எளிய தமிழில்

இரா.கதிர்வேல்

PHP இணைய தளங்களை அட்டகாசமான வசதிகளோடு உருவாக்கும் ஒரு சிறந்த, ஆனால் மிக எளிய நுட்பம். விக்கிப்பீடியா, வேர்டுபிரஸ் போன்ற பல முக்கிய வலைத்தளங்கள் இந்த மொழியிலேயே உருவாக்கப் பட்டுள்ளன.

இதை, இந்த நூல் எளிமையாக அறிமுகம் செய்கிறது.

தமிழில் கட்டற்ற மென்பொருட்கள் பற்றிய தகவல்களை "கணியம்" மின் மாத இதழ், 2012 முதல் வெளியிட்டு வருகிறது.இதில் வெளியான PHP பற்றிய கட்டுரைகளை இணைத்து ஒரு முழு புத்தகமாக வெளியிடுவதில் பெரு மகிழ்ச்சி கொள்கிறோம்.

உங்கள் கருத்துகளையும், பிழை திருத்தங்களையும் editor@kaniyam.com க்கு மின்னஞ்சல் அனுப்பலாம்.

http://kaniyam.com/learn-php-in-tamil-ebook என்ற முகவரியில் இருந்து இந்த நூலை பதிவிறக்கம் செய்யலாம். உங்கள் கருத்துகளையும் இங்கே பகிரலாம்.

படித்து பயன் பெறவும், பிறருடன் பகிர்ந்து மகிழவும்

வேண்டுகிறோம்.

கணியம் இதழை தொடர்ந்து வளர்க்கும் அனைத்து அன்பர்களுக்கும் எமது நன்றிகள்.

த.சீனிவாசன்
tshrinivasan@gmail.com

ஆசிரியர்
கணியம்
editor@kaniyam.com

எளிய தமிழில் PHP

முதல் பதிப்பு பிப்ரவரி 2016

பதிப்புரிமம் © 2016 கணியம்.

ஆசிரியர் – இரா.கதிர்வேல் – linuxkathirvel.info@gmail.com

பிழை திருத்தம்: த.சீனிவாசன் - tshrinivasan@gmail.com

வடிவமைப்பு: த.சீனிவாசன்

அட்டைப்படம் - மனோஜ் குமார் - socrates1857@gmail.com

இந்த நூல் கிரியேடிவ் காமன்ஸ் என்ற உரிமையில் வெளியிடப்படுகிறது . இதன் மூலம், நீங்கள்

- யாருடனும் பகிர்ந்து கொள்ளலாம்.
- திருத்தி எழுதி வெளியிடலாம்.
- வணிக ரீதியிலும்யன்படுத்தலாம்.

ஆனால், மூலப் புத்தகம், ஆசிரியர் மற்றும் www.kaniyam.com பற்றிய விவரங்களை சேர்த்து தர வேண்டும். இதே உரிமைகளை யாவருக்கும் தர வேண்டும். கிரியேடிவ் காமன்ஸ் என்ற உரிமையில் வெளியிட வேண்டும்.

நூல் மூலம் :

http://static.kaniyam.com/ebooks/learn-php-in-tamil/learn-php-in-tamil.odt

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 Unported License

முன்னுரை

தமிழ்வழி கல்வியில் படித்தவர்கள் ஒரு தொழில்நுட்பத்தைக் கற்றுக்கொள்ள வேண்டும் என ஆசைப்படும் போது அவர்கள் கற்றுக்கொள்ள விரும்பும் தொழில்நுட்பம் தமிழிலேயே இருக்கும் பட்சத்தில் அவர்களால் அதை நன்கு புரிந்துகொண்டு எளிமையாக கற்றுக்கொள்ள முடியும். தமிழ் மொழி தெரிந்தவர்கள் ஒன்றை கற்றுக்கொள்ள வேண்டும் என நினைக்கும் போது, ஆங்கிலம் அதற்கு தடையாக இருக்கக்கூடாது. தற்போது ஆங்கிலம் அதற்கு தடையாக இருப்பதாக நான் நினைக்கிறேன். இந்நிலையில் அந்த தடையை உடைக்க என்னால் முடிந்த பங்களிப்பை இந்த புத்தகத்தின் மூலமாக அளித்திருக்கிறேன். இன்றைக்கு எதற்கெடுத்தாலும் இணையத்தை நாடிச்செல்லும் நிலைமை வந்து விட்டது. அப்படிப்பட்ட இணையத்தில் எது தொடர்பாக தமிழில் தேடினாலும் தகவல்கள் கிடைக்க வேண்டும். அந்த நிலைமையை நாம் அனைவரும் இணைந்து உருவாக்க வேண்டும்.

நான் சென்னைக்கு வேலைதேடி வந்தபோது PHP Developer ஆக வேலைக்குச் செல்லவேண்டும் எனும்

முடிவில் HTML, CSS, JavaScript, Bootstrap, jQuery ஆகியவைகளைப் பற்றி படித்துக்கொண்டிருந்தேன். இவைகளை படித்து முடித்துவிட்டு அதன்பிறகு PHP பற்றி படிக்கலாம் என நினைத்துக்கொண்டிருந்தேன். கணியம் இதழின் ஆசிரியர், ஸ்ரீனிவாசன் அவர்களிடம் நான் வேலைக்குச் செல்வது தொடர்பான ஆலோசனைகளை கேட்பதற்காக. சென்னைக்கு வந்துள்ளது தொடர்பாகவும், PHP தொடர்பாக படித்துக்கொண்டிருப்பதையும் தெரிவித்தேன். "அப்படியா மகிழ்ச்சி, அப்படியே PHP யைப் பற்றி கணியத்திற்கு கட்டுரைகள் எழுதிக்கொடுங்கள்" எனக் கூறினார். தினமும் காலை 11-மணியிலிருந்து இரவு 7-மணி வரை ஒரு வாரத்திற்கு தீவிரமாக கட்டுரைகளை எழுதி கணியத்திற்கு அனுப்பி வைத்தேன். அடுத்த ஒரு சில வாரங்களில் எதிர்பாராத விதமாக நான் Python Developer ஆக பணியில் சேர்ந்துவிட்டதால் அதன்பிறகு மீதமிருந்த ஒரு சில பகுதிகளை எழுதிமுடிக்கமுடியவில்லை. நேரம் கிடைக்கும் போதெல்லாம் கொஞ்சம் கொஞ்சமாக அனைத்து பகுதிகளையும் எழுதி முடித்து கணியம் இதழிற்கு அனுப்பி வைத்தேன். அந்த அனைத்து பகுதிகளும் கணியத்தில் வெளியிடப்பட்டு, அவைகள் தொகுக்கப்பட்டு இப்போது புத்தகமாக வெளிவந்துள்ளது. PHP பற்றி கணியம் இதழில் எழுத வாய்ப்பளித்து, ஊக்கமளித்த ஸ்ரீனிவாசன் அவர்களுக்கும், கணியம் இதழுக்கும், கணியம் குழுவினருக்கும் என மனமார்ந்த நன்றியையும்

தெரிவித்துக்கொள்கிறேன்.

ஒரு நல்ல நோக்கத்திற்காக பலர் இணைந்து ஒரு செயலைச் செய்யும் போது அந்த குழுவில் நாமும் இருந்தோம் என்பது எவ்வளவு பெருமையான விஷயம். அந்தவகையிலே கணியம் குழுவில் நானும் இணைந்திருப்பது பெருமையாக இருக்கிறது. கணியம் மிகப்பெரிய நோக்கத்துடன் இயங்கிக்கொண்டிருக்கிறது. அந்த நோக்கத்தின் சிறிய பங்களிப்பாக PHP பற்றிய தொடர்களை கணியத்தில் எழுதினேன்.

இந்த தொடரை எழுதுவதற்கு பல வழிகளிலும் எனக்கு உதவி செய்த என்னுடைய வழிகாட்டிகள் அன்பு(எ)மணிகன்டண், அண்ணன் வை.சிதம்பரம், சோம.நீலகண்டன் ஆகியோருக்கும், சென்னையில் நான் தங்கியிருக்கும் என் அறை நண்பர்கள் கார்த்திக், வினோத், மணிமாறன், மதன், வெங்கட் ஆகியோருக்கும், அலுவலக நண்பர்கள் கிருஷ்ணன், ராஜாசிங், பிரபாகரன், வினோத், முத்துராஜ் ஆகியோருக்கும் எனது மனமார்ந்த நன்றியை தெரிவித்துக்கொள்கிறேன்.

இரா.கதிர்வேல்

சித்தாதிக்காடு,

28.12.2015

வலைப் பதிவு: http://gnutamil.blogspot.in

மின்னஞ்சல்: linuxkathirvel.info@gmail.com

பொருளடக்கம்

முன்னுரை..8

1 PHP யின் வரலாறு...37

1.1 PHP யின் வரலாறு...37

1.2 PHP உருவான விதம்..38

1.3 PHP 3 யின் வெற்றி...41

1.4 PHP 4 - விஸ்பரூபம்...42

1.5 PHP 5 – Object Orientation , Error Handling and XML...43

1.6 PHP பிரபலமாக உள்ளது எப்படி?...................44

2 PHP அறிமுகம்...46

2.1 PHP என்றால் என்ன?...46

2.2 PHP -யால் என்னென்னவெல்லாம் செய்ய முடியும்?...47

2.3 ஏன் PHP?...48

2.4 PHP எப்படி வேலை செய்கிறது?.......................49

2.5 PHP ஏன் மிகவும் பயனுள்ளதாக இருக்கிறது?..57

3 LAMP Server -ஐ உபுண்டு 12.04 -ல் நிறுவுதல்.......59

3.1 முதல் படி:..60

3.2 Apache Server -னை சோதனை செய்து பார்க்க:
....64

3.3 PHP - யினை சோதனை செய்து பார்க்க:........65

4 PHP Script உருவாக்குதல்...69

4.1 PHP Script உருவாக்குதல்..................................69

4.2 PHP நிரல் எழுத தேவையானவைகள்:..............69

4.3 PHP நிரல் வரம்புச்சுட்டி (Code Delimiters):....70

4.4 PHP உங்கள் கணினியில் நிறுவப்பட்டுள்ளதா?
....72

4.5 PHP நிரல் பொதிதல் முறைகள்:.........................76

5 Comments - குறிப்புரைகள்...............82

5.1 ஒற்றைவரி குறிப்புரை:.................85

5.2 பலவரி குறிப்புரை:.......................86

6 மாறிகள் (Variables).......................88

6.1 மாறிகளுக்கு பெயர் வைத்தல்:....88

6.2 மாறிகளுக்கு மதிப்புகளை அளித்தல்:...91

6.3 மாறிகளின் மதிப்புகளை அணுகுதல்:...92

6.4 PHP மாறியினுடைய வகையை மாற்றுதல்:...95

6.5 மாறி மதிப்புகளை வைத்திருக்கிறதா என சோதித்தல்(Check Whether a variable is set):...................97

7 PHP மாறி வகைகள்:........................100

7.1 முழு எண் மாறி வகை (Integer Variable Type):...................................100

7.2 மிதவை எண் மாறி வகை (Float Variable Type):....................................101
7.3 பூலியன் வகை மாறி (Boolean Variable Type):....................................103
7.4 சர மாறி வகை (String Variable Type):..104
8 மாறிலி (Constants)........................108
8.1 மாறிலியை வரையறுத்தல் (Defining a Constant):........................109
8.2 முன் வரையறுக்கப்பட்ட மாறிலிகள் (Predefined Constants): 113
9 Operators (வினைக்குறி)................119
9.1 எண்கணி மற்றும் வழங்குதல் வினைக்குறிகள் (Assignment

Operators):..121

9.2 கணித வினைக்குறிகள்

(Arithmetic Operators):.....................127

9.3 ஒப்பீடு வினைக்குறி (Comparison

Operators):..129

9.4 ஏரண வினைக்குறிகள் (Logical

Operators):..132

9.5 ஏறுமான மற்றும் இறங்குமான

வினைக்குறிகள் (Increment and

Decrement Operators):.....................133

9.6 சரத்தொடர் இணைப்பு

வினைக்குறி (String Concatenation

Operator):..136

9.7 செயற்படுத்தும் வினைக்குறி —

வழங்கியில் கட்டளைகளை செயற்படுத்துதல் (Execution Operator – Executing Server Side Commands)..137
10 Flow Control and Looping..........139
10.1 Conditional Statements........140
10.2 கண்ணி கூற்றுகள் (Looping Statements)..147
10.3 switch கூற்று (switch statements)...158
10.4 கண்ணி முறிப்பு (Breaking a Loop):...165
11 Functions..171
11.1 Function (செய்லகூறு) என்றால் என்ன?..171

11.2 செயல்கூறை(function) எப்படி எழுதுவது?..172
11.3 செயல்கூறில் இருந்து மதிப்புகள் திரும்புதல் (Returning a Value from a function)..174
11.4 செயல்கூறுக்கு அளபுருக்களை செலுத்துதல் (passing parameters to a function)..176
11.5 செயல்கூறை அழைத்தல் (calling functions)..178
11.6 Passing Parameters by Reference..181
11.7 Functions and Variable Scope

....186
12 Arrays..188
12.1 Numerical Array......................190
12.2 Associative Array....................190
12.3 Array உருவாக்குதல் (Creating a Array)..190
12.4 Empty Array உருவாக்கம் (empty array creation).....................................191
12.5 Array - யின் உறுப்புகளை அணுகுதல்..192
12.6 Associative Array யை உருவாக்குதல் (Creating an Associative Array)..194
12.7 Associative Array – யின்

உறுப்புகளை அணுகுதல் (Accessing Elements of an Associative Array)…195

12.8 Array சுட்டியைப் பயன்படுத்துதல்(Using Array Pointers) 197

12.9 Array யின் உறுப்புகளை மாற்றுதல், சேர்த்தல் மற்றும் நீக்குதல்(Changing, Adding and Removing Array Elements)................202

12.10 Looping மூலம் array – யின் உறுப்புகளை அணுகுதல்(Looping through array Elements)....................210

12.11 Replacing Sections of an Array

....215

12.12 Array - யை வரிசைப்படுத்துதல்.

....216

12.13 Associative Array - யை வரிசைப்படுத்துதல்...............................224

12.14 Array – யைப் பற்றிய தகவல்களைப் பெறுதல் மற்றும் இதர array செயல்கூறுகள்(functions).........225

13 Working with Strings and Text in PHP...226

13.1 எழுத்துக்களை மாற்றுதல் (Changing the Case of a PHP String)

....227

13.2 ASCII மதிப்புக்கு மாற்றுதல்

மற்றும் ASCII மதிப்புகளிலிருந்து மாற்றுதல்...231

13.3 வடிவுறு சரங்களை அச்சிடுதல் (Printing Formatted Strings).............238

13.4 சரத்தின் நீளத்தை கண்டுபிடித்தல் (Finding the Length of a String)..241

13.5 சரத்தை Array யாக மாற்றுதல் (Converting a String into a Array)....242

13.6 சரத்தின் முன்னும் பின்னும் இருக்கக்கூடிய whitespace ஐ நீக்குதல் (Removing Leading and Trailing Whitespace from a String)................244

13.7 சரங்களை ஒப்பிடுதல் (Comparing Strings)..........................246
13.8 சரத்தை அணுகுதல் மற்றும் மாற்றுதல் (Accessing and Modifying Characters in String)..........................249
13.9 சரத்திற்குள் உருவை தேடுதலும் , பகுதிச்சரமாக பிரித்தலும் (Searching for Characters and Substrings in a String) 251
13.10 Extracting and Replacing Substrings..254
13.11 Replacing All Instances of a Word in a String.................................256
14 கோப்பு முறைமையும், கோப்புகள்

உள்ளீடும் / வெளியீடும் (File systems and File I/O)..259
14.1 கோப்புகளை திறத்தலும் உருவாக்குதலும் (Opening and Creating Files)..260
14.2 கோப்புகளை மூடுதல் (Closing Files)..262
14.3 கோப்பில் எழுதுதல் (Writing to a File)..264
14.4 கோப்பிலிருந்து தகவல்களைப் படித்தல் (Reading From a File)..........266
14.5 கோப்பு இருக்கிறதா என சோதித்தல் (Checking Whether a File Exists)..268

14.6 கோப்புகளை பிரதியெடுத்தல், நகர்த்துதல் மற்றும் அழித்தல் (Moving, Copying and Deleting Files)..............269
14.7 கோப்புகளின் பண்புகளை அணுகுதல்(Accessing File Attributes) 271
14.8 வெளியீட்டு வைப்பகம் (Output Buffering)..275
15 அடைவுகளுடன் பணியாற்றுதல் (Working with Directories)....................280
15.1 புதிதாக அடைவுகளை உருவாக்குதல் (Creating Directories) 281
15.2 அடைவை நீக்குதல் (Deleting

Directory)..283
15.3 Finding and Changing the Current Working Directory................285
15.4 அடைவிற்குள் இருக்கும் கோப்புகளை பட்டியலிடுதல் (Listing Files in a Directory)............................287
16 HTML Forms ஒரு பார்வை...........289
16.1 HTML படிவங்கள் உருவாக்குதல் (Creating HTML Forms)......................290
16.2 HTML Text Object (உரை பொருள்)..292
16.3 HTML TextArea Object (உரைப்பகுதி பொருள்).......................294
16.4 The HTML Button Object

(பொத்தான் பொருள்).........................298
16.5 HTML check Boxes.................301
16.6 HTML Radio Button...............304
16.7 HTML Drop-down / Select Object...306
16.8 HTML Password Object.........312
17 PHP and HTML Forms..................314
17.1 படிவம் உருவாக்குதல் (Creating the Form)...314
17.2 PHP ஐ பயன்படுத்தி படிவத்தின் தகவலை Process செய்தல் (Processing Form Data Using PHP).......................318
17.3 Processing Multiple Selections

with PHP(பல தேர்வுகளை

செயல்படுத்துதல்)................................323

18 PHP and Cookies – Creating, Reading and Writing (குக்கீஸ் உருவாக்குதல், படித்தல் மற்றும் எழுதுதல்) 328

18.1 குக்கீஸ்......................................329

18.2 The Difference Between

Cookies and Sessions (Cookies and

Sessions இரண்டிற்குமான வேறுபாடு)

