में हम **DOM Tree** कह सकते हैं, जो कि किसी भी Web Page के सारे Structure को Represent करता है।

जो Web Page हमें Web Browser में दिखाई देता है, वह Web Page, DOM Tree के रूप में Device की Memory में Stored रहता है और तब तक Stored रहता है, जब तक कि वह Web Page, Web Browser में Loaded रहता है।

चूंकि DOM Tree, Client Side Web Browser की Memory में Stored रहता है और उसी DOM Tree के अनुसार Web Page, Web Browser में Render होता है, इसलिए यदि हम DOM Tree में किसी तरह का कोई परिवर्तन करते हैं, तो वह परिवर्तन Instantly Web Browser में Reflect होता है।

JavaScript एक Client Side Programming Language है और इस Language का प्रयोग करके हम **DOM Tree** को Access व Modify कर सकते हैं और हम जैसे ही DOM Tree में कोई Modification करते हैं, उसका Effect तुरन्त Web Browser में Reflect हो जाता है।

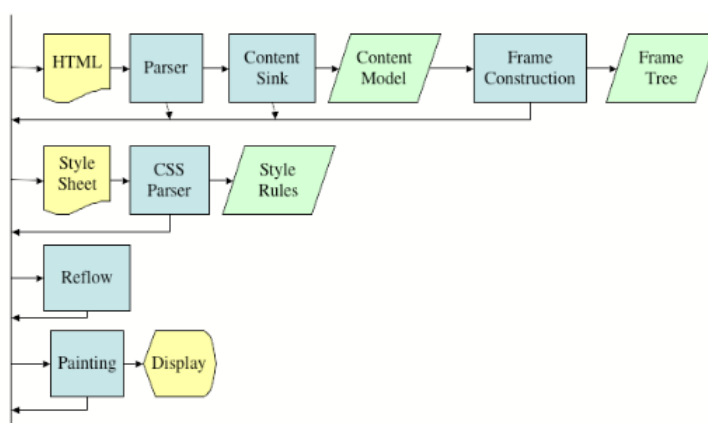
सारांश के रूप में एक Web Page में निम्न क्रम में विभिन्न Processes Perform होती हैं और Web Page दिखाई देता है:

- 1 जब किसी HTML Document को Web Browser में Load किया जाता है, तब एक HTTP Call द्वारा वह HTML Page Browser की Memory में Load होना शुरू होता है।
- 2 जैसे-जैसे HTML Document Browser में Load होना शुरू होता है, Web Browser का एक HTML Parser “Content Tree” या DOM Create करना शुरू करता है, जिसका हर Node, HTML Page के एक Element को Represent करता है।
- 3 जब कोई Script, Style, Image या Font आदि अन्य प्रकार के Resources Web Browser में Load होते हैं, तब विभिन्न Web Browsers विभिन्न प्रकार के Rules को Follow करते हुए, विभिन्न स्थितियों के आधार पर किसी Resource की Loading को कुछ समय के लिए Block करते हैं अथवा HTML Parser को किसी Resource के आने तक Wait करने के लिए Instruct करते हैं।
- 4 एक बार पूरा HTML Document Web Browser में Load हो जाने के बाद Content Tree पर Web Browser के Default Style Sheet की Styles Apply होती हैं। कई बार HTML Document के अन्दर भी कुछ Styles को Apply किया गया होता है अथवा HTML Document में External Style Sheets Specified होती हैं। इन Styles को Handle करने के लिए एक नई Tree बनती है, जिसे “**Rendering Tree**” कहते हैं।

इस Rendering Tree में केवल वे ही Elements होते हैं, जिन्हें Web Browser में Display करना होता है। जो Elements, Web Browser की Screen में Display नहीं होने हैं, जैसे कि Head, Script, Styles, आदि Elements अथवा जिन Elements की Display Property को None किया गया है, वे Elements इस Render Tree में शामिल नहीं होते। Render Tree के सभी Nodes, Style Information को Represent करते हैं।

- 5 ज्यादातर Render हो सकने वाले Elements CSS Box Model के Rules को Follow करते हैं और उनकी कुछ Width, Height, Border, Padding, Spacing, Margin व Position Values होती हैं। इन Objects के लिए एक Rectangular Box Create होता है, जिन्हें Frame कहते हैं। हर Object के लिए Frame हो, ऐसा जरूरी नहीं हैं। जैसे SVG Image की कोई Frame नहीं होती, बल्कि इसे एक iframe के अन्दर Place किया जाता है और इस iframe का एक Rectangular Box यानी Frame होता है। एक Frame में इस बात की सारी Information होती है कि किसी Element Object को Web Browser में किस प्रकार से Render होना है।
- 6 चूंकि एक ही HTML Document विभिन्न प्रकार की Devices जैसे कि Laptop, PC, iPhone आदि पर Display हो सकता है, जिनकी Screen Size, Color Depth व Resolution अलग-अलग हो सकते हैं, इसलिए ये जरूरी होता है कि Web Browser, HTML Page को Render करने से पहले Device के आधार पर इस बात का निर्णय ले कि Screen पर कौनसा Object कहां दिखाई देगा। इसलिए सभी Objects की Rendering Information प्राप्त करने के बाद Web Browser विभिन्न Elements की Screen पर Positioning की Calculation करता है कि कौनसा Element किस जगह पर दिखाई देगा।
- 7 अन्त में जब Web Browser को ये पता चल जाता है कि कौनसा Object Exactly किस Location पर Draw करना है, वह Rendering Tree के सभी Objects को Screen पर Draw कर देता है, जिसे Painting करना कहा जाता है।

सामान्यतया ये प्रक्रिया लगभग सभी Web Browsers में थोड़े बहुत Changes के साथ एक समान ही होती है, जो कि एक Static HTML Document को Render करने की प्रक्रिया है। लेकिन HTML Documents Dynamic भी हो सकते हैं और वे Dynamic तब हो जाते हैं, जब DOM Tree के साथ किसी प्रकार की प्रक्रिया की जाती है और DOM Tree के साथ किसी भी प्रकार की प्रक्रिया करने के लिए JavaScript एक अच्छी Programming Language है, जिसे वर्तमान समय में सभी Web Browsers Support करते हैं क्योंकि सभी Web Browsers में JavaScript Interpreter होता है। HTML Document Pages निम्न स्थितियों में से किसी भी स्थिति में Dynamic हो सकते हैं :



- 1 यदि DOM Tree में कोई Element Add या Delete किया जाए।
- 2 यदि किसी Element Object के Style Attribute को Modify किया जाए।

इन दोनों स्थितियों में HTML Document की Rendering की उपरोक्त सारी प्रक्रिया फिर से दोहराई जाती है। कुछ स्थितियों में पूरे DOM Tree की Rendering करने की जरूरत नहीं पड़ती है, जैसे किसी Object का केवल Color Change करने पर केवल Repainting की जरूरत पड़ती है। लेकिन किसी Object की Location Change करने पर पूरी Rendering Process फिर से Repeat होती है।

किसी भी Web Browser में Images की तुलना में Style Sheet व Scripts पहले Load होती हैं। Stylesheet का प्रयोग Rendering Tree Create करने के लिए होता है, लेकिन इसका कोई भी प्रभाव Content Tree पर नहीं पड़ता। इसलिए HTML Parsing व JavaScript Execution उस समय Continue हो रहा हो सकता है, जिस समय CSS Download हो रहा होता है।

किसी Script को जब उस समय Style Information की जरूरत होती है, जिस समय वह Stylesheet Download हो रही होती है व Rendering Tree बन रही होती है। इस प्रकार की स्थिति में Web Browser हमें Error दे सकता है। इसलिए इस प्रकार की स्थिति में Stylesheet को Script की तुलना में पहले Load करना चाहिए।

INTERNET RELATED GENERAL TERMS

Internet Related General Terms

Network

जब विभिन्न प्रकार के बहुत सारे Electronic Devices (जैसे कि Computers, NetBooks, Laptops, Mobile Phones, iPads, Tablet PCs आदि) आपस में किसी माध्यम (जैसे कि Telephone Lines, Wireless Techniques, Satellites आदि) के द्वारा एक दूसरे से Connected होते हैं और एक दूसरे के Data व Information को आपस में Share करते हुए एक दूसरे के साथ Communication करते हैं, तो इस प्रकार से आपस में जुड़े हुए Devices के Group को **Network** कहा जाता है। Networks मूल रूप से तीन प्रकार के होते हैं:

LAN – Local Area Network or Intranet

जब कुछ Electronic Devices आपस में एक छोटे से कमरे या Building के अन्दर Information Sharing व Communication करने के लिए एक दूसरे से Connected होते हैं, तो इस प्रकार के छोटे से Network को **LAN** या **Intranet** कहा जाता है।

MAN – Metropolitan Area Network

जब कुछ LANs दो या दो से ज्यादा शहरों में स्थित होते हैं और आपस में Information Sharing व Communication करने के लिए एक दूसरे से Connected होते हैं, तो इस प्रकार के मध्यम आकार के Network को **MAN** कहा जाता है।

WAN – Wide Area Network or Internet

जब बहुत सारे LANs व MANs विभिन्न देशों में स्थित होते हैं और आपस में Information Sharing व Communication करने के लिए एक दूसरे के साथ Connected होते हैं, तो इस प्रकार के बड़े Network को **WAN** या **Internet** कहा जाता है।

WWW – World Wide Web

World Wide Web एक ऐसा तरीका है, जिसमें दुनिया भर के लाखों Computers व अन्य प्रकार की Electronic Devices जैसे कि Mobile Phones आदि आपस में Connected होते हैं और आपस में विभिन्न प्रकार की Information को Share करते हैं, जो कि Web Pages या HTML Pages के रूप में Web पर स्थित होते हैं।

Protocol

दो या दो से ज्यादा Electronic Devices के बीच आपस में Communication करवाने के लिए कुछ छोटे-छोटे Standard Software बनाए गए हैं। इन Software को **Protocol** कहा जाता है। विभिन्न प्रकार की जरूरतों को पूरा करने के लिए विभिन्न प्रकार के Protocols Develop किए गए हैं।

उदाहरण के लिए यदि Internet से किसी File को एक Computer से दूसरे Computer पर Transfer करना हो, तो ये काम **FTP** (File Transfer Protocol) करता है, यदि E-Mail Send करना हो तो **SMTP** (Simple Mail Transfer Protocol) व E-Mail को प्राप्त करना हो, तो **POP** (Post Office Protocol) Use किया जाता है। जबकि यदि हमें World Wide Web पर उपलब्ध

किसी Resource को Access करना हो, तो इस जरूरत को **HTTP** (Hyper Text Transfer Protocol) पूरा करता है।

इसी तरह से विभिन्न प्रकार के अन्य Internet या Network Related कामों को पूरा करने के लिए विभिन्न प्रकार के Protocols को Develop किया गया है। इन Protocols के समूह को एक विशेष नाम से Identify किया जाता है, जिसे **TCP/IP** यानी **Internet Control Protocol / Internet Protocol** कहा जाता है।

IP Address

जब भी कोई Device Internet से Connect होता है, तो Internet द्वारा एक विशेष तरीके का प्रयोग करके Connect होने वाले हर Device को एक Unique Number प्रदान कर दिया जाता है। Internet द्वारा हर Device को दिए जाने वाले इस Unique Number को उस Device का **IP Address** कहा जाता है।

ये एक 32-Bit Number होता है, जिसमें चार 8-Bit Numbers होते हैं और चारों Numbers 0 से 255 की Range के बीच हो सकते हैं। इन चारों Numbers को एक Dot का प्रयोग करके एक दूसरे से अलग किया जाता है।

उदाहरण के लिए **170.17.8.192** किसी समय किसी Computer का एक IP Address हो सकता है। इस IP Number के दो हिस्से होते हैं: पहला हिस्सा उस **Network** को Identify करता है जिस में **Host Exist** है और दूसरा हिस्सा किसी Particular **Host** को Identify करता है। IP Address दो प्रकार के होते हैं:

Dynamic IP Address

हमारे Network द्वारा हमारे किसी Device को Provide किया गया ये IP Address Number तभी तक हमारे Device को Refer करता है, जब तक हम Net से Connected रहते हैं। जैसे ही हम Net से Disconnect होते हैं, हमें Allot किया गया IP Number किसी अन्य Device को Provide किया जा सकता है।

यदि हम फिर से Net से Connect होते हैं, तो फिर से हमें वही IP Number प्राप्त नहीं होगा, बल्कि Internet द्वारा हमें एक नया Number दे दिया जाएगा। इस स्थिति में हमारा Device तो एक ही होता है, लेकिन अलग-अलग समय पर Net से Connect होने के कारण कई IP Numbers द्वारा Identify हो सकता है। इस प्रकार के IP Address को **Dynamic IP Address** कहा जाता है।

Static IP Address

यदि हम चाहें तो हम हमारे Host यानी Server के लिए एक Static IP Address प्राप्त कर सकते हैं, लेकिन Static IP Address काफी महंगा होता है। ये एक ऐसा IP Address होता है, जो Unique होता है और कभी भी बदलता नहीं है चाहें Net से Connected रहें अथवा Disconnected रहें।

Host or Server

Network पर स्थित एक विशिष्ट प्रकार का Computer **Host** या **Server** कहलाता है। चूंकि Host सामान्यतया **Server** का काम करता है, इसलिए Host हमेशा बाकी के अन्य Computers की तुलना में अधिक Powerful होता है। TCP/IP Network के हर Host का एक Unique IP Address होता है, जिससे उस Host की Network पर एक Unique पहचान होती है।

कोई भी Powerful Configuration वाला Computer Host या Server हो, ऐसा नहीं होता बल्कि जिस Computer पर एक Special Type का Software जिसे **Web Server**, **Mail Server** अथवा **File Server** कहते हैं, Installed होता है, उसी Computer को Host अथवा Server कहा जा सकता है।

वर्तमान समय में मूल रूप से IIS व Apache नाम के दो Web Servers सबसे ज्यादा उपयोग में लिए जाते हैं। IIS Web Server Windows Operating System के लिए Microsoft Company द्वारा बनाया गया Web Server है, इसलिए इस पर Microsoft Technology की Programming Languages जैसे कि ASP या ASP.NET में बनाए गए Web Applications ज्यादा आसानी व सुविधापूर्ण तरीके से Run होते हैं, जबकि Apache Linux के लिए Develop किया गया Web Server है, जो Server Side Scripting Language के रूप में PHP को ज्यादा बेहतर तरीके से Access करता है।

Hostname or Domain

किसी भी TCP/IP Network के किसी Powerful Computer को Host बनाया जाता है, जो उसके Clients को Services Provide करता है। हर Host का एक Unique IP Address होता है, जिससे उस Host को अन्य Clients Identify करते हैं। लेकिन जब हम Internet से जुड़ते हैं, तब लाखों TCP/IP Networks आपस में Connected होते हैं, इसलिए विभिन्न प्रकार के Hosts को Identify करने के लिए IP Address को याद रखना जरूरी होता है, ताकि एक Client Required Host से किसी Service के लिए Request कर सके।

चूंकि IP Address वास्तव में एक 32-Bit Number होता है और विभिन्न Hosts के IP Numbers को याद रखना एक कठिन काम है, इसलिए विभिन्न Hosts को Internet की एक विशेष Service द्वारा एक **Logical Symbolic** नाम दे दिया जाता है और हमें किसी Host को उसके IP Number के स्थान पर उसके नाम से याद रखना होता है, जो कि तुलनात्मक रूप से सरल होता है। किसी Host के IP Address के साथ एक नाम Associate करने का काम **DNS** (Domain Name Service) व Sun Microsystems Company का **NIS** (Network Information Services) करता है।

जब हम Net से किसी Web Site के किसी Web Page को प्राप्त करना चाहते हैं, तब हमें उस Web Site का Web Address लिखना होता है। इसी Web Address में हमारे उस Host का नाम होता है, जिससे हम Connect होना चाहते हैं। Internet की जिस **DNS** व **NIS** Service का प्रयोग करके किसी Host को एक नाम Provide किया जाता है, वही Service हमारे Web Address में से Host के नाम को भी प्राप्त करने के बाद Host के नाम के आधार पर उस नाम से सम्बंधित IP Address या IP Number को प्राप्त करता है और हमें उस Host पर पहुंचा देता है, जिस पर हमारी Required Site उपलब्ध होती है। इस नाम को ही **Hostname** या **Domain Name** कहा जाता है।

जैसे www.bccfalna.com एक Domain Name है, जिसका कोई न कोई **Unique IP Address** भी होगा। लेकिन यदि हम इस Web Site पर पहुंचना चाहते हैं, तो हमें केवल इस नाम को याद रखना होता है। शेष काम **DNS** Automatically करता है।

