

PHP in Hindi

BccFalna.com
097994-55505

Kuldeep Chand

In this EBook, I have covered approximately all Core Concepts of PHP. I have described Object Oriented PHP Concepts too in great details, because these days "OOPS" is the mainly used Programming Pattern in each Programming or Scripting Language because of the benefit it provides us for Code Reusing and Object Orientation.

After learning Core Concepts of PHP, you can easily move to any PHP Framework like Symfony, WordPress, Joomla, Drupal, etc... and you can easily develop Theme or Plugin of these most used CMS and Frameworks.

In this EBook, I have not only covered Core Concepts of PHP but also I have tried to introduce various Programming Features we require in any Professional Web Application like Logging System.

I think, this is the Only EBook available on NET with Great Details of CORE PHP in Hindi Language and after reading this EBook, you would really be in Good Situation to work on a Professional Web Application.

PHP

In Hindi



Kuldeep Chand

Betalab Computer Center
Falna

Core PHP 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 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.....	12
Web Development Sequence and Used Technologies	20
PHP – History and Introduction	29
Types of Programming Languages.....	29
Environment Setup	31
Creating First PHP Script.....	36
Setting PHP Interpreter Path	38
Interpreting First PHP Script	41
PHP Fundamentals	47
<?php ... ?>	47
Output Statements.....	47
print() Statement.....	48
echo() Statement.....	48
Case Sensitive.....	48
Statements and Semicolons	49
Block Statement.....	49
Whitespace and Line Breaks	49
Comments	50
“C” Style Comment or Multiline Comment.....	50
“C++” Style Single Line Comment.....	50
Unix Shell Style Single Line Comment.....	51
Literals	51
Identifier	51
Variables	52
Constants	52
Function Names	53
Class Names	54
Keywords	54
Data Types	55
Scalar Data Types.....	55
Compound Data Types	68
Special Data Types	68
Variable.....	69
Variable Initialization and Assignment Types.....	72
Variable Variables	75
Expressions	77
Operand	77
Operators	77
Control Statements	96
Types Of Control Statement.....	97
Conditional Statements	98
Un-Conditional Statements	107
Iteration - Loops	110
Jumping Statements.....	119
Arrays	124
Associative Array.....	129
Multidimensional Array	138
Array Related Functions.....	144
Function	198
Built-In or Library Function and User Defined Functions.....	200

Function Nesting	200
Calling Function and Called Function.....	201
Function Creation or Function Definition	201
Types of User Defined Functions	203
Default Arguments.....	209
Returning Multiple Values	212
Variable Number of Arguments	212
Variable Functions.....	214
Anonymous Functions.....	216
Recursive Functions.....	218
Variable Scope and Lifetime	218
User Defined Function Library.....	225
File Inclusion Statements	226
String	231
String Printing Statements.....	231
String Length.....	239
String Comparision.....	240
String Case.....	243
String Cleaner	244
String Padding.....	246
Characters and Words Counting.....	247
Character ⇔ ASCII Code.....	250
Array ⇔ String.....	251
Hexadecimal ⇔ Binary.....	255
Searching and Manipulating.....	256
Miscellaneous String Functions.....	270
Type Casting.....	272
Converting to Boolean.....	273
Converting to Integer.....	274
Converting to Float.....	275
String Conversion to Numbers	275
Converting to String.....	276
Converting to Array	277
Converting to Object.....	277
Converting to Resource.....	278
Converting to NULL.....	278
Converting to Binary.....	279
Conversion Functions	279
floatval() Function.....	279
doubleval() Function.....	280
intval() Function.....	280
strval() Function.....	281
Testing Functions	282
isset() Language Construct	282
empty() Language Construct.....	283
unset() Function	284
is_ Functions	285
PHP Character Functions.....	288
Date and Time	291
Unix Timestamp	291
PHP Date and Time Library	292
Summary	299

Object Oriented PHP	301
Object Oriented Programming Concepts	301
Encapsulation	303
Inheritance	304
Polymorphism	304
Abstraction	304
Class	306
Object	310
Access Scope Specifier	312
Initializing Data Members	318
Scope Resolution Operator (::)	318
Constant Data Members	318
Constructors	319
Destructors	321
Static Class Members	322
Object Cloning	326
PHP Overloading	334
__set() Method	334
__get() Method	338
__isset() Method	340
__unset() Method	341
__call() Method and __callStatic() Method	341
Inheritance	342
Method Overriding	346
protected Scope Access Specifier	351
Inheritance and Constructors	352
Abstract Class	354
Final Class	359
Interface	361
Autoloading Objects	367
Traits	371
Precedence of Methods	374
Multiple Traits	376
Conflict Resolution	377
Changing Method Visibility	383
Traits in Trait	383
Abstract Traits	385
Static Trait Members	386
Static Methods	387
Trait Properties	388
Type Hinting	388
Object Iteration	390
Object Serialization	392
Serialization Format	397
Magic Methods	399
__toString() Method	399
__sleep() and __wakeup() Methods	400
__invoke() Method	402
__set_state() Method	403
Late Static Binding	406
instanceof Keyword	409
Object / Class Functions	410

class_exists() Function.....	410
get_class() Function.....	412
get_class_methods() Function.....	414
get_class_vars() Function.....	415
get_declared_classes() Function.....	417
get_object_vars() Function.....	417
get_parent_class() Function.....	418
interface_exists() Function.....	419
is_a() Function.....	420
is_subclass_of() Function.....	421
method_exists() Function.....	422
class_alias() Function.....	423
get_called_class() Function.....	424
get_declared_interfaces() Function.....	424
get_declared_traits() Function.....	425
property_exists() Function.....	426
trait_exists() Function.....	427
Function Handling Function.....	427
Callback Functions.....	428
Namespaces.....	437
Creating Namespaces.....	438
Sub-Namespace.....	440
Using Namespace.....	441
PHP Rules for Accessing Namespaces.....	445
Exception Handling.....	446
Default Constructor.....	448
Methods.....	449
User Defined Exception Class.....	452
Multiple catch Block.....	453
Summary.....	455
PHP Web Facilities.....	458
HTTP Basics.....	458
PHP Super Global Variables.....	460
\$_COOKIE Global Array.....	460
\$_GET Global Array.....	460
\$_POST Global Array.....	461
\$_FILES Global Array.....	461
\$_ENV Global Array.....	461
\$_SERVER Global Array.....	461
Processing Forms.....	462
Methods.....	462
Parameters.....	463
File Inclusion.....	469
Self Processing Pages.....	472
Multivalve Parameters.....	476
\$_SERVER – Server Information.....	479
SERVER_SOFTWARE Key.....	479
SERVER_NAME Key.....	479
SERVER_INTERFACE Key.....	479
SERVER_PROTOCOL Key.....	479
SERVER_PORT Key.....	479
REQUEST_METHOD Key.....	479

PATH_INFO Key	480
PATH_TRANSLATED Key	480
SCRIPT_NAME Key	480
QUERY_STRING Key	480
REMOTE_HOST Key	480
REMOTE_HOST Key	480
AUTH_TYPE Key	480
REMOTE_USER Key	480
CONTENT_TYPE Key	481
CONTENT_LENGTH Key	481
Setting Response Headers	482
Different Content Types	483
Downloadable File	486
Uncatchable File	486
Redirection	486
Page Expiration	487
Maintaining State	487
Cookies	489
Sessions	497
File Management System	504
Data Organization	505
Working with Files	507
Opening a File - fopen() Function	508
Closing a File - fclose() Function	513
Writing to File	514
Navigation in File	516
Reading from File	519
Formatted File Writing and Reading	530
Remote Data Sources	536
allow_url_fopen	538
allow_url_include	538
safe_mode	538
Creating Socket Connection	539
Sending Request Message to Remote Server over Socket	540
Receiving Response Message from Remote Server over Socket	542
PHP File System Related Functions	544
Path Related Functions	545
Size Related Functions	547
File Related Time Functions	548
Reading Directory Contents	550
General Functions	553
Shell Command Functions	554
Dynamic Website	558
MVC Pattern	558
Database Fundamental	560
Primary Key	562
Foreign Keys	562
MySql with phpMyAdmin	564
MySql Data Types	573
Datatype Attributes	579
MySql Storage Engines	582
Using MySQL with PHP	585

Establishing Connection between MySQL and PHP	585
Executing SQL Queries on MySQL through PHP	587
Closing the Connection	588
Retrieving Query Results from MySQL Database	593
Database Security	596
mysqli_num_rows() Function	599
Updating MySQL Records with PHP	600
PHP with MySQL in Detail	601
Connection with Database	602
Interaction with Database	604
Parsing Query Results	608
Working with Prepared Statements	611
HTML Related Special Functions	614
addslashes() Function	614
stripslashes() Function	615
addslashes() Function	616
stripslashes() Function	616
htmlspecialchars() Function	616
htmlspecialchars_decode() Function	618
htmlentities() Function	619
html_entity_decode() Function	619
nl2br() Function	620
Last but not Least. There is more	621

WEB DEVELOPMENT

Web Development

हम Web को दो हिस्सों में Divide कर सकते हैं।

- 1 **Web Site**
- 2 **Web Application**

Website सामान्यतया Advertisement के लिए उपयोगी होती है जबकि Web Application, Data को Manage करने के लिए। Web Sites को हम एक अन्य तरीके से फिर से दो भागों में बांट सकते हैं:

- 1 **Static Web Site**
- 2 **Dynamic Web Site**

Static Web Site ऐसी Web Site होती है, जिसके Contents को केवल एक बार Develop किया जाता है और बहुत कम बार Modify किया जाता है। ऐसे Content को बार-बार Modify करने की जरूरत नहीं होती। जबकि **Dynamic Web Site** ऐसी Web Site होती है, जिसके Content समय-समय पर और बार-बार जरूरत के अनुसार बदलते रहते हैं।

उदाहरण के लिए किसी Company के विभिन्न Employees की Information या Company के विकास की Information, किसी School के विभिन्न विद्यार्थियों की Personal Information, Teachers की Personal Information आदि ऐसी बातें हैं, जो लम्बे समय तक नहीं बदलती। इसलिए इस प्रकार की Information को जब Web Site के माध्यम से Represent किया जाता है, तो बनने वाली Web Site एक **Static Web Site** होती है।

जबकि किसी Company के विभिन्न Employees की Salary की Information या Company के Growth से संबंधित Information जैसे कि Balance Sheet आदि बार-बार बदलती रहती है। इसी तरह से किसी School के विभिन्न Students की Mark-Sheet के Numbers व Results हर साल बदलते रहते हैं। जब इस प्रकार की Information को किसी Web Site के माध्यम से Represent किया जाता है, तब जिस प्रकार की Web Sites बनानी पड़ती हैं, उन्हें **Dynamic Web Sites** कहते हैं।

यदि इस प्रकार की Web Sites को ज्यादा बेहतर शब्द से Represent करें, तो इसे **Web Applications** भी कह सकते हैं। क्योंकि ये एक Full Flash Software होते हैं जो किसी एक Specific Type की जरूरत को Best तरीके से पूरा करते हैं व किसी एक समस्या से संबंधित विभिन्न प्रकार की Information को Best तरीके से Manage करते हैं। Dynamic Website को भी हम 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।

एक ऐसी Web Site जिसे Visit करते समय, User उस Web Site के साथ किसी तरह का Interaction भी कर सकता है, किसी Item को Click कर सकता है, अधिक जानकारी के लिए किसी तरह के Animation को देख सकता है या अपनी जरूरत के अनुसार Content को Client Side में Modify करने में सक्षम होता है। यानी जब User Client Side में किसी

Web Site के साथ किसी तरह का कोई Interaction कर सकता है, तो इस प्रकार की Web Sites को **Interactive Web Site** कहते हैं।

जबकि **Server Side Dynamic Website** में Webpage पर दिखाई देने वाले Contents User की जरूरत के आधार पर Server से बनकर या Modify होकर Client Web Browser में Display होते हैं। यानी ये Web Page ऐसे Web Page होते हैं, जो User की जरूरत के अनुसार Web Server पर Dynamically बनते हैं।

उदाहरण के लिए जब कोई Student अपनी Mark Sheet को Internet से प्राप्त करना चाहता है, तो वह किसी Web Site के किसी Web Page पर दिखाई देने वाले Form में अपना Name या Roll Number Enter करता है और उसे केवल उसी की Mark Sheet प्राप्त होती है।

जबकि हम जानते हैं कि उसी Form पर किसी अन्य Roll Number या नाम को Specify करने पर वही Web Page फिर से दिखाई देगा, लेकिन उसके Content पिछले वाले Result की तुलना में बिल्कुल अलग होंगे, क्योंकि सभी Students का Result व Mark Sheet एक समान नहीं होता।

इस प्रकार की Web Site जो कि अलग-अलग Input के लिए अलग-अलग Result प्रदान करे, **Dynamic Website** कहलाती है और जब User को दिखाई देने वाला Web Page किस तरह का दिखाई देगा, ये निर्णय User द्वारा Input किए गए Data के आधार पर **Web Server** लेता है, तो इस प्रकार की Dynamic Web Site को **Server Side Dynamic Website** कहा जाता है।

एक Server Side Dynamic Web Site के भी दो हिस्से होते हैं। जो हिस्सा Client Side के Web Browser में User के सामने दिखाई देता है, वह हिस्सा **Front End** कहलाता है जो कि सामान्यतः **Web Page** होता है, जबकि उस **Front End Web Page** पर User को क्या और कैसा दिखाई देना चाहिए, इसे Dynamic Web Site के जिस हिस्से द्वारा Control किया जाता है, या Generate किया जाता है, उस हिस्से को **Back End** कहा जा सकता है।

किसी User को दिखाई देने वाला Website का हिस्सा यानी **Front End** भी तीन भागों में बांटा जा सकता है :

1. **Structure of Web Page**
2. **Style of Web Page**
- 3rd **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। जबकि Webpage को Interactivity व Dynamic बनाने का काम **JavaScript** का होता है।

इसी तरह से Website के Back End को तीन हिस्सों में Divide किया जा सकता है:

1. **Web Server or Host**
2. **Server Side Scripting Language**
- 3rd **Server Side Database**

सामान्यतः नए **Programmers Web Clients** व **Web Server** को ठीक से नहीं समझ पाते जो उन्हें तेजी से **Web Programming** सीखने में काफी परेशानी में पैदा करता है। इसलिए सबसे पहले ये समझना जरूरी है कि आखिर ये **Web Site** होती क्या है और लोग **Web Site** क्यों बनवाते हैं।

यदि हम **Root Level** पर समझें, तो एक **Web Sites** केवल किसी **Information** को **Represent** करने का **Electronic Medium** है। यदि **Professional** शब्दों में कहें तो एक **Web Site** वास्तव में एक सबसे कम खर्चे वाला तथा **Customers** के लिए सबसे ज्यादा सुविधापूर्ण तरीके से किसी **Product** के बारे में **Detail** से **Information** देने वाला **Advertising Medium** है। जिसे विभिन्न **Businessman** अपने किसी **Product** को **Internet** के माध्यम से **Directly Sell** करने अथवा किसी **Product** को **Indirectly Promote** करने के लिए **Develop** करवाते हैं।

क्या आपने कभी सोचा है कि **Advertising** का मूल उद्देश्य क्या होता है? **Advertising** का मूल उद्देश्य किसी **Product** की **Direct Selling** करना अथवा **Indirect Promoting** करना ही होता है, ताकि उस **Product** के **Owner** का व्यापार बढ़ सके।

Internet किसी व्यापार को बढ़ाने में **Newspaper, Radio, Television** आदि की तरह ही एक बहुत बड़ा माध्यम है, जहां करोड़ों लोग किसी भी समय **Internet Surfing** करते हुए उपलब्ध रहते हैं और इसीलिए विभिन्न **Business Owners** अपनी **Web Site** बनवाते हैं।