....330

18.3 குக்கீயினுடைய அமைப்பு (The

Structure of Cookie).........................332

18.4 குக்கீஸ் காலாவதியாகும்

நேரத்தை அமைத்தல்(Cookie Expiration Setting)..332

18.5 குக்கீயின் பாதை அமைப்பு(Cookie path Setting)..........333

18.6 குக்கீ domain அமைப்பு(Cookie domain Setting)..................................333

18.7 குக்கீயின் பாதுகாப்பு அமைப்பு(Cookie Security Setting)....334

18.8 குக்கீ உருவாக்குதல்(Creating a Cookie in PHP)...................................334

18.9 குக்கீயினைப் படித்தல்(Reading a Cookie in PHP)...................................336

18.10 குக்கீயை அழித்தல்(Deleting a

Cookie)..338
19 அமர்வு (Understanding PHP Sessions)..342
19.1 Session என்றால் என்ன?.......342
19.2 PHP Session உருவாக்குதல் (Creation a PHP Session)..................343
19.3 Session மாறிகளை உருவாக்குதல் மற்றும் படித்தல்(Creating and Reading Session Variables)345
19.4 Session தகவல்களை கோப்பில் எழுதுதல்(Writing Session Data to a File)................348
19.5 கோப்பில் சேமிக்கப்பட்ட session தகவல்களை படித்தல் (Reading Saved Session)..........................354
20 பொருள் நோக்கு நிரலாக்கம் (Object Oriented Programming)..356
20.1 Object என்றால் என்ன?..............................356
20.2 Class என்றால் என்ன?..................................357
20.3 Class –லிருந்து Object ஐ உருவாக்குவது

எப்படி?...357
20.4 sub-classing என்றால் என்ன?......................358
20.5 PHP class ஐ வரையறை செய்தல்...............359
20.6 PHP class உருவாக்குதல் மற்றும் சிதைத்தல்
(class constructors and destructors).......................360
20.7 PHP class இல் உறுப்பினர்கள்(members)
உருவாக்குதல்...365
20.8 Methods ஐ வரையறை செய்தல் மற்றும்
அழைத்தல்(Defining and Calling Methods).............369
20.9 Subclassing in PHP.......................................373
20.10 ChildClass மூலமாக ParentClass இன்
method ஐ பயன்படுத்திக்கொள்ளுதல்......................375
20.11 PHP Object Serialization...........................377
20.12 PHP Object பற்றிய தகவல்களைப் பெறுதல்
....379
21 PHP யும் தரவுத்தளமும் (Using PHP with MySQL)
....384
21.1 PHP உடன் MySQL ஐ இணைத்தல் (Connect

with PHP to a MySQL Server)....................................385
21.2 MySQL தரவுதளத்திலிருந்து PHP மூலமாக பதிவேடுகளை(Record) தேர்வு செய்தல் (Selecting Records from a MySQL Database Using PHP):......388
21.3 பதிவேட்டில் தகவல்களை சேர்த்தல் Adding Records to MySQL Database using PHP.................391
21.4 Using PHP to get Information about a MySQL Database...394
22 PHP மற்றும் SQLite (PHP and SQLite)................398
22.1 PHP வழியாக SQLite Database உருவாக்குதல் (Creating an SQLite Database with PHP)...............398
22.2 PDO (PHP Data Objects) மூலமாக SQLite DB ஐ உருவாக்குதல்...398
22.3 PHP மூலமாக SQLite இல் Table உருவாக்குதல் (Using PHP to Create Table to an SQLite Database) 400
22.4 Using PHP to Add Records to an SQLite Database...402
22.5 PHP மூலமாக Records களை தேர்வு செய்தல்

(Using PHP to Select Records from an SQLite Database)..404
23 முடிவுரை..408
24 ஆசிரியர் பற்றி..411
25 கணியம் பற்றி...415
இலக்குகள்...415
பங்களிக்க...416
விண்ணப்பங்கள்..419
வெளியீட்டு விவரம்...420
26 நன்கொடை...421

1 PHP யின் வரலாறு

1.1 PHP யின் வரலாறு

பிரச்சனைகள் ஏற்படும் போதே அதன் தீர்வுகளும் தேடப்படுகிறது. எங்கு தேடியும் தீர்வுகள் கிடைக்காத பட்சத்தில், அதற்கான தீர்வை தாமாகவே முயன்று கண்டுபிடிப்பர். அவருக்கு ஏற்பட்ட அந்த பிரச்சனை வேறு ஒருவருக்கு ஏற்படும் போது, மற்றவர்களுக்கும் அவரைப் போல கஷ்டபடாமல் இருப்பதற்காக, கண்டுபிடித்த அந்த தீர்வை அனைவரும் தெரிந்து கொள்ளுவதற்காக இலவசமாக கிடைக்கும் வகையில் ஏற்பாடு செய்கிறார்.

மற்றவர்களும் அந்த தொழில்நுட்பத்தை(தீர்வு) ஏற்றுக்கொண்டு அதை மெருகேற்றும் போது அந்த தீர்வை கண்டுபிடித்தவரே கற்பனை செய்ய முடியாத அளவிற்கு அந்த தொழில்நுட்பம் வளர்ந்து நிற்கும்.

1.2 PHP உருவான விதம்

PHP யின் முதல் பதிப்பு **1995** ஆம் ஆண்டு **Ramus Lerdof** அவர்களால் உருவாக்கப்பட்டது. Rasmus தற்போது Yahoo நிறுவனத்தில் பொறியாளராக பணிபுரிந்து வருகிறார். அவருடைய இணையதளத்தை எளிமையாக உருவாக்க HTML உடன் இணைந்து நன்றாக வேலை செய்யும் ஏதோ ஒன்று தேவைப்பட்டது. முக்கியமாக இணைய உலாவியிலிருந்து வழங்கிக்கு தகவல்களை அனுப்பவும், வழங்கியில் இருந்து இணைய உலாவியில் தகவல்களைப் பெறவும் எளிமையான ஒரு தொழில்நுட்பம் அல்லது மொழி தேவைப்பட்டது.

அந்த தேவையை பூர்த்தி செய்யும் வகையில் **Perl** மொழியினைக் கொண்டு ஒரு தொழில்நுட்பத்தை உருவாக்கினார். அந்த தொழில்நுட்பத்திற்கு அவர் இட்ட பெயர் '**Personal Home Page / Form Interpreter'**. Rasmus Lerdof அவர்கள் உருவாக்கிய அத்தகைய தொழில்நுட்பம் இணைய உள்ளடக்கங்களையும், இணைய படிவங்களையும் செயல்முறைப்படுத்துவதற்கா வசதியான வழிகளை ஏற்படுத்தி கொடுத்தது.

'Personal Home Page / Form Interpreter' என்னும் பெயர் பின்பு PHP/FI என்று சுருக்கி அழைக்கப்பட்டது. இறுதியாக '**PHP: Hypertext Preprocessor'** என பெயர்

மாற்றம் செய்யப்பட்டது. 'GNU's Not Unix' என்பது எப்படி GNU என்று சுருக்கமாக அழைக்கப்படுகிறதோ, அதே போல 'PHP: Hypertext Preprocessor' என்பதும் **PHP** என்று சுருக்கமாக அழைக்கப்படுகிறது.

PHP/FI யின் முதல் பதிப்பு(Version 1.0) Rasmus அவர்களுடைய சொந்த இணையதளத்தைத் தாண்டி வேறு எங்கும் பயன்படுத்தப்படவில்லை. காரணம் அவரினுடைய சொந்த தேவைக்காக அவர் அதை உருவாக்கினார். **PHP/FI 2.0** னுடைய அறிமுகம் அதை மாற்ற தொடங்கியது.

ஆனால் **PHP 3 பதிப்பு 1997** ஆம் ஆண்டு வெளியிடப்பட்ட போது அனைவரின் நம்பிக்கையையும் சுக்குநூறாக்கி யாரும் எதிர்பாராத விதமாக PHP அனைவரின் மத்தியிலும் புகழ் பெற்றது.

1.3 PHP 3 யின் வெற்றி

1997 ஆம் ஆண்டு வாக்கில் இணையதளங்களின் வளர்ச்சி அசுர வேகமெடுத்தது. அவ்வாறு வளர்ச்சி பெற்ற இணையதளங்கள் அதே சமயத்தில் **Apache Web Server**

–ஐ பயன்படுத்தி வந்தன. அதே காலகட்டத்தில்தான் PHP யின் அடுத்த கட்ட வளர்ச்சிக்காக **Andy Gutmans** மற்றும் **Zeev Suraski** ஆகிய இருவரும் PHP 3 திட்டத்தை அறிமுகப்படுத்தினர். குறிப்பாக Apache Web Server உடன் PHP இணைந்து செயல்படும் விதமாக PHP வலிமையாக உருவாக்கி வெளியிடப்பட்டது.

அதேசமயத்தில்PHP + Apache Web Server கூட்டணி வெற்றியை நோக்கி பயணித்தது. இணைய உலகில் 10% க்கு மேலான இணையதளங்கள் PHP தங்களுடைய இணையதளத்தில் பயன்படுத்த ஆரம்பித்தன.

1.4 PHP 4 - விஸ்பரூபம்

Andi Gutmans and Zeev Suraski ஆகிய இருவராலும் PHP 3 மறுபடியும் மெருகேற்றப்பட்டது. PHP3-இன் மெறுகேற்றப்பட்ட பதிப்பு **PHP4** ஆக வெளிவந்தது. அவ்வாறு கட்டமைக்கப்பட்ட PHP4 ஒரு சிறிய தொழில்நுட்பத்துடன் வெளிவந்தது. அவ்வாறு இணைக்கப்பட்ட அந்த சிறிய தொழில்நுட்பம் Zend Engine என்று அழைக்கப்பட்டது.

மெருகேற்றப்பட்ட அம்சங்கள்:

- *மற்ற இணைய வழங்கிகளுக்கு* (Microsoft's Internet Information Server – IIS) *ஆதரவு அளிக்கும் வகையில் உருவாக்கப்பட்டது.*
- *நினைவக மேலாண்மையை திறம்பட செய்தல்*
- *பெரிய திட்டங்கள், வணிக பயன்பாடு மற்றும்* mission critical *பயன்பாடுகளுக்கு ஆதரவு*

1.5 *PHP 5 – Object Orientation , Error Handling and XML*

OOP *க்கு ஆதரவளிக்கும் வகையில் மேலும் மெருகேற்றப்பட்டது.* Java, Python *போன்று மற்ற மொழிகளில் உள்ளது போல* try/catch error and exception handling *வசதிகள் ஏற்படுத்தப்பட்டது. தரவுகளை கையாளுதல் குறிப்பாக* **XML** *மற்றும்* **SQLite** *போன்றவைகளை கையாளுதல் மற்றும் தரவுதளத்துடன் இணைந்து பணியாற்றுவதற்கான எளிய வழிமுறைகள் போன்ற வசதிகள் கொண்டு வரப்பட்டது.*

1.6 *PHP பிரபலமாக உள்ளது எப்படி?*

PHP *என்பது ஒரு* **Server Side** Scripting *மொழி* (Language) *என்பது நாம் அனைவரும் அறிந்த ஒன்றே.* PHP *இணைய பயன்பாடிற்காக வடிவமைக்கப்பட்டிருந்தாலும் அது ஒரு பொதுப் பயன்பாட்டிற்கான மொழியாகவும், அனைவராலும் விரும்பி பயன்படுத்தப்பட்டு வருகிறது. இன்றைக்கு இருக்கக்கூடிய அனைத்து இணைய வழங்கிகளிலுமே* Apache+PHP *உள்ளது. புதிதாக நிறுவப்படும் இணையவழங்கிகளில்* PHP *என்பது தவிர்க்க முடியாத ஒன்றாக உள்ளது.* 2013 *ஆம் ஆண்டில் எடுக்கப்பட்ட புள்ளிவிபர கணக்குப்படி,* 240 *மில்லியனுக்கு* (1 *மில்லியன்*=10 *லட்சம்,* 240 *மில்லியன்*=240000000) *அதிகமான இணையதளங்களில்* PHP *பயன்படுத்தப்பட்டு வருகிறது.* 2.1 *மில்லியன் இணைய வழங்கிகளில்* PHP *நிறுவப்பட்டு உள்ளது.*

Perl, Python *போன்ற மொழிகளை கற்றவர்கள்,* PHP–*ஐ கற்றுக்கொள்வது என்பது மிகவும் எளிது. அதுபோல எளிமையான* Syntax, *அனைத்து தகவல்தளங்களுடனும் ஒத்து இயங்குவது மற்றும் பாதுகாப்பு அம்சங்கள் மிகுந்திருப்பது என பல்வேறு அம்சங்கள்* PHP – *இல் இருப்பதால், இன்றைய இணைய உலகில்* Web

Developement ற்கு தகுந்த மொழியாக PHP பிரபலமடைந்துள்ளது.

2 PHP அறிமுகம்

2.1 PHP என்றால் என்ன?

PHP என்பது ஒரு Server Side Scripting language. எளிமையாக சொல்ல வேண்டுமானால் உங்களுக்கு ஒரு புத்தகம் தேவைப்படுகிறது. அதை இணையமூலம் வாங்குவதற்காக ஏதோ ஒரு பதிப்பகத்தின் இணையதளத்திற்கு செல்கிறீர்கள். அந்த பதிப்பகத்தின் இணையதளம் PHP மூலம் உருவாக்கப்பட்டதெனில். அந்த இணைதளத்தில் செய்யும் அனைத்து வேலைகளும் உங்களுடைய கணினியில் (Client Side) Process ஆகாமல், பதிப்பகத்தின் இணையதளம் எந்த வெப் சர்வரில்(Server Side) இருக்கிறதோ அங்கு Process செய்யப்பட்டு உங்களுக்கு தேவையான விபரங்களை இணையதளம் கொடுக்கும். அவ்வாறு Server இல் செயல்படுத்தப்படும் நிரல்கள் Server Side Scripting Language எனப்படும். PHP நிரல்கள் அனைத்தும் Server Side இல் Process செய்யப்படுவதால். PHP ஒரு server side scripting language ஆகும்.

Ruby, Python, Perl ஆகிய மொழிகளும் Server Side

Scripting Language ஆக பயன்படுத்தப்படுகிறது. அதுபோலவே PHP யும் இருந்தாலும், PHP சில தனிச்சிறப்புகளைக் கொண்டுள்ளது. அது என்னவெனில் நம்முடைய இணையதள உருவாக்க வேலைகளை எளிமையாக செய்வதெற்கென நிறைய Extension களை வைத்திருக்கிறது.

குறிப்பாக Database இல் தகவல்களை சேமிப்பதற்கும், Database இல் இருக்கும் தகவல்களை இணையதளத்தின் மூலம் பெறுவதற்கும், இணையதளங்களை Dynamic ஆக வடிவமைக்கவும், Content களை திறம்பட கையாள்வதற்கும் மிகவும் எளிமையான வழிகளை PHP கொண்டுள்ளது. அதனால் மேற்காணும் வேலைகளை நாம் மற்ற மொழிகளில் செய்வதை விட PHP யில் எளிமையாக செய்யலாம்.

2.2 PHP -யால் என்னென்னவெல்லாம் செய்ய முடியும்?

- Dynamic Page Content களை உருவாக்க முடியும்.
- Web Server இல் கோப்புகளை உருவாக்குதல், அழித்தல், நீக்குதல், திறத்தல், எழுதுதல் ஆகியவைகளை செய்ய முடியும்.
- படிவத்தின் தகவல்களை (Form Data) சேகரிக்க

முடியும்.

- Cookies களை அனுப்ப மற்றும் பெய முடியும்.
- தகவல்தளத்தில் (Database) தகவல்களை சேர்த்தல், நீக்குதல், மாற்றுதல் ஆகியவைகளை செய்ய முடியும்.
- பயனர்களினுடைய (Users) செயல்பாடுகளை கட்டுப்பாட்டிற்குள் வைக்க முடியும்.
- தகவல்களை Encrypt செய்ய முடியும்.
- HTML ஆக மட்டுமில்லாமல், Images, PDF Files, Flash Movies XML, XHTML ஆகிய வடிவங்களிலும் வெளியீடுகளை கொண்டு வர முடியும்.

2.3 ஏன் PHP?

- பல்வேறு இயங்குதளங்களில் PHP – ஐ இயக்க முடியும். (உதாரணமாக. Windows, Linux, Unix, Mac OS X, etc…)
- இன்றைக்கு பயன்பாட்டில் உள்ள அனைத்து Server (Apache, IIS, etc) களுடனும் ஒத்து இயங்கக்கூடியது.
- MySQL, SQLite, Postgres, Oracle, MS SQL போன்ற அனைத்து தகவல்தளங்களையும் PHP ஆதரிக்கிறது.

- PHP என்பது அனைவருக்கும் இலவசம். PHP யினுடைய அதிகாரப்பூர்வமான இணைதளத்தில்(www.php.net) இருந்து அனைவரும் இலவசமாகவே தரவிறக்கம் செய்து கொள்ளலாம்.

- கற்றுக்கொள்ள எளிமையான மொழியாகவும், Server Side இல் சிறப்பாக இயங்க்ககூடிய மொழியாகவும் PHP இருக்கிறது.

2.4 PHP எப்படி வேலை செய்கிறது?

பயனர் தன்னுடைய கணினியில் இருக்கும் இணைய உலாவியைத் திறந்து, உலாவியினுடைய முகவரிப்பட்டையில் இணையதளத்தின் முகவரியை கொடுத்து இயக்கும் போது, உலாவி வலைப்பக்கத்தின் பிரதியை கேட்டு இணைய வழங்கிக்கு கோரிக்கை அனுப்புகிறது.