Resources

Internet पर विभिन्न प्रकार की ऐसी Files होती हैं जिन्हें Web Browser Support करता है। ये Files कोई HTML, XML या अन्य प्रकार का Document हो सकता है, कोई Text File हो सकती है, कोई Document File हो सकती है अथवा कोई Media File जैसे कि Image, Sound अथवा Video की File हो सकती है। इन विभिन्न प्रकार की Files को सामान्यतया एक शब्द में “Resources” कहा जाता है।

URL – Uniform Resource Locator

चूंकि Internet पर कई प्रकार की Files Available हैं जिन्हें Web Browser द्वारा Access व Use किया जा सकता है। इन विभिन्न प्रकार के Resources का एक Unique Address होता है, जिसका प्रयोग करके इन Resources को Web Browser में प्राप्त व Access किया जा सकता है। Resources के इन Unique Address को ही URL या Uniform Resource Locator कहा जाता है।

यानी <http://www.bccfalna.com/home.html> home.html नाम के Resource या Document File का एक Unique Address है। इस Address को Web Browser के Address Bar में Specify करके हम Directly इस Web Page पर पहुंच सकते हैं। ये Web Address ही URL या Uniform Resource Locator है क्योंकि हम जब भी कभी इस Address को उपयोग में लेंगे, हम हमेशा home.html Document पर ही पहुंचेंगे। किसी भी URL के हमेशा तीन भाग होते हैं :

Protocol
Server Name
Resource with Path

हमारे उपरोक्त उदाहरण URL में **http://** Protocol है, www.bccfalna.com Web Server है और **home.html** Resource है। यानी

Protocol	http://
Server Name	www.bccfalna.com
Resource with Path	/home.html

जब हम इस पूरे URL को Web Browser के Address Bar में लिखते हैं, तब Web Browser इस Address से तीन बातें समझता है।

पहला ये कि हम जिस Resource को Web Server से प्राप्त करना चाहते हैं, वह एक HTML Document है क्योंकि HTML Document को ही **Hypertext Document** भी कहा जाता है और Hypertext Document को प्राप्त करने के लिए Web Browser को **HTTP Protocol Use** करना पड़ता है।

दूसरी Information Web Browser को ये मिलती है कि हम हमारा Resource जिस Web Server से प्राप्त करना चाहते हैं, उस Web Server का नाम www.bccfalna.com है। अब इस एक नाम के भी तीन हिस्से हैं:

- जहां पहला हिस्सा **www** Web Browser को ये बता रहा है कि हमारा Document जिस Host Computer पर रखा है, वह एक ऐसा Computer है, जो Internet यानी **World**

HTML5 WITH CSS3 IN HINDI

Wide Web का एक हिस्सा है क्योंकि वह World Wide Web यानी Internet से Connected है जबकि

- दूसरा हिस्सा उस Computer का नाम है, जिस पर हमारा Document Placed है जो कि **bccfalna** है और
- तीसरा हिस्सा ये बता रहा है कि ये एक **Commercial Web Document** है और इस Web Site में Directly या Indirectly कुछ न कुछ बेचने की कोशिश की जा रही है।

URL से तीसरी Information Web Browser को ये मिलती है कि हम इस Web Site से जिस Document को प्राप्त करना चाहते हैं, उस Resource का नाम **home.html** है और ये Document Web Server के Root Folder में है क्योंकि “/” Character किसी भी Computer System के Root या Home को Represent करता है।

मानलो कि हमारा वांछित Document एक **sunlight.jpg** नाम की Image File है और ये Image File हमारे Web Server www.bccfalna.com पर किसी **images** नाम के एक Folder में है Stored है। अब इस Image File को प्राप्त करने के लिए हमें जो URL Use करना पड़ेगा वह निम्नानुसार होगा:

<http://www.bccfalna.com/images/sunlight.jpg>

चूंकि हमारा Document एक Image File है और Image File एक Linkable Document है, इसलिए इसे **http://** Protocol द्वारा Access किया जा सकता है। साथ ही हमारी Image File www.bccfalna.com नाम के Web Server पर है और इस Web Server के Root पर **images** नाम का एक Folder है, इसलिए हमने इस Domain Name के बाद उस Folder के नाम को Specify किया है और अन्त में अपनी **sunlight.jpg** नाम की Resource File को Refer किया है।

HTML

HYPER TEXT MARKUP LANGUAGE

HTML (Hyper Text Markup Language)

1990 से पहले तक किसी भी तरह की Information को Internet से Access करना काफी Technical काम होता था। Internet से Information को Access करने का ये तरीका काफी Tricky होता था जो कि एक आम आदमी के बस की बात नहीं होती थी। इसलिए एक ऐसे तरीके की जरूरत हुई, जिससे Internet के माध्यम से Information को Access करना सरल हो सके। ये काम Tim Berners-Lee नाम के एक Physicist ने किया।

Tim ने एक ऐसा तरीका विकसित किया जिससे “**Hypertext**” Links के रूप में Internet से Text Information को आसानी से Cross-Reference किया जा सकता था। ये विचार वास्तव में Tim का नहीं था, लेकिन उन्होंने एक ऐसी Markup Language का विकास किया, जिससे इस विचार को Implement किया जाना सम्भव हो सका।

Hypertext का मतलब ऐसे Text से होता है, जिसे Electronic Form में Store किया गया हो और जिसे Links के माध्यम से Cross-Reference किया जा सकता हो। यानी जिस Information को Electronic Form में Convert करके Links द्वारा Cross-Reference किया जा सकता है, ऐसे Text को “**Hypertext**” कहा जाता है।

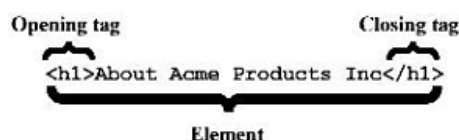
HTML (Hypertext Markup Language) Web Pages की मूल भाषा है। Internet पर आज जितनी भी Web Sites हैं, वे सभी Web Sites HTML Pages का Collection मात्र है, जिनमें HTML Elements का प्रयोग किया जाता है। HTML के इन Elements को ही **Markup** भी कहते हैं, क्योंकि ये Elements Web Page के विभिन्न प्रकार के Contents को विभिन्न तरीकों से Web Browser में Render होने के लिए Mark करते हैं और Web Browser को इस बात का Indication करते हैं, कि Web Browser में किस Content को किस जगह पर और किस तरह से Render करना है।

ये एक Markup Language है जो कि पूरी तरह से Content या Text के Organization से संबंधित है। उदाहरण के लिए किसी Newspaper में हम जो Content देखते हैं, उसमें कुछ Text बड़े होते हैं जो Heading की तरह दिखाई देते हैं जबकि कुछ छोटे होते हैं जो Sub-Heading या Paragraph की तरह दिखाई देते हैं तो कुछ Texts List के रूप में होते हैं।

Elements

HTML, किसी Document के Texts की विभिन्न प्रकार की Marking करता है और ये तय करता है कि कौनसा Text Heading के रूप में दिखाई देगा, कौनसा Text Sub-Heading के रूप में दिखाई देगा। कौनसा Text List के रूप में होना चाहिए और कौनसा Text Paragraph के रूप में होना चाहिए।

Document के Contents की Marking करने के लिए HTML Specification में जिस तरीके को उपयोग में लिया जाता है, उसे **Element** कहते हैं। इसे हम निम्न चित्रानुसार समझ सकते हैं:



HTML5 WITH CSS3 IN HINDI

विभिन्न प्रकार के HTML Elements को एक **Angle Brackets** के Pair के रूप में Specify किया जाता है। हर HTML Element के दो भाग होते हैं, जिन्हें Tag कहा जाता है। पहला Tag Opening Tag होता है और ये Opening Tag किसी Feature को **On** करता है, जबकि दूसरा Tag Closing Tag होता है और ये Tag उस **On** किए गए Feature को **Off** करता है।

उदाहरण के लिए यदि हमें हमारे Web Page में “Hello WWW” शब्द को मुख्य **Heading** के रूप में दिखाना हो, तो हमें इस Text को Web Browser में Render करने के लिए HTML Source Page में **<h1> Element** को निम्नानुसार Use करना होगा:

```
<h1>Hello WWW</h1>
```

जब Web Browser इस Line को Interpret करता है, तब उसे **<h1> Tag** का Pair का पहला Opening Tag प्राप्त होता है। यह Tag Web Browser को इस बात का Signal देता है, कि इस Tag से आगे, जो भी Text है, उसे Web Browser में **Heading** के रूप में **Render** करना है।