सामान्यतः नए **Web Programmers** एक और गलती करते हैं। उनका तर्क ये होता है कि सभी **Web Sites** हमेशा किसी **Product** की **Selling** या **Promoting** नहीं करतीं, बल्कि ज्यादातर **Web Sites** पर वे जो **Information** देखते हैं, उनमें कहीं भी किसी भी तरह के **Product** का जिक्र नहीं होता। उदाहरण के लिए **Google, Yahoo** आदि **Web Sites** हैं, जो बिना किसी तरह की **Fees** लिए हुए हमें **Internet** पर **Searching** करने की सुविधा देते हैं।

जबकि सच्चाई ये है कि हर **Web Site** किसी ना किसी **Product** की **Selling** के लिए ही **Develop** की जाती है और जो **Web Site Online Selling** या **Promotion** नहीं करतीं, वे **Web Site** ज्यादा समय तक **Available** नहीं रहतीं।

नए **Web Programmers Marketing Fundas** व **Advertising Tricks** से अनभिज्ञ होते हैं। उन्हें हर **Web Site** पर **Selling** के लिए **Product** दिखाई नहीं देते, इसलिए वे समझते हैं कि वह **Web Site** मुफ्त में सारी जानकारी दे रहा है और यहीं नए **Web Programmers** गलती करते हैं। सामान्यतः वे समझते हैं कि हर **Product Physical** होता है, जो कि उनकी सबसे बड़ी भूल है।

Product हमेशा **Physical** ही हो, ऐसा जरूरी नहीं है। उदाहरण के लिए यदि मानलें कि आपने अपनी **School** की पढाई पूरी कर ली और अब आप किसी **Best MBA College** में **Admission** प्राप्त करना चाहते हैं। ये जानने के लिए कि सबसे अच्छा **MBA College** कौनसा है और वहां क्या **Fees** है, कितने साल का कोर्स है, क्या **Subjects** पढाए जाते हैं, पुराने **Students** की **Job Placements** की क्या स्थिति है, आदि विभिन्न प्रकार की जानकारियों को प्राप्त करने के लिए आप लोगों से पूछते हैं अथवा आप **Internet** पर **Surfing** का प्रयोग करते हैं।

जब आप **Internet** द्वारा किसी **College** का **Selection** करना चाहते हैं, ताकि आप **Best College** में **Admission** प्राप्त कर सकें, तो वास्तव में आप विभिन्न **Colleges** की **Advertising** ही तो देख रहे होते हैं। क्योंकि आप जिस किसी भी **School** या **College** में

Admission लेंगे, वह School या College आपसे Fees के रूप में पैसा वसूल करेगा और यदि उस College ने अपनी Web Site न बनवाई होती, तो आपको उस College के बारे में जानकारी कैसे मिलती। यदि आप उस College के बारे में Internet के माध्यम से जान नहीं पाते, तो आप वहां Admission कैसे लेते और यदि आप उस Internet द्वारा Selected College में Admission नहीं लेते, तो वह College आपसे Fees कैसे वसूलता।

यदि ध्यान से देखा जाए, तो यहां आपने उस College से कुछ नहीं खरीदा, फिर भी फीस के रूप में आपने उसे पैसा दिया। तो आप कैसे कह सकते हैं कि हर Web Site अपना Product Sell नहीं करती क्योंकि किसी College के लिए उसके Course ही उसके Products हैं और कोई Service या Course कोई Physical वस्तु नहीं होती, फिर भी उसे खरीदा जाता है।

यानी मूल रूप से समझने वाली बात ये है कि Internet दुनिया का एक सबसे बड़ा **Advertising Medium** है और हर Web Site किसी न किसी Product की Advertising के लिए ही बनायी गई होती है, फिर चाहे वह Product Physical हो अथवा Virtual.

जब Product Physical होता है, तब Web Site उस Product को Directly Offer करती हैं, ताकि जो User उस Web Site को देखे, वे उस Product को Directly Internet के माध्यम से Online खरीद सकें जबकि जो Product Directly Sellable नहीं होते, जैसे कि कोई Service या Course, उन्हें Internet के माध्यम से Promote किया जाता है, ताकि लोग उन Offer की गई Services के बारे में जानें और Web Site Owner को ज्यादा से ज्यादा Customers यानी Clients मिलें, ताकि उसका व्यापार बढ़ सके। इस प्रक्रिया को हम **Indirect Selling** या **Promotion** कह सकते हैं।

तो सारांश के रूप में कहें तो हर Web Site किसी ना किसी Businessman की ही होती है और हर Businessman चाहता है कि उसे ज्यादा से ज्यादा Customers मिलें। इसलिए एक Businessman के नजरिए से समझें, तो उसकी Web Site को Visit करने वाला हर User, उसका एक Customer या **Client** है। क्योंकि जो User किसी Owner की Web Site को Visit करता है, उस User को भी **Client** कहा जा सकता है, जबकि User जिस Owner की Web Site को Visit करता है, उस Web Site Owner को **Server** भी कहा जा सकता है, क्योंकि वह Owner अपने User को किसी न किसी तरह की **Physical Product** या **Virtual Service** Provide करता है।

जिस प्रकार से Real World में **Client** व **Server** होते हैं, जैसाकि हमने उपरोक्त Discussion द्वारा समझा, उसी प्रकार से जब हम Web Development की बात करते हैं, तब भी Client व Server होते हैं, जिन्हें सामान्यतः **Web Client** व **Web Server** कहा जाता है।

Web Client व Web Server को Web Development के नजरिए से ठीक से समझने के लिए हमें दो **Perspectives** को ध्यान में रखना होता है, क्योंकि Web Client व Web Server, दोनों दो Layers का समूह होते हैं:

Hardware Layer **Software Layer**

User जिस Computer या Device जैसे कि Computer, Laptop, Notebook, Mobile Phone आदि के माध्यम से Internet को Use करता है, वह माध्यम **Hardware Layer** को Represent करता है। जबकि वह User अपनी Device में Installed जिस Software के माध्यम से Internet Surfing करता है, वह माध्यम **Software Layer** को Represent करता है, जो कि सामान्यतः **Web Browser** होता है।

यानी वह Device, जिसके द्वारा User Internet Use करता है, **Hardware Client** है। साथ ही उस Device में Installed वह Software जो कि सामान्यतः Web Browser होता है, जिसके माध्यम से User Internet Surfing करता है, **Software Client** है।

Software हमेशा **Hardware** पर निर्भर होते हैं और ये दोनों हमेशा साथ में होते हैं तभी उपयोगी होते हैं। यानी यदि आपके पास Computer हो, लेकिन उस Computer में कोई **Web Browser** जैसे कि Internet Explorer, Mozilla, Firefox, Safari, Chrome आदि न हो, तो आप Internet Surfing नहीं कर सकते, क्योंकि कोई भी Device बिना उपयुक्त Software के Electronic पुर्जों के एक Box के अलावा और कुछ नहीं होता इसलिए बिना Web Browsers के आपका Computer आपको Web Surfing नहीं करवा सकता।

जबकि यदि दूसरे तरीके से देखें, तो बिना किसी Hardware के किसी Software का कोई औचित्य ही नहीं होता क्योंकि Software हमेशा किसी न किसी Hardware के अन्दर ही होता है। परिणामस्वरूप यदि आपके पास Internet Surfing करने के लिए कोई Device ही नहीं है, तो फिर Software ही नहीं सकता।

Hardware व Software के इस Combination को ही **Web Client** या **Web Server** कहा जाता है। यानी एक **User**, जो कि किसी Web Site को Visit करता है, एक **Device**, जिसके माध्यम से User किसी Web Site को Visit करता है और वह **Web Browser Software** जिसके बिना User किसी Web Site को Visit नहीं कर सकता, तीनों का Combined रूप **Web Client** को Represent करता है, लेकिन एक Web Developer के नजरिए से हम केवल Web Browser को ही **Web Client** या **Client Software** कहते हैं क्योंकि Web Browser किसी Device पर निर्भर नहीं होता इसलिए हर Device में समान या भिन्न Web Browser हो सकता है और एक Web Programmer के रूप में हमें केवल Web Browser के बारे में ही सोचना होता है।

अब हम Web Server के बारे में समझते हैं। Web Client यानी **Device + Web Browser Software** किसी User को ये सुविधा देते हैं, कि वह Internet पर किसी तरह की Request Perform कर सके। उदाहरण के लिए जब User किसी Web Site का URL किसी Web Browser के Address Bar में Fill करके Enter Key Press करता है अथवा HTML Web Page पर दिखाई देने वाले किसी Link को Click करता है, तो वास्तव में वह एक प्रकार की Request कर रहा होता है, जो इस बात को Indicate करता है कि वह उस URL या Link से Associated Information को जानना चाहता है।

चूंकि सामान्यतः एक Device को केवल एक ही User Use कर रहा होता है और वह User एक बार में केवल एक ही Request करता है, इसलिए User के Device का High Quality व High Performance का होना जरूरी नहीं होता, लेकिन User जिस Web Site को Visit कर रहा होता है, उसी समय उसी Web Site को लाखों लोग Use कर रहे हो सकते हैं।

उदाहरण के लिए जिस समय आप Google पर कुछ Search कर रहे होते हैं, उसी समय लाखों लोग उसी Google के उसी Home Page पर किसी ना किसी तरह की Searching कर रहे होते हैं। इस स्थिति में एक ही समय पर लाखों लोगों की Requirements को सामान्य से Computer या Mobile Phone Device द्वारा पूरा किया जाना सम्भव नहीं हो सकता।

इसलिए User जिस Web Site को Visit करता है, उस Web Site को एक बहुत ही High Power व High Quality के Computer System पर Host किया जाना जरूरी होता है, जो

कि हर समय On रहे। इस High Power Configuration वाले Computer System को सामान्यतः **Web Host** कहा जाता है।

हर **High Configuration** वाला Computer Web Host होता है, ऐसा समझना गलत है। आप अपने स्वयं के Computer को भी Web Host की तरह Use कर सकते हैं और आगे आने वाले Contents में हम ऐसा करेंगे भी। लेकिन क्योंकि एक Web Host को लाखों लोगों की Requests को समान समय पर पूरा करने की जरूरत हो सकती है, इसलिए Web Host Computers का किसी भी अन्य Computer System की तुलना में ज्यादा Powerful होना जरूरी होता है।

फिर से ध्यान दें कि एक High Power Configuration वाला Computer System ठीक उसी तरह से आ सकने वाली लाखों Requests को अकेले Handle नहीं कर सकता, जिस तरह से एक User का Device बिना Client Software के Web Surfing नहीं कर सकता। यानी इस High Power Configuration वाले Computer System को भी एक Software की जरूरत होती है, जो आने वाली Requests को Handle करता है। ये High Power Configuration वाला Computer System तो केवल उन Requests को पूरा करने की गति को बढ़ा देता है, ताकि कम से कम समय में ज्यादा से ज्यादा Users की Requests को पूरा किया जा सके। इस Special Software को **Web Server Software** कहते हैं।

Web Server Software का हमेशा किसी High Configuration वाले Computer System पर ही Install किया जा सकता है, ऐसा नहीं है बल्कि हम किसी भी सामान्य से Computer System पर भी इन **Web Server Softwares** को Install कर सकते हैं और जिस Computer System पर किसी Web Server Software को Install करते हैं, उसी Computer को **Web Server** कहा जा सकता है, फिर भले ही वह Computer सामान्य सा Pentium1 Processor वाला Computer ही क्यों न हो।

Web Server Software ही वह Software होता है, जो User द्वारा आने वाली Request को Accept करता है और User को उसका वांछित परिणाम Web Page के रूप में फिर से Serve करता है या फिर से भेजता है, जिसे User का Web Browser Receive करके User के सामने Render करता है।

इस तरह से अब यदि हम सारांश के रूप में समझें, तो **User + User Device + Web Browser** का Combination **Web Client** को Represent करता है, जबकि एक Web Developer के लिए Coding के लिहाज से केवल **Web Browser** महत्वपूर्ण होता है।

जबकि **Host Computer System + Web Server + Web Developer + Web Site Owner** का Combination **Web Server** कहलाता है, लेकिन एक Web Developer के लिए Coding के लिहाज से केवल **Web Server** को महत्वपूर्ण होता है, हालांकि हमें Web Browser की तुलना में Web Server के साथ बहुत कम काम करना होता है।

वर्तमान समय में मूल रूप से **IIS** व **Apache** नाम के दो **Web Servers** सबसे ज्यादा उपयोग में लिए जाते हैं।

IIS, 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 Perl, PHP आदि को ज्यादा बेहतर तरीके से Access करता है।

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 **MSSQL, MySql** आदि 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 लेकिन Scripting Language द्वारा आने वाले Data को DBMS Software में Store व Manage कैसे करना है, इस बात का निर्णय पूरी तरह से DBMS Software लेता है।

चूंकि Internet पूरी तरह से **Client-Server Architecture Technology** पर आधारित है, जिसके हमेंशा दो और थोड़ा और गहराई में जाने पर तीन हिस्से होते हैं, जिन्हें **2-Tier** व **3-Tier Architecture** कहा जाता है।

2-Tier Architecture में मूल रूप से **Client** व **Server** होते हैं, जिनके बारे में आप उपरोक्त Discussion द्वारा अच्छी तरह से समझ गए होंगे। जबकि **3-Tier Architecture** में Client व Server के अलावा एक **Business Tier** या **Logic Tier** भी होता है, जो कि विभिन्न प्रकार के Business Logics को Handle करता है। सामान्यतः ये तीसरा Tier, DBMS Software का हिस्सा होता है और Client Tier व Server Tier के बीच में अपना Role Play करता है।

चलिए, अब हम उपरोक्त Discussion को सारांश के रूप में एक बार Revise करते हुए समझते हैं कि क्या और कैसे होता है?