இணைய வழங்கி அந்த கோரிக்கையை பெற்றுக்கொண்டு அந்த வலைப்பக்கத்தினை தேடி கண்டுபிடித்து பயனரினுடைய உலாவிக்கு அனுப்பி வைக்கிறது. இவையனைத்தும் இணையத்தின் மூலம் கனக்கச்சிதமாக நடைபெறும்.

இணைய வழங்கி வலைப்பக்கத்தின் உள்ளடக்கங்களைப் பற்றி எந்த கவலையும் கொள்ளாது. கேட்ட பக்கத்தினை உலாவிக்கு கொடுப்பதோடு சரி வழங்கியின் வேலை முடிகிறது. உலாவிதான் உள்ளடக்கங்களை காண்பிக்கும் செயல்களில் ஈடுபடுகிறது.

HTML, CSS, JavaScript, jQuery என பல தொழில்நுட்பங்களைக் கொண்டு இன்றைக்கு இணையதளங்கள் வடிவமைக்கப்படுகின்றன. மேற்காணும் தொழில்நுட்பங்களின் நிரல்வரிகளைத்தான் உலாவிகளால் புரிந்து கொள்ள முடியுமே தவிர. PHP போன்ற நிரல்களை எவ்வாறு காண்பிப்பது என்பது உலாவிக்கு தெரியாது.

ஒரு வலைப்பக்கத்தில் PHP யின் நிரல்கள் இருந்தால், PHP யின் நிரலை உலாவி மறுபடியும் இணைய வழங்கிக்கு அனுப்பி வைக்கும் அந்த நிரல்கள் PHP pre-processing module க்கு அனுப்பி வைக்கப்படும். வலைப்பக்கத்தை வடிவமைத்தவர் என்ன நிரல் எழுதியிருக்கிறாரோ அதற்கான வெளியீட்டை PHP pre-processing module Web Server க்கு அனுப்பி வைக்கும். அதன்பின்பு Web Server ஆனது வலைப்பக்கத்தில் PHP நிரல் இருக்கும் இடத்தில் PHP pre-processing module அனுப்பி வைத்ததை Substitutes செய்யும். அதற்கேற்றாற்போல் உலாவியானது வலைப்பக்கத்தை நமக்கு காண்பிக்கும்.

சுருக்கமாக சொல்ல வேண்டுமானால் ஒரு வலைப்பக்கத்தில் இருக்கும் php நிரல் அந்த உலாவியால் process செய்யப்படாது. php pre-processing module ஆல் process செய்யப்பட்டு அதில் கிடைக்கும் வெளியீட்டைத்தான் உலாவி காண்பிக்கும்.

இதை ஒரு சின்ன உதாரணத்தின் மூலம் காண்போம். கீழ்காணும் நிரலில் <?php ?> எனும் சிறப்புக்குறியீடுகள் இருக்கிறது. இந்த குறியீடுதான் உலாவிக்கு php நிரலை உணர்த்துவதற்கான குறியீடு.

<?php – எனும் குறியீடு php நிரல் ஆரம்பமாவதையும், ?> எனும் குறியீடு php நிரல் முடிவடைவதையும் குறிக்கிறது.

```
 <!–test1.php –>
<!DOCTYPE html>
<html>
<head>
```

```
<title>PHP –
Learning</title>
</head>
<body>
<h1>Hello PHP!</h1>
<?php
echo “Hello World!”;
echo “<br />”;
echo “Hello PHP!”;
?>
</body>
</html>
```

test1.php

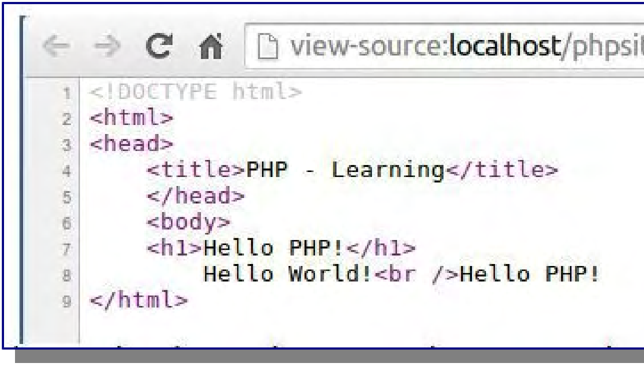
<?php இந்த குறியீட்டிற்கு முன்பு உள்ள வரிகள் அனைத்தும் HTML இன் நிரல் வரிகள். அதன்பின் தொடரும் வரிகள் php நிரல்வரிகள். இந்த வரிகளை web server கண்டுபிடித்து அதற்கான வெளியீட்டை உடனடியாக web browser க்கு அனுப்பி வைக்கிறது. அதை உலாவி நமக்கு காண்பிக்கிறது.

Test1.php நிரலின் வெளியீடு

page source – இன் வெளியீடு

```
<!DOCTYPE html>
<html>
<head>
<title>PHP -
Learning</title>
</head>
<body>
<h1>Hello PHP!</h1>
Hello World!<br />Hello PHP!
```

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



சிவப்பு நிறத்தில் இருப்பவைதான் php *நிரலின் வெயியீடு*.

மேற்காணும் வெளியீட்டை HTML *மூலமாகவே செய்து விடலாமே ஏன் தனியாக* php – *ஐ பயன்படுத்த வேண்டும்*? *என்று இங்கு உங்களுக்கு ஒரு சந்தேகம் எழலாம்*. *இதை ஒரு சின்ன உதாரணத்தின் மூலம் நாம் புரிந்து கொள்ளலாம்*.

வங்கியினுடைய வாடிக்கையாளருக்கு அவர்களுடைய வங்கி எண், பெயர், கணக்கில் இருக்கும் தொகை ஆகியவைகளை காண்பிப்பதற்காக ஒரு வலைப்பக்கத்தை வடிவமைக்கிறோம். அதை HTML இல் வடிவமைத்தால் மற்ற வாடிக்கையாளர்களினுடைய விபரங்களையும் எளிமையாக தெரிந்து கொள்ளலாம். காண்பிக்கும் பக்கத்தின் மீது வைத்து view page source கொடுத்தால் அந்த விபரங்கள் தெரிந்துவிடப் போகிறது. இதை தடுக்கும் விதமாக ஒரு வாடிக்கையாளருக்கு ஒரு பக்கம் என வடிவமைத்தால் அது மலையளவு கஷ்டமான வேலை. ஒரு வங்கியில் 2-லட்சம் வாடிக்கையாளர் இருந்தால் ஒருவருக்கு ஒரு பக்கம் என 2-லட்சம் பக்கங்களை வடிவமைக்க வேண்டும்.

இன்னொன்று என்னவென்றால் HTML வைத்து உருவாக்கும் பக்கங்கள் static ஆக இருக்கும் ஆகையால் ஒரு பக்கத்திற்காக என்ன வடிவமைத்தமோ அதன் content கள் மாறாது.

அதே நேரத்தில் வாடிக்கையாளரின் விபரங்களை தகவல்தளத்தில் சேமித்து வைத்து அந்த விபரங்களை php மூலமாக பெறும் போது ஒரு குறிப்பிட்ட வாடிக்கையாளரைத் தவிர வேறு யாருடைய தகவல்களையும் யாரும் தெரிந்து கொள்ள முடியாது. அதோடு php யினுடைய இறுதி வெளியீட்டைத்தான்

நம்மால் தெரிந்துகொள்ள முடியுமே தவிர அதற்காக உள்ளீடுகளையோ, நிரல்வரிகளையோ தெரிந்து கொள்ள முடியாது. இதனால்தான் PHP பயன்படுத்தப்படுகிறது. இது மாதிரியான பாதுகாப்பு அம்சங்களையும், வசதிகளையும் கொண்டதால்தான் PHP சிறந்து விளங்குகிறது.

இப்ப சொல்லுங்க PHP அவசியம் வேணுமா? வேண்டாமா?

2.5 *PHP* ஏன் மிகவும் பயனுள்ளதாக இருக்கிறது?

- HTML மூலமாக உருவாக்கப்படும் பக்கங்கள் அனைத்தும் Static Page என்று அழைக்கப்படுகிறது. அதாவது வலைப்பக்கத்தின் உள்ளடக்கங்களில் எந்த மாற்றமும் ஏற்படாமல் அப்படியே இருக்கும். JavaScript -ஐக் கொண்டு Dynamic Page -களை வடிவமைக்கலாம். Dynamic Page வடிவமைப்பதற்கான சக்தி மிகுந்த இயந்திரத்தை JavaScript கொண்டிருந்த போதிலும் அதன்மூலமாக Client Side மட்டுமே மாற்றங்களை நிகழ்த்த முடியும்.
- JavaScript -ஐக் கொண்டு Web Server உடன்

தொடர்புகொள்ள முடியாது. Web Browser க்குள் மட்டுமே மாற்றங்களைக் கொண்டு வர முடியும். உதாரணத்திற்குச் சொல்ல வேண்டுமானால் JavaScript – ஆல் Database இல் உள்ள தகவல்களை பிரித்து அதன் வெளியீட்டை Web Page இல் காண்பிக்க முடியாது.

- ஆனால் Database இல் உள்ள தகவல்களை PHP மூலமாக திறமையாக கையாள முடியும். PHP Server Side Scripting Language ஆக இருப்பதோடு, எளிமையாக கற்றுக்கொள்ளும் வகையிலும் இருப்பதால். PHP மிகுந்த பயனுள்ளதாக இருக்கிறது. குறிப்பிட்டுச் சொல்ல வேண்டுமானால் MySQL தகவல்தளத்துடன் PHP சிறப்பாக ஒத்து இயங்குகிறது. MySQL யில் உள்ள தகவல்களை நான் PHP -ஐக் கொண்டு எளிமையாக பெறமுடியும்.

3 LAMP Server -ஐ உபுண்டு 12.04 -ல் நிறுவுதல்

Linux **A**pache **M**ySQL **P**HP - *என்பதன் சுருக்கமே* LAMP *ஆகும்.* LAMP *எனபது மிகவும் பிரபலமானதொரு இணையதள உருவாக்க/வடிவமைப்புச் சூழல்.*

இதில் Linux *என்பது லினக்ஸ் இயங்குதளம் (எந்தவொரு லினக்ஸ் வழங்கலாகவும் இருக்கலாம்),* Apache *என்பது இணைய வழங்கி*(Web Server), MySQL *என்பது* RDBMS *தகவல்தளம்,* PHP *என்பது மாறக்கூடிய இணையப் பக்கங்களுக்கான* (Dynamic Web Page)Scripting Language.

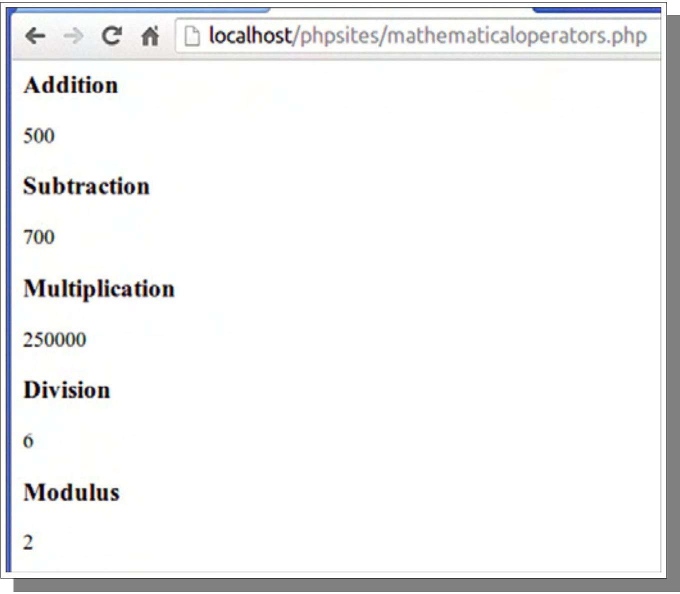
LAMP -*ல் நாம் நிறுவ வேண்டியவை* Apache, MySQL, PHP *ஆகியவைகள் மட்டுமே. நம்மிடம்தான் உபுண்டு இயங்குதளம் இருக்கே. இவையனைத்தையும் தனித்தனியாக நிறுவவேண்டியதில்லை, அப்படி நிறுவ முற்பட்டாலும் அது நமக்கு கொஞ்சம் களைப்பான செயலாகத்தான் அமையும் ஆகையால் இவையனைத்தையும் ஒரே வரிக் கட்டளையில் நிறுவ முடிந்தால் நன்றாக இருக்குமல்லவா!*

ஒற்றை வரி கட்டளையின் மூலம் LAMP Server னை நிறுவக்கூடிய வசதில் உபுண்டு 12.04 LTS - ல் இருக்கிறது. அது எப்படி என பார்ப்போம்.

கணினியில் இணைய இணைப்பு இருக்க வேண்டும் அதுதான் இங்கு முக்கியமாக குறிப்பிட வேண்டியது. ஏன் இதைச் சொல்கிறேன் என்றால் நான் பாலிடெக்னிக் படித்த காலத்தில் LINUX For You Magazine -ல் மென்பொருள்கள் நிறுவக் கொடுத்திருக்கும் கட்டளைகளை அப்படியே முனையத்தில் தட்டச்சு செய்வேன் ஆனால் மென்பொருள் நிறுவப்படாது. காரணம் தெரியவில்லை, இணைய இணைப்பின் மூலம்தான் இந்தக் கட்டளை வேலை செய்யும் என காலப்போக்கில் தான் எனக்கு தெரியும். இப்பபொழுது இருக்கக்கூடிய இணையவசதியெல்லாம் அப்பொழுது இல்லை. இப்பொழுது இணையவசதியினை மிகவும் எளிதாக GPRS மூலம் லினக்ஸிற்குள் கொண்டு வந்து விடலாம். அறியாத வயசுதானே அத விடுங்க.

3.1 முதல் படி:

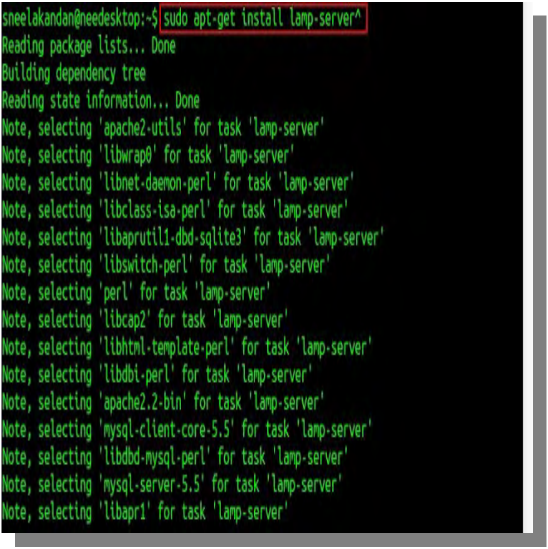
முனையத்தை திறந்து கொள்ளவும், அதில் கீழ்கண்ட கட்டளைகளைக் கொடுக்கவும்.




```
sudo apt-get update
```

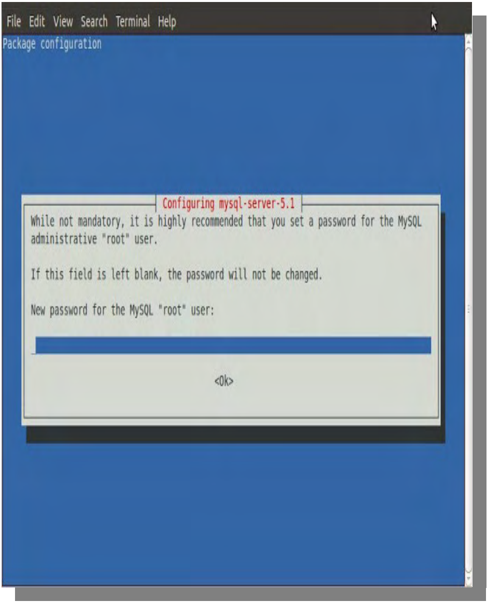
```
sudo apt-get install lamp-server^
```

கவனிக்க: ^ இந்தக் குறியிடு Keyboard -ல் இருக்கும் Number key - 6 -ல் இருப்பது இதைக் கட்டாயம் கொடுக்க வேண்டும்



உங்களின் இணைய இணைப்பின் வேகத்தினைப் பொறுத்து நிறுவுதல் முடியும். நிறுவுதல் முடியும் வரை காத்திருக்கவும். நிறுவுதல் முடியும் தருவாயில் MySQL Database னுடைய root பயனாளருக்கான

கடவுச்சொல்(password) கேட்கும், கடவுச்சொல்லை உள்ளிடவும்.



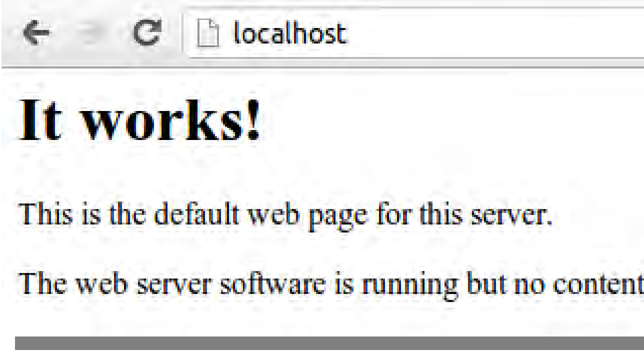
நிறுவுதல் முடிந்தபின் அனைத்தும் சரியாக நிறுவப்பட்டிருக்கிறதா என சோதனை செய்து பார்க்க வேண்டியது அவசியமல்லவா!

3.2 Apache Server -னை சோதனை செய்து பார்க்க:

முனையத்தில் **sudo service apache2 restart** கட்டளையினை இயக்கவும். இந்தக் கட்டளையினைக் கொடுத்தவுடன், Apache Server மறுதொடக்கம் செய்யப்படும்.

உங்கள் கணினியில் இருக்கும் ஏதாவதொரு இணைய உலாவியினைத் திறந்து, முகவரிப் பட்டையில் **http://localhost/** எனக் கொடுத்து

இயக்கவும், இயக்கியவுடன் கீழ்காணுவது உலாவியில் தெரிந்தால் சரியாக இயங்குகிறது என்று அர்த்தம்.