Web Browser इस पहले Tag से आगे लिखे गए Texts को तब तक Heading के रूप में Render करता रहता है, जब तक कि उसे Tag Pair का Closing Tag **</h1>** प्राप्त नहीं हो जाता। HTML Element के **<h1> Pair** व उस Pair के बीच लिखे गए Contents के पूरे समूह को HTML का **Element** कहा जाता है।

यानी यदि हम पिछले Syntax को देखें, तो ये एक HTML Element का उदाहरण है। एक HTML Element में सामान्यतया एक **Opening Tag** एक **Closing Tag** व दोनों Elements के बीच में लिखा गया कुछ **Content** होता है लेकिन कई HTML Elements ऐसे भी होते हैं।

HTML Specification में विभिन्न प्रकार के Elements Define किए गए हैं और इन Elements को समझ कर उन Elements के अनुसार Document को Display या Render करने का काम Web Browser करता है। यानी Web Browser किसी Document के HTML Codes को Web Server से प्राप्त करता है और उनकी **Parsing** करके Contents को Web Browser में Render कर देता है।

उदाहरण के लिए किसी HTML Document में यदि कोई Content Heading के रूप में दिखाई देना चाहिए, तो हम उस Content को Heading Element के बीच Specify करते हैं। जब Web Browser उस Document को Read करता है, तो जहां पर भी उसे HTML Elements प्राप्त होते हैं, वहां के Content को वह Element की Specification के अनुसार Modify कर देता है। इसी प्रक्रिया को सरल शब्दों में Document की **Parsing** होना कहते हैं।

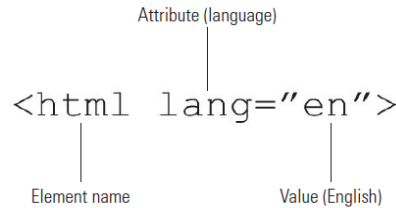
HTML वास्तव में कोई Programming Language नहीं है, बल्कि ये एक Markup Language है, जो मूल रूप से Web Browser में दिखाई देने वाले Web Page के Contents की Organization करने का काम करता है। Web Browser में हम जिस Page को देखते हैं, वह Page वास्तव में HTML Elements के आधार पर ही Render या Display होता है। यदि हम HTML Elements की File को Source Program मानें, तो Web Browser उस Source Program को Interpret करने वाले Interpreter की तरह काम करता है।

Attributes

विभिन्न Elements सामान्य काम करने के अलावा कुछ Special प्रकार के काम भी कर सकते हैं। इन Special प्रकार के कामों को करने के लिए इन Elements के Opening Tag में कुछ Properties को Specify किया जाता है। इन Properties को **Attributes** कहते हैं। ये

HTML5 WITH CSS3 IN HINDI

Attributes, Element के विभिन्न Optional या Compulsory Features को Represent करते हैं, जिसे हम निम्न चित्रानुसार समझ सकते हैं :



Element के हर Attribute के दो भाग होते हैं। पहले भाग को **Name** से व दूसरे भाग को **Value** से Represent किया जाता है।

Element के किसी Attribute की **Name Property** वह Property होती है, जिसे हम Set करना चाहते हैं और **Value** वह मान होता है, जिससे हम किसी Name Property को Set करते हैं।

उदाहरण के लिए `<h1>` एक Element है, जिसे पूरे Document Page पर Uniquely Identify करने के लिए हम उसे एक ID दे सकते हैं। इस स्थिति में **"id"** एक **Name Property** है जबकि उसमें Specified नाम एक **Value** है।

```
<h1 id="MainSite"></h1>
```

Elements and Attributes के Name को हमेंशा Small Case Letters में Specify करना चाहिए तथा Attributes में Set किए जाने वाले मान को हमेंशा Double Quotes के बीच लिखना चाहिए, क्योंकि Double Quotes के बीच हम निम्नानुसार Single Quotes को Use कर सकते हैं:

```
<h1 title="Main Site's Heading"></h1>
```

लेकिन Single Quotes के बीच हम Double Quotes को Use नहीं कर सकते।

```
<h1 title='Main Site's Heading'></h1>
```

हालांकि हम Single Quotes के बीच Double Quotes को Use कर सकते हैं। लेकिन HTML Document में Strings, Text या Name Values को Double Quotes में ही Specify करना चाहिए, ताकि Server Side Scripting Language में Single Quotes को String के लिए आसानी से उपयोग में लिया जा सके। (Optional)

HTML5

CONTENT

CATEGORIES

Content Categories

HTML5 में “Document का Contents किस प्रकार का है”, इस बात को ध्यान में रखते हुए Content को अच्छी तरह से Manage करने के लिए कई Rules बनाए गए हैं। ये Rules कई Groups के रूप में Define किए गए हैं, जिन्हें Content Models कहा जाता है और हर Group में किसी एक प्रकार के Content को Handle करने से संबंधित कई Elements Define किए गए हैं। इस Content Model में हर HTML Element Zero, One या Many Content Model से संबंधित हो सकता है। मूल रूप से HTML5 में Content को तीन मुख्य Content Categories में Divide किया गया है:

Main Content Category

इस Category में बहुत सारे Elements द्वारा Share किए जाने वाले Common Rules को Describe किया गया है, जो निम्नानुसार हैं:

Metadata Content

इस Category से संबंधित Elements Document के **Presentation** व **Behavior** को Modify करने व अन्य Documents से Links Setup करने का काम करते हैं। इस Category के मुख्य Elements निम्नानुसार हैं:

<base> , <command> , <link> , <meta> , <noscript> , <script> , <style> and <title>

Flow Content

इस Category में उन Elements को रखा गया है जो मूल रूप से Text अथवा Embedded Content को Hold करने का काम करते हैं। ये Elements निम्नानुसार हैं:

Text	<a>	<abbr>	<address>	<article>	<aside>	<audio>
<bdo>	<blockquote>	 	<button>	<canvas>	<cite>	<code>
<datalist>		<details>	<dfnrr>	<div>	<dl>	
<fieldset>	<figure>	<footer>	<form>	<h1>	<h2>	<h3>
<h5>	<h6>	<header>	<hgroup>	<hr>	<i>	<iframe>
<input>	<ins>	<kbd>	<keygen>	<label>	<map>	<mark>
<menu>	<meter>	<nav>	<noscript>	<object>		<output>
<pre>	<progress>	<q>	<ruby>	<samp>	<script>	<section>
<small>			<sub>	<sup>	<svg>	<table>
<time>		<var>	<video>	<wbr>		<command>
<h4>		<math>	<p>	<select>	<textarea>	<embed>

कुछ अन्य Elements भी इस Category से संबंधित हैं, लेकिन वे कुछ विशेष स्थितियों में ही उपयोग में लिए जाते हैं, जो निम्नानुसार हैं:

- <area> ये <map> Element का Descendent Element होता है।
- <link> ये तभी Use करते हैं जब **itemprop** Attribute Present हो।
- <meta> ये भी तभी Use करते हैं जब **itemprop** Attribute Present हो।
- <style> इसे तब Use करते हैं, जब **scoped** Attribute Present हो।

Sectioning Content

इस Group से Associated Elements, Current Document की Outline Create करने के लिए उपयोगी होते हैं, जो कि **<header>**, **<footer>** व अन्य **Heading Elements (h1, h2, h3, h4, h5, h5, hgroup)** का Scope Define करते हैं। इस Category से संबंधित Elements निम्नानुसार हैं:

<article>, <aside>, <nav>, <section>

Heading Content

इस Group के Elements किसी Section में लिखे गए Content का Title Define करते हैं। Section को दो तरीकों से Mark किया जा सकता है। पहला तरीका Manual होता है, जिसमें विभिन्न Sectioning Elements को Use किया जाता है। इन Elements को Use करके लिखा गया Content, **Sectioning Content** कहलाता है, जबकि दूसरे तरीके में उस समय Automatically Invisible Section बन जाते हैं, जब हम किसी Heading Element (h1, h2, h3, h4, h5, h6, hgroup) का प्रयोग करते हैं। इस Category से संबंधित Elements निम्नानुसार हैं:

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <hgroup>

हालांकि Sectioning Content कुछ Headings को Hold करता है, लेकिन **<header>** Element को इस Group में शामिल नहीं किया गया है।