- 1 सबसे पहले User किसी Web Site का Address Web Browser के Address Bar में Place करके Enter Key Press करता है अथवा किसी Web Site के HTML Page पर दिखाई देने वाले Link पर Click करता है।
- 2 Web Browser User द्वारा Specified URL को Web Server पर भेजता है और उस Resource के लिए Web Server से Request करता है।
- 3 Web Server, Web Browser से आने वाले Request को Identify करता है और देखता है कि वह Resource कोई Static Web Page है या Dynamic Web Page है।
- 4 यदि Requested Resource **Static Web Page** होता है, तो Web Server उस Resource को Specified URL के अनुसार अपने Web Host पर Search करता है और Resource मिल जाने की स्थिति में वह Resource फिर से Web Browser को भेजते हुए Request को पूरा करता है।

जबकि Resource के Host पर Available न होने की स्थिति में एक Error Return करता है, जो इस बात को Specify करता है कि Specified Resource Host पर Available नहीं है।

- 5 यदि Requested Resource **Dynamic Web Page** होता है, तो Web Server उस Resource को Specified URL के अनुसार अपनी Scripting Language पर Parsing के लिए भेजता है।

यदि Windows का Web Host हो, तो Scripting Language के रूप में सामान्यतः ASP या ASP.NET Scripting Language आने वाली Request को Process करता है जबकि यदि Linux का Web Host हो, तो PHP, Perl जैसी Scripting Languages आने वाली Request की Processing करते हैं।

यदि Data को Store या Access करने के लिए Server Side में किसी DBMS Software को Use किया गया हो, तो Scripting Languages अपने Associated DBMS Software पर Data को Store या Access करने के लिए DBMS Software से Request करता है।

DBMS Software, Scripting Language द्वारा आने वाली Request को Fulfill करने के लिए अपने Business Tier में Specify किए गए **Business Rules** व **IO Rules** को Data पर Apply करता है और Business Rules व IO Rules के पूरी तरह से Satisfy होने की स्थिति में Scripting Language को Requested Data Return करता है अथवा आने वाले Processed Data को DBMS Software में Store करके Scripting Language को इस बात की जानकारी देता है कि उसने अपना काम पूरा कर दिया है।

जबकि यदि DBMS Software पर आने वाली Request से DBMS Software के Business Tier पर Specified किसी तरह का Business या IO Rules का Violation मिलता है, तो DBMS Software, Scripting Language को एक Appropriate Error Message Return करता है।

दोनों ही स्थितियों में Scripting Language को DBMS Software से कोई Output मिलता है, जिसके आधार पर वह अपना Resultant Web Page Reformat करता है और Web Server को इस बात का Instruction देता है कि वह Web Browser द्वारा Requested Resource को Serve कर सकता है।

- 6 Scripting Language से Formatted Resultant Web Page तैयार हो जाने की जानकारी मिल जाने के बाद Web Server उस Resultant Web Page को फिर से Web Browser को Return कर देता है।
- 7 Web Browser, Web Server से आने वाले Resultant Web Page को फिर से Render कर देता है। सबसे पहले Web Browser आने वाले Web Page के HTML Codes के अनुसार Web Page को Structure करता है। फिर उस पर विभिन्न Inline व Outline CSS Rules Apply करता है और अन्त में JavaScript के Behaviors को Apply करके User के सामने Interactive Web Page Render कर देता है।

इस प्रकार से User द्वारा एक Request पूरी होने में उपरोक्त सभी Steps Follow होते हैं। चूंकि Static Web Page की Request पूरी होने में Dynamic Web Page की तुलना में कम Steps Follow होते हैं, इसलिए Static Site की Speed, Dynamic Site की Speed से हमेशा कम होती है।

Web Development Sequence and Used Technologies

उपरोक्त Discussion से एक और बात सामने आती है कि एक Dynamic Web Site कम से कम 6 Techniques के Mixture से बनती है और यदि हम थोड़ा और गहराई में जाएं, और Web Site को थोड़ा सा भी Dynamic व Interactive बनाना चाहें, तो और भी बहुत सारी Technologies अपना Role Play करती हैं। चलिए, थोड़ा सा इस विषय में भी जान लेते हैं।

जब किसी Web Site को बनाना होता है, तो सबसे पहले उस Web Site के Look को तय किया जाता है कि आखिर वह Web Site बनने के बाद अन्त में User को कैसी दिखाई देगी। चूंकि एक Web Site को अच्छा दिखाने के लिए कई तरह के **Colors, Graphics व Fonts** आदि Use किए जाते हैं, इसलिए सबसे पहले जरूरत पड़ती है एक **Graphics Designer** की।

Graphics Designer सबसे पहले Businessman की जरूरतों को समझते हुए किसी भी Web Site का एक Drawing Create करता है। ये Drawing Create करने के लिए वह विभिन्न प्रकार के Graphics Tools जैसे कि **Photoshop, CorelDraw, Illustrator, Fireworks, GIMP** आदि Use करता है और Web Site का **Logo** व विभिन्न प्रकार के अन्य **Graphics** के साथ Web Site का **Layout** भी Design करता है और Web Site Owner यानी उस Businessman को दिखाता है, जो Web Site बनवाना चाहता है।

जब Site Owner अपनी Web Site के **Design, Layout व Graphics** से पूरी तरह से सन्तुष्ट हो जाता है, तब वह Graphics Designer अपने Graphics को Web Site के Front End Designer को देता है।

यदि Graphics Designer को अपने काम का अच्छा ज्ञान हो, तो सामान्यतः वह Front Designer को अपने Graphics के साथ उस Graphics के **Slice** Create करके भी देता है, जिससे Front Designer को इस बात का पता चल जाता है कि किस Slice को कहां Use करना है।

Graphics Designer का काम यहां समाप्त हो जाता है। अब शुरू होता है Front End Designer का काम। Front End Designer Web Site के Layout के **Drawing** के आधार पर HTML Coding को Use करते हुए Web Site का **Structure** Create करता है और इस Structure के साथ CSS को Use करते हुए Web Site की Styling करता है।

Web Site बिल्कुल वैसी ही दिखाई दे, जैसा Graphics Designer ने बनाया है, इसके लिए Front End Designer, Graphics Designer द्वारा दिए गए Graphics Slices को अपने CSS में जरूरत के अनुसार Use करता है और बिल्कुल वही Look **HTML + CSS** द्वारा Generate करता है, जैसा Graphics Designer ने बनाया है।

कई बार Web Sites में Businessman की इच्छानुसार Animation जैसी सुविधा प्राप्त करनी होती है। इस स्थिति सामान्यतः Flash Designer की जरूरत पड़ती है, क्योंकि सामान्यतः Animation का काम Flash Designers ही करते हैं। वैसे अब नई Technology के अनुसार HTML5 में JavaScript API द्वारा SVG Technology का प्रयोग करके भी Graphics व Animation Develop किया जाने लगा है।

यदि Front End Designer HTML व CSS के अलावा JavaScript भी जानता हो, तो वह Web Site के Front End को और बेहतर व Interactive बनाने के लिए JavaScript के Codes को Use करता है। सामान्यतः JavaScript के स्थान पर jQuery, Dojo, YUI जैसे

किसी Framework को भी Use कर सकता है, जो कि Front End को आसानी से Interactive बनाने के लिए Develop किए गए Frameworks हैं।

जब बात JavaScript की आती है, तब Front End में कई और Technologies जुड़ जाती हैं। वर्तमान समय में ऐसी Web Sites बहुत ज्यादा बनाई जाने लगी हैं, जिसमें विभिन्न प्रकार की जरूरतों को पूरा करने के लिए Web Site बार-बार Web Browser में Reload नहीं होता बल्कि Web Browser समान Web Page में ही अलग-अलग Contents को Display करता रहता है। इस जरूरत को पूरा करने के लिए सामान्यतः AJAX तकनीक का प्रयोग किया जाता है।

AJAX एक ऐसी तकनीक है, जो कि JavaScript व XML का मिश्रण है, जो कि User की जानकारी के बिना Current Web Page में ही Server से नए Content की Request करता है और आने वाले नए Content को बिना Web Page को फिर से Web Browser में Reload किए हुए User के सामने Render कर देता है।

इसलिए जब हम **AJAX (Asynchronous JavaScript and XML)** की बात करते हैं, तब हमें XML को भी थोड़ा बहुत समझना जरूरी हो जाता है, अन्यथा हम AJAX Technology को बेहतर तरीके से Use नहीं कर सकते, जो कि Current Market की Requirement है।

Front End को तेजी से Develop करना किसी भी Web Development Company की मूल जरूरत होता है ताकि जल्दी से जल्दी वह अपने Client से अपनी Develop की गई Web Site का पैसा वसूल कर सके और Fast Front End Development के लिए जरूरी है कि Front End Developer JavaScript द्वारा नहीं बल्कि किसी JavaScript Framework को Use करके Front End को Interactive बनाए।

सामान्यतः यदि बहुत ही ज्यादा जरूरत न हो, तो किसी भी Company में अब Directly JavaScript के Codes Create नहीं किए जाते, बल्कि JavaScript के स्थान पर इसके Frameworks जैसे कि **jQuery, MooTools, Dojo, YUI, Prototypes** आदि को Use किया जाता है, क्योंकि ये Frameworks जिस काम को **1 Line** के Codes से पूरा कर देते हैं, उन्हीं कामों को यदि Pure JavaScript द्वारा पूरा किया जाए तो कम से कम 10 से 20 Lines का Code लिखना पड़ेगा साथ ही अलग-अलग Web Browsers के लिए अलग-अलग JavaScript Codes लिखने की जरूरत भी पड़ सकती है, जबकि ये Frameworks Cross Browser Format में Develop किए गए हैं। यानी आपको अलग-अलग Web Browsers के लिए अलग-अलग Framework Codes लिखने की जरूरत नहीं रहती है।

जब हम JavaScript Frameworks की बात करते हैं, तब हमें JavaScript के Object Oriented Concept पर ध्यान देना पड़ता है और JavaScript के Object Oriented Concept में विभिन्न प्रकार के Data को जिस Format में Use व Access किया जाता है, वह एक Special Format है, जिसे **JSON (JavaScript Object Notation)** कहा जाता है और एक Front End Designer को इसे भी समझने की जरूरत पड़ती है।

JavaScript का प्रयोग केवल Web Page को Interactive बनाने के लिए ही नहीं किया जाता, बल्कि इसका विकास तो मूल रूप से Client Side Validation के लिए किया गया था और आज भी JavaScript इस काम को बखूबी करता है। लेकिन जब Client Side Validation की बात आती है, तब बात आती है HTML Forms की और HTML Forms यानी Data, User Input करेगा और चूंकि Data, User Input करेगा, तो हम User द्वारा Input किए जाने वाले Data पर कभी विश्वास नहीं कर सकते।

इसलिए हमें Client Side में ही ये तय करना पड़ता है कि User, Form के किसी Field में ऐसी कोई Information न Fill करे, जो कि गलत हो या हमारी Web Site के लिए हानिकारक हो सकती हो। फलस्वरूप हमें Client Side के Form के Fields में Entered Text को Validation के लिए Check करने की जरूरत पड़ती है और Client Side में ये काम **Regular Expressions** द्वारा किया जाता है।

ये तो हुई Client Side की बात, अब चलते हैं Server Side में। जब हम Server Side की बात करते हैं तब **XML, JSON** व **Regular Expression** फिर से काम आते हैं, लेकिन Client Side की तुलना में Server Side में इनकी ज्यादा जरूरत पड़ती है। क्योंकि Client Side में जो Data, Server से भेजा जाता है, ज्यादातर परिस्थितियों में वह Data **XML** या **JSON** Format में ही भेजा जाता है, ताकि Client Side में JavaScript उस Data को User के Web Browser में जरूरत के अनुसार Render कर सके। यानी हम XML व JSON को छोड़ नहीं सकते। हमें इनके बारे में भी जरूरत के अनुसार थोड़ा बहुत तो जानना ही होगा।

जब हम Server Side Scripting की बात करते हैं, तब हमें थोड़ा-बहुत **Apache** या **IIS** Web Servers के बारे में भी जानने की जरूरत पड़ती है, ताकि हम Special Types की जरूरतों को Web Server के माध्यम से भी पूरा कर सकें। सामान्यतः Web Server के साथ भी हमें **Regular Expressions** को Use करने की जरूरत पड़ती है।

Server Side Scripting के विषय में बात करें, तो बिना DBMS Software के कोई भी Dynamic Web Site नहीं बनाई जा सकती। इसलिए हमें किसी न किसी DBMS Software को भी ठीक से समझना जरूरी हो जाता है।

लगभग सभी DBMS Softwares 80% Common होते हैं, लेकिन फिर भी यदि हम Microsoft Technology पर आधारित Web Site बना रहे हैं, तो हमें IIS, ASP या ASP.NET तथा MSSQL Server या MS-Access के बारे में जानने की जरूरत पड़ती है क्योंकि Microsoft Technology में इन्हीं Server Side Scripting Languages, Web Servers व DBMS Softwares को Use व Access करना होता है।

यदि हम **ASP.NET** की बात करें तो हमें Server Side Language के रूप में VB.NET या C#.NET को Use करना पड़ता है, क्योंकि ASP.NET में Scripting Language के रूप में इन्हीं में से किसी एक या दोनों को ज्यादा Use किया जाता है। हालांकि ये दोनों **Programming Languages** Windows Operating System के Desktop Applications बनाने के लिए भी उपयोगी होते हैं व वर्तमान समय में बहुत Use किए जाते हैं।

जबकि यदि हम Linux Web Host Use करते हैं, तो हमें Scripting Language के रूप में **PHP, Perl** जैसी Languages को Use करना पड़ता है जबकि DBMS Software के रूप में **MySql** को ज्यादा Use किया जाता है साथ ही हमें Apache Web Server को भी थोड़ा बहुत समझना जरूरी हो जाता है।

जब इतनी सारी Technologies की जरूरत एक Web Site बनाने के लिए पड़ती है, तो इतनी सारी Technologies को ठीक से Manage व Maintain करने के लिए भी एक Special Software की जरूरत पड़ती है, जिसे IDE (Integrated Development Environment) कहते हैं।

IDE के रूप में आपको **MS-Visual Studio, Eclipse, NetBeans, DreamWeaver** में से एक या एक से ज्यादा को सीखने की जरूरत पड सकती है, क्योंकि ज्यादातर **Companies** में इन्हीं में से एक या एक से अधिक **IDEs** में काम किया जाता है, ताकि **Development** को **Fast** व **Manageable** तरीके से किया जा सके।

तो क्या आप अन्दाजा लगा पाए कि कितनी **Technologies** की जरूरत पड सकती है एक **Web Site** बनाने के लिए, जबकि सभी प्रकार की जरूरतों को पूरा करने के लिए इनके अलावा भी कई अन्य **Technologies** हैं, जिन्हें सीखने की जरूरत पड सकती है। चलिए, देखते हैं:

- 1 Photoshop (Illustrator, Fireworks, CorelDraw, GIMP)
- 2 Adobe Flash
- 3 HTML (Hyper Text Markup Language)
- 4 CSS (Cascading Style Sheets)
- 5 JavaScript
- 6 JSON (JavaScript Object Notation)
- 7 XML (eXtensible Markup Language)
- 8 AJAX (Asynchronous JavaScript and XML)
- 9 Regular Expressions
- 10 Apache or IIS Web Server
- 11 PHP/Perl or ASP.NET (VB or C# or Both)
- 12 MySql or MSSQL Server
- 13 MS-Visual Studio, Eclipse, NetBeans, DreamWeaver

क्या आपको लगता है कि ये सभी **Technologies** आप स्वयं अकेले सीखें और फिर अपने स्तर पर स्वयं पूरी **Web Site** बनाएं। यदि आप ऐसा सोचते हैं, तो पहली बात तो ये है कि इतनी **Technologies** को अच्छी तरह से सीखने के लिए आपके लिए 5 साल भी कम पड़ेंगे और दूसरी बात ये है कि जब तक आप पहली **Technology** से आखिरी **Technology** तक सीखेंगे, तब तक पांचवी **Technology** तक इतने नए **Versions** आ जाएंगे, कि आपको फिर से पहली **Technology** को सीखना पड़ेगा और ये प्रक्रिया **Recursive** तरीके से पूरी जिन्दगी चल सकती है। यानी आप कभी भी सभी **Technologies** को स्वयं अकेले **Mastering Level** तक नहीं सीख सकते।

इसीलिए किसी भी **Company** में कई तरह के **Departments** होते हैं और हर **Department** में अपनी तरह का **Development** होता है। उदाहरण के लिए **Graphics Designer** का अपना काम होता है और उसे **Front End** या **Back End Coding** से कोई मतलब नहीं होता।

Front End Department को **Back End Department** के **Developers** व **Graphics Designer** से कोई मतलब नहीं होता।

इसी तरह से **Back End Designer** को **Front End** व **Graphics Designer** से कोई मतलब नहीं होता।

यहां तक कि **Back End Script Writer** को **Back End Database Designer** व **Analyst** से भी कोई मतलब नहीं होता।

यानी सरल तरीके से कहें, तो उपरोक्त सभी विषयों को मूल रूप से चार भागों में बांटा जा सकता है और चारों भागों के लोग केवल अपने काम को ही **Best** तरीके से पूरा करते हैं, कर सकते हैं:

Graphics Designer

Graphics Designer का पूरा ध्यान Graphics Develop करने पर होता है और एक Graphics Designer को Photoshop, Illustrator, Fireworks, CorelDraw, GIMP, Flash आदि Technologies को ही Best तरीके से सीखना होता है।

Front End Designer

Front End Designer का पूरा ध्यान Web Site का Front यानी Layout बनाने पर होता है और एक Front End Designer को HTML, CSS, JavaScript, JSON, XML, AJAX, Regular Expressions आदि Front End Designing से संबंधित Technologies को ही Best तरीके से सीखना होता है।

Back End Designer

Back End Designer का पूरा ध्यान Back End Technologies पर होता है और एक Back End Designer को JSON, XML, AJAX, Regular Expressions, Apache or IIS Web Server, PHP/Perl or ASP.NET (VB or C# or Both) को ही Best तरीके से सीखना होता है।

यहां भी यदि Microsoft Technology को महत्व दिया जा रहा है, तो PHP व Perl जैसी Languages को सीखना जरूरी नहीं है, जबकि Linux Technology को महत्व देने की स्थिति में ASP.NET, VB, C# को सीखना जरूरी नहीं है।

Database Designer

Database Designer का मुख्य काम Web Site Owner की जरूरत के अनुसार विभिन्न प्रकार के Data को Best तरीके से Database में Store करने, Access करने की सुविधा देने व Database को Maintain करने व Database की Performance को बनाए रखने से संबंधित होता है इसलिए एक Database Designer को केवल इन्हीं जरूरतों को पूरा करने से संबंधित Technologies को अच्छी तरह से सीखना होता है।

यदि एक Database Designer Windows Technology को Handle करता है, तो उसे केवल MS-Window, MSSQL Server या MS-Access के बारे में Best तरीके से जानना होता है जबकि Linux Technology को Use करने की स्थिति में उसे Linux तथा MySql जैसे Database Software को अच्छी तरह से समझना होता है।

अब सवाल ये है कि क्या हर Company में ये चारों हिस्से होते हैं और क्या हर Company में इतने सारे प्रकार के Developers होने जरूरी होते हैं? तो जवाब है, हां। लगभग हर Company में इतने प्रकार के Developers जरूर होते हैं।

तो अब दूसरा सवाल ये है कि क्या हम बिना इन विभिन्न प्रकार के Developers को Hire किए हुए छोटे स्तर पर अपना Web Development का काम शुरू नहीं कर सकते? तो इस सवाल का जवाब है हां और दूसरा जवाब है नहीं।

हमें इन सभी प्रकार के Developers की जरूरत जरूर होती है, तभी कोई Web Site ठीक से बन सकती है और लम्बे समय तक Maintain की जा सकती है, लेकिन इसका मतलब ये नहीं है कि ये सभी Developers Physically हमारे पास हों। मतलब?