3.3 PHP - யினை சோதனை செய்து பார்க்க:

முதலில் /var அடைவிற்குள் www எனும் பெயருடன் ஒரு அடைவு உருவாக்கப் பட்டிருக்கிறதா என சரிபார்த்துக்

கொள்ளுங்கள். அப்படி உருவாகி இருக்கவில்லையென்றால்

முனையத்தில்,

```
cd /var
sudo mkdir www
```

எனக் கொடுத்து www எனும் அடைவினை உருவாக்கிக் கொள்ளுங்கள்.

அதன்பிறகு, முனையத்தில் கீழ்காணும் கட்டளையினைக் கொடுங்கள்,

```
echo "<?php phpinfo(); ?>"
| sudo tee
/var/www/testing.php
```

முனையத்தில்

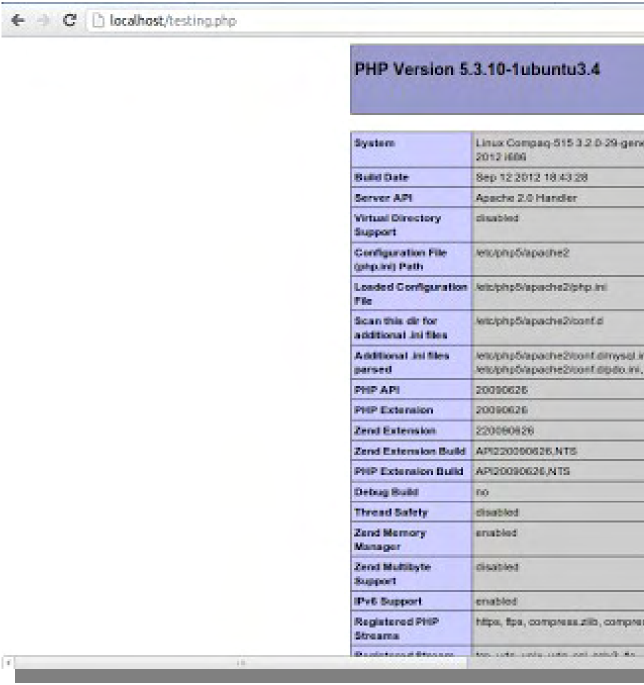
```
sudo service apache2 restart
```

கட்டளையினைக் கொடுத்து ஒருமுறை Apache Server -னை மறுதொடக்கம் செய்து கொள்ளுங்கள்.

அடுத்து, இணைய உலாவி ஏதாவது ஒன்றைத் திறந்து முகவரிப் பட்டையில் கீழ்காணும் முகவரியினைக் கொடுங்கள்,

http://localhost/testing.php

படத்தில் உள்ளது போன்று உங்களுக்கு செய்தி கிடைத்தால் PHP -யும் சரியாக நிறுவப்பட்டிருக்கிறது என முடிவு செய்துக் கொள்ளலாம்.



வெற்றிகரமாக இப்பொழுது LAMP Server *-னை உபுண்டு வில் நிறுவியிருப்போம்.*

4 PHP Script உருவாக்குதல்

4.1 PHP Script உருவாக்குதல்

இதற்கு முந்தைய பகுதிகளில் PHP எப்படி வேலை செய்கிறது என்று பார்த்தோம். இந்த பகுதியில் PHP Script – ஐ எப்படி உருவாக்குவது என்று பார்ப்போம். PHP நிரலை எழுத தொடங்குவதற்கு முன் PHP நிரலை எழு என்னென்னெவெல்லாம் தேவை என்று பார்ப்போம்.

4.2 PHP நிரல் எழுத தேவையானவைகள்:

1. Web Server
2. PHP
3. Browser
4. Text Editor
5. Database

Web Server, PHP, Database ஆகியவைகளை எப்படி நிறுவுவது என்பதைப் பற்றி தெரிந்து கொள்ள இந்த (gnutamil.blogspot.in/2012/12/lamp-server-1204.html)

பக்கத்திற்கு செல்லுங்கள் விபரங்கள் மிகவும் தெளிவாக தமிழில் கொடுக்கப்பட்டுள்ளது. அல்லது PHP யின் அதிகாரப்பூர்வ தளமான php.net/manual/en/install.php -க்குச் செல்லலாம்.

Browser மற்றும் Editor ஆகிய இரண்டும் அனைத்து இயங்குதளங்களிலும் இயல்பாகவே நிறுவப்பட்டு இருக்கும். வேண்டுமானால் நீங்கள் கூடுதலாக Mozilla Firefox, Google Chrome உலாவிகளை நிறுவிக்கொள்ளுங்கள்.

4.3 PHP நிரல் வரம்புச்சுட்டி (Code Delimiters):

ஒரு வலைப்பக்கத்திற்குள் மற்ற நிரல் வரிகளிலிருந்து php நிரல்வரிகளைச் சுட்டிக்காட்டுவதற்காக **<?php** எனும் Opening Tag – ம், **?>** எனும் Closing Tag – ம் பயன்படுத்தப்படுகிறது. இந்த இரண்டு Tag களுக்கு உள்ளே நீங்கள் எவ்வளவு php நிரல் வரிகளை வேண்டுமானாலும் எழுதிக்கொள்ளலாம்.

கீழ்காணும் முறைகளிலும் எழுதிக்கொள்ளலாம்.

முறை 1:

```
<?
//php codings
?>
```

முறை 2:

```
<?php
//php codings
php?>
```

முறை 3:

```
<script language="”php”">
//php codings
</script>
```

முறை 4:

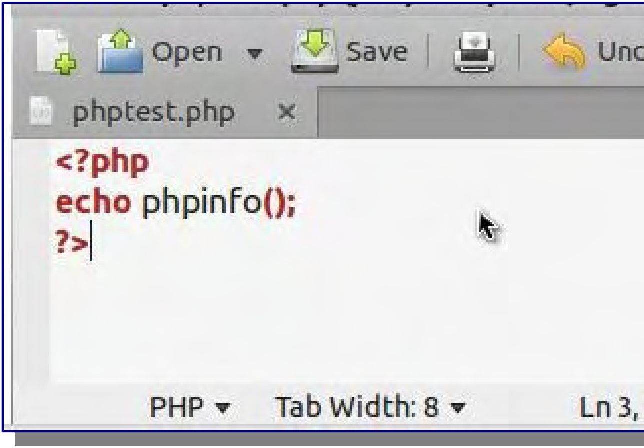
```
<?php
//php codings
?>
```

கடைசியாக இருக்கும் **<?php //php codings ?>** *முறையே பரிந்துரைக்கப்படுகிறது.*

4.4 PHP உங்கள் கணினியில் நிறுவப்பட்டுள்ளதா?

PHP உங்களது கணினியில் நிறுவப்பட்டுள்ளதா என்பதை கீழ்காணும் சின்ன நிரல் மூலம் சோதிக்கலாம்.

Text Editor திறந்து கொள்ளுங்கள்



```
<?php
echo phpinfo();
?>
```

என கொடுத்து phptest.php என்ற பெயருடன் கோப்பை /var/www/ எனும் அடைவிற்குள் சேமியுங்கள். உலாவியைத் திறந்து முகவரிப்பட்டையில் localhost/phptest.php என்று கொடுங்கள் கீழ்காணும் வெளியீடு கிடைத்தால் உங்களது கணினியில் PHP நிறுவப்பட்டுள்ளது என்று அர்த்தம்.

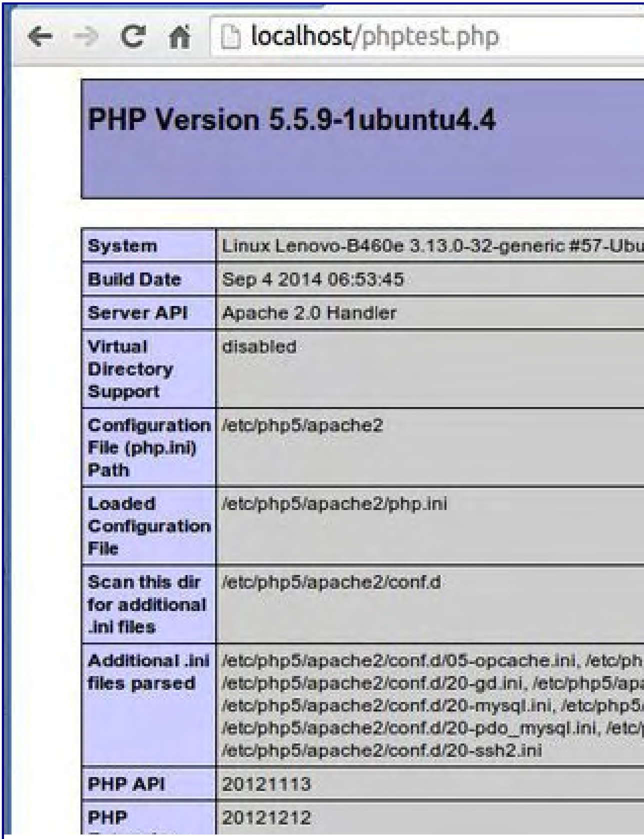
கவனத்திற்கு:

இனிமேல் நாம் பார்க்கப் போகும் அனைத்து நிரல்களையுமே நீங்கள் உங்களுக்கு பிடித்தமான ஏதாவது ஒரு Text Editor – ஐக் கொண்டு எழுதிக்கொள்ளுங்கள். அவ்வாறு எழுதிய கோப்பினை கட்டாயம் நீங்கள் /var/www/ எனும் அடைவிற்குள்தான் சேமிக்க வேண்டும். இதனைத்தவிர்த்து வேறு எந்த அடைவிற்குள் நீங்கள் கோப்பினைச் சேமித்தாலும் அதிலுள்ள PHP நிரல் மட்டும் வேலை செய்யாது. அதே சமயத்தில் கோப்பிற்குள் இருக்கும் HTML நிரலுக்கான வெளியீடு உங்களுக்கு கிடைக்கும்.

இது DocumentRoot எனப்படும்.

/var/www என்பது டெபியன், உபுண்டுவிற்கு.
/var/www/html ஃபெடோரா, ரெட்ஹாட்டிற்கு

இதை Apache configuration file மூலம் மாற்றலாம்


localhost/phptest.php

PHP Version 5.5.9-1ubuntu4.4

System	Linux Lenovo-B460e 3.13.0-32-generic #57-Ubu
Build Date	Sep 4 2014 06:53:45
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/ph /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apa /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/ /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/ /etc/php5/apache2/conf.d/20-ssh2.ini
PHP API	20121113
PHP	20121212

4.5 *PHP நிரல் பொதிதல் முறைகள்:*

1. HTML File -க்குள் PHP *நிரலை பொதிதல்*
2. PHP File – *க்குள்* HTML *நிரலை பொதிதல்*

HTML File க்குள் PHP நிரலை பொதிதல்(PHP into an HTML File) :

PHP *நிறுவப்பட்டுள்ளதா எனும் சோதனை செய்வதற்காக ஒரு* PHP *நிரலை எழுதிய அனுபவம் உங்களுக்கு ஏற்கனவே இருக்கிறது. இப்பொழு நாம் மற்றொரு* PHP *நிரலை எழுதப் போகிறோம். நாம் எற்கனவே எழுதிய நிரலில் எந்தவிதமான* HTML Tag *குகளையும்*

பயன்படுத்தவில்லை. ஆனால் இப்பொழுது நாம் ஒரு HTML File – க்குள் PHP நிரலை எழுத்தப்போகிறோம்.

உங்களுடைய Editor – ஐத் திறந்து கீழ்காணும் HTML File ஐ உருவாக்குங்கள்

```
<!DOCTYPE html>
<html>
<head>
<?php
echo “<title>PHP in
Tamil</title>”;
?>
</head>
 <body>
<?php
echo “<p>This content was
generated by PHP</p>”;
?>
</body>
</html>
```

firstscript.php என்ற பெயருடன் கோப்பினை சேமியுங்கள். உலாவியைத் திறந்து உங்களது நிரலை

இயக்கிப் பார்த்தால் கீழ்காணும் வெளியீடு கிடைக்கும்.

<u>கவனிக்கவும்:</u>

PHP *கோப்புகள்* .php *எனும்* file extentsion *உடன் இருக்கும். நிரல்வரிகள் அனைத்தும்* ; (semicolon) *உடன் முடியும். இந்த* semicolon PHP *யின் ஒரு நிரல் வரி முடிவுறுவதை உணர்த்துவதற்கு* (line separator) *பயன்படுகிறது.*

PHP நிரலுக்குள் HTML நிரலை பொதிதல் (Embedding HTML into a PHP Script):

முந்தைய உதாரணத்தில் ஒரு HTML *பக்கத்திற்குள்* PHP *நிரலை எழுதுவதைப் பற்றி பார்த்தோம். இப்பொழுது*

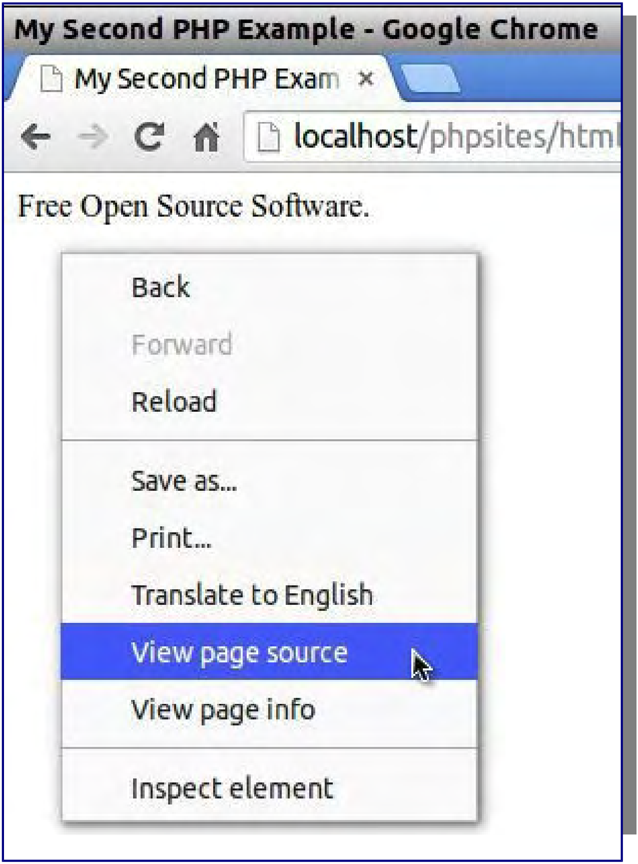
அதை அப்படியே தலைகீழாக செய்யப்போகிறோம். PHP நிரலுக்குள் HTML -ஐ எழுதப்போகிறோம்.

கீழ்காணும் நிரலை எழுதுங்கள்.

```
<?php
echo “<html>\n”;
echo “<head>\n”;
echo “<title>My Second PHP
Example</title>\n”;
echo “</head>\n”;
echo “<body>\n”;
echo “<p>Free Open Source
Software.</p>\n”;
echo “</body>\n”;
echo “</html>\n”;
?>
```

htmlintophp.php எனும் பெயருடன் சேமியுங்கள். உலாவியில் இயக்குங்கள் கீழ்காணுவது போல் உங்களுக்கு வெளியீடு கிடைக்கும்.

View Page Source கொடுத்து பார்த்தீர்களேயானால் கீழ்காணுவது போல உங்களுக்கு வெளியீடு கிடைக்கும்.


My Second PHP Example - Google Chrome
My Second PHP Exam
localhost/phpsites/html
Free Open Source Software.
Back
Forward
Reload
Save as...
Print...
Translate to English
View page source
View page info
Inspect element

```
<html>
<head>
<title>My Second PHP
Example</title>
</head>
<body>
<p>Free Open Source
Software.</p>
</body>
</html>
```

ஏன் இப்படி தெரிகிறது. *காரணம்* PHP pre-processor *இதைப் பற்றி நாம் ஏற்கனவே பார்த்திருக்கிறோம்*. *ஞாபகம் வருகிறதா*? *வந்தால் மகிழ்ச்சி*!

5 Comments - குறிப்புரைகள்

அனைத்து கணினி நிரல் மொழிகளிலுமே குறிப்புரை (comments) வசதி இருக்கிறது. இந்த குறிப்புரை -இல் எழுத்தப்படும் வரிகள் நிரலின் பகுதியாக கருதப்படாது. அதாவது comment இல் எழுதப்படும் வரிகள் நிரல் வரிகளாக கருத்தில் கொண்டு படிக்கவோ/இயக்கவோ

பட மாட்டாது. நிரலை எழுதியவரைத் தவிர மற்றவர்கள் அந்த நிரலைப் பார்வையிடும் போது இந்த குறிப்புரை பயன்படுகிறது அவ்வளவுதான்.

PHP -ஐப் பொறுத்தமட்டிலே இந்த குறிப்புரை வரிகள் PHP pre-processor ஆல் புறக்கணிக்கப்படும். முழுக்கமுழுக்க மனிதர்கள் புரிந்து கொள்வதற்காக மட்டுமே இந்த குறிப்புரை.

கீழ்கண்ட வழிகளில் குறிப்புரை பயன்படலாம்:

1. நிரல் வரிகள் என்ன காரணத்திற்காக எழுத்தப்பட்டுள்ளது என்பதை உணர்த்தும் குறிப்பாக பயன்படலாம்.
2. குறிப்பிட்ட காலம் கழித்து நீங்கள் எழுதிய நிரலை, நீங்களே பார்வையிடும் போது அதை எதற்காக எழுதினீர்கள் என்பது மறந்து போகலாம். அந்த சமயத்தில் இந்த குறிப்புரை உங்களுக்கு மிகவும் உதவியாக இருக்கும்.
3. நீங்கள் குறிப்புரை ஐக் கொண்டு நிரல்களை உருவாக்கினால் அந்த நிரலை மற்றவர்கள் எளிதில் புரிந்து கொள்வார்கள்.
4. ஏதோ ஒரு காரணத்திற்காக நீங்கள் எழுதிய நிரலை பாதியோடு விட்டு விட்டுச்

செல்கிறீர்கள், அந்த நேரத்தில் உங்கள் நிறுவனத்தில் பணிபுரியும் வேறொரு பணியாளர் அந்த நிரலை தொடர்ந்து எழுத முற்படும் போது நீங்கள் கொடுத்த குறிப்புரை அவருக்கு மிகப்பெரிய உதவியாகவும். நிரலை விரைவாக எழுதவும் உதவும்.