Phrasing Content

इस Group के अन्दर वे Elements आते हैं, जो Text अथवा अन्य Mark-ups को Hold करने का काम करते हैं। Phrasing Content सामान्यतया Paragraphs Create करते हैं। इस Category से संबंधित Elements निम्नानुसार हैं:

<abbr>	<audio>		<bdo>	
	<button>	<canvas>
<cite>	<code>	<command>	<datalist>	<dfn>		<embed>
<i>	<iframe>		<input>	<kbd>	<keygen>	<label>
<mark>	<math>	<meter>	<noscript>	<object>	<output>	<progress>
<q>	<ruby>	<samp>	<script>	<select>	<small>	
	<sub>	<sup>	<svg>	<textarea>	<time>	
<var>	<video>	<wbr>	Plain Text			

कुछ विशेष परिस्थितियों में हम कुछ और Elements को इस Category के Elements के रूप में Use कर सकते हैं, जो कि निम्नानुसार हैं:

<a>, <area>, , <ins>, <link>, <map>, <meta>

Embedded Content

इस Category में वे Elements होते हैं, जो किसी अन्य Resource अथवा Content को किसी अन्य Mark-up Language अथवा Namespace से Import करते हैं। इस Category से Related Elements निम्नानुसार हैं:

<audio> , <canvas> , <embed> , <iframe> , , <math> , <object> , <svg> , <video>

Interactive Content

इस Category में वे Elements होते हैं जो User Interaction के लिए Design किए गए हैं। ये Elements निम्नानुसार हैं:

<a> , <button> , <details> , <embed> , <iframe> , <keygen> , <label> , <select> , <textarea>

कुछ अन्य Elements कुछ विशेष परिस्थितियों में इस Category में Participate करते हैं, जो निम्नानुसार हैं:

- <audio>** यदि controls Attribute Present हो।
- ** यदि usemap Attribute Present हो।
- <input>** यदि type Attribute Hidden न हो।
- <menu>** यदि type Attribute Toolbar State में हो।
- <object>** यदि usemap Attribute Present हो।
- <video>** यदि controls Attribute Present हो।

Form-Associated Content

इस Category में वे Elements आते हैं जो किसी Form Element के अन्दर Nested होते हैं। सामान्यतया इस Category के Elements के साथ **form** Attribute में **id** Specified होता है अथवा ये Form के अन्दर होते हैं। ये Elements निम्नानुसार हैं:

<button> , <fieldset> , <input> , <keygen> , <label> , <meter> , <object> , <output> , <progress> , <select> , <textarea>

इस Category के Elements की कुछ Sub-Category भी है, जो निम्नानुसार है:

listed

इस Category के Element निम्नानुसार हैं:

<button> , <fieldset> , <input> , <keygen> , <object> , <output> , <select> , <textarea>

labelable

जो Elements `<label>` Element के साथ Associated हो सकते हैं, इस Category में आते हैं। इस Category के Element निम्नानुसार हैं:

```
<button> , <input> , <keygen> , <meter> , <output> , <progress> , <select> ,  
<textarea>
```

submittable

जो Elements, Form Submitting के समय Form Data Set Create करने के लिए Use होते हैं, इस Category में आते हैं। इस Category के Element निम्नानुसार हैं:

```
<button> , <input> , <keygen> , <object> , <select> , <textarea>
```

resettable

जो Elements, Form Resetting के समय Affected होते हैं, इस Category में आते हैं। इस Category के Element निम्नानुसार हैं:

```
<input> , <keygen> , <output> , <select> , <textarea>
```

Transparent Content Model

`` व `<ins>` Elements Transparent Elements हैं। ये वे Elements होते हैं, जिन्हें HTML Document से Remove कर देने पर भी Final Output Rendered Content पर कोई प्रभाव नहीं पड़ता है।

ARCHITECTURE OF HTML DOCUMENT

Architecture of HTML Document

<!DOCTYPE >

किसी भी HTML5 Document का ये सबसे पहला HTML Element होता है। ये Element Web Browser को बताता है कि Current Document एक HTML5 Supported Document है और इसमें HTML5 Specification के Standards यानी Elements का प्रयोग किया गया है।

यदि हम ये Line न भी लिखें, तो भी Web Browser हमारे Document को HTML5 Document की तरह ही Process करता है, क्योंकि Web Browser हमारे Document को DOCTYPE के आधार पर नहीं बल्कि Web Browser के Internal Architecture के आधार पर Parse करता है और Web Browser में यदि HTML5 Element को Handle करने के लिए Functionality को Specify किया गया है, तो Web Browser उस Functionality को Document पर जरूर Apply करेगा, भले ही हमने DOCTYPE Specify किया हो या नहीं।

लेकिन जब हम बिना DOCTYPE वाले Document की Validating करते हैं, तब हमें Errors प्राप्त होती है। इसलिए हर HTML5 Document में निम्न Line लिखकर ये जरूर Specify करना चाहिए, कि हमारा Current HTML Web Page एक HTML5 Elements Supported Document है:

```
<!DOCTYPE html>
```

HTML5 DOCTYPE Case Sensitive नहीं है, इसलिए हम इसे निम्नानुसार भी लिख सकते हैं:

```
<!doctype html>
```

<html> Element

हमारा पूरा Document एक HTML Document है, इसलिए हमारे पूरे Document के सारे Elements को हमें <html> Element के Opening व Closing Tags के बीच ही लिखना होता है। ये Element किसी भी Web Page का **Root Element** होता है। सभी अन्य Elements इस Element के **Descendants** होते हैं। इसलिए किसी भी HTML Document को Represent करने के लिए हमें सबसे पहले **html** Element को Use करना होता है। इस Element के मुख्यतः दो भाग होते हैं अथवा दूसरे शब्दों में कहें तो इस Root **html** Element के दो और केवल दो Child Elements होते हैं:

The <head> Element

ये Element हमेशा HTML Page के Top पर होता है। इस Element में वह Information होती है, जो Argument के रूप में Browser को Pass होती है। इस Element के बीच जो भी कुछ होता है, वह अन्य HTML Elements की तुलना में सबसे पहले Web Browser में Load होता है।

सामान्यतया इस Element में Web Page का **Title** होता है, जो Browser के Title Bar पर दिखाई देता है। इसके अलावा इस Element में Page का **Description** व Page के **Keywords** को Define करने वाले Elements को भी सम्मिलित किया जाता है। इन Elements का प्रयोग Search Engines हमारे Web Page को खोजने के लिए करते हैं। इस Element के अन्दर निम्न Elements का प्रयोग किया जा सकता है:

How to Buy from BccFalna.com

इस Website पर उपलब्ध सभी **Saleable Hindi EBooks** के साथ "ADD TO CART" नाम का एक Button Attached है। आप जो भी पुस्तक खरीदना चाहते हैं, उसके साथ Associated **ADD TO CART** Button को Click करते ही वह पुस्तक आपके **Shopping Cart** में Add हो जाती है:

My Shopping Cart

Data Structure and Algorithms in Hindi	
Delete	
Advance JavaScript in Hindi	
Delete	
Java Programming Language in Hindi	
Delete	
Total Amount	Rs. 1000.00
Discount Amount	Rs. 200.00
Total Payable Amount	Rs. 800.00

[View Cart](#)
[Checkout](#)

Hindi EBooks – Add to Cart => Checkout => Pay => Download


[View Cart](#) ⇨ "Java Programming Language in Hindi" was successfully added to your cart.