मतलब ये है कि इतने प्रकार के High Profile Developers को Hire करना काफी महंगा काम हो सकता है, जबकि लगभग 80% Web Sites इतनी Typical नहीं होतीं, कि उनके लिए अलग से Database Designers व Back End Developers की जरूरत हो और इन लोगों के Replacement के रूप में हमें दूसरा Option मिलता है **Frameworks** का।

Frameworks ऐसे Software Packages होते हैं, जो बड़ी ही आसानी से विभिन्न प्रकार की Back End जरूरतों को Internally पूरा कर देते हैं, जिनको Develop करने के लिए हमें अलग से Designers Hire करने की जरूरत नहीं पड़ती। यानी एक Front End Designer बड़ी ही आसानी से इन Frameworks का प्रयोग करके Backend Requirements को Fulfill कर सकता है।

Frameworks दो प्रकार के होते हैं। पहला **Content Management System** कहलाता है जबकि दूसरा **Application Management System** कहलाता है।

Content Management System के रूप में WordPress, Drupal, Joomla को ज्यादा उपयोग में लिया जाता है, जबकि Application Management Framework के रूप में Symphony, CodeIgnitor, CakePHP, आदि को Use किया जाता है।

Server Side Back End Developer Requirement को तो एक Front End Designer विभिन्न प्रकार के Frameworks का प्रयोग करके पूरा कर सकता है, लेकिन Front End Designer के लिए तो फिर भी कई Technologies को सीखना जरूरी होगा। यदि आप ऐसा सोच रहे हैं, तो आप गलत सोच रहे हैं।

जिस तरह से Server Side जरूरतों को पूरा करने के लिए Frameworks हैं, उसी तरह से Client Side जरूरतों को पूरा करने के लिए भी Frameworks हैं। Client Side में मूल रूप से HTML व CSS ऐसी Technologies हैं, जो सभी Front End Designers को सीखनी ही चाहिए, अन्यथा वे Front End को ठीक से Control नहीं कर सकते। लेकिन जब बात JavaScript की आती है, तब JavaScript के कई Frameworks हैं, जिनका प्रयोग JavaScript के स्थान पर किया जा सकता है।

jQuery, MooTools, Dojo, YUI, Prototypes आदि विभिन्न प्रकार के JavaScript Frameworks के उदाहरण हैं, जिनमें से **jQuery** मुझे Personally बहुत पसन्द है क्योंकि इसे सीखना व Use करना बाकी सभी अन्य Frameworks की तुलना में आसान है। यदि आप CSS जानते हैं, तो समझ लीजिए कि आप बहुत ही आसानी से jQuery को उपयोग में ले सकते हैं और बहुत ज्यादा तेजी से अपनी Web Site की Interactivity व Validation Related जरूरतों को पूरा कर सकते हैं।

जब आप इन में से किसी Framework को Use करते हैं, तब भी यदि आप अन्य Technologies को ठीक से समझने के लिए सीखते हैं, तो अच्छा है लेकिन जरूरी नहीं है। उदाहरण के लिए यदि आप केवल jQuery को ठीक से समझ लेते हैं, तो आप आसानी से AJAX संबंधित Dynamic जरूरतों को 4 – 5 Line के Code द्वारा पूरा कर सकते हैं। आपको इसके लिए अलग से JavaScript व XML सीखने की जरूरत नहीं है।

XML, JSON, Regular Expression आदि को jQuery या अन्य Frameworks स्वयं Internally Handle करता है, इसलिए इन Technologies को ज्यादा गहराई से समझने की जरूरत नहीं रह जाती। Pure JavaScript की जरूरत लगभग समाप्त ही हो जाती है, जबकि इन Frameworks का प्रयोग करके आप Flash जैसा Animation भी प्राप्त कर सकते हैं।

जहां तक Graphics Designer की बात है, तो Internet पर हजारों ऐसी Web Sites हैं, जो Free Web Site Templates Provide करती हैं। किसी भी अच्छे से Template को Download करके बड़ी ही आसानी से अपनी जरूरत के अनुसार उसे Modify किया जा सकता

है। इसलिए यदि Graphics Designer के नजरिए से देखें, तो हमें अलग से किसी Graphics Designer की भी जरूरत Compulsory रूप से नहीं है।

यानी यदि अब हम ये जानना चाहें कि हमें कुल कितनी तकनीकों को एक Web Site बनाने के लिए जरूरी रूप से सीखना होगा, तो ये List अब काफी छोटी हो सकती है और ये List निम्नानुसार है:

- 1 HTML (Hyper Text Markup Language)
- 2 CSS (Cascading Style Sheets)
- 3 JavaScript Frameworks like **jQuery**, MooTools, Dojo, YUI, Prototypes, etc...
- 4 Server Side Framework like Symphony, WordPress, MODx, CodeIgnitor, etc...
- 5 PHP/Perl or ASP.NET (VB or C# or Both)
- 6 MS-Visual Studio, Eclipse, NetBeans, DreamWeaver IDE

उपरोक्त List को देखें तो ये List अब पहले की तुलना में आधी हो चुकी है। परिणामस्वरूप विभिन्न प्रकार के Professional Developers की जरूरत भी लगभग समाप्त हो चुकी है।

हालांकि Frameworks का प्रयोग करके हम बड़ी ही आसानी से कम समय में ज्यादा Development कर सकते हैं, लेकिन फिर भी **JavaScript** व **PHP** को जरूर अच्छी तरह से सीखना चाहिए। क्योंकि सभी Front Side Frameworks पूरी तरह से JavaScript पर आधारित होते हैं जबकि Linux Based लगभग ज्यादातर Back End Frameworks **PHP Based** होते हैं।

Window Based Web Servers के लिए हमें **VB.Net** या **C#.Net** को सीखना जरूरी होता है, क्योंकि Windows Based Frameworks हालांकि बहुत कम हैं, लेकिन जो भी हैं वे पूरी तरह से इन्हीं दोनों Languages पर आधारित हैं।

JavaScript को ठीक से समझा तो किसी भी Framework को बड़ी ही आसानी से उपयोग में लेना सीख सकते हैं जबकि PHP को समझ कर बड़ी ही आसानी से किसी भी Server Side Framework को तेज गति से सीख सकते हैं।

इन दोनों Languages को अच्छी तरह से सीखना इसलिए भी जरूरी है क्योंकि अलग-अलग Companies में अलग-अलग तरह की जरूरतों को पूरा करने के लिए Frameworks Use करने पड़ते हैं, जिनका Decision, Company Owner Project की जरूरत के आधार पर लेता है। इस स्थिति में किसी एक या दो Framework को सीख कर Company में Long Term Job की उम्मीद नहीं की जा सकती।

लेकिन यदि सभी Frameworks के आधार को सीख लिया जाए, तो Long Term Job की Guarantee होती है, क्योंकि उस स्थिति में हम बड़ी ही आसानी और बहुत ही तेज गति से उन Root Languages पर आधारित किसी भी Framework को सीख सकते हैं।

तो अब यदि हम मूल रूप से ये जानना चाहें कि किन Technologies को Compulsory रूप से सीखना जरूरी है, तो वे Technologies निम्नानुसार होंगी:

- 1 HTML (Hyper Text Markup Language)
- 2 CSS (Cascading Style Sheets)
- 3 JavaScript

4 PHP or ASP.NET

इस List में हमने किसी IDE को Specify नहीं किया है क्योंकि जब हम किसी भी Language में Coding करना सीखते हैं, तब इनमें से किसी भी IDE में काम करना शुरू कर सकते हैं और ये IDE Automatically धीरे-धीरे समझ में आ जाते हैं। यानी इन्हें अलग से सीखने की जरूरत नहीं होती, इसलिए हमने इन्हें हमारी List से हटा दिया है।

इस तरह से आपको मूलतः उपरोक्त 4 Technologies को ठीक से सीखना होता है, ताकि आप एक Web Developer बन सकें और जैसाकि आप जानते हैं कि हम इस पुस्तक में PHP सीखने वाले हैं क्योंकि इस पुस्तक को आपने PHP सीखने के लिए ही खरीदा है। हालांकि PHP पूरी तरह से HTML और मूल रूप से HTML के Forms से संबंधित है। इसलिए इस पुस्तक को ठीक से समझने के लिए आपको HTML का और विशेष रूप से HTML के **Forms Part** का अच्छा ज्ञान होना जरूरी है।

चूंकि JavaScript AJAX तकनीक का प्रयोग करते हुए PHP Pages की भी Request कर सकता है, इसलिए यदि आपको JavaScript का भी अच्छा ज्ञान हो, तो PHP को JavaScript की AJAX तकनीक के साथ Use करते हुए आप और भी बेहतर व ज्यादा Interactive Web Site बना सकते हैं, लेकिन JavaScript का ज्ञान होना Compulsory नहीं है।

PHP

HISTORY

&

INTRODUCTION

PHP – History and Introduction

PHP को **Rasmus Lerdorf** नाम के Canada निवासी Programmer ने विकसित किया था। वास्तव में **Rasmus** ने सबसे पहले Perl Scripts का एक समूह Create किया और उस Perl Scripts के समूह को “**Personal Home Page Tools**” (PHP Tools) नाम दिया था।

इस समूह को Rasmus ने अपने Personal Homepage को Maintain करने के लिए Develop किया था। ये Scripts Rasmus के Resume व Web Pages को Display व Maintain करने के लिए Develop किया था। जबकि Rasmus ने **PHP** की Public Announcing **8 June, 1995** को किया।

बाद में Rasmus ने इन Scripts को CGI Binaries के रूप में “**C**” **Programming Language** में लिखा, ताकि इन Scripts द्वारा वे HTML Forms व Database के साथ Communication कर सकें और इसी Scripts के समूह को “**Personal Home Page/Forms Interpreter**” या **PHP/FI** नाम दिया। इन Scripts का प्रयोग करके आसानी से Simple Dynamic Applications बनाए जा सकते थे। इसी PHP/FI के First Version को Rasmus ने 8 June 1995 को Publicly Announce किया था।

इस Release में वे सभी Basic Functionalities थीं, जो आज के PHP में हैं। इसमें Perl की तरह Variables थे तथा Form Handling की सुविधा व Embedded HTML की सुविधा थी। इसके Syntax पूरी तरह से Perl Language की तरह थे। PHP/FI के पहले दो Versions को Rasmus ने स्वयं ही Develop किया था, लेकिन तीसरे Version को Develop करने के लिए पूरी एक Team ने काम किया था। फलस्वरूप इस Scripting Language का नाम बदल कर “**Hypertext Pre-Processor**” रखा गया जिसका Short Form **PHP** है।

PHP के पांचवे Version तक PHP बिल्कुल भी Stable नहीं था, लेकिन पांचवे Version के आने के बाद आज सबसे ज्यादा Web Sites Based हैं और PHP के 6th Version के Market में Launch होने की तैयारी चल रही है। लेकिन वास्तव में **PHP** है क्या? चलिए, जानने की कोशिश करते हैं।

Types of Programming Languages

Programming Languages मूलतः दो प्रकार की होती हैं:

Compiled Language Interpreted Language

Compiled Languages ऐसी Programming Languages होती हैं, जिनके Programs जिस Computer Architecture (Intel, AMD Athelon, Solaris, Single Core, Dual Core Quad Core, x86, x64, ...etc) व Operating System (Linux, Unix, Windows, MacOS, Wrap, OS/2, etc...) के लिए Develop किए जाते हैं, उन्हीं Computer Architecture के अनुसार पूरी तरह से Binary Codes में Convert हो जाते हैं।

यानी एक बार किसी Program को Compile कर देने के बाद उस Program के Source Codes की जरूरत नहीं रहती है क्योंकि उस Program के Codes पूरी तरह से उस Computer Architecture व Operating System के आधार पर Binary Form या

Machine Codes में Convert हो जाते हैं, जिन्हें बिना Source Codes के बार-बार Execute किया जा सकता है।

ये Compiled Codes पूरी तरह से Machine Dependent होते हैं, इसलिए जिस Architecture के लिए किसी Program को Compile किया जाता है, उस Architecture के अलावा किसी भी अन्य Architecture पर वह Compiled Program Run नहीं होता।

यदि हम किसी अन्य Platform पर उस Program को Execute करना चाहें, तो हमें उस नए Computer Architecture व Operating System के लिए उस Program के Source Codes को फिर से Compile करना पड़ता है।

चूंकि Compiled Codes पूरी तरह से Machine Dependent होते हैं और पूरी तरह से Native Machine के अनुसार Converted होते हैं, इसलिए Compiled Programs के Execute व Run होने की Speed तेज होती है, क्योंकि इन Programs को बार-बार Machine Codes में Convert होने की जरूरत नहीं होती है।

जबकि दूसरी प्रकार की Programming Languages को Interpreted या Interpreter Based Programming Language कहा जाता है। JavaScript, PHP, Perl, आदि Interpreter Based Programming Languages हैं। इस प्रकार की Programming Languages में बने Programs को जब भी Execute किया जाता, इन Programs के Source Codes हर बार Current Computer Architecture व Operating System के अनुसार Machine Codes में Convert होते हैं और Program Execution के बाद समाप्त हो जाते हैं।

यानी यदि किसी Interpreter Based Program को Run करना हो, तो हमें हर बार इन Programs के Source Codes की जरूरत पड़ती है।

चूंकि Interpreter Based Programming Languages को हर बार Machine Codes में Convert होना पड़ता है, इसलिए इन Programming Languages की Speed Compiled Programs की Speed की तुलना में कुछ कम होती है।