5. ஒரு நல்ல நிரலாளர் என்பவர் குறிப்புரை இல்லாமல் நிரல் எழுதமாட்டார். குறிப்புரை கொண்டு நிரல் எழுதுவதே ஒரு நல்ல நிரலாளருக்கு அழுகு.

PHP யில் இரண்டு விதமான குறிப்புரைகள் இருக்கின்றன.

1. Single line குறிப்புரை (ஒற்றை வரி குறிப்புரை)
2. Multi line குறிப்புரை (பலவரி குறிப்புரை)

PHP யினுடைய குறிப்புரை C, C++ and Java நிரல்களின் குறிப்புரை முறைகளை ஒத்தே இருக்கிறது. இந்த மொழிகளில் ஏற்கனவே பரிச்சயம் உள்ளவர்களுக்கு PHP குறிப்புரை வியப்பாகத் தோன்றாது.

5.1 ஒற்றைவரி குறிப்புரை:

இரண்டு முன்னோக்கிய சாய்வுகளைக் கொண்டு இருக்கும்.

//This is single line comment.

உதாரணம்:

```
<?php
//php coding start here
echo “Hello PHP!”;
    //echo is used to print
the statement
//php coding end here
?>
```

ஒற்றைவரி குறிப்புரை ஒரு புதிய வரியாகவும் இருக்கலாம் அல்லது ஒரு நிரல் வரியினுடைய இறுதியில் இருந்தும் தொடங்கலாம். தற்காலிகமாக ஒரு நிரல்வரியை நீக்குவதற்கு ஒற்றைவரி குறிப்புரை பயன்படும்.

உதாரணம் :

```
<?php
//echo “Hello PHP!”;
echo “Hello Linux!”;
?>
```

5.2 பலவரி குறிப்புரை:

பலவரி குறிப்புரை / மற்றும் */ ஆகிய இரண்டு குறியீட்டிற்குள் இருக்கும். /* குறியீடு குறிப்புரையின் ஆரம்பத்தையும், */ குறியீடு குறிப்புரையின் முடிவையும் உணர்த்துகிறது.*

கீழ்காணும் உதாரணத்தை பாருங்கள்.

```
<?php
/* This is a
```

```
multiline
comments
*/
echo “Hello PHP”;
?>
```

ஒரு வரிகளுக்கு மேலாக குறிப்புரை எழுத வேண்டிய அவசியம் உள்ள போது பலவரி குறிப்புரை உதவியாக இருக்கும். ஒன்றிற்கு மேற்பட்ட நிரல்வரிகளை தற்காலிகமாக நிறுத்தி வைக்கவும். பலவரி குறிப்புரை உதவும்.

6 மாறிகள் (*Variables*)

PHP *மாறிகள்* (Variables)

variable *என்பதற்கு தமிழில் மாறி என்று அர்த்தம். தகவல்களோடு நாம் வேலை செய்யும் போது அத்தகைய தகவல்களை சேமித்து வைப்பதற்கு வசதியான ஒரு வழி வேண்டும். அத்தகைய வசதியான ஒரு வழிதான் மாறிகள். மாறிகள் மதிப்புகளைக் கொண்டிருக்கும். நிரல்கள் இயங்கும் போது மாறிகள் கொண்டிருக்கும் மதிப்புகள் மாறலாம்.*

6.1 மாறிகளுக்கு பெயர் வைத்தல்*:*

மாறிகளை உருவாக்குவதற்கு முன்பு மாறிகளுக்கு எப்படி பெயரிட வேண்டும் என்பதை பார்த்துவிடுவது அவசியமானது.

அனைத்து PHP மாறிகளும் $ குறியீட்டைக் முன்னொட்டாக கொண்டே தொடங்கும்.

இந்த $ *முன்னொட்டு அதைத் தொடர்ந்து வருவது ஒரு மாறி என்பதை* PHP pre-processor *க்கு தெரிவிக்கும்.*

மாறியின் முதல் எழுத்து கட்டாயமாக ஒரு எழுத்தை கொண்டோ அல்லது _ (underscore) *கொண்டுதான் தொடங்க வேண்டும்.*

முதல் எழுத்தைத் தொடர்ந்து வரும் எழுத்துக்கள் எண்ணாகவோ, எழுத்தாகவோ அல்லது _ (underscore) *ஆகவோ இருக்கலாம்.*

மற்ற எதைக்கொண்டு மாறிக்கு பெயர் வைத்தாலும் அது பிழையாக கருதப்படும்.

கவனிக்கவும்:

PHP *ஒரு* case sensitive scripting language *ஆகையால் சிறிய எழுத்து மற்றும் பெரிய எழுத்துக்கள் கவனத்தில் எடுத்துக்கொள்ளப்படும்.*

$myName என்பதும் $myname என்பதும் வெவ்வேறானவை. இரண்டும் ஒன்றல்ல.

சரியான முறை பெயரிடல்:

$_myName
$myName
$__myName
$myVar12

தவறான முறை பெயரிடல்:

$_1myName – underscore க்கு அடுத்து எழுத்துதான் வர வேண்டும்.
$1myName – முதல் எழுத்து எழுத்தாகத்தான் இருக்க வேண்டும், எண்ணாக இருக்கக் கூடாது.
$my-Name – எண், எழுத்து, underscore ஐத் தவிர மற்ற எதையும் பயன்படுத்தக்கூடாது.

6.2 மாறிகளுக்கு மதிப்புகளை அளித்தல்:

assignment operator -ஐப் பயன்படுத்தி மாறிகளுக்கு மதிப்புகள் அளிக்கப்படுகிறது. Assignment operator என்பது = (சமம்) குறியீடு ஆகும். மாறிகளுக்கு மதிப்புகள் கொடுக்கும் போது மாறிகள் இடதுபுறமாகவும் அதன்பின் = குறியீடும் அதனைத் தொடர்ந்து மாறிக்காக மதிப்பும் இருக்க வேண்டும். இறுதியாக ; (semicolon) உடன் முடிய வேண்டும்.

```
$myName = “stallman”;
$foss = “Free Open Source Software”;
$examNumber = 1002;
$cyclePrice = 1500.36;
```

இங்கு

$myName என்ற மாறிக்கு stallman என்ற மதிப்பும்
$foss என்ற மாறிக்கு Free Open Source Software என்ற மதிப்பும்
$examNumber என்ற மாறிக்கு 1002 என்ற மதிப்பும்
$cyclePrice என்ற மாறிக்கு 1500.36 என்ற மதிப்பும் கொடுக்கப்பட்டுள்ளது.

6.3 மாறிகளின் மதிப்புகளை அணுகுதல்:

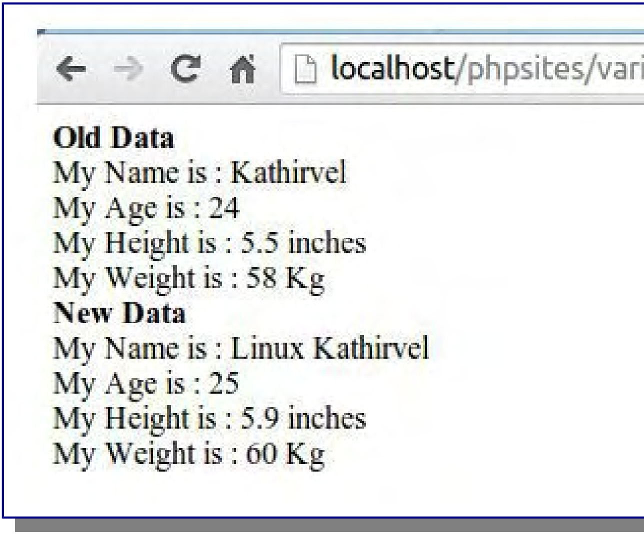
இதுவரை மாறிகளை உருவாக்குவது, பெயரிடுவது, மதிப்புகள் கொடுப்பது பற்றி பார்த்தோம். இப்போது மாறகளுக்கு கொடுக்கப்பட்டிருக்கும் மதிப்புகளை எப்படி அணுகுவது என்று பார்ப்போம். மாறிகளின் மதிப்புகளை அணுகுவது என்பது மிகவும் எளிமையான ஒன்று. உங்களுக்கு எந்த இடத்தில் மாறியின் மதிப்பு வேண்டுமோ அந்த இடத்தில் மாறியினுடைய பெயரை இட்டால் போதும். அந்த இடத்தில் மாறியின் மதிப்பு அளிக்கப்படும்.

கீழ்காணும் நிரலை இயக்கிப் பார்த்தால் இந்த கருத்தாக்கங்கள் நன்றாக புரியும்.

```
<?php
$myName = “Kathirvel”;
$myAge = 24;
$myHeight = 5.5;
$myWeight = 58;
echo “<b>Old Data</b>”;
echo “<br>”;
echo “My Name is : $myName”;
echo “<br>”;
echo “My Age is : $myAge”;
```

```
echo “<br>”;
echo “My Height is : ” .
$myHeight . ” inches”;
echo “<br>”;
echo “My Weight is : ” .
$myWeight . ” Kg”;
echo “<br>”;
$myName = “Linux Kathirvel”;
$myAge = 25;
$myHeight = 5.9;
$myWeight = 60;
echo “<b>New Data</b>”;
echo “<br>”;
echo “My Name is : $myName”;
echo “<br>”;
echo “My Age is : $myAge”;
echo “<br>”;
echo “My Height is : ” .
$myHeight . ” inches”;
echo “<br>”;
echo “My Weight is : ” .
$myWeight . ” Kg”;
?>
```

மேற்காணும் நிரலை variables.php எனும் பெயருடன் சேமிக்கவும். மேற்காணும் நிரலை இயக்கும் போது கீழ்காணும் வெளியீடு கிடைக்கும்.



இங்கு . (dot) ஆனது concatenation character ஆகும். அதாவது இரண்டு statement களை இணைப்பது. மேற்காணும் நிரலில் முதல்பகுதியில் பயன்படுத்தப்பட்ட அதே மாறிகள் புதிய மதிப்புகளுடன் பயன்படுத்தப்பட்டு உள்ளது. அப்படியென்றால் என்ன அர்த்தம். மாறிகளின் மதிப்பு நிலையானது அல்ல. மாறக்கூடியது.

6.4 PHP மாறியினுடைய வகையை மாற்றுதல்:

PHP பல்வேறு வகையான Data Type களை ஆதரிக்கிறது. குறிப்பாக integer, float, boolean, array, object, resource and string. இவையனைத்தையும் பற்றி நாம் பின்பு விரிவாக பார்க்க இருக்கிறோம். இப்போது மாறியினுடைய வகையை மாற்றுவது பற்றி பார்ப்போம்.

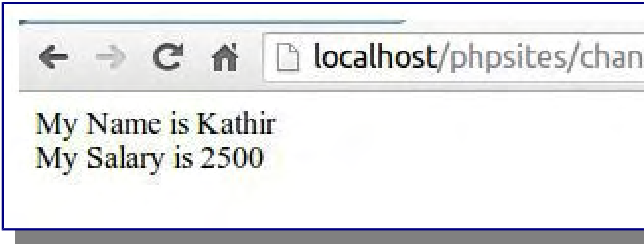
PHP ஒரு Loosly typed language JavaScript ஐப் போல. Loosly types language என்றால் ஒரு குறிப்பிட்ட data type இல் இருக்கும் variable ஐ வேறொரு data type க்கு மாற்றிக்கொள்ளலாம். interger லிருந்து float க்கு, float லிருந்து integer க்கு என மாற்றிக்கொள்ளலாம்.

Java, C, C++ போன்ற மொழிகள் **Strongly Typed Languages**. இந்த மொழிகளில் ஒரு குறிப்பிட்ட வகை data type லிருந்து வேறொரு வகை data type ற்கு மாற்றிக்கொள்ள முடியாது.

கீழ்காணும் நிரலைப் பாருங்கள்:

```
<?php

$myName = “Kathir”;
echo “My Name is $myName”;

$myName = 2500;
echo “<br>”;

echo “My Salary is $myName”;

?>
```

output:



$myName என்பது முதலில் String Data Type ஆகவும், பின்பு Integer Data type ஆகவும் தானாகவே மாறியுள்ளது.

6.5 மாறி மதிப்புகளை வைத்திருக்கிறதா என சோதித்தல்(*Check Whether a variable is set*):

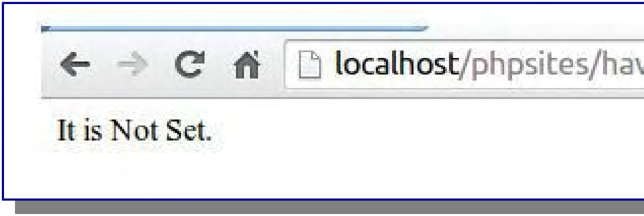
மாறிகளுடன் நாம் வேலை செய்யும் போது மாறிகள் மதிப்புகளை வைத்திருக்கிறதா என்று அடிக்கடி சோதனை செய்து பார்க்க வேண்டியிருக்கும். இதற்காகவே PHP வழங்கியிருக்கும் function தான் isset(). Isset() function ஐப் பயன்படுத்தி மாறி மதிப்புகளை வைத்திருக்கிறதா இல்லையா என்பதை எளிமையாக கண்டுபிடித்துவிடலாம். Function பகுதியில் இதைப் பற்றி விரிவாக தெரிந்து கொள்ள இருக்கிறோம்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$myName;
if (isset($myName)) {
```

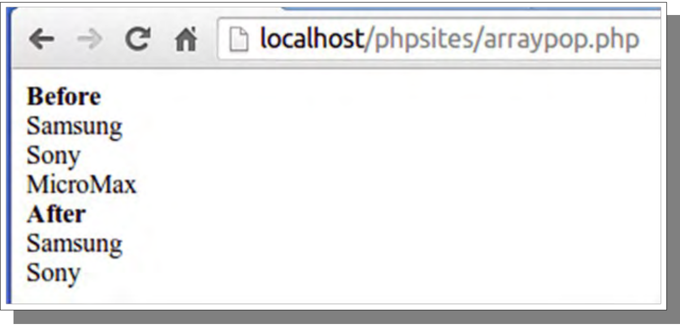
```
echo “It is Set.<br>”;}

else {
echo “It is Not Set.”;}

?>
```

வெளியீடு:



7 PHP மாறி வகைகள்:

இந்த பகுதியில் Integer, String, Float, Boolean மாறி வகைகளை காண இருக்கிறோம். Array, Object பின்வரும் பகுதியில் விரிவாக காணலாம்.



7.1 முழு எண் மாறி வகை (Integer Variable Type):

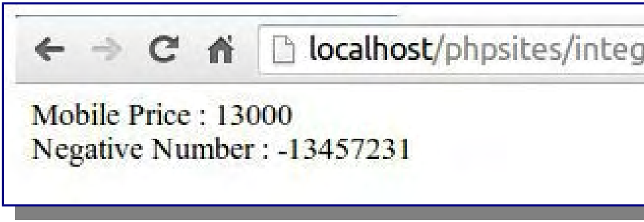
முழுஎண் மாறிகள் -2147483648 லிருந்து 2147483647 வரையில் உள்ள முழு எண்களை கொண்டிருக்கும். எதிர்முழு எண்கள் கழித்தல் (-) குறியை எண்ணிற்கு முன்னதாக கொண்டிருக்கும். மேற்காணும் மதிப்புகளை தாண்டும் போது இயல்பாகவே அது மிதவை (Float

Point) வகைக்கு மாற்றப்படும்.

கீழ்காணும் உதாரண நிரலைப் பாருங்கள்:

```
<?php
$mobilePrice = 13000;
$myNegative = -13457231;
echo "Mobile Price :
$mobilePrice";
echo "<br>";
echo "Negative Number :
$myNegative";
?>
```

வெளியீடு:

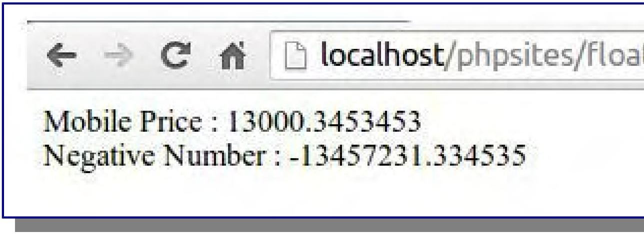


7.2 மிதவை எண் மாறி வகை (*Float Variable Type*):

தசம எண்களே மிதவை எண்கள். உதாரணமாக 1.067, 0.25, 423454567098, 84664435.9576
கீழ்காணும் நிரலை பாருங்கள்

```
<?php

$mobilePrice =
13000.3453453;
$myNegative =
-13457231.3345354;

echo “Mobile Price :
$mobilePrice”;
echo “<br>”;
echo “Negative Number :
$myNegative”;

?>
```

வெளியீடு:



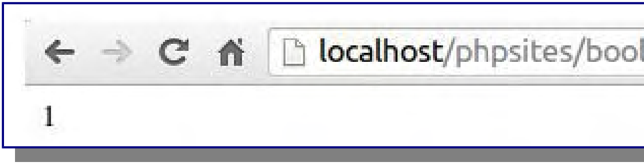
7.3 பூலியன் வகை மாறி (Boolean Variable Type):

பூலியன் வகை மாறிகள் true அல்லது false ஆகிய இரண்டு மதிப்புகளை மட்டும் கொண்டிருக்கும். Flow control and Looping இல் பூலியன் வகை மாறிகளைப் பற்றி விரிவாக காண்போம். குறிப்பாக if -ஐப் பற்றி பார்க்கும் போது காணலாம். வெளிப்படையாக நாம் true or false என்று சொன்னாலும். PHP நிரலுக்குள் அது 0 or 1 என்றுதான் எடுத்துக்கொள்ளும்.

கீழ்காணும் நிரலை காணுங்கள்