Product	Price	
C Programming Language In Hindi	Rs. 350.00	ADD TO CART
C++ Programming Language In Hindi	Rs. 350.00	ADD TO CART
Java Programming Language In Hindi	Rs. 350.00	ADD TO CART
C# Programming Language In Hindi	Rs. 400.00	ADD TO CART
ADO.NET With C# In Hindi	Rs. 350.00	ADD TO CART
Oracle 8i-9i SQL/PLSQL In Hindi	Rs. 350.00	ADD TO CART

यदि आप अपने Shopping Cart में कई पुस्तकें Add करते हैं, तो **Extra Discount** प्राप्त होता है, जो कि **Discount Amount** व **Discount Amount** घटाने के बाद सभी पुस्तकों के **Total Payable Amount** के रूप में इसी **My Shopping Cart** में उपरोक्त चित्रानुसार दिखाई देता है।

सभी वांछित पुस्तकें अपने **Shopping Cart** में Add करने के बाद अपना **Order Place** करने हेतु **Checkout** Button को Click करना होता है। परिणामस्वरूप निम्नानुसार **Checkout Page** Display होता है, जहां आपको अपनी **Billing Details** को Specify करके अपना **Payment Mode** Select करना होता है:

Billing Details		Ordered EBooks	
Country *		Product	Total
India		C in Hindi × 1	Rs. 350.00
Full Name *		C++ in Hindi × 1	Rs. 350.00
Kuldeep		Cart Subtotal	Rs. 595.00
Address *		Order Total	Rs. 595.00
Subhash Road Nilkanth Nagar			
Town / City *			
Falna Station			
State / County *	Postcode / Zip *		
Rajasthan	306116		
Email Address *	Mobile No. *		
bccfalna@gmail.com	09799455505		

☒ Online Payment 

नीचे दिए गए "Place order" Button पर Click करते हुए अपना Order Place कीजिए और अपने Debit Card (ATM Card, Credit Card), Cash Card, Internet Banking या Mobile Banking द्वारा Payment करते हुए खरीदी गई पुस्तकें Download कीजिए, तुरन्त।

Payment करने से सम्बंधित किसी भी प्रकार की समस्या के समाधान के लिए 097994-55505 पर Call/Miss Call करें।

☐ Offline Manual Payment

Place order

यदि आप इन पुस्तकों को खरीदने के लिए **Total Payable Amount** का भुगतान अपने Debit Card (ATM, Credit Card), Cash Card अथवा Net Banking) द्वारा घर बैठे करना चाहते हैं, तो आपको **Online Payment** Option Select करना होता है।

लेकिन यदि आपके पास किसी प्रकार का CC-Avenue Supported **Debit Card** (ATM, Credit Card) या **Cash Card** नहीं है, न ही आपके पास किसी Bank की Internet या Mobile Banking सुविधा है, तो उस स्थिति में अपना Offline Order Place करने के लिए **Offline Manual Payment** Option को Select कर सकते हैं।

Online Payment using CCAvenue

जब आप **Online Payment** Option को Select करते हुए **"Place order"** Button पर Click करते हैं, तो आपके सामने निम्नानुसार Page Display होता है:

BccFalna - Hindi EBooks

Billing Information

ORDER DETAILS

Order #: 6382_14060730

Coupon Code [Apply](#)

Order Amount 2.00

Order Total **INR 2.00**

Payment Information

Debit Cards >
Net Banking
Cash Card
Mobile Payments

We Accept

Card Number

Expiry Date

CVV

Issuing Bank Name

INR 2.00 (Total payable)

Fill your Debit Card / Net Banking / Cash Card or Mobile Payments information here and click on "Make Payment" Button

Powered by
CCAvenue®

अपनी सुविधा अपने *Debit Card (ATM, Credit Card)*, *Net Banking*, *Cash Card* या *Mobile Payments* Option को Select कीजिए और उपरोक्त चित्र में दर्शाए अनुसार सभी जरूरी Payment Information को Fill करके **Make Payment** Button पर Click कीजिए तथा आगे आने वाले Payment Security से सम्बंधित Step (Login/Password/Pin) Follow कीजिए, ताकि आपका Total Payable Amount आपके Bank A/c से हमारे Bank A/c में Transfer हो सके।

आपका Payment Transfer होते ही आप निम्न चित्रानुसार [My Account](#) Page पर पहुंच जाते हैं, जहां पर आप द्वारा Order की गई सभी पुस्तकों के Download Links होते हैं, साथ ही आपको Automatically एक Email भी Send कर दिया जाता है, जिसमें आप द्वारा खरीदी गई पुस्तकों के **Download Links** होते हैं:

My Account



Thank you for shopping with us. Your account has been charged and your transaction is successful. We will be shipping your order to you soon.

Hello **Kuldeep** (not Kuldeep? [Sign out](#)). From your account dashboard you can view your recent orders, manage your shipping and billing addresses and [edit your password and account details](#).

Available downloads

- | | |
|--|-----------------------|
| ⬇ C in Hindi – C Programming Language in Hindi | 6 downloads remaining |
| ⬇ C in Hindi – Windows Programming with C in Hindi | 6 downloads remaining |
| ⬇ C++ in Hindi – C++ Programming Language in Hindi | 6 downloads remaining |

Recent Orders

Order	Date	Status	Total	
#6382	June 7, 2014	Completed	Rs. 2.00 for 2 items	View

Offline Payment using Manual Ways

जब आप **Offline Manual Payment** Option को Select करते हुए “Place order” Button पर Click करते हैं, तो Click करते ही आपका Order Place हो जाता है और आपके सामने निम्नानुसार Page Display होता है:

Checkout

Thank you. Your order has been received.

ORDER: #6472

DATE: June 10, 2014

TOTAL: Rs. 1,160.00

PAYMENT METHOD: Offline Manual Payment

How to Deposit Payment पर दिए गए किसी भी तरीके को Use करते हुए **Our Bank Accounts** पर Specified किसी भी Bank A/c में “Order Total” (Total Payable Amount) का Payment Deposit करने के बाद अपने Payment Deposit करने की जानकारी देने के लिए **097994-55505** पर Call/Miss Call करें और अपने Order ID (जैसे कि Order: #6388) तथा अपने Payment Deposit करने से सम्बंधित जानकारी दें।

जैसे ही आपका Payment हमारे किसी भी Bank A/c में Deposit होगा और हमें आपका Payment Confirmation Call/Miss Call प्राप्त होगा, 10 Minute में आप द्वारा Order की गई EBooks के Download Links का EMail आपको Send कर दिया जाएगा, जहां से आप अपनी Purchase की गई पुस्तकों को Download कर सकेंगे। इन पुस्तकों को आप हमारी Website के **My Account** Menubar Option Page से भी Download कर सकते हैं।

Order Details

Product	Total
C in Hindi × 1	Rs. 280.00
C++ in Hindi × 1	Rs. 280.00
C# in Hindi × 1	Rs. 320.00
Cart Subtotal:	Rs. 1,160.00
Order Total:	Rs. 1,160.00

ये Webpage आप द्वारा Place किए गए Order की Information के साथ ही Order की गई पुस्तकों के **Download Links** प्राप्त करने के लिए Follow किए जाने वाले अगले Step की जानकारी भी देता है, साथ ही इस Page पर दिखाई देने वाली सारी Information आपको आपके Email पर भी Send कर दी जाती है, जिन्हें Follow करते हुए आप अपने **Total Payable Amount** का **Offline Manual Payment** करके अपनी Order की गई पुस्तकों के Download Links प्राप्त कर सकते हैं।

जब आप इस **Offline Manual Payment** Option को Select करते हुए Order Place करते हैं, तो आपका Order तब तक **On-Hold** Status में रहता है, जब तक कि आप **Offline Manual Payment** Page पर Specified किसी भी तरीके का प्रयोग करते हुए अपना **Total Payable Amount**, हमारे **Bank A/c** में Transfer/Deposit नहीं कर देते।

अपना **Total Payable Amount** हमारे **Bank A/c** में Transfer/Deposit करने के बाद आपको हमारे Mobile No.: **097994-55505** पर Call/Miss Call करके अपने Payment Transfer/Deposit करने से सम्बंधित जानकारी देनी होती है। जैसे ही आपका Call/Miss Call हमें प्राप्त होता है, हम अपना Bank A/c Check करते हैं और जैसे ही आपका Payment हमारे Bank A/c में Transfer/Deposit होता है, हम आप द्वारा Order की गई पुस्तकों का Download Link Manually Activate कर देते हैं।

परिणामस्वरूप आपको Automatically एक EMail प्राप्त होता है, जिसमें आप द्वारा Order की गई सभी पुस्तकों के Download Links होते हैं, जिन्हें आप अगले 48 घण्टों के दौरान Download कर सकते हैं। साथ ही आपके Download Links के Activate होने की Information हम आपको Call/SMS के माध्यम से भी देते हैं।

जबकि अपने Order की Current Status देखने के लिए आप Website के Menubar में दिखाई देने वाले [My Account](#) Menu Option पर Click कर सकते हैं, जहां आपके Order की Current Status Information निम्न चित्रानुसार दिखाई देती है:

My Account

Hello **Kuldeep Mishra** (not Kuldeep Mishra? [Sign out](#)). From your account dashboard you can view your recent orders, manage your shipping and billing addresses and [edit your password and account details](#).