लेकिन Interpreter Based Programming Languages का एक फायदा भी है और वो फायदा ये है कि इन Programming Languages में बने Programs किसी भी Platform या Computer Architecture पर निर्भर नहीं होते इसलिए किसी भी प्रकार के Computer Architecture या Operating System पर आसानी से Run हो जाते हैं, क्योंकि ये हर बार Interpret होते हैं यानी इनका Interpreter इन Programs को हर बार Current Computer Architecture व Operating System के अनुसार Native Machine Codes में Convert करता है।

Interpreter Based Programming Languages को ही **Scripting Language** भी कहा जाता है और PHP एक Scripting Language है, क्योंकि जब भी हम किसी PHP Program को Execute करते हैं, वह PHP Program अपने Interpreter पर Parse होता है और अपनी Coding के अनुसार Appropriate Results Generate करता है।

PHP का प्रयोग केवल Web Pages को Dynamic बनाने के लिए ही किया जा सकता है, ऐसा नहीं है। हम PHP का प्रयोग कई अन्य प्रकार की जरूरतों को पूरा करने के लिए भी कर सकते हैं। PHP के साथ GTK का प्रयोग करके हम Platform Independent Desktop

Application बना सकते हैं, जो कि किसी भी Platform या Computer Architecture पर बिना किसी परेशानी के Execute हो सकता है।

PHP का प्रयोग करके हम Adobe Flash व PDF Files को Use कर सकते हैं अथवा Programmatically, नई PDF File Create कर सकते हैं। POSIX व Perl Based Regular Expression Libraries का प्रयोग करके हम Complex String Operations कर सकते हैं।

PHP के साथ Command Line Scripts को Run कर सकते हैं और विभिन्न प्रकार के System Administrative कामों को Automatically पूरा कर सकते हैं।

PHP केवल Linux पर ही Run होता हो, ऐसा नहीं है। बल्कि PHP किसी भी प्रकार के Operating System व Architecture पर Execute होता है और विभिन्न प्रकार के Web Servers के साथ मिलकर काम कर सकता है। यानी हम PHP का प्रयोग केवल Apache Web Server के साथ नहीं बल्कि IIS, Netscape/iPlanet आदि के साथ भी कर सकते हैं।

साथ ही हम PHP द्वारा HTML Documents के साथ विभिन्न अन्य प्रकार के Formats जैसे कि PDF, GIF, JPG, PMG, Flash Movies, Text Files, XML Files आदि के साथ भी PHP का प्रयोग करके अपनी विभिन्न प्रकार की जरूरतों को पूरा कर सकते हैं। PHP में इन सभी प्रकार के Formats के साथ काम करने के लिए Built-In Support है और हमें अलग से किसी Library को Include करने की जरूरत नहीं है।

PHP MySQL ही नहीं बल्कि विभिन्न प्रकार के DBMS Softwares जैसे कि PostgreSQL, Oracle, Sybase व ODBC Compliant Database के साथ आसानी से Integrate हो जाता है। यानी PHP का प्रयोग करते हुए हम इन में से किसी भी Database को अपने Data को Store करने के लिए Use कर सकते हैं। हालांकि MySQL PHP के साथ Best Matching करता है और सबसे ज्यादा उपयोग में लिया जाता है।

Environment Setup

जब हम PHP Programming शुरू करना चाहते हैं, तो सबसे पहले हमें दो Basic Softwares की जरूरत होती है और ये दोनों ही Softwares Free हैं। ये Softwares निम्नानुसार हैं:

- 1 **Text Editor** (Notepad++) or IDE (**Aptana Studio 3**)
- 2 **Web Server** (**WAMP, XAMPP**)

चूंकि, 90% से ज्यादा लोग Windows Use करते हैं, इसलिए हम यहां केवल Windows के बारे में ही बात करेंगे। Windows Operating System पर WAMP या XAMPP Web Server Software को Install किया जा सकता है। ये Web Server Software हमारे Local Computer पर Install करने के बाद हमारा Local Computer एक Web Server Computer बन जाता है।

Notepad++ को <http://notepad-plus-plus.org/> Website से Download किया जा सकता है, जबकि **XAMPP** Web Server को <http://www.apachefriends.org/> से तथा **WAMP** को <http://www.wampserver.com/> से Download किया जा सकता है। इनके अलावा हम <http://www.apтана.com/> से “**Aptana Studio 3**” Download कर सकते हैं।

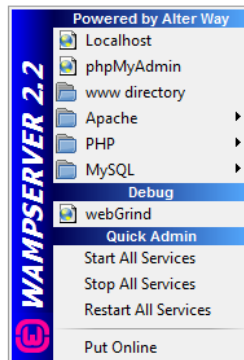
ये एक Eclipse आधारित Development IDE है, जिसका प्रयोग Fast Development के लिए किया जा सकता है।

यदि आप चाहें, तो केवल WAMP Download कर सकते हैं, क्योंकि PHP Program बनाने के लिए हमें केवल एक Text Editor व Web Server की जरूरत होती है। IDE का प्रयोग हम केवल हमारी सुविधा के लिए कर रहे हैं। WAMP Server Install करने के बाद Task Bar में हमें निम्न चित्रानुसार एक Icon दिखाई देता है।



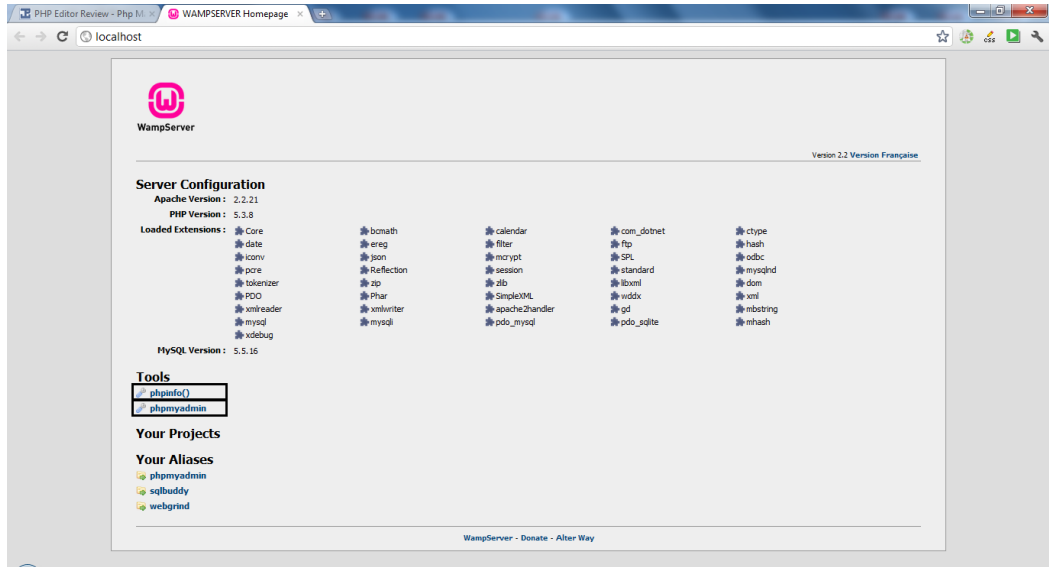
जब Icon Red Color में दिखाई देता है, तो इस बात को Indicate करता है कि Web Server बन्द है। Green Color का दिखाई देना इस बात का Indication है कि Web Server Online है यानी चालू है और Offline स्थिति में Icon Orange Color का दिखाई देता है। यदि Icon Orange Color का दिखाई दे रहा है, तो ये Web Server के Working Condition में न होने की स्थिति को Represent करता है।

इस Icon को Click करने पर हमारे सामने निम्नानुसार एक Popup Window Display होता है:



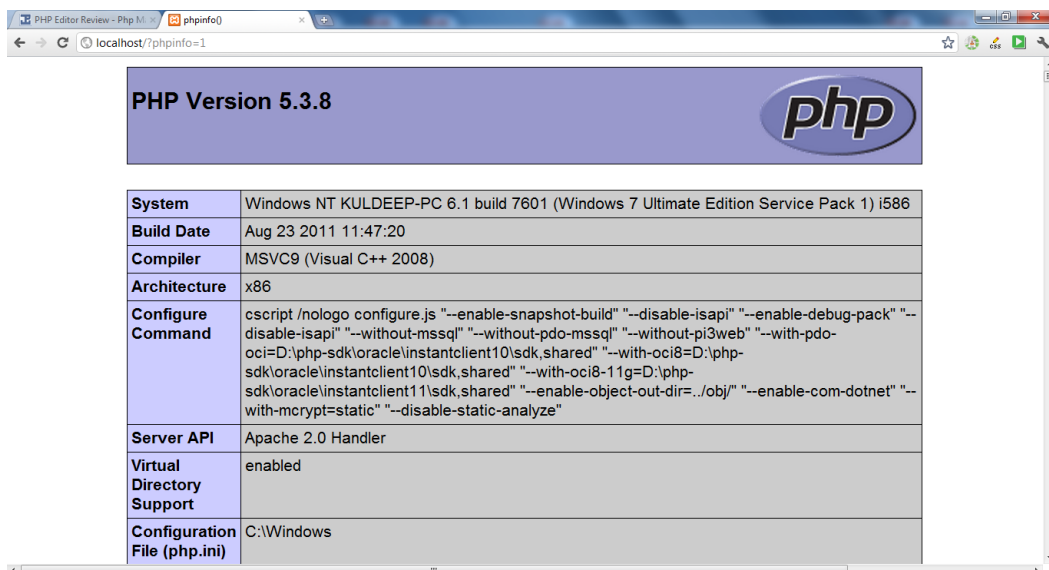
दिखाई देने वाले विभिन्न Links को Click करके हम इसके विभिन्न Features को देख सकते हैं व अपनी सुविधानुसार Web Server को Start, Stop या Restart कर सकते हैं अथवा किसी Specific Feature को On या Off कर सकते हैं।

Web Server ठीक से काम कर रहा है या नहीं, इस बात को Confirm करने का एक तरीका ये है कि हम उपरोक्त Popup Box में दिखाई दे रहे Localhost Option को Click करें। जैसे ही हम इसे Click करते हैं, हमारे सामने निम्नानुसार Web Page Display होता है, जिसमें Installed WampServer व उससे Related विभिन्न Featurs, Softwares व उनके Versions दिखाई देते हैं।



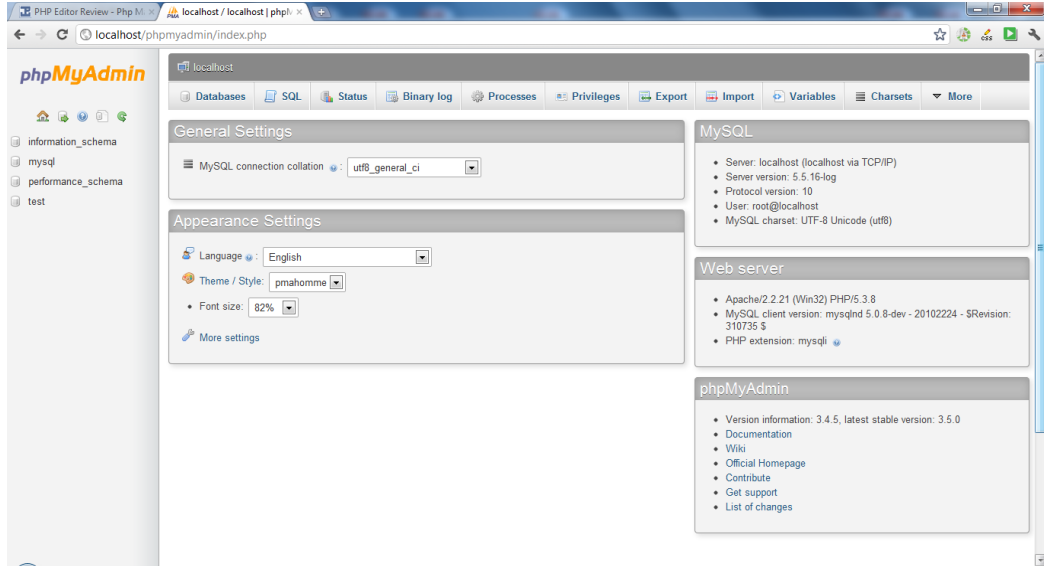
इस Web Page का दिखाई देना, इस बात का Indication है कि Web Server ठीक से काम कर रहा है। ये Web Page हमें और भी कई प्रकार की जानकारियां देता है, जिनमें सबसे महत्वपूर्ण जानकारी उन Extensions के बारे में देता है, जो WAMP Server को Install करते ही Automatically Load हो जाती हैं। उपरोक्त Web Page में देखें तो *Core, date, PDO, json, mysql, ftp, gd, dom, xml* आदि कई ऐसे जरूरी Extensions हैं, जो WAMP Server के साथ Automatically Install व Load हो जाते हैं।

अब हमें पता करना होता है कि PHP ठीक से काम कर रहा है या नहीं और इस बात का पता लगाने के लिए हमें केवल इस Web Page पर दिखाई देने वाले Tools **phpinfo()** Link को Click करना होता है। इसे Click करते ही हमारे सामने निम्नानुसार Web Page Open होना चाहिए:



यदि ये Web Page Open होता है, तो इसका मतलब है कि Apache Web Server के साथ-साथ PHP भी ठीक तरह से काम कर रहा है। अन्त में हमें ये पता करना होता है कि MySQL ठीक से काम कर रहा है या नहीं और इस बात का पता लगाने के लिए हमें पिछले

Web Page पर दिखाई देने वाले **phpmyadmin** नाम के Link को Click करना होता है, जिसे Click करते ही हमारे सामने निम्नानुसार Web Page Display होना चाहिए:

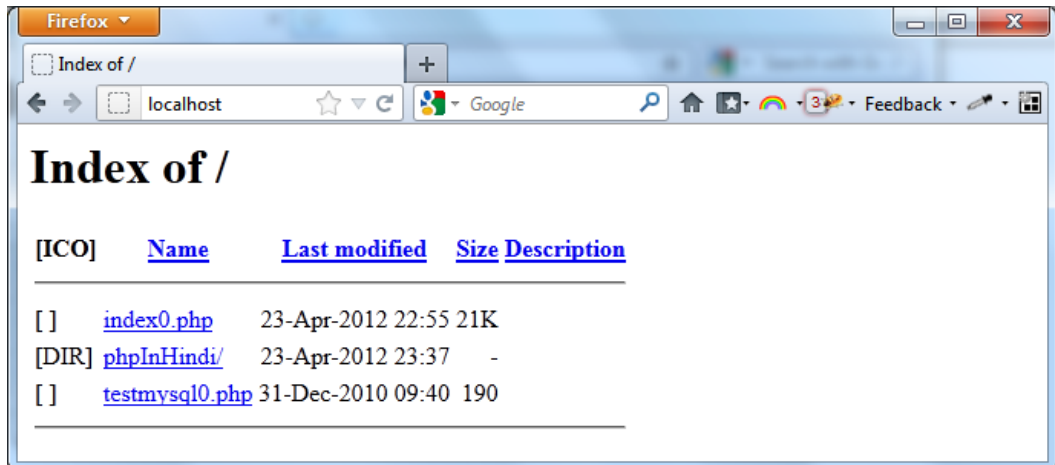


इस Web Page का दिखाई देना इस बात का Indication है कि हमारा Web Server Apache, Scripting Language PHP व Database MySQL तीनों ठीक तरह से काम कर रहे हैं और हम आगे बढ़ सकते हैं तथा Development का काम शुरू कर सकते हैं।