```
<?php
$myName = "Stallman";
echo isset($myName);
?>
```

வெளியீடு



isset($myName) என்பது true ஆக இருப்பதால், அதனுடைய 1 என்று வந்திருக்கிறது.

7.4 சர மாறி வகை (String Variable Type):

வார்த்தைகள் மற்றும் வாக்கியங்களை (words and sentences) வைத்துக் கொள்ள சர மாறி வகைகள் பயன்படுகிறது. சர மாறியிலிருந்து தேவைப்படும் போது அதனுடைய பகுதிகளை பிரித்தெடுக்க PHP

அனுமதிக்கிறது.

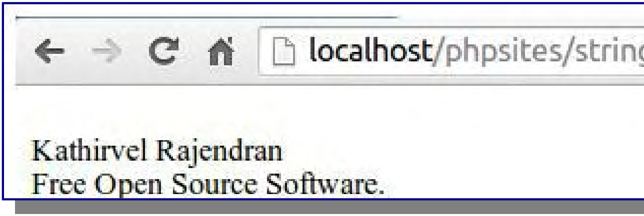
சர மாறி வகைக்கான மதிப்புகள் (‘) Single Quotes அல்லது (“) Double Quotes க் கொண்டு இருக்கும்.

கீழ்காணும் நிரலை பாருங்கள்

```
<?php

$myName = ‘Kathirvel
Rajendran’;
$foss = “Free Open Source
Software.”;
$string1 = “This string
contains ‘single quotes'”;
$string2 = ‘This string
contains “double quotes”‘;

echo “<br>”;
echo $myName;
echo “<br>”;
echo $foss;
echo “<br>”;
echo $string1;
echo “<br>”;
echo $string2;
```

```
?>
```

வெளியீடு

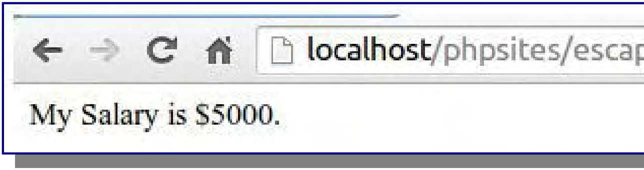


backslash (\) *பின்னோக்கிய சாய்விற்கு பின் வருவது* escape character *எனப்படும். உதாரணமாக நீங்கள்* $ *குறியீட்டை வெளியீடாக பயன்படுத்த வேண்டுமெனில் அதற்கு பின்னோக்கிய சாய்வைதான் பயன்படுத்த வேண்டும்.*

கீழ்காணும் நிரலைப் பாருங்கள்:

```
<?php
echo “My Salary is \$5000.”;
?>
```

நிரலின் வெளியீடு



சில குறிப்பிடத்தகுந்த Escape Sequences.

\n – New Line
\r – Carriage Return
\t – Tab
\\ – Backslash Character
\” – Double Quotation Mark
\$ – Dollar sign (prevents text from being treated as a variable name)
\034 – Octal ASCII value
\xOC – Hexadecimal ASCII Value

மாறிலி (Constants)

அடிக்கடி பயன்படுத்தக்கூடிய, மாறாத மதிப்புகளுக்கு நீங்கள் மாறிலியைப் பயன்படுத்திக் கொள்ளலாம். உதாரணமாக, வருடத்தின் நாட்கள், பூமியின் விட்டம், 1000 மி.லி = 1 லிட்டர், கணிதத்தில் பயன்படுத்தும் பை போன்றவைகளைக் கூறலாம். என்றைக்கும் இவைகளின் மதிப்பு மாறாமல் அப்படியே இருக்கும்.

மாறிலிகள் (constants) global scope -ஐக் கொண்டது. Global scope என்பதின் அர்த்தம் என்னவென்றால் global scope -இல் இருக்கும் மதிப்புகளை, உங்களுடைய நிரலின் function, object மற்றும் எந்த இடத்திலும் பயன்படுத்திக் கொள்ளலாம்.

நமது நிரல் எழுதும் வேலைகளை எளிதாக்குவதற்காக நிறைய உள்ளமைக்கப்பட்ட (built-in) constants களைக் PHP கொண்டுள்ளது.

கவனிக்க: *மாறி*(variable) *என்பதும் மாறிலி*(constants) *என்பதும் ஒன்றல்ல. இரண்டும் வேறு வேறானவை.*

8.1 மாறிலியை வரையறுத்தல் *(Defining a Constant):*

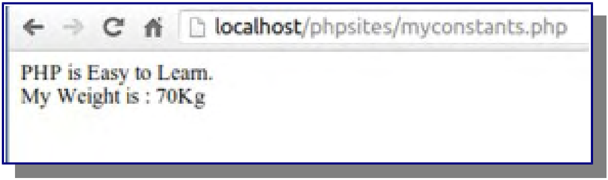
மாறியை (variable) *வரையறுப்பதற்கு நாம் மாறியினுடைய பெயருக்கு முன்பு* $ *குறியீட்டைப் பயன்படுத்துவோம். ஆனால், மாறிலியை*(constants) *வரையறுப்பதற்கு* define() function *ஐப் பயன்படுத்த வேண்டும். மாறிலியை வரையறுக்க* $ *முன்னொட்டு தேவையில்லை.* define function *இரண்டு* arguments *களைக் கொண்டிருக்கும். அதில் ஒன்று* constant *-இன் பெயராகவும், மற்றொன்று* constant *-இன் மதிப்பாகவும் இருக்கும்.*

மாறிலியின் பெயர்கள் case sensitive *கொண்டது. ஆனாலும் இது ஒரு பிரச்சனையாக இருக்கப் போவதில்லை காரணம் என்னவென்றால். வசதிக்காக, மற்ற கணினி நிரல் மொழிகளில் உள்ளதைப் போன்றே மாறிலியின் பெயர்கள் பெரிய எழுத்துக்களைக்* (upper case letters) *கொண்டே பெயரிட்டப்படுகிறது.*

கீழ்காணும் நிரலைப் பார்த்தால் உங்ளுக்கு நன்றாக புரியும்.

```
<?php
define('WELCOME_MESSAGE',
"PHP is Easy to Learn.");
define('MY_WEIGHT',70);

echo WELCOME_MESSAGE;
echo "<br>";
echo "My Weight is :
".MY_WEIGHT."Kg";
?>
```

நிரலின் வெளியீடு கீழ்காண்பது போன்று இருக்கும்



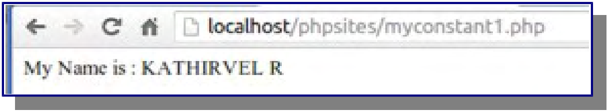
constant (மாறிலி) இன் மதிப்புகளை நிரலில் எங்கு வேண்டுமானாலும் பயன்படுத்திக் கொள்ளலாம். மாறிலிக்கு பெயரிடும்போது எப்படி $ குறியீடு இல்லாமல் பெயரிடுகிறமோ அதே போல அதன் மதிப்பை பயன்படுத்தவும் $ குறியீடு தேவையில்லை. நேரடியாக பெயரைக் குறிப்பிட்டு மதிப்புகளைப் பயன்படுத்திக் கொள்ளலாம்.

மாறிலி வரையறுக்கப்பட்டுள்ளதா என சோதித்தல்
defined () function ஐப் பயன்படுத்தி மாறிலி வரையறுக்கப்பட்டுள்ளதா இல்லையா என்பதைக் கண்டுபிடிக்கலாம். define() function மாறிலியின் பெயரை argument ஆக எடுத்துக் கொள்கிறது. மாறிலி வரையறுக்கப்பட்டிருந்தால் true எனவும் வரையறுக்கப்படவில்லை எனில் false எனவும் வெளியீட்டைக் கொடுக்கும். உதாரணமாக MY_NAME எனும் மாறிலி வரையறுக்கப்பட்டுள்ள என்பதை சோதிக்க வேண்டுமெனில் defined() function – ஐப் பயன்படுத்தி அதை எளிமையாக செய்யலாம்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
define('MY_NAME','KATHIRVEL
R');
if (defined('MY_NAME')) {
echo "My Name is :
".MY_NAME;
echo "<br>";
}
else {
echo "What is your name?";
echo "<br>";
}
?>
```

நிரலுக்கான வெளியீடு



8.2 முன் வரையறுக்கப்பட்ட மாறிலிகள் (*Predefined Constants*):

Web Developer இன் வேலையை எளிமைப்படுத்துவதற்காக PHP நிறைய உள்ளமைக்கப்பட்ட (built-in) மாறிலிகளை தன்னகத்தே கொண்டுள்ளது என்று இந்த பகுதியின் தொடக்கத்திலேயே பார்த்தோம். அதைப் பற்றி விரிவாக இந்த பகுதியில் பார்க்கலாம். குறிப்பாக அடிக்கடி பயன்படுத்தப்படும் சில பயனுள்ள முன் வரையறுக்கப்பட்ட மாறிலிகளைப் பற்றி பார்ப்போம்.

Script மற்றும் சூழல் தொடர்பான மாறிலிகள் (Script and Environment Related Constants):

PHP நிரல் இயங்கக்கூடிய இணைய வழங்கிகள், நிரல் இயங்கக்கூடிய கணினி (Client) மற்றும் நிரலைப் பற்றிய விபரங்களை வழங்குவதற்காக PHP இல் நிறைய மாறிலிகள் உள்ளன. கீழ்காணும் மாறிலிகள் அடிக்கடி பயன்படுத்தப்படுபவை.

மாறிலியின் பெயர் ***விளக்கம்***

__LINE__ நிரலில் மொத்தம் எத்தனை நிரல் வரிகள் இருக்கிறது என்ற விபரத்தை அளிக்கிறது.
__FILE__ நிரல் இருக்கும் கோப்பின் விபரங்களை அளிக்கிறது.
__FUNCTION__ தற்போது செயல்படுத்தப்பட்டு கொண்டிருக்கும் செயல்கூறைப் (function) பற்றிய விபரங்களை அளிக்கிறது.
__CLASS__ தற்பொழுது பயன்பாட்டில் இருந்து கொண்டிருக்கும் class பற்றிய விபரங்களை அளிக்கிறது.
__METHOD__ தற்பொழுது இயக்கத்தில் இருக்கும் class இல் இருக்கக்கூடிய method பற்றிய விபரங்களை அளிக்கிறது.
PHP_VERSION PHP யினுடைய பதிப்பு(Version) விபரங்களை அளிக்கிறது.
PHP_OS PHP pre-processor இருக்கும் இயங்குதளத்தின் விபரங்களை அளிக்கிறது.
PHP_EOL புதிய வரிக்கான உருவைக்(character) கொண்டிருக்கிறது.
DEFAULT_INCLUDE_PATH include கோப்புகளுக்காக PHP பார்வையிடும் கொடாநிலை(default) பாதையை அளிக்கிறது.

கணித மாறிலிகள்(Mathematical Constants):

நிரல் எழுதும் நேரத்தையும், கணக்கீடுகள் செய்யும் நேரத்தையும் சேமிப்பதற்காக பயனுள்ள பல கணித மாறிலிகளைக் PHP கொண்டுள்ளது. கீழ்காணும் அட்டவணை கணித மாறிலிகளை வரிசைப்படுத்தப்பட்டுள்ளது.

***மாறிலி*(Constant)**	***விளக்கம்***
M_E	e – *னுடைய மதிப்பு*
M_EULER	Euler's *மாறிலியின் மதிப்பு*
M_LNPI	PI – *யின் மடக்கை மதிப்பு*
M_LN2	2 -*ன் மடக்கை மதிப்பு*
M_LN10	10 -*ன் மடக்கை மதிப்பு*
M_LOG2E	*அடிமானம்* 2 *உடைய மடக்கையில்* E -*னுடைய மதிப்பு*
M_LOG10E	*அடிமானம* 10 *உடைய மடக்கையில்* E – *னுடைய மதிப்பு*
M_PI	PI *யின் மதிப்பு*
M_PI_2	PI/2 *வின் மதிப்பு*
M_PI_4	PI/4 -*இன் மதிப்பு*
M_1_PI	1/PI – *இன் மதிப்பு*
M_2_PI	2/PI – *இன் மதிப்பு*

M_SQRTPI	PI *யின் வர்க்கமூலம்*
M_2_SQRTPI	2/PI *யின் வர்க்கமூலம்*
M_SQRT2	2 – *இன் வர்க்கமூலம்*
M_SQRT3	3 – *இன் வர்க்கமூலம்*
M_SQRT1_2	1/2 – *இன் வர்க்கமூலம்*

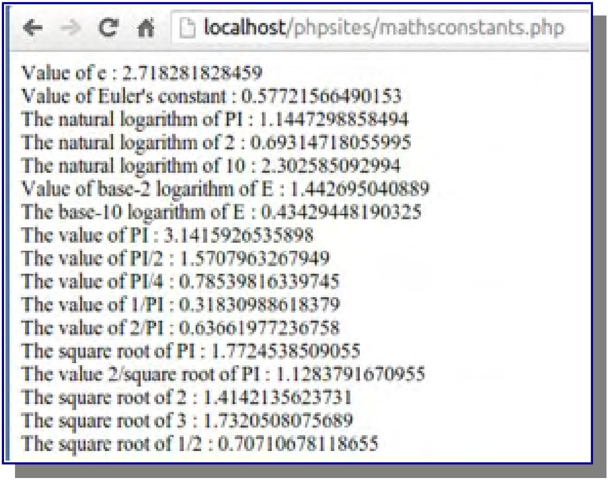
கீழ்காணும் நிரலைப் பாருங்கள்:

```
<?php
echo “Value of e : “.M_E;
echo “<br>”;
echo “Value of Euler’s
constant : “.M_EULER;
echo “<br>”;
echo “The natural logarithm
of PI : “.M_LNPI;
echo “<br>”;
echo “The natural logarithm
of 2 : “.M_LN2;
echo “<br>”;
echo “The natural logarithm
of 10 : “.M_LN10;
echo “<br>”;
echo “Value of base-2
logarithm of E : “.M_LOG2E;
echo “<br>”;
echo “The base-10 logarithm
```

```
of E : “.M_LOG10E;
echo “<br>”;
echo “The value of PI :
“.M_PI;
echo “<br>”;
echo “The value of PI/2 :
“.M_PI_2;
echo “<br>”;
echo “The value of PI/4 :
“.M_PI_4;
echo “<br>”;
echo “The value of 1/PI :
“.M_1_PI;
echo “<br>”;
echo “The value of 2/PI :
“.M_2_PI;
echo “<br>”;
echo “The square root of
PI : “.M_SQRTPI;
echo “<br>”;
echo “The value 2/square
root of PI : “.M_2_SQRTPI;
echo “<br>”;
echo “The square root of 2 :
“.M_SQRT2;
echo “<br>”;
echo “The square root of 3 :
“.M_SQRT3;
echo “<br>”;
echo “The square root of 1/2
: “.M_SQRT1_2;
```

```
?>
```

வெளியீடு


localhost/phpsites/mathsconstants.php
Value of e : 2.718281828459
Value of Euler's constant : 0.57721566490153
The natural logarithm of PI : 1.1447298858494
The natural logarithm of 2 : 0.69314718055995
The natural logarithm of 10 : 2.302585092994
Value of base-2 logarithm of E : 1.442695040889
The base-10 logarithm of E : 0.43429448190325
The value of PI : 3.1415926535898
The value of PI/2 : 1.5707963267949
The value of PI/4 : 0.78539816339745
The value of 1/PI : 0.31830988618379
The value of 2/PI : 0.63661977236758
The square root of PI : 1.7724538509055
The value 2/square root of PI : 1.1283791670955
The square root of 2 : 1.4142135623731
The square root of 3 : 1.7320508075689
The square root of 1/2 : 0.70710678118655

9 Operators (வினைக்குறி)

Operators (வினைக்குறி)

மாறிகள் (variables) மற்றும் மதிப்புகள் (values) ஆகியவற்றின் மீது கூட்டல், கழித்தல், பெருக்கல், வகுத்தல் மற்றும் இணைத்தல் மற்றும் இன்னும் பல வேலைகளை செய்வதற்கு வினைக்குறிகள் பயன்படுகின்றன. இது PHP யில் மட்டுமல்ல அனைத்து நிரல் மொழிகளிலேயுமே இருக்கின்றது. வினைக்குறிகள் தனியாக மட்டுமல்லாது ++, –, += போன்று இணைந்த வடிவிலும் இருக்கின்றது.

ஒரு குறிப்பிட்ட வேலைக்காக நிர்ணயிக்கப்பட்டு இருக்கும் மாறிகள் அல்லது மதிப்புகளுடன் இருக்கக் கூடியவை **வினைஏற்பிகள்** எனப்படும். அத்தகைய வினைஏற்பிகளுடன் (Operands) வினைக்குறிகள் (Operators) தனக்கான வேலைகளைச் செய்கின்றது.

எந்த வகையான வினைக்குறியை (Operator) நாம் பயன்படுத்தப் போகிறோம் என்பதைப் பொருத்தே

வினைஏற்பிகளின் இடமும், எண்ணிக்கையும் நிர்ணயிக்கப்படுகிறது.

உதாரணமாக கீழ்காணும் கோவையை(expression) பாருங்கள்

1 + 3;

இ ் ந்த கோவையில் (expression) நாம் ஒரு வினைக்குறியையும், இரண்டு வினைஏற்பிகளையும் வைத்திருக்கிறோம். இந்த '+' வினைக்குறி இரண்டு வினைஏற்பிகளின் மதிப்புகளை கூட்டி அந்த மதிப்பை நமக்கு முடிவாக தருகின்றது.