Recent Orders

Order	Date	Status	Total	
#6472	June 10, 2014	On-hold	Rs. 1,160.00 for 4 items	View
#6381	June 7, 2014	Completed	Rs. 1.00 for 1 item	View

चूंकि ये सारा Process हमें व आपको Manually Follow करना होता है, इसलिए इस Offline Manual Payment द्वारा Order करने की स्थिति में पुस्तकों का Download Link प्राप्त होने में 5 से 10 मिनट का समय लगता है।

एक बार Download Link Activate हो जाने के बाद आप अपनी खरीदी गई पुस्तकों को अपने [My Account](#) Page से भी Download कर सकते हैं, जहां Download Link Activate होने के बाद आपको अपना [My Account](#) Page निम्न चित्रानुसार दिखाई देने लगता है:

My Account

Hello **Kuldeep Mishra** (not Kuldeep Mishra? [Sign out](#)). From your account dashboard you can view your recent orders, manage your shipping and billing addresses and [edit your password](#) and account details.

Available downloads

- ⬇ C in Hindi – C Programming Language in Hindi 8 downloads remaining
- ⬇ C in Hindi – Windows Programming with C in Hindi 8 downloads remaining

Recent Orders

Order	Date	Status	Total	
#6520	June 11, 2014	Completed	Rs. 350.00 for 1 item	View
#6472	June 10, 2014	Processing	Rs. 1,160.00 for 4 items	View
#6381	June 7, 2014	Completed	Rs. 1.00 for 1 item	View

इसके अलावा किसी Particular Order की Details प्राप्त करने के लिए आप इस [My Account](#) Page पर दिखाई देने वाले **View** Button को भी Click कर सकते हैं।

Offline Manual Methods to Pay “Total Payable Amount”

अपना **Total Payable Amount** Pay करने के लिए आप अपनी सुविधानुसार निम्न में से किसी भी तरीके को Use कर सकते हैं:

Fund Transfer Using ATM Machine

वर्तमान समय में लगभग सभी Banks अपनी ATM Machine द्वारा **Fund Transfer** करने की सुविधा Provide करते हैं, जहां आप अपने ATM Card द्वारा हमारे किसी भी Bank Account में अपनी पुस्तकों का **Total Payable Amount** Transfer कर सकते हैं। अतः यदि आपके पास निम्न में से किसी भी State Bank का **Debit Card** है:

- **SBI** (*State Bank of India*)
- **SBBJ** (*State Bank of Bikaner and Jaipur*)
- **SBH** (*State Bank of Bikaner and Hyderabad*)
- **SBP** (*State Bank of Bikaner and Patiala*)
- **SBM** (*State Bank of Mysore*)
- **SBT** (*State Bank of Travancore*)

अथवा PNB (*Punjab National Bank*) या BOB (*Bank of Baroda*) का **Debit Card** है, तो आप SBI/PNB/BOB के **ATM Machine** से भी अपना Payment हमारे SBI/PNB/BOB Bank A/c में Transfer कर सकते हैं।

यदि आप SBI ATM Machine से हमारे SBI Bank A/c में Payment Transfer करना चाहते हैं, तो आपको निम्न Steps को Follow करना होता है:

- SBI ATM Machine में अपना **Debit Card, Swipe** कीजिए।
- ATM Screen के **Bottom Right Corner** में दिखाई देने वाले **Transfer** नाम के Option को Select कीजिए।
- अपने **Debit Card** का **PIN Number** Enter कीजिए।
- अब **Card to Card Transfer** नाम के Option को Select कीजिए।
- अब हमारे **SBI Debit Card Number (6220180786800030243)** को Enter कीजिए।
- अब हमारे **SBI Debit Card Number (6220180786800030243)** को दोबारा Enter कीजिए।
- अब **Transferable Amount** के रूप में **Total Payable Amount** Specify कीजिए।
- अब अपने **Account Type** (Savings or Checking) को Select कीजिए।
- उपरोक्त सभी Steps सही तरीके से Follow होने की स्थिति में आपका **Transaction Complete** हो चुका है और **Total Payable Amount** लगभग तुरन्त हमारे SBI Bank A/c में जमा हो जाता है।

ठीक इसी तरह के Steps आपको उस समय भी Follow करने होते हैं, जब आप **PNB या BOB** के **ATM Machine** के माध्यम से हमारे **PNB Bank A/c** में **Total Payable Amount, Card to Card Transfer** करते हैं।

हालांकि आप **HDFC, IDB, ICICI** जैसे कई अन्य **ATM Machines** द्वारा भी **Card to Card Transfer** कर सकते हैं, लेकिन यदि दोनों **Debit Cards** समान Banks (PNB, SBI or BOB) के न हों, तो Transaction Perform होने में 24 से 96 घण्टे का समय लगता है।

इसलिए इस स्थिति में बेहतर यही होता है कि यदि आपके पास **Net Banking, Mobile Baking, AirTel Money** या **SBI/PNB/BOB Debit Card** किसी भी तरह की सुविधा न हो, तो आप **Total Payable Amount** का Payment करने के लिए **Cash Deposit** तरीके को ही Use करें अथवा Bank में जाकर **NEFT Transfer** भी कर सकते हैं, जिसमें आपका Payment अधिकतम 4 घण्टे के दरम्यान हमारे Bank A/c में Deposit हो जाता है।

Payment Transfer Using Net-Banking

यदि आपके पास **Net-Banking** की सुविधा है, तो आप Payment Transfer करने के लिए अपने Account में Login करके निम्न में से किसी भी Bank A/c में Payment Deposit कर सकते हैं:


 भारतीय स्टेट बैंक State Bank of India <i>With you - all the way</i>		
SBI Bank A/c no.	:	31154882587 (Saving A/c)
Account Name	:	Namita Mishra
Branch Name	:	Falna
Address	:	Near Railway Crossing, Falna Station – 306116
IFSC Code	:	SBIN0007868
Branch Code	:	007868
MICR Code	:	306002100
ATM Debit Card No.	:	6220180786800030243 (Maestro) <i>(For Card to Card Transfer using ATM Machine)</i>



PNB Bank A/c no.	:	4445000100034960 (Saving A/c)
Account Name	:	Kuldeep Chand
Branch Name	:	Falna
Address	:	College Road, Falna Station – 306116
IFSC Code	:	PUNB0444500
Branch Code	:	444500
MICR Code	:	306024200
ATM Debit Card No.	:	5126 5200 1286 3655 (MasterCard) (For Card to Card Transfer using ATM Machine)



BOB Bank A/c no.	:	35260100003212 (Saving A/c)
Account Name	:	Namita Sharma
Branch Name	:	Falna
Address	:	Sanderao Road, Falna, Dist. Pali (Raj.)- Pin-306116
IFSC Code	:	BARB0FALNAX
Branch Code	:	FALNAX
MICR Code	:	NON-MICR
ATM Debit Card No.	:	4029850310081366 (VISACard) (For Card to Card Transfer using ATM Machine)

 स्टेट बैंक ऑफ बीकानेर एण्ड जयपुर State Bank of Bikaner and Jaipur <i>The Bank with a vision</i>		
SBBJ Bank A/c no.	:	61089986732 (Saving A/c)
Account Name	:	Kuldeep Chand Mishra
Branch Name	:	Bali
Address	:	Sr. Secondary School Road, Bali- 306701
IFSC Code	:	SBBJ0010193
Branch Code	:	010193
MICR Code	:	306003193

जब आप **Net-Banking** के माध्यम से Payment करना चाहते हैं, तो आपको लगभग **8 से 24 घण्टे पहले** हमारे उस Account को **Beneficiary** के रूप में अपने Bank A/c से Link करना पड़ता है, जिसमें आप **Payment Transfer** करना चाहते हैं। जब एक बार हमारा Bank Account Beneficiary के रूप में Activate हो जाता है, उसके बाद आप उस Bank Account में अपना **Total Payable Amount** Transfer कर सकते हैं।

Pay with Mobile-Banking or AirTel Money

यदि आपने अपने **Mobile Number** पर **AirTel Money** नाम की Service को Activate किया हुआ है, तो आप अपने Mobile द्वारा AirTel Money Account के माध्यम से भी हमें Payment कर सकते हैं। जबकि यदि आपने अपने Bank से Mobile Banking की सुविधा को Activate करवाया हुआ है, तो आप अपने Mobile द्वारा हमें **Mobile-Banking** के माध्यम से भी Payment Transfer कर सकते हैं।