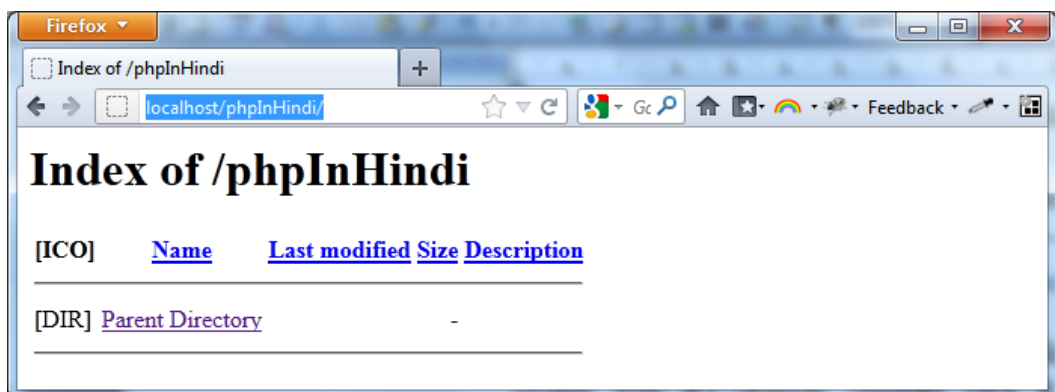
हम हमारे इस Local Web Server पर जो भी Web Pages या Script Files Create करते हैं, उन्हें हमें इस Web Site के एक Special Folder में ही Store करना होता है, तभी हम उन Files को Web Browser के Address Bar में <http://localhost> द्वारा Access कर सकते हैं। WAMP Server में इस Special Folder का नाम **www** होता है, जबकि XAMPP में इस Folder का नाम **htdocs** होता है।

इस Directory तक पहुंचने के लिए हमें WAMP के Icon पर Click करने पर दिखाई देने वाले Popup Menu के “**www directory**” Option को Click करना होता है और हम सीधे ही उस Directory में पहुंच जाते हैं, जहां हमें हमारी Local Web Server Files को Store करना होता है। यदि WAMP Installation के समय Default Path को Change न किया गया हो, तो सामान्यतः ये Directory “**C:\wamp\www**” Path में होती है।

इस Directory में सामान्यतः *index.php* व *testmysql.php* नाम की दो Files पहले से ही होती हैं, जिनकी वजह से हमें Web Browser में उपरोक्त सभी Screens दिखाई देते हैं। इन Files को सामान्यतः Rename कर देना चाहिए, ताकि ये Files Web Browser में Directly Run न हों। जब हम इन Files को Rename कर देते हैं और Web Browser के Address Bar में <http://localhost> Type करते हैं, तब हमें हमारा Local Web Server यानी **www** Folder व उसकी विभिन्न Files निम्नानुसार दिखाई देती हैं:



यदि हम **www** Folder में कोई नया Folder Create करते हैं, तो वह Folder भी हमें यहां दिखाई देता है। जैसाकि उपरोक्त चित्र में **phpInHindi** नाम का एक नया Folder Create किया गया है, जो कि दिखाई दे रहा है। यदि हम इस Folder पर Click करें, तो Address Bar में भी निम्नानुसार परिवर्तन होता है और हम इस phpInHindi नाम के Folder में पहुंच जाते हैं:

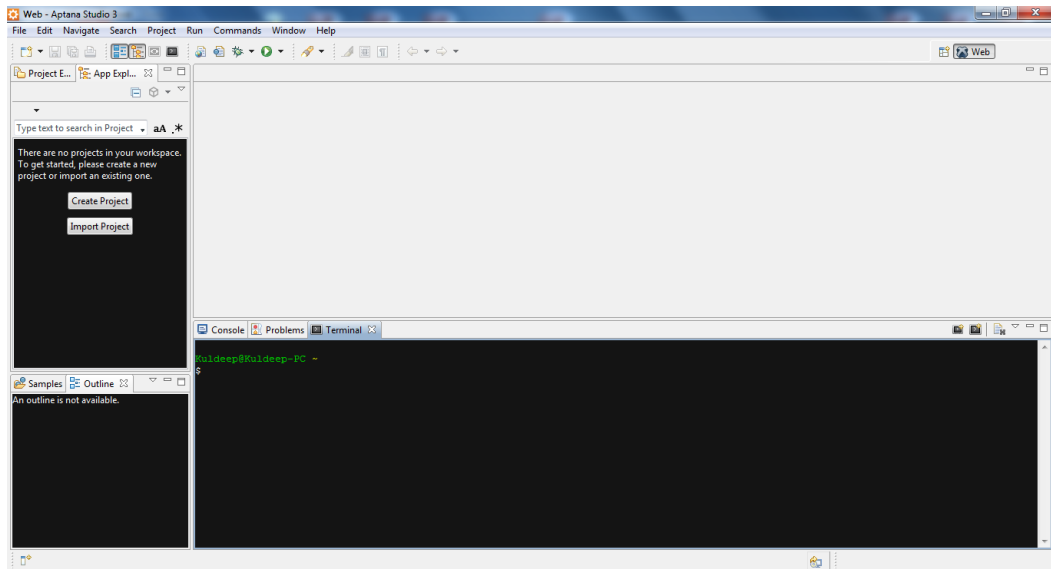


उपरोक्त चित्र के Address Bar में हम अपने Newly Created Folder **phpInHindi** के नाम को Absolute URL "<http://localhost/phpInHindi/>" के रूप में देख सकते हैं।

Web Server Setup करने के बाद हमें "**Aptana Studio 3**" को Install करना होता है, जो कि Eclipse IDE है और इसे मुख्य रूप से Web Development की जरूरतों को पूरा करने के लिए Modified किया गया है।



Aptana को Install करने के बाद जब हम इसे Open करते हैं, तो ये हमें निम्नानुसार दिखाई देता है:

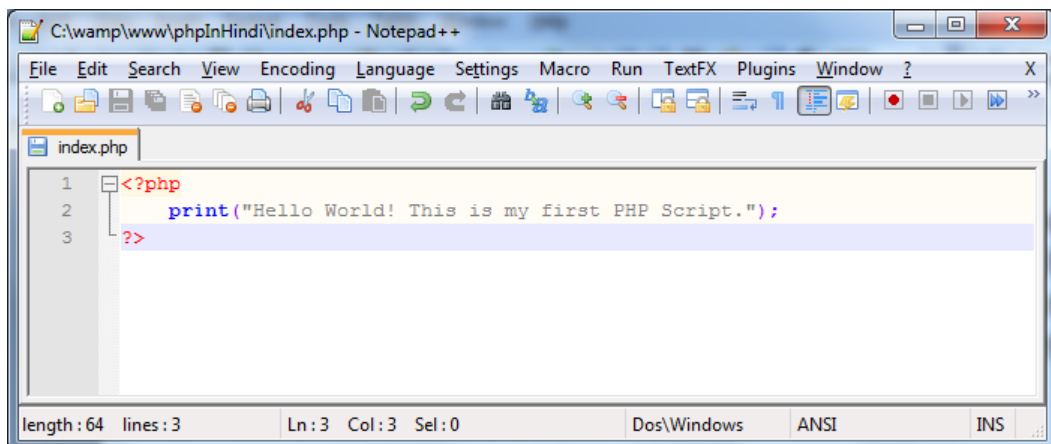


Aptana का प्रयोग करके हम पूरा PHP Application Develop व Maintain कर सकते हैं और चूंकि हमें Eclipse, NetBeans जैसे कुछ बहुत ज्यादा Use किए जाने वाले IDEs को भी Use करना सीखना चाहिए, इसलिए हम Eclipse को अपने PHP Learning Course के दौरान Use करेंगे।

वैसे PHP सीखने के लिए हमें केवल एक Web Server और एक Text Editor की ही जरूरत होती है, इसलिए पहले हम ये जानेंगे कि किस तरह से हम मात्र एक Text Editor का प्रयोग करके किसी PHP Program को Create व Interpret कर सकते हैं, फिर हम ये जानेंगे कि किस तरह से Aptana Studio हमारे काम को सरल बनाने में उपयोगी साबित हो सकता है। तो चलिए, अब हम हमारा सबसे पहला PHP Program बनाते हैं और उसे Interpret करना सीखते हैं।

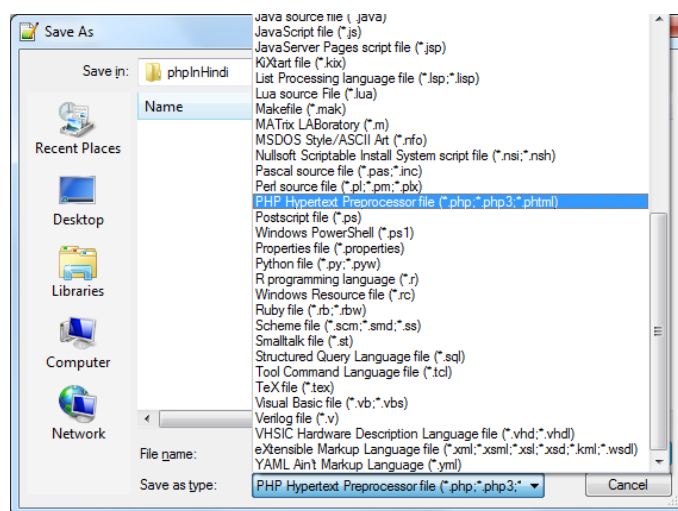
Creating First PHP Script

चूंकि, फिलहाल हम Aptana को Use नहीं कर रहे हैं, इसलिए हम Notepad++ Text Editor Open करेंगे और उसमें निम्नानुसार पहला PHP Script Code लिखेंगे:

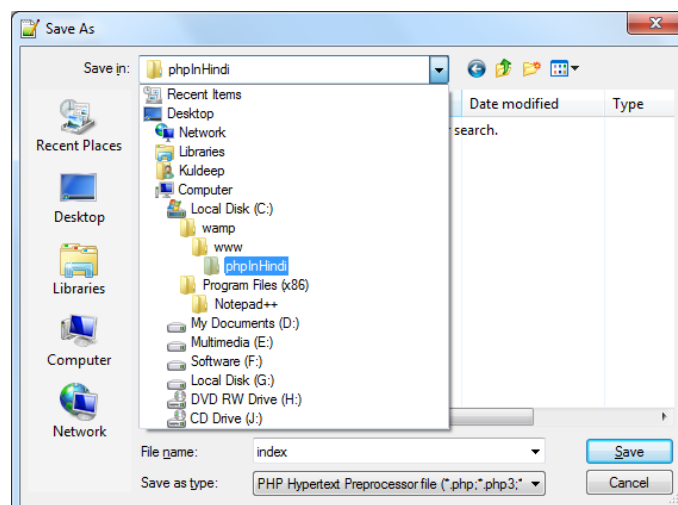


अब हमें इस File को Save करना है। चूंकि हम **PHP Script File Create** कर रहे हैं, इसलिए हमें इस File को **.php Extension** के साथ Save करना होगा, साथ ही इस File का नाम हम **index.php** रखेंगे, क्योंकि **“index”** नाम किसी भी Web Document File का Default नाम होता है, जिसे Web Browser के Address Bar में Specify न किया जाए, तब भी Web Browser **index** नाम की File को Directly Interpret या Render कर देता है।

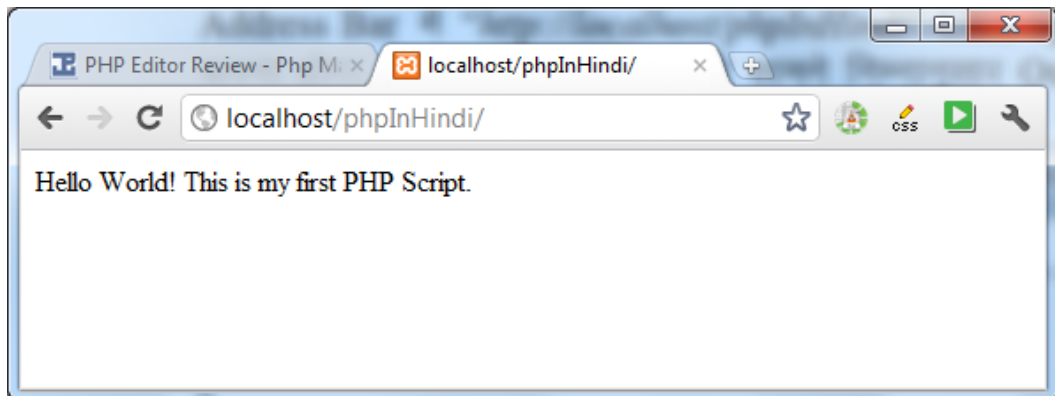
अपनी File को **PHP File** के रूप में Save करने के लिए हमें Notepad++ के **“Save As”** Dialog Box में निम्न चित्रानुसार **“Save as type:”** के Combo Box से **“PHP Hypertext Preprocessor file (*.php;*.php3;*.phtml)”** Option को Select करना होगा, जैसाकि निम्न चित्र में दिखाया गया है।



चूंकि हमें हमारी File को Web Browser में **“localhost”** Address से Access करने की जरूरत पड सकती है, इसलिए हमें हमारी Script File को हमारे Local Web Site के **www** नाम के Folder में ही Save करना होगा। इसलिए हमने हमारी PHP File के Saving Path को निम्न चित्रानुसार **“C:\wamp\www\phpInHindi”** Specify किया है।



यदि सबकुछ सही तरीके से किया गया, तो File Save हो जाने के बाद हमारा PHP Code Notepad++ के चित्र में दिखाए अनुसार Colors में दिखाई देने लगेगा। अब इस PHP Script को Interpret करने के लिए हमें इस File को Web Browser में Open करना होगा। Web Browser में इस File को Interpret करते हुए Open करने के लिए हमें Web Browser के Address Bar में "<http://localhost/phpInHindi/>" URL Specify करके Enter Key को Press करना होगा और ऐसा करते ही हमारे सामने निम्नानुसार Output Render होगा:



यदि हमने हमारी File का नाम "**index**" के अलावा कुछ भी जैसे कि **demo** Specify किया होता, तो फिर यही Result प्राप्त करने के लिए हमें Web Browser के Address Bar में "<http://localhost/phpInHindi/demo.php>" URL Specify करके Enter Key को Press करना पड़ता।

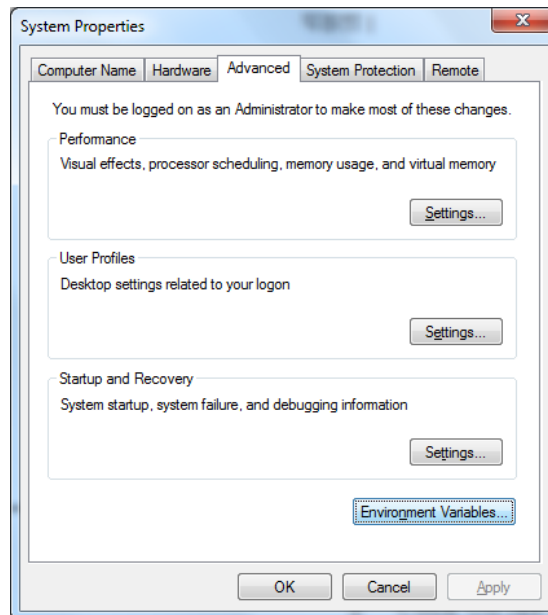
हालांकि इस तरीके को Use करके भी हम हमारे किसी भी PHP Script का Output देख सकते हैं, लेकिन PHP को जल्दी सीखने का केवल एक ही तरीका है कि हम PHP को Command Prompt का प्रयोग करते हुए सीखें व PHP की विभिन्न Scripts को PHP Command Prompt पर Interpret व Run करें।

Setting PHP Interpreter Path

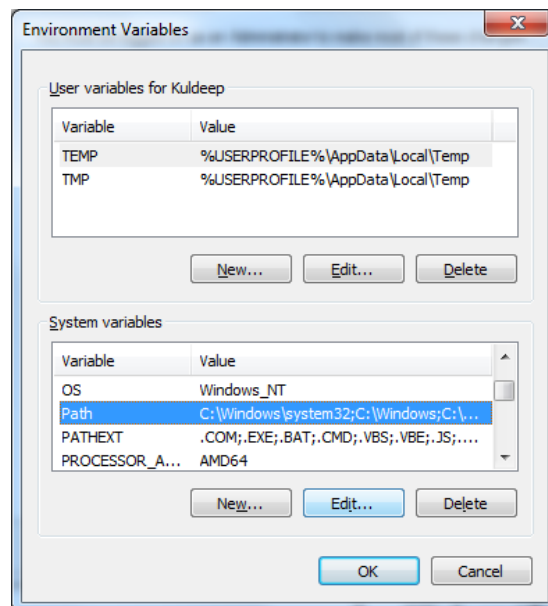
जब हम PHP Scripts को Command Prompt द्वारा Run करना चाहते हैं, तो सबसे पहले हमें **php.exe** File को "**Path**" Environment Variable में Set करना होता है, ताकि हम हमारे Computer में कहीं से भी PHP Interpreter को Invoke कर सकें।

चूंकि हमने WAMP Install किया है और हमारे Computer में PHP "**C:\wamp\bin\php\php5.3.8**" Path पर Installed है, जहां 5.3.8 PHP का Version है, जो कि बदल सकता है, इसलिए इस Path को हमें "**Path**" नाम के Global Variable में Set करना होगा। इसे Set करने के लिए हमें निम्न Steps Follow करने होते हैं:

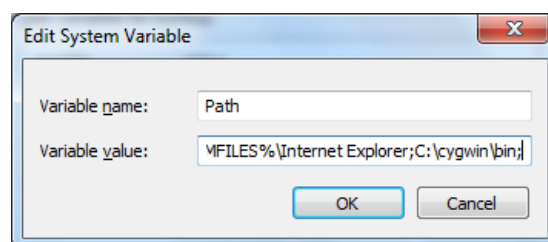
- 1 Right Click on My Computer and Select "**Properties**" Option
- 2 Click On **Advance System Settings** Option. A New Dialog Box will Display.