ஒரு கோவை (expressions) முழுமை பெறவதற்காக ஒரு வினைக்குறி மற்ற வினைக்குறியுடன் இணைந்திருக்க முடியும். உதாரணத்திற்கு கீழ்காணும் கோவையை பாருங்கள்

$myAddition = 1 + 3;

மேலே காணும் உதாரணத்தில், $myAddition மாறியால்

(variable) அடையாளப்படுத்தப்பட்ட இரண்டு வினைஏற்பிகளின் மதிப்புகள் கூட்டப்பட்டு அதன் முடிவு வழங்குதல் (assignment) வினைக்குறியால் (operator) (=) $myAddition மாறியில் சேமிக்கப்படுகிறது.

ஒவ்வொரு வினைக்குறியும் அவைகளின் வினைஏற்பிகளுடன் எப்படி தொடர்புபடுத்தப்படுகிறது என்பதைப் பற்றி இந்த பகுதியில் பார்க்க இருக்கிறோம்.

9.1 எண்கணி மற்றும் வழங்குதல் வினைக்குறிகள் (Assignment Operators):

= (equals) வினைக்குறியால் அடையாளப்படுத்தப்படும் மாறிக்கு மதிப்புகளை வழங்குவதற்காக வழங்குதல் வினைக்குறி (assignment operator) பயன்படுத்தப்படுகிறது. கணித வினைக்காக (Mathematical Operation) வழங்குதல் வினைக்குறி, எண்கணித வினைக்குறியுடன் சில நேரங்களில் இணைந்தும் இருக்கலாம்.

கீழ்காணும் அட்டவணையி ் ல் PHP -யில் உள்ள ஏழு வழங்குதல் வினைக்குறிகளும்(Assignment Operators)

பட்டியலிடப்பட்டுள்ளது.

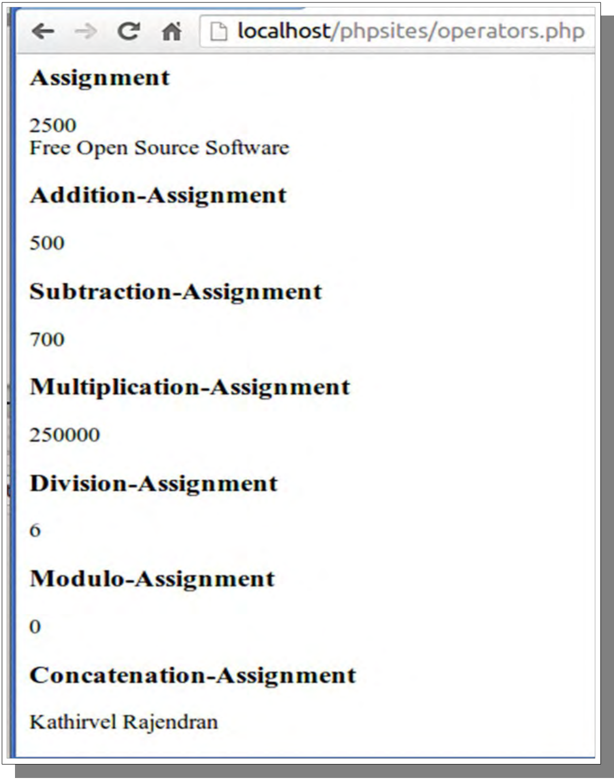
வினைக்குறி (Operator)	**வகை(Type)**	**உதாரணம்(Example)**
=	வழங்குதல்	$myVar = 30;
+=	கூட்டுதலும்-வழங்குதலும்	$myVar = 10;$myVar +=5
-=	கழித்தலும்-வழங்குதலும்	$myVar = 10;$myVar -= 5;
*=	பெருக்குதலும்-வழங்குதலும்	$myVar = 10;$myVar *= 5;
/=	வகுத்தலும்-வழங்குதலும்	$myVar = 10;$myVar /= 10;
%=	மீதியும்-வழங்குதலும்	$myVar = 10;$myVar %= 5;
.=	சரத்தொடர்இணைப்பும்-வழங்குதலும்	$myName = “Kathirvel”;$myName .= “ Rajendran”;

உதாரண நிரல்:

```
<?php
echo "<h3>Assignment</h3>";
$myVarA = 2500;
$myVarB = "Free Open Source
Software";
echo $myVarA;
echo "<br>";
echo $myVarB;
//————————

echo "<h3>Addition-
Assignment</h3>";
$myVarC = 240;
$myVarC += 260;
echo $myVarC;
//————————

echo "<h3>Subtraction-
Assignment</h3>";
$myVarD = 1200;
$myVarD -= 500;
echo $myVarD;
//————————

echo "<h3>Multiplication-
Assignment</h3>";
$myVarE = 500;
$myVarE *= 500;
echo $myVarE;
//————————
echo "<h3>Division-
```

```
Assignment</h3>";
$myVarF = 1200;
$myVarF /= 200;
echo $myVarF;
//————————

echo "<h3>Modulo-
Assignment</h3>";
$myVar = 10;
$myVar %= 5;
echo $myVar;
//————————

echo "<h3>Concatenation-
Assignment</h3>";
$myName = "Kathirvel";
$myName .= " Rajendran";
echo $myName;
?>
```

வெளியீடு:


localhost/phpsites/operators.php
Assignment
2500
Free Open Source Software
Addition-Assignment
500
Subtraction-Assignment
700
Multiplication-Assignment
250000
Division-Assignment
6
Modulo-Assignment
0
Concatenation-Assignment
Kathirvel Rajendran

9.2 கணித வினைக்குறிகள் (Arithmetic Operators):

எண்கணித வினைக்குறிகள் கணிதம் தொடர்பான வினைகளை செய்கிறது. கீழ்காணும் அட்டவணையில் எண்கணித வினைக்குறிகள் பட்டியலிடப்பட்டுள்ளது.

வினைக் குறி (Operator)	**வகை(Type)**	**உதாரணம்(Example)**
+	கூட்டல்	$total = 100 + 250;
–	கழித்தல்	$total = 250 – 100;
*	பெருக்கல்	$total = 100 * 200;
/	வகுத்தல்	$total = 200 / 5;
%	மீதி	$total = 200 % 3;

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
echo “<h3>Addition</h3>”;
$myAdd1 = 240;
$myAdd2 = 260;
$myAddition = $myAdd1 +
$myAdd2;
echo $myAddition;
//————————

echo “<h3>Subtraction</h3>”;
$mySub1 = 1200;
$mySub2 = 500;
$mySubtraction = $mySub1 –
$mySub2;
echo $mySubtraction;
//————————

echo
“<h3>Multiplication</h3>”;
$myMul1 = 500;
$myMul2 = 500;
$myMultiplication = $myMul1
* $myMul2;
echo $myMultiplication;
//————————

echo “<h3>Division</h3>”;
$myDiv1 = 1200;
$myDiv2 = 200;
$myDivision = $myDiv1 /
$myDiv2;
```

```
echo $myDivision;
//————————

echo “<h3>Modulus</h3>”;
$myMod1 = 10;
$myMod2 = 4;
$myModulus = $myMod1 %
$myMod2;
echo $myModulus;
?>
```

வெளியீடு

9.3 ஒப்பீடு வினைக்குறி (Comparison Operators):

இரண்டு மதிப்புகளை ஒப்பிட்டு பார்ப்பதற்காக ஒப்பீடு வினைக்குறி (Comparison Operators) பயன்படுகிறது. ஒப்பீடக்கூடிய இரண்டு மதிப்புகளின் நிலையைப் பொறுத்து true அல்லது false எனும் விடையை அளிக்கிறது. எண்களுடனோ அல்லது சரத்துடனோ (strings) ஒப்பீடு வினைக்குறிகளைப் பயன்படுத்திக்

கொள்ளலாம். இரண்டு வினைஏற்பிகளுடன் (Operands) ஒப்பீடு வினைக்குறி பயன்படுத்தப்படுத்தப்படுகிறது. ஒப்பீடு வினைக்குறிகளின் செயல்பாடுகளைப் பற்றி கீழே இருக்கும் அட்டவணையில் தெளிவாக தெரிந்து கொள்ளலாம்.

Operator (வினைக் குறி)	**Type(வகை)**	**Examples(உதாரணம்)**
==	சமம்	$myVar = 10;if ($myVar == 10) echo ‘myVar equals 10’;
!=	சமமில்லாத	$myVar = 10;if ($myVar != 20) echo ‘myVar does not equal 10’;
<>	சமமில்லாத	$myVar = 10;if ($myVar <>20) echo ‘myVar does not equal 10’;

===	ஒரேமாதிரி யான	$myVar = 10;$myString = “10”; if ($myVar === $myString) echo ‘myVar and myString are same type and value’;
!==	ஒரேமாதிரி அல்லாத	$myVar = 10;$myString = “10”; if ($myVar !== $myString) echo ‘myVar and myString are not same type and value.
<	விடக் குறைவு	$myVar = 10;if ($myVar <20) echo ‘myVar if less than 20’;
>	விட அதிகம்	$myVar = 20;if ($myVar >10) echo ‘myVar if less than 20’;
<=	விடக் குறைவு	$myVar = 10;if($myVar <= 5)

	அல்லது சமம்	echo ‘myVar is less than or equal to 5’;
>=	விட அதிகம் அல்லது சமம்	$myVar = 10;if ($myVar >= 5) echo ‘myVar is greater than or equal to 5’;

9.4 ஏரண வினைக்குறிகள் (*Logical Operators*):

ஏரண வினைக்குறிகள் (Logical Operators) *பூலியன் வினைக்குறிகள் என்றும் அழைக்கப்படுகிறது ஏனென்றால் கோவையினுடைய*(expression) *பகுதிகளை மதிப்பீடு* (evaluate) *செய்து* true (1) *அல்லது* false (0) *எனும் மதிப்பை வெளியீடாக தருகிறது*. PHP *இல் உள்ள ஏரண வினைக்குறிகள்* (Logical Operators) *கீழே உள்ள அட்டவணையில் பட்டியலிடப்பட்டுள்ளது*.

Operator (வினைக்குறி)	Type(வகை)	Examples(உதாரணம்)
&&	AND	If ($a <25) &&($b >45))
||	OR	If (($a <25) \|\| ($b >45))
Xor	XOR	If (($a <25) xor ($b >45))

9.5 ஏறுமான மற்றும் இறங்குமான வினைக்குறிகள் (*Increment and Decrement Operators*):

```
$myMark = 100;

$myMark = $myMark – 1;
```

என்று கொடுப்பதற்கு பதில் ஏறுமான மற்றும் இறங்குமான வினைக்குறிகளைப் பயன்படுத்தி மிகவும் விரைவாக மேலே உள்ள கணக்கீட்டைச் செய்யலாம். ஏறுமானத்திற்கு ++ என்ற வினைக்குறியையும், இறங்குமானத்திற்கு — என்ற வினைக்குறியையும் PHP பயன்படுத்துகிறது.

இரண்டு வழிகளில் இந்த வினைக்குறியைப் பயன்படுத்தலாம் ஒன்று pre (முன்) மற்றொன்று post (பின்). pre முறையானது (mode) ஏறுமான அல்லது இறங்குமானத்தை கோவையினுடைய (expression) மீதமிருக்கும் பகுதிகளை செயல்படுத்துவதற்கு முன்பு செய்கிறது. உதாரணமாக, ஒரு மாறியின் மதிப்பை வேறொரு மாறிக்கு அளிப்பதற்கு முன்பாகவே அந்த மாறியின் மதிப்பை ஏறுமானம் அல்லது இறங்குமானம் அதாவது அதிகப்படுத்த அல்லது குறைக்க வேண்டுமெனில் நாம் pre ஏறுமானம் அல்லது இறங்குமானத்தைப் பயன்படுத்தலாம்.

post முறையானது (mode) ஏறுமான அல்லது இறங்குமானத்தை கோவையினுடைய (expression) மீதமிருக்கும் பகுதிகளை செயல்படுத்திய பின்பு செய்கிறது. உதாரணமாக, ஒரு மாறியின் மதிப்பை வேறொரு மாறிக்கு அளித்த பின்பு அந்த மாறியின் மதிப்பை ஏறுமானம் அல்லது இறங்குமானம் அதாவது அதிகப்படுத்த அல்லது குறைக்க வேண்டுமெனில் நாம் post ஏறுமானம் அல்லது இறங்குமானத்தைப் பயன்படுத்தலாம்.

வினைக்குறி ஒரு மாறியின் முன்பாகவோ, பின்பாகவோ அல்லது எங்கு அமைகிறது என்பதைப் பொறுத்து pre அல்லது post என்பது அமைகிறது. உதாரணமாக

$myMark++ , இங்கு ++ என்பது மாறிக்கு பின்பாக அமைந்திருப்பதால் இது post increment எனப்படுகிறது. ++$myMark என்பதில் ++ என்பது மாறிக்கு முன்பாக அமைந்திருப்பதால் அது pre increment எனப்படுகிறது.

கீழ்காணும் அட்டவணை அதை தெளிவாக விளக்குகிறது.

Operator (வினைக்குறி)	**Type(வகை)**	**Equivalent(சமமானது)**
++$var	முன் ஏறுமானம் (pre increment)	$var = 10;$var2 = $var + 1;
–$var	முன் இறங்குமானம் (pre decrement)	$var = 10;$var2 = $var – 1;
$var++	பின் ஏறுமானம்	$var = 10;$var2 = $var; $var = $var + 1;
$var–	பின் இறங்குமானம்	$var = 10;$var2 = $var;

		$var = $var – 1;

9.6 சரத்தொடர் இணைப்பு வினைக்குறி (String Concatenation Operator):

இரண்டு மதிப்புகளை இணைத்து ஒரு சரத்தை (string) உருவாக்க சரத்தொடர் இணைப்பு வினைக்குறி (operator) பயன்படுகிறது. (.) நிறுத்தற்குறியின் மூலமாக சரத்தொடர் இணைப்பு வினைக்குறி அடையாளப்படுத்தப்படுகிறது. மதிப்புகள் (values), மாறிகள் (variable), மாறிலிகள் (constants), சரங்கள் (strings) என்று எதில் வேண்டுமானாலும் இதைப் பயன்படுத்திக் கொள்ளலாம்.

எண்களையும், சரத்தையும் இணைத்தல்

```
echo 4589 . ' is my bike registration number';
```

மேலே உள்ள நிரலின் வெளியீடு கீழ்காண்பது போன்று

இருக்கும்.

4589 is my bike registration number.

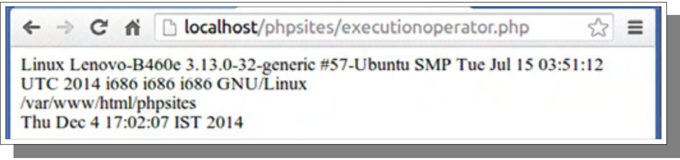
9.7 செயற்படுத்தும் வினைக்குறி — வழங்கியில் கட்டளைகளை செயற்படுத்துதல் (*Execution Operator — Executing Server Side Commands*)

முனையத்தில் நாம் இயக்கக்கூடிய கட்டைளைகள் அனைத்தையும், செயற்படுத்தும் வினைக்குறியைப் பயன்படுத்தி இயக்கலாம். PHP யின் பலத்தில் இதுவும் ஒன்று. நமது இணையதளம் எந்த இணைய வழங்கியில் வைக்கப்பட்டுள்ளதோ அந்த இயங்குதளத்தின் கட்டளைகளை இயக்கி நம்மால் வெளியீட்டைப் பெற முடியும்.

(`) குறியீடு செயற்படுத்தும் வினைக்குறியாக பயன்படுத்தப்படுகிறது. நாம் இயக்க வேண்டிய கட்டளைகளை (`) குறியீட்டிற்குள் கொடுக்க வேண்டும். கீழ்காணும் நிரலைப் பாருங்கள். (`) குறியீடு என்பது ஒற்றை மேற்கோள்குறி அல்ல(single quotes). (~) குறியீடு இருக்கும் பொத்தானில் இருக்கக் கூடியது.