यदि आप अपना **Total Payable Amount** Pay करने के लिए **AirTel Money** या **Mobile Banking** Transfer सुविधा को Use करते हैं, तो आपका Payment तुरन्त हमारे Account में Transfer हो जाता है। इसलिए तुरन्त EBooks प्राप्त करने हेतु **Payment Transfer** करने का ये सबसे तेज तरीका है। जबकि इसके अलावा जो दूसरा सबसे तेज तरीका है, वह ATM Machine द्वारा **Fund Transfer** सुविधा का उपयोग करते हुए Payment Transfer करना है।

जबकि **Internet-Banking** उस स्थिति में काफी धीमा Process है, जब आप पहली बार Payment कर रहे होते हैं, क्योंकि First Time Payment करने से पहले आपको Beneficiary के रूप में हमारे किसी एक Bank Account को अपने Bank Account से Link करना पड़ता है और इस Process में Payment Transfer हेतु हमारा Bank Account **Activate** होने में कम से कम 8 से 24 घण्टे का समय लगता है। हालांकि एक बार Account Activate हो जाने के बाद आपका Transfer तुरन्त हो जाता है।

जबकि **Cash Deposit** का तरीका सबसे धीमा तरीका है, जहां आप Bank Holidays को Payment Deposit नहीं कर सकते और Business Days में भी Payment Deposit करने का एक निश्चित समय **10PM to 4AM** होता है।

Cash Deposit in Bank Branch

यदि आपके पास Net-Banking या Mobile-Banking की सुविधा नहीं है, तो आप हमारे किसी भी Bank A/c में **Total Payable Amount**, **Cash Deposit** भी कर सकते हैं अथवा आप Bank Branch में जाकर **NEFT Transfer** के माध्यम से भी Payment कर सकते हैं, जो कि Cash Deposit के समान ही होता है।

जब आप **Direct Deposit** करना चाहते हैं, तब आपको आपके किसी भी नजदीकी Bank Branch में जाकर एक **Payment Deposit Slip** Fill-Up करना होता है, जिसमें आपको हमारे किसी भी Bank A/c की Information को Fill करना होता है, जबकि **Payment Deposit** करवाने के लिए उसी **Bank** में आपका स्वयं का **Account** होना जरूरी नहीं है।

उदाहरण के लिए यदि आप हमारे SBI Bank A/c में अपनी Selected पुस्तकों का **Total Payable Amount** Pay करने के लिए Bank में जाकर Direct Deposit करना चाहते हैं, तो आप जो **Payment Deposit Slip** Fill-Up करेंगे, वह अगले चित्रानुसार करना होता है।

SB/C/C/RD/DL/TL A/C PAY-IN-SLIP

Date 15/4/12

भारतीय स्टेट बैंक प्रगतिनगर (8053) शाखा
State Bank of India, Pragatinagar (8053) Branch

पूरा नाम/ FULL NAME Namita Mishra

खाता क्र. A/C No. 31154882587

रोकड़ / चेको का विवरण राशि AMOUNT
DETAILS OF CASH/CHEQUES रु. Rs. पै. Ps.

	600	a
कुल / Total	600	a

रुपये / Rupees 600

अधिकारी / OFFICER

टिप्पणी : अंतरण लिखनों को वसूली के बाद जमा किया जाएगा।
Note : Transfer instruments will be credited after realisation.

इस चित्र द्वारा आप समझ सकते हैं कि Payment, Direct Deposit करने के लिए आपको हमारे किसी Bank A/c की Information को *Payment Deposit Slip* में Specify करना होता है, इसलिए उस Bank में आपका स्वयं का Bank A/c होना जरूरी नहीं होता।

इसी तरह से यदि आप चाहें, तो हमारे किसी भी Bank A/c में Check द्वारा भी **Total Payable Amount** का **Check Deposit** कर सकते हैं।

यानी आप किसी भी तरीके से हमारे किसी भी Bank A/c में *Total Payable Amount* Deposit कर सकते हैं। लेकिन हम **Money-Order, Demand-Draft** या **Check** जैसे Manual माध्यमों से Payment Accept नहीं करते, क्योंकि इस तरह का Payment Clear होने में बहुत समय लगता है।

जबकि **ATM Fund Transfer, Cash Deposit, Mobile Banking** अथवा **Net-Banking** के माध्यम से तुरन्त Payment Transfer हो जाता है, जिससे हम आपको आपकी Purchased EBooks **10 से 30 Minute** के दरम्यान आपके Order में Specified **Email Address** पर Send कर देते हैं।

अपना Payment करने के लिए आप जिन **Offline Manual** तरीकों को उपयोग में ले सकते हैं, उनकी **Detailed Information** आप <http://www.bccfalna.com/how-to-deposit-payment/> से भी प्राप्त कर सकते हैं, जहां आपको Payment करने से सम्बंधित किसी भी तरह का Latest Update प्राप्त होता है।

Pay with PayPal if you live Out Of India

यदि आप **India** में नहीं रहते लेकिन ये **Hindi EBooks** खरीदना चाहते हैं, तो आप अपनी वांछित पुस्तकों के **Total Payable Amount** का भुगतान हमें **PayPal** के माध्यम से bccfalna@gmail.com पर भी Send कर सकते हैं।

चूंकि **International Payment Processing** में विभिन्न प्रकार के **Extra Charges** Pay करने होते हैं, इसलिए PayPal के माध्यम से Payment करते समय आपको **Total Payable Amount (In USD) + \$2** का Extra Payment Send करना जरूरी होता है।

अपना Payment Send करने के बाद आप अपने Payment की Information हमें SMS या Email के माध्यम से दे सकते हैं। जैसे ही आपका **Email/SMS** हमें प्राप्त होगा, आप द्वारा Order की गई पुस्तकों का Download Link आपके **Email Address** पर जितना जल्दी सम्भव होगा, उतना जल्दी Send कर दिया जाएगा।

Confirm the Payment

जब आप अपनी Order की गई पुस्तकों को खरीदने के लिए उपरोक्तानुसार किसी भी **Offline Manual** तरीके से “*Total Payable Amount*” हमारे किसी भी Bank A/c में Deposit/Transfer कर देते हैं, तो Payment Deposit/Transfer करते ही आपको हमें उसी **Mobile Number** से एक **Call/Miss Call/SMS** करना होता है, जिसे आपने Order Place करते समय “**Order Form**” में Specify किया था।

इसी Mobile Number के माध्यम से हमें पता चलता है कि आपने किन पुस्तकों के लिए कौनसा Order किया है और उनका *Total Payable Amount* कितना है। साथ ही हमें ये भी पता चल जाता है कि आप द्वारा Purchase की जा रही पुस्तकें किस **Email Address** पर Send करनी है।

आपके *Total Payable Amount* को हम Net-Banking के माध्यम से अपने Bank A/c में Check करते हैं और यदि आपका *Total Payable Amount* हमारे किसी भी Bank A/c में Deposit/Transfer हुआ होता है, तो हम आपको **10 Minute** के दरम्यान आपकी Order की गई EBooks आपके Email Address पर Send कर देते हैं, जिसे आप अगले 2 दिनों में कभी भी Download कर सकते हैं।

If you have any problem

यदि पुस्तकें खरीदने से सम्बंधित किसी भी प्रकार की कोई बात आपको ठीक से समझ में न आई हो या किसी भी तरह का Confusion हो, तो आप **097994-55505** पर **Call/Miss Call/SMS** कर सकते हैं। यथा सम्भव तुरन्त आपकी समस्या का समाधान किया जाएगा।

चूंकि ये सारी पुस्तकें **PDF Format Softcopy Ebooks** हैं इसलिए इन पुस्तकों का **Download Link** आपको आपके Email पर ही **Send** किया जाता है, जिन्हें Click करते ही ये पुस्तकें आपके Computer पर Download होना शुरू हो जाती हैं।

एक बार इन पुस्तकों को Download करने के बाद आप इन्हें किसी भी PDF Supported *Computer, Mobile, Smart Phone, Tablet PC, Net-Book, Notebook* या *Laptop* जैसी Device के माध्यम से पढ़ सकते हैं अथवा यदि आप चाहें, तो अपने Printer द्वारा इन पुस्तकों का Hard Copy Printout निकाल सकते हैं।

चूंकि इन पुस्तकों के Download Links आपको आपके Email Address पर ही प्राप्त होते हैं, इसलिए जरूरी है कि उपरोक्त "**Order Form**" पर आप अपना जो **Email Address** व **Mobile Number** Specify करते हैं, वह Working और एकदम सही हो। क्योंकि किसी भी तरह की परेशानी होने की स्थिति में हम आपको आपके [Mobile Number](#) अथवा [Email Address](#) द्वारा ही Contact करते हैं।