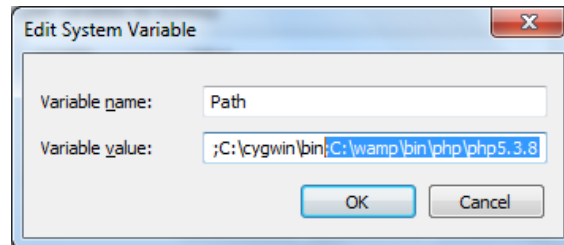
- 3 Click on “**Environment Variables...**” Button. A New Dialog Box will Display.



- 4 Select “**Path**” from “**System variables**” List and click on “**Edit...**” Button.

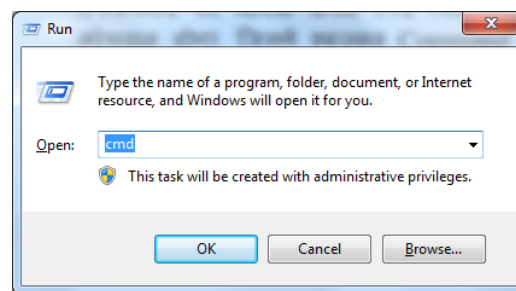


- 5 Go to the end of the “**Variable value:**” Text Box and place a Semicolon. Type the full path with Semicolon “;C:\wamp\bin\php\php5.3.8” at the end of the text as following:

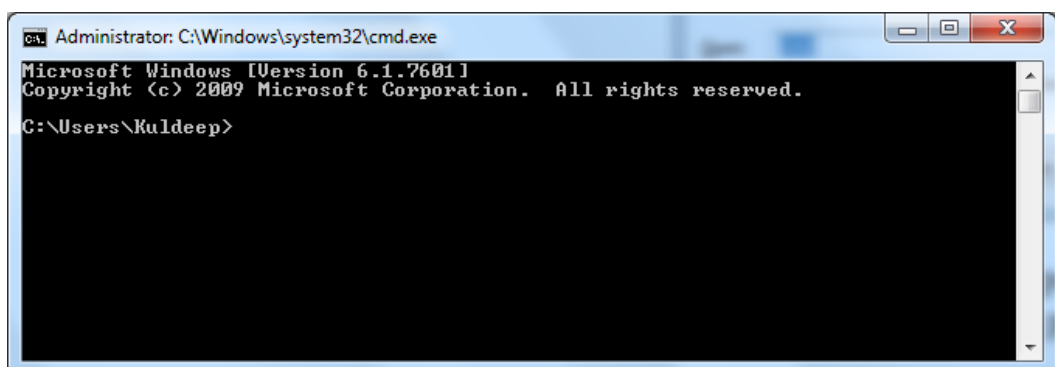


- 6 Click “OK”, “OK” and “OK”.

उपरोक्त प्रक्रिया पूरी करने के बाद अब हम हमारे Computer में कहीं से भी Command Prompt द्वारा **php.exe** Interpreter को Execute कर सकते हैं। अब **Start Button** पर Click करके **Run** Option को Select कीजिए। निम्नानुसार एक “Run” Dialog Box Display होगा:



यहां “**cmd**” Command Type करके Enter Key Press कीजिए अथवा “**OK**” Button पर Click कीजिए। आपके सामने निम्नानुसार Command Prompt Display हो जाएगा:

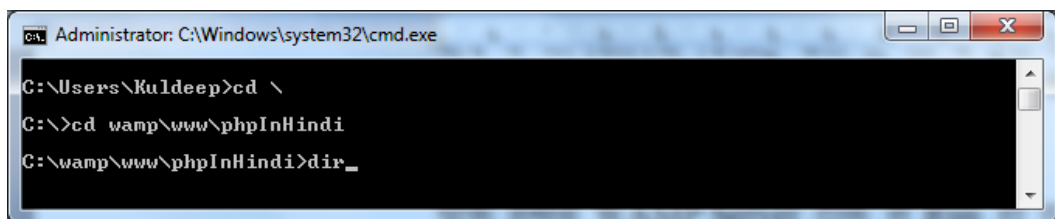


इस Command Prompt पर “**php -v**” Type करके Enter Key Press करने पर आपको उपरोक्त चित्रानुसार PHP का Version व विभिन्न प्रकार की अन्य जानकारियां दिखाई देंगी। यदि ये जानकारियां दिखाई देती हैं, तो ये इस बात का Signal है कि आपके PHP Interpreter का Path पूरी तरह से Set है और आप कहीं से भी PHP Interpreter को Invoke कर सकते हैं।

चूंकि हमारा WAMP Server ठीक से Run हो रहा है साथ ही हमने हमारे PHP Interpreter का Path भी Set कर दिया है, इसलिए अब हम Command Prompt द्वारा अपने PHP Program को Interpret कर सकते हैं।

हमने हमारा पहला PHP Script “C:\wamp\www\phpInHindi” नाम के Folder में Create किया था, इसलिए सबसे पहले हमें उस Folder में पहुंचना होगा। इस काम को करने के लिए हमें निम्नानुसार तीन Commands Fire करने होंगे:

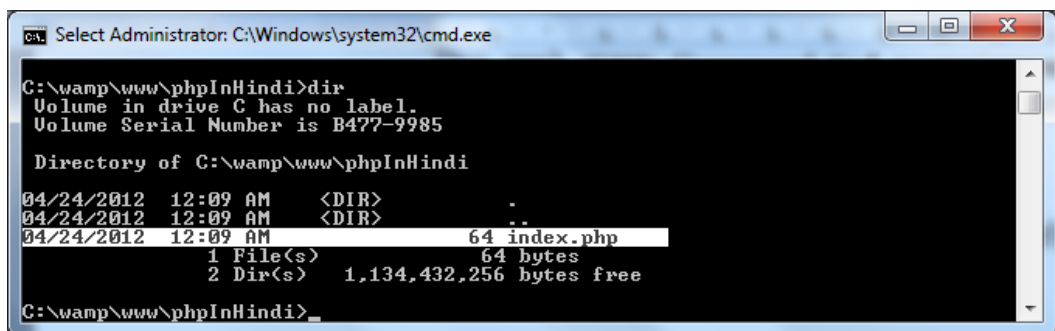
सबसे पहले हमने “cd \” Command Fire किया है, जो हमें Root Directory यानी “C:” Prompt पर ले जाता है।



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\Kuldeep>cd \
C:\>cd wamp\www\phpInHindi
C:\wamp\www\phpInHindi>dir_
```

फिर हमने दूसरा Command “cd wamp\www\phpInHindi” Fire किया है, जो हमें हमारे WAMP Server के www Folder के अन्दर स्थित “phpInHindi” नाम के Folder में ले जाता है।

अन्त में हमने “dir” Command किया है, जो हमें हमारे Current Folder की सभी Files Display करता है।

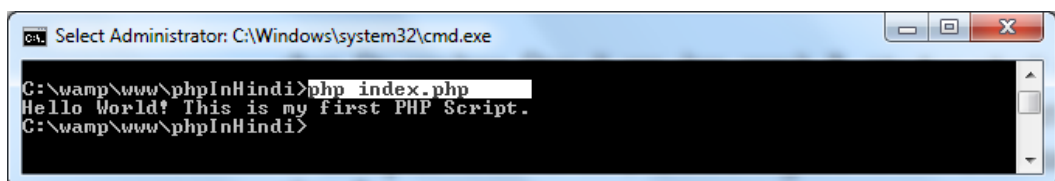


```
Select Administrator: C:\Windows\system32\cmd.exe
C:\wamp\www\phpInHindi>dir
Volume in drive C has no label.
Volume Serial Number is B477-9985

Directory of C:\wamp\www\phpInHindi
04/24/2012  12:09 AM    <DIR>          .
04/24/2012  12:09 AM    <DIR>          ..
04/24/2012  12:09 AM                64 index.php
               1 File(s)                64 bytes
               2 Dir(s)  1,134,432,256 bytes free
C:\wamp\www\phpInHindi>_
```

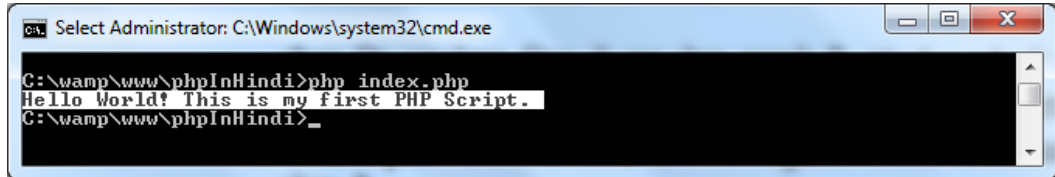
Interpreting First PHP Script

जैसा कि उपरोक्त चित्र में हम देख सकते हैं, “index.php” नाम की File इस Folder में स्थित है, इसलिए इस PHP Script File को हम यहां से Interpret कर सकते हैं। इस File को यहां से Interpret करने के लिए हमें निम्नानुसार अगला Command “php index.php” Fire करना होता है:



```
Select Administrator: C:\Windows\system32\cmd.exe
C:\wamp\www\phpInHindi>php index.php
Hello World! This is my first PHP Script.
C:\wamp\www\phpInHindi>
```

इस Command को Fire करते ही हमारी PHP Script File Interpret हो जाती है और हमें निम्नानुसार Output दिखाई देता है, जिसे हमने हमारी Script File में Execute होने के लिए लिखा था:



```
CA: Select Administrator: C:\Windows\system32\cmd.exe
C:\wamp\www\phpInHindi>php index.php
Hello World! This is my first PHP Script.
C:\wamp\www\phpInHindi>_
```

इस प्रकार से Command Prompt द्वारा हम किसी भी PHP Script File को Interpret कर सकते हैं और उसका Output देख सकते हैं।

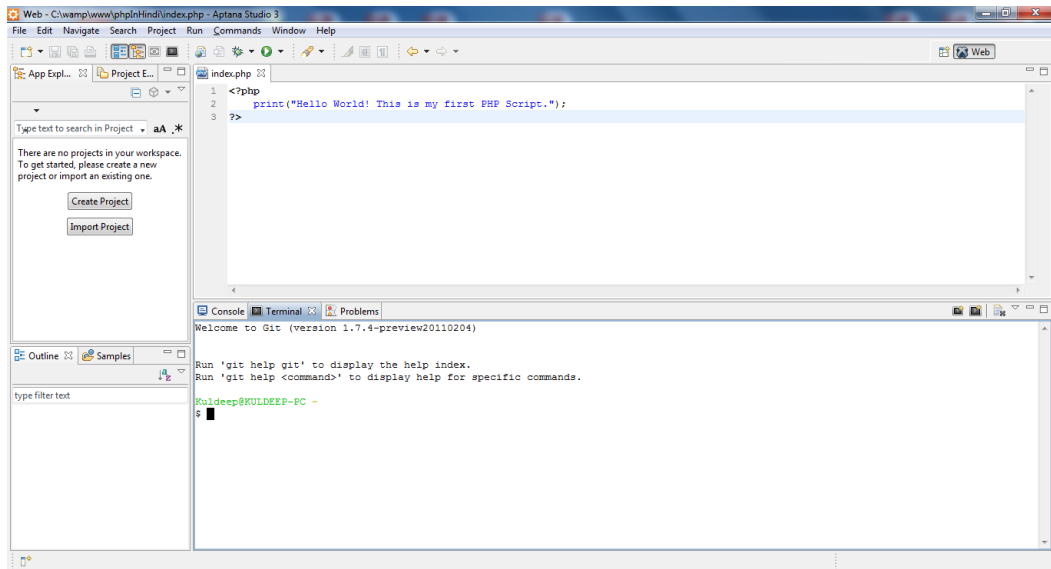
अब हम सारांश के रूप में उपरोक्त पूरी प्रक्रिया को समझते हैं, जो निम्नानुसार है:

- 1 सबसे पहले हमें हमारे PHP Interpreter का **Path** Operating System के **Global Variable** में **Set** करना होता है, ताकि PHP Interpreter को हम कहीं से भी **Invoke** करके अपनी PHP Script को **Interpret** कर सकें। इस **Path** को केवल एक ही बार **Set** करना होता है। इसे हर बार **Set** करने की जरूरत नहीं होती है।
- 2 फिर हमें एक **Text Editor** में अपनी PHP Script को **Type** करना होता है और अपने **Web Server** के **Root Folder** अथवा किसी **Sub-Folder** में **.PHP** Extension के साथ **Save** करना होता है, ताकि हम हमारे **Local Web Server** के **“localhost”** Path द्वारा भी उस **Script File** को **Access** व **Interpret** कर सकें। **.php** Extension **Specify** करना इसलिए जरूरी है, क्योंकि PHP Interpreter **.php** Extension वाली Files को ही **Interpret** करता है।
- 3 फिर हमें **Run Dialog Box** से **Command Prompt** को **Open** करना होता है और उस **Folder** तक पहुंचना होता है, जहां पर हमारी PHP File को हमने **Save** किया है।
- 4 अन्त में PHP Interpreter को **Use** करके हमें हमारी PHP Script File को **Interpret** करना होता है।

उपरोक्त Discussion द्वारा आप ये तो समझ ही गए होंगे कि एक PHP Script को Handle करने के लिए हमें कम से कम दो Programs पहला **Text Editor** व दूसरा **Command Prompt** को **Start** करके रखना होता है। जबकि **Aptana** का प्रयोग करके हम इन दो Programs से मुक्त हो सकते हैं। साथ ही **Aptana** हमें **Coding Related** कई प्रकार की **Help** भी करता है।

हालांकि हम चाहे **Command Prompt** Use करें या **Aptana Studio**, दोनों के लिए ही हमें PHP का **Path** Set करना जरूरी होता है। तो चलिए, समझने की कोशिश करते हैं कि किस तरह से हम **Aptana** का प्रयोग करके किसी PHP Script को आसानी से **Create** व **Interpret** कर सकते हैं।

सबसे पहले **“Aptana Studio 3”** Software को **Start** कीजिए व **File Menu** में जाकर अपनी **index.php** File को **Open** कीजिए। ऐसा करने पर आपको **Aptana Studio** निम्नानुसार दिखाई देगा:



इस Window में आपको तीन हिस्से दिखाई दे रहे हैं। Left Side के Tabs को Minimize किया जा सकता है क्योंकि फिलहाल हमारी जरूरत केवल Top व Bottom Window से ही पूरी हो जाएगी। Top Window में हमें हमारी *index.php* File के Code दिखाई दे रहे हैं जबकि Bottom Window एक **Terminal** Window है, जो Command Prompt की जरूरत को पूरा करता है।

चूंकि, ये Terminal Window Linux के Commands को Windows Operating System में Use करने की सुविधा देता है, इसलिए इस Window में “**dir**” Command के स्थान पर हमें “**ls**” Command को Use करना पड़ता है।

चूंकि हमारी *index.php* File “*C:\wamp\www\phpInHindi*” Path पर Stored है, तो इस Path पर जाने के लिए हमें दिखाई देने वाले Terminal Window में निम्नानुसार Command Fire करना होता है:

```
$ cd /c/wamp/www/phpInHindi
```

चूंकि हम एक ऐसे Terminal में हैं, जो कि Commands को Linux Format में Accept करता है, इसलिए हमें हमारे Path तक पहुंचने के लिए उपरोक्त Command को Use करना पड़ता है।

चूंकि Linux में Root Directory तथा Path को “/” से Refer किया जाता है, इसलिए इस Command में “/c/” “C:” Drive को Represent कर रहा है, जबकि शेष **Path** समान ही है।



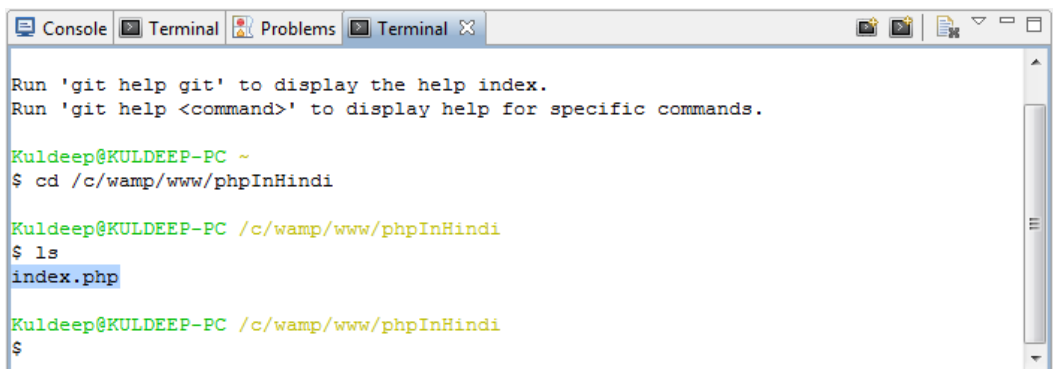
```
Console Terminal Problems Terminal X
Welcome to Git (version 1.7.4-preview20110204)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Kuldeep@KULDEEP-PC ~
$ cd /c/wamp/www/phpInHindi

Kuldeep@KULDEEP-PC /c/wamp/www/phpInHindi
$
```

हम Currently किस Path पर हैं, इसकी जानकारी हमें Green Color व Orange Color में दिखाई देने वाले Path से प्राप्त हो जाती है, जो कि “/c/wamp/www/phpInHindi” Path को Represent कर रहा है। इस Directory में Stored सभी Files व Directories की List देखने के लिए हम “ls” Command Fire कर सकते हैं, जो कि “dir” Command के समान ही है। इस Command को Fire करने पर हमें निम्न चित्रानुसार Files व Directories की List प्राप्त होती है:



```
Console Terminal Problems Terminal X

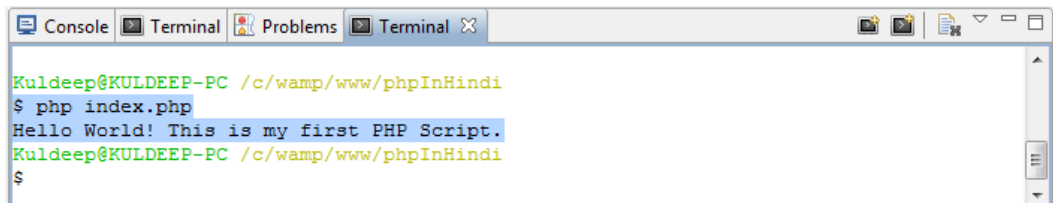
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Kuldeep@KULDEEP-PC ~
$ cd /c/wamp/www/phpInHindi

Kuldeep@KULDEEP-PC /c/wamp/www/phpInHindi
$ ls
index.php

Kuldeep@KULDEEP-PC /c/wamp/www/phpInHindi
$
```

उपरोक्त चित्र में हम देख सकते हैं कि हमारे Current Folder में *index.php* नाम की केवल एक ही File है। अब इस PHP File को Interpret करने के लिए हमें वही Command यहां Fire करना होता है, जिसे हमने Command Prompt में Fire किया था और हमें Interpreted PHP Script का Result इसी Terminal Window में दिखाई दे जाता है। जैसे:



```
Console Terminal Problems Terminal X

Kuldeep@KULDEEP-PC /c/wamp/www/phpInHindi
$ php index.php
Hello World! This is my first PHP Script.
Kuldeep@KULDEEP-PC /c/wamp/www/phpInHindi
$
```

उपरोक्त Discussion में हमने दो तरीकों को Use करते हुए एक PHP Script को Create करके Interpret किया। आप जो चाहें वो तरीका Use कर सकते हैं। मुझे Personally “Aptana Studio 3” का Terminal व Text Editor, Windows के Command Prompt व Notepad++ की तुलना में ज्यादा पसन्द है, क्योंकि ये मुझे कई प्रकार की Coding Help Provide करता है, जो कि मुझे Notepad++ से प्राप्त नहीं होती।

तो, हमने हमारा PHP Environment पूरी तरह से Setup कर लिया। यानी अपना Local Web Server, IDE, Global Path व Command Prompt को Use करने के बारे में पर्याप्त जानकारी प्राप्त कर ली साथ ही हमने हमारी पहली PHP Script File दो तरीकों से Create करके, उसे दोनों तरीकों से Interpret भी कर ली। यानी अब हम PHP का Basic समझने के लिए पूरी तरह से तैयार हैं। तो, चलिए! अब हम PHP का **Core** समझते हैं।

PHP

FUNDAMENTALS

PHP Fundamentals

किसी भी Programming Language के हमेशा कुछ नियम होते हैं, जिन्हें Follow किए बिना हम उस Programming Language को ठीक से नहीं समझ सकते। इन नियमों के समूह को ही Language का Core या Fundamental अथवा Basics कहा जाता है।

लगभग सभी Procedural Languages किसी न किसी तरह से “C” Language से तथा Object Oriented Languages किसी न किसी तरह से “C++” Language से ही Inspired है। चूंकि PHP एक ऐसी Programming Language है, जिसे हम एक Procedural Language की तरह भी Use कर सकते हैं और Object Oriented Languages की तरह भी।

इसलिए यदि आप पहले “C” Language सीख चुके हैं, तो PHP सीखना आपके लिए काफी आसान हो जाएगा, साथ ही यदि आपको Java का भी थोड़ा बहुत ज्ञान है, तो फिर बहुत ही आसानी से व तेजी से आप PHP को समझ पाएंगे और Professionally Use कर सकेंगे।

<?php ... ?>

किसी भी PHP Script के विभिन्न Statements को हमें <?php ... ?> के बीच ही लिखना होता है, क्योंकि PHP Interpreter इन्हीं Tags के बीच Enclosed Statements को PHP Interpreter पर Interpret होने के लिए भेजता है। इसलिए हम जितने भी PHP Programs बनाते हैं, उन सभी Programs के PHP Codes को हमें इन्हीं Tags के बीच लिखना होता है, भले ही हमारा Code केवल एक Single Line का ही क्यों न हो।

Output Statements

बिना Output Statements को समझे हुए हम आगे ही नहीं बढ़ सकते क्योंकि किसी भी Program के तीन Compulsory Parts होते हैं, जिन्हें **Input**, **Process** व **Output** कहा जाता है।

जब हम किसी Programming Language को सीख रहे होते हैं, तब Input के रूप में हम हमारे किसी Program में स्वयं के Data Specify कर सकते हैं, इसलिए Input हमारे Control में होता है। यदि हम चाहें तो हमारे Program में हम किसी भी तरह की Processing किए बिना भी कुछ काम कर सकते हैं, इसलिए किसी Data को Process करना है या नहीं, ये भी पूरी तरह से हमारे Control में होता है।

लेकिन हमने जो Data Input किया और उस पर जो Processing की है, उसे Output के रूप में दिखाने का काम पूरी तरह से Programming Language के Control में होता है। इसलिए Output के लिए हम हमेशा Programming Language द्वारा Provide किए जाने वाले Functions व Statements पर निर्भर होते हैं।

PHP हमें ऐसे बहुत सारे Functions व Statements Provide करता है, जिनका प्रयोग करके हम हमारे Program के Output को देख सकते हैं। PHP हमें विभिन्न प्रकार की परिस्थितियों में Generate होने वाले विभिन्न प्रकार के Outputs को Handle करने के लिए कई Functions व Statements Provide करता है, जिनके बारे में हम आगे जानेंगे, लेकिन इनमें से कुछ Functions व Statements निम्नानुसार हैं, जिनका प्रयोग हम हमारे Program का Output देखने के लिए कर सकते हैं:

print() Statement

ये एक Statement है न कि Function, जिसका Syntax निम्नानुसार होता है:

```
int print(string $argument);
```

जैसा कि इस Syntax में हम देख सकते हैं कि **print()** Statement Argument के रूप में किसी भी Data Type अथवा Mixed Data Type के मान को Accept करता है और जो भी मान इसे Argument के रूप में Specify किया जाता है, ये Function उस मान को Web Browser में अथवा Terminal पर Display कर देता है।

साथ ही ये Function एक Integer Return करता है। यदि Integer के रूप में 1 Return होता है, तो इसका मतलब ये है कि Function ठीक से Run हुआ जबकि यदि **print()** Statement ठीक से Run नहीं हो पाता, तो 1 के अलावा कुछ भी Return कर सकता है।

चूंकि **print()** एक Statement है न कि Function, इसलिए यदि हम चाहें, तो इसके साथ Specify किए गए Parenthesis को बिना Specify किए हुए भी इसे उपयोग में ले सकते हैं।

echo() Statement

print() Statement की तरह ही ये भी एक Function नहीं बल्कि Statement है और इसे भी Exactly Function की तरह उपयोग में नहीं लिया जा सकता। चूंकि ये भी Function नहीं है, इसलिए **print()** Statement की तरह ही, इसके साथ भी Paranthesis का प्रयोग करना जरूरी नहीं है। इसका Syntax निम्नानुसार होता है:

```
void echo ( string $arg1 [, string $... ] )
```

ये Statement भी बिल्कुल **print()** Statement की तरह ही काम करता है, लेकिन ये किसी तरह का कोई मान Return नहीं करता। हम **print()** या **echo()** दोनों में से किसी भी Statement को Use करके अपने Program के Output को प्राप्त कर सकते हैं। इन दोनों में से कोई भी Special नहीं है। बस हमें केवल इतना ध्यान रखना होता है, कि ये दोनों ही Functions नहीं हैं, इसलिए इन्हें Function मान कर Use नहीं करना चाहिए।

Case Sensitive

PHP भी "C" की तरह ही एक Case Sensitive Programming Language है। यानी "C" Language में भी विभिन्न User Defined Identifier Names Case Sensitive होते हैं। लेकिन PHP में सबकुछ Case Sensitive नहीं होता। उदाहरण के लिए User Defined Classes व Functions के नाम Case Sensitive नहीं होते। इसी तरह से PHP में पहले से बने हुए Built-In Functions, Classes आदि तथा Keywords Case Sensitive नहीं होते और हम इन्हें निम्नानुसार किसी भी तरह से लिख सकते हैं:

```
print "Hello World";  
PRINT "Hello World";  
Print "Hello World";
```

PrInT “Hello World”

उपरोक्त चारों Lines में हमने **print** Function को चार अलग तरीकों से लिखा है, लेकिन ये चारों Statements PHP Parser पर बिना किसी परेशानी के Execute हो जाते हैं, क्योंकि ये PHP का Built-In Function है। लेकिन PHP में हम जो Variable Create करते हैं, वे Case Sensitive होते हैं। उदाहरण के लिए \$name, \$Name, \$NAME, व \$NaMe चार अलग Variables हैं और इन्हें एक दूसरे के Replacement के रूप में Use नहीं किया जा सकता।

NOTE: PHP को भी “C” Language की तरह पूर्ण रूप से Case Sensitive मान कर ही Programming करना बेहतर रहता है, नहीं तो कई स्थानों पर हम Variables को भी Case Insensitive मान कर Coding करने लगते हैं, फलस्वरूप हमारे Codes में Bugs आने लगते हैं।

Statements and Semicolons

PHP Program की हर Line को एक Statement कहा जाता है और PHP के हर Statement का अन्त एक Semicolon से होता है। PHP Interpreter इसी Semicolon से Statement के अन्त को Identify करता है।

यदि हम किसी Statement के अन्त में Semicolon न लगाए, तो PHP Parser हमें Error देता है और हमारी PHP Script को Parse नहीं करता। इसीलिए हमने पिछले Code Segment में सभी Print Statements के अन्त में एक Semicolon का प्रयोग किया है।

Block Statement

PHP में जब हमें कुछ Statements को एक Group के रूप में Represent करना होता है, तब “C”, “C++” व Java की तरह ही उन सभी Statements को Opening व Closing Curly Braces के बीच Enclose किया जाता है और Closing Curly Brace के बाद Semicolon नहीं लगाया जाता। जैसे:

```
{
    print “Hello World”;
    PRINT “This is my first PHP Script.”;
}
```

Whitespace and Line Breaks

PHP में Coding को अच्छी तरह से Format करने के लिए हम जिस तरह से चाहें उस तरह से Whitespaces व Line Breaks का प्रयोग कर सकते हैं। PHP को इस बात से कोई फर्क नहीं पड़ता। उदाहरण के लिए उपरोक्त Code Segment को हम निम्नानुसार भी लिख सकते हैं:

```
print “Hello World”; PRINT “This is my first PHP Script.”;
```

How to Get Complete PDF EBook

आप **Online Order** करके **Online** या **Offline** Payment करते हुए इस Complete EBook को तुरन्त Download कर सकते हैं।

Order करने और पुस्तक को Online/Offline Payment करते हुए खरीदने की पूरी प्रक्रिया की विस्तृत जानकारी प्राप्त करने के लिए आप BccFalna.com के निम्न Menu Options को Check Visit कर सकते हैं।

How to Make Order

[How to Order?](#)

How to Buy Online

[How to Pay Online using PayUMoney](#)

[How to Pay Online using Instamojo](#)

[How to Pay Online using CCAvenue](#)

How to Buy Offline

[How to Pay Offline](#)

[Bank A/c Details](#)

जबकि हमारे Old Buyers के [Reviews](#) भी देख सकते हैं ताकि आप इस बात का निर्णय ले सकें कि हमारे Buyers हमारे PDF EBooks से कितने Satisfied हैं और यदि आप एक से अधिक EBooks खरीदते हैं, तो [Extra Discount](#) की Details भी Menubar से प्राप्त कर सकते हैं।