```
<?php
echo `uname -a` . "<br>";
echo `pwd` . "<br>";
echo `date` . "<br>";
?>
```

வெளியீடு:



Flow Control and Looping

PHP போன்ற நிரல்மொழிகளைப் பயன்படுத்துவதன் நோக்கமே, வலை (web) அடிப்படையிலான தகவல்களில் தர்க்கம் மற்றும் நுண்ணறிவு நுணுக்கங்களை கட்டமைக்க வேண்டும் என்பதாகும். தர்க்கம், நுண்ணறிவு என்று வந்துவிட்டாலே சூழலுக்கு ஏற்ப தகவல்களின் அடிப்படையில் அடிக்கடி முடிவுகளை எடுக்க வேண்டி வரும்.

உதாரணமாக, நிரலினுடைய ஒரு குறிப்பிட்ட பகுதியை பலமுறை இயக்க வேண்டிய அவசியம் ஏற்படலாம் அல்லது குறிப்பிட்ட நிபந்தனை பொருந்தும் போது மட்டும் நிரலை இயக்க வேண்டிய அவசியம் ஏற்படலாம் அல்லது நிபந்தனைகள் பொருந்தாத போது மட்டும் நிரல் வரியை இயக்க வேண்டிய சூழல் ஏற்படலாம். (சரியான பயனர் பெயர் மற்றும் கடவுச்சொல்லை மட்டும் உள்ளிட்டால் மட்டுமே தகவலை அணுக முடியும் என்ற நிபந்தனையை இதற்கு உதாரணமாக கூறலாம்). நிரல் மொழியில் இதை flow control and looping என்று அழைப்பார்கள்.

நிபந்தனைக் கூற்றுகள் (conditional Statements)

1. if statements
2. if ... else ... statements

கண்ணி கூற்றுகள் (looping Statements)

1. while loops
2. do ... while loops

Switch Statements

இந்தப் பகுதியில் இவைகளைப் பற்றி விரிவாக, உதாரணங்களுடன் பார்ப்போம்.

10.1 Conditional Statements

வாழ்க்கையில் ஒவ்வொரு விஷயமும் முடிவுகளைச் சுற்றியே சுழல்கிறது. ஒரு நாளில் நாம் எத்தனை முடிவுகளை எடுக்கிறோம் என்று கணக்குப் பார்த்தால்

நமக்கே ஆச்சர்யமாக இருக்கும். ஒரு நாளில் நாம் எத்தனையோ முடிவுகளை எடுக்கிறோம், அவை சின்ன முடிவுகளிலிருந்து பெரிய முடிவுகளை வரை இருக்கலாம். எந்த ஆடையை அணிய வேண்டும்? என்ன சாப்பிட வேண்டும்? எந்த சாலையின் வழியாக பயணிக்க வேண்டும்? இது போன்ற எத்தனையோ முடிவுகளை நாம் தினம் தினம் எடுத்துக் கொண்டிருக்கிறோம்.

இதுபோன்று கணினியை முடிவு எடுக்க வைப்பதற்காக அதற்கென தனியாக ஆணைகள் (instructions) இருக்கின்றது. நிபந்தனைக் கூற்று (conditional statements) முடிவுகளை உருவாக்க உதவுகிறது. Conditional statements ஒரு குறிப்பிட்ட கோவையின் (expression) முடிவுகளின் அடிப்படையில் நிரலின் பகுதியை இயக்க வேண்டுமா அல்லது வேண்டாமா என்று கட்டுப்படுத்துகிறது. கோவையானது (expression) true அல்லது false இந்த இரண்டு பூலியன் மதிப்புகளில் ஏதாவது ஒன்றை அளிக்கும்.

PHP யில் இரண்டு வகையான நிபந்தனை கட்டமைப்புகள் இருக்கின்றது. அவைகள் if மற்றும் if … else.

If statement (கூற்று)

நிபந்தனைகளைக் கொண்ட நிரல்களை எழுதுவதற்கான அடிப்படை if *கூற்றிலிருந்தே தொடங்குகிறது*. If *கூற்றின் முதல் வரி* if *கூற்றையும் அதனைத் தொடர்ந்து பிறை வளைக்குள்* (parentheses) *கோவைகளையும்* (expressions) *கொண்டிருக்கும்*.

உதாரணம்:

$myMark = 98;

if ($myMark < 100)

மேற்காணும் உதாரணத்தில், $myMark *எனும் மாறி* (variable)*யின் மதிப்பு 2 ஐ காட்டிலும் குறைவாக இருக்கிறதா? என சோதிக்கப்படுகிறது. இருந்தால்* true *எனவும் இல்லையென்றால்* false *எனவும் மதிப்பீடு செய்யப்படுகிறது*.

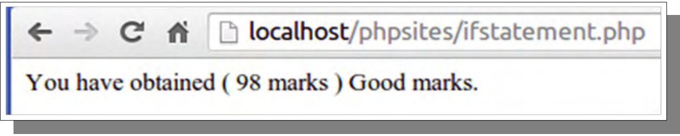
If *கூற்றில் இரண்டாவது படி என்னவென்றால்*, *கோவை* (expression) true (*சரி*) *என்றால் என்ன செய்ய வேண்டும்*

என்பதைக் குறிப்பதாகும். அவ்வாறு கோவை சரி என்றால் செய்ய வேண்டிய வேலைகள் if கூற்றைத் தொடர்ந்து வரும் open and closing braces க்குள் கொடுக்கப்படும். braces க்குள்தான் கொடுக்க வேண்டும் என்பது கட்டாயமல்ல. Braces இல்லாமல் கொடுத்தாலும் அது ஏற்றுக் கொள்ளப்படும். ஆனாலும் if கூற்றைத் தொடர்ந்து ஒற்றை வரி நிரல் இருந்தாலும் அதை braces க்குள் கொடுப்பதையே பரிந்துரைக்கின்றனர். அப்பொழுதுதான் நிரல்கள்களை எளிமையாக படிக்கவும், பொதுவாக ஏற்படும் தவறுகளை தவிர்க்கவும் முடியும்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$yourMark = 98;
if( $yourMark > 90 ) {
echo “You have obtained ( “.
$yourMark.” marks ) Good
marks.”;
}
?>
```

வெளியீடு:



if … else கூற்று (if … else statements)

நாம் மேலே பார்த்த if கூற்றில் கோவை (expression) true (சரி) என்றால் என்ன செய்ய வேண்டும் என்பதை மட்டும்தான் அனுமதிக்கிறது. கோவை (expression) false (தவறு) என்றால் என்ன செய்ய வேண்டும் என்பதை நாம் if கூற்றில் சொல்ல முடியாது. அதற்காகத்தான் if…else கூற்று. கோவை சரி என்றால் என்ன செய்ய வேண்டும் என்பதையும் மற்றும் கோவை தவறு என்றால் என்ன செய்ய வேண்டும் என்பதையும் if…else கூற்றில் நாம் சொல்ல முடியும்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$milkPacketColor = ‘blue’;
if ( $milkPacketColor ==
‘blue’ ) {
$milkPrice = 45;
echo “Milk Price is Rs.”.
$milkPrice.” p/l”;
echo “<br>”;
}
else {
$milkPrice = 40;
echo “Milk Price is Rs.”.
$milkPrice.” p/l”;
echo “<br>”;
}
?>
```

வெளியீடு

if … else *கூற்றை* if … else … if *கூற்றாகவும் நீட்டிக்க*

முடியும்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$customerName = “Kathirvel”;
if ( $customerName ==
“Kathir” ) {
echo “Customer Name : “.
$customerName;
echo “<br>”;
}
else if ( $customerName ==
“Kathirvel” ) {
echo “Customer Name : “.
$customerName;
echo “<br>”;
}
else {
echo “Sorry!”;
echo “<br>”;
}
?>
```

வெளியீடு

10.2 கண்ணி கூற்றுகள் (*Looping Statements*)

கணினி ஒரே வேலையை எத்தனை முறை வேண்டுமானாலும் திரும்ப திரும்ப அலுக்காமல் செய்யும் என்பது அனைவரும் அறிந்ததே. கணினியைப் பற்றி படிக்கும் அனைவருக்கும் இது தெரிந்ததுதான். அதேபோல் நிரலில் ஒரு குறிப்பிட்ட பகுதியை நிபந்தனையை எட்டும் வரையில் திரும்ப திரும்ப செய்வதற்கு looping statements (*கண்ணி கூற்றுகள்*) *பயன்படுகிறது.*

PHP – *யில் மூன்று வகையான கண்ணி கூற்றுகள் இருக்கின்றது. அவைகள்*

1. for loop
2. while loop
3. do … while loop

for loops (for கண்ணி)

ஒரு எண் தன்னைத்தானே பத்து முறை கூட்டிக்கொள்ள வேண்டுமெனில் அதற்கான PHP *நிரல் கீழ்கண்டவாறு இருக்கும்.*

```
<?php
$myInterest = 1;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
$myInterest += $myInterest;
echo "My Interest Amount
Rs.".$myInterest;
echo "<br>";
?>
```

வெளியீடு

நாம், மேலே உள்ள ஒரு எண்ணை 10 முறை தன்னைத்தானே கூட்டிக்கொள்ளும்படி அமைத்துள்ளோம். ஒரு வேளை 1000 முறை கூட்ட வேண்டும் அல்லது 10000 முறை கூட்ட வேண்டுமெனில். மேலே உள்ளது போன்று நிரலை எழுதிக் கொண்டிருந்தால் நிலைமை என்னவாகும்? நிரலினுடைய வரிகள் அதிகமாவதோடு, முக்கியமாக அதிகமான நேரம் வீணாகும். இதுபோன்ற சூழலை தவிர்ப்பதற்குத்தான் for கண்ணி பயன்படுகிறது.

For கண்ணியினுடைய (loop) தொடரியல் (syntax):

```
for ( initializer; conditional expression; loop
expression )

{

//PHP statements to be executed go here
```

}

initializer வழக்கமாக counter variable ஐ initialize செய்கிறது. இதற்கு $i மாறியே வழக்கமாக பயன்படுத்தப்படுகிறது. உதாரணமாக $i = 0 ஐ சொல்லலாம். இது $i இன் மதிப்பை 0 என அமைக்கிறது.

எத்தனை முறை loop இயங்க வேண்டும் என்பதை conditional expression நிர்ணயிக்கிறது. Loop 1000 முறை இயங்க வேண்டுமென்றால் $i < 1000 என கொடுக்க வேண்டும்.

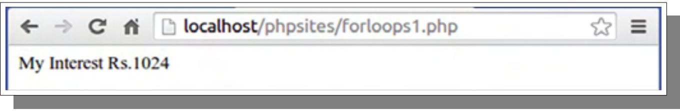
இறுதியாக, loop expression counter variable மீது நடக்க வேண்டிய செயலைப் பற்றி கூறுகிறது. உதாரணத்திற்கு counter variable இன் மதிப்பு ஒவ்வொரு முறையும் 1 அதிகமாவதைக் கூறலாம்.

$i++

நாம் மேலே பார்த்த அனைவற்றையும் ஒருங்கிணைத்து ஒரு நிரல் எழுதுவோமா? கீழே உள்ள நிரலைப் பாருங்கள்.

```
<?php
$myInterest = 1;
for ( $i=0 ; $i < 10 ; $i+
+ ) {
$myInterest += $myInterest;
}
echo “My Interest Rs.”.
$myInterest;
echo “<br>”;
?>
```

வெளியீடு



While loops (while கண்ணி)

for loop இல் loop *ஆனது நிபந்தனையைத் தொடுவதற்கு எத்தனை முறை திரும்ப திரும்ப இயங்க வேண்டும்*

என்பதை முன்னாடியே சொல்லிவிடுகிறோம். ஒரு வேளை நிபந்தனையைத் தொட எத்தனை முறை loop ஆனது இயங்க வேண்டும் என்பது நமக்கு தெரியவில்லையென்றால் என்ன செய்வது? இங்குதான் while loop பயன்படுகிறது.

While loop –இன் தொடரியல் (syntax of while loop)

```
<?php

while ( condition )

{

// PHP statements go here

}

?>
```

condition true ஆக இருக்கும் வரை while loop வேலை

செய்யும். false ஆகும் போது loop இயங்காது.

கீழ்காணும் உதாரண நிரலை பாருங்கள்

```
<?php
$myInterest = 1;
$j = 1;
while ( $j <= 10 ) {
$myInterest += $myInterest;
$j++;
}
echo “My Interest Rs.”.
$myInterest;
echo “<br>”;
?>
```

வெளியீடு

do … while loops(do … while கண்ணி)

do … while loop என்பது while –க்கு அப்படியே நேரெதிரானதாக இருக்கும் என நீங்கள் நினைக்கலாம். while loop ஆனது while loop ற்குள் இருக்கும் நிரலை இயக்குவதற்கு முன் முதலில் while loop இல் கொடுக்கப்பட்டிருக்கும் நிபந்தனையை மதிப்பீடு செய்து சோதித்துப் பார்க்கிறது. முதல் சோதனையிலேயே நிபந்தனை தவறு என்றால் அதன் பின்பு loop ற்குள் இருக்கும் நிரல்களை இயக்காமல் loop ஐ விட்டு வெளியேறிவிடும். இந்நிலையில், ஒருமுறையேனும் loop ஆனது இயங்க வேண்டும் அல்லது நிபந்தனை இறுதியாகத்தான் சோதிக்கப்பட வேண்டும் என்ற தேவை உங்களுக்கு வரும்போது அதை do … while loop நிறைவேற்றி வைக்கிறது. Do … while loop இன் செயல்பாடும் அதுதான்.

do … while கண்ணியின் இன் தொடரியல் (do … while loop syntax):

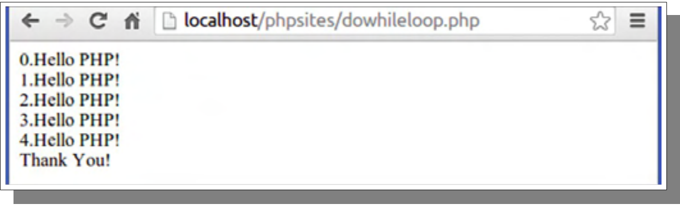
```
<?php
do
{
PHP statements
}
while (conditional expression)
```

```
?>
```

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$myInit = 0;
do
{
echo “$myInit.Hello PHP!”;
echo “<br>”;
$myInit++;
}
while($myInit < 5);
echo “Thank You!”;
?>
```

வெளியீடு



10.3 switch கூற்று (switch statements)

நாம் ஏற்கனவே if...else கூற்றுகளைப் பற்றி பார்த்துள்ளோம். குறைவான நிபந்தனைகள் என்றால் if... else கூற்றுகளைப் பயன்படுத்தலாம். அதே நேரத்தில் நிபந்தனைகள் அதிகமாக இருக்கும் போதும் if...else கூற்றைக் கொண்டு அதைச் செய்யும்போது, நேரம் வீணாவதோடு அவ்வாறு செய்து கொண்டிருப்பது தேவையில்லாத செயலாகவும் அமையும். ஆகையால் நிபந்தனைகள் அதிகமாகும்போது நாம் தாராளமாக switch கூற்றைப் பயன்படுத்தி அந்த வேலையை எளிமையாக முடிக்கலாம்.

switch கூற்றின் தொடரியல் (syntax)

switch (”value”)

{

case “match1”:

PHP statements

break;

case “match2”:

PHP statements

break;

case “match3”:PHP Essentials

PHP statements

```
break;

case "match4":

PHP statements

break;

case "match5":

PHP statements

break;

default:

PHP statements

break;

}
```

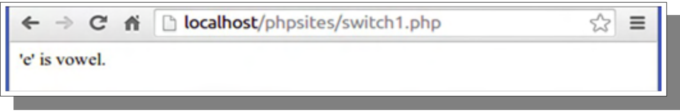
switch (‘value’) என்பதற்குள் நாம் சோதிக்க வேண்டிய உள்ளீடு அல்லது நிபந்தனையைக் கொடுக்க வேண்டும். எத்தனை case கூற்று (statements) வேண்டுமானாலும் நீங்கள் கொடுத்துக் கொள்ளலாம். case கூற்றுக்குள் இருக்கும் நிபந்தனையோடு உள்ளீடு பொருந்தினால், அதன்பின் இருக்கக்கூடிய நிரல் வரிகள் இயக்கப்படும். பொருந்தாவிட்டால் அடுத்த case கூற்றை ஆராய்துப் பார்க்கும். இது இறுதியாக இருக்கும் case கூற்று வரை தொடரும். எதுவுமே பொருந்தவில்லை என்றால் இறுதியாக இருக்கும் default கூற்றின் கீழ் உள்ள நிரல்கள் இயக்கப்படும். இங்கு முக்கியமாக கவனிக்க வேண்டியது என்னவென்றால் break கூற்றைத்தான். break இல்லாமல் கொடுத்தால் எந்த கூற்று பொருந்தியதோ அதற்கு அடுத்து உள்ள கூற்றும் செயல்படுத்தப்படும்.

உதாரணமாக, ஆங்கிலத்தில் இருக்கக்கூடிய a,e,i,o,u எழுத்துக்கள் வந்தால் அது உயிரெழுத்து என்று நமக்கு செய்தி கிடைக்க வேண்டும். இதற்கு if...else ஐ வைத்து ஒரு நிரல் எழுதினால் எப்படி இருக்கும் என்பதைப் பார்ப்போம்.

```
<?php
$inputChar = “e”;

if ( $inputChar == “a” ) {
echo “‘$inputChar’ is
vowel.”;
echo “<br>”;
}

elseif ( $inputChar == “e” )
{
echo “‘$inputChar’ is
vowel.”;
echo “<br>”;
}

elseif ( $inputChar == “i” )
{
echo “‘$inputChar’ is
vowel.”;
echo “<br>”;
}

elseif ( $inputChar == “o” )
{
echo “‘$inputChar’ is
vowel.”;
echo “<br>”;
}

elseif ( $inputChar == “u” )
```

```
{
echo “‘$inputChar’ is
vowel.”;
echo “<br>”;
}

else {
echo “Input Character is not
a vowel”;
}
?>
```

இதன் வெளியீடு



மேற்காணும் நிரலை switch கூற்று கொண்டு எழுதும் போது அது எவ்வளவு எளிமையாக அமைகிறது என்று பாருங்கள்.

```
<?php
$inputChar = “e”;

switch ($inputChar) {

case “a”:
echo “‘$inputChar’ is a
vowel”;
break;

case “e”:
echo “‘$inputChar’ is a
vowel”;
break;

case “i”:
echo “‘$inputChar’ is a
vowel”;
break;

case “o”:
echo “‘$inputChar’ is a
```

```
vowel”;
break;

case “u”:
echo “‘$inputChar’ is a
vowel”;
break;

default:
echo “‘$inputChar’ is not a
vowel”;
break;
}
?>
```

வெளியீடு

10.4 கண்ணி முறிப்பு (*Breaking a Loop*):

loop ஐ break பண்ண வேண்டிய அவசியம் நமக்கு

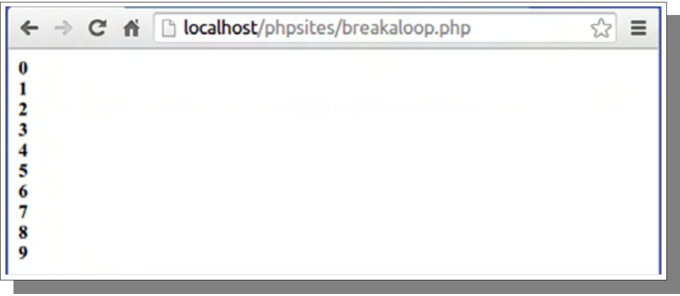
எப்போதாவது ஏற்படலாம். இந்த தேவையை நாம் break கூற்று மூலமாக நிறைவேற்றி கொள்ளலாம்.

1000 வரை எண்களை அச்சிடும்படி கீழ்காணும் நிரலை எழுதியிருக்கிறோம். ஆனாலும் loop ஆனது 10 என்ற எண்ணை அடைந்தவுடன் break கூற்று மூலமாக முறிக்கப்படுகிறது.

நிரல்:

```
<?php
for ( $i = 0; $i < 1000; $i+
+) {
if ($i == 10) {
break;
}
echo "<b>$i</b><br>";
}
?>
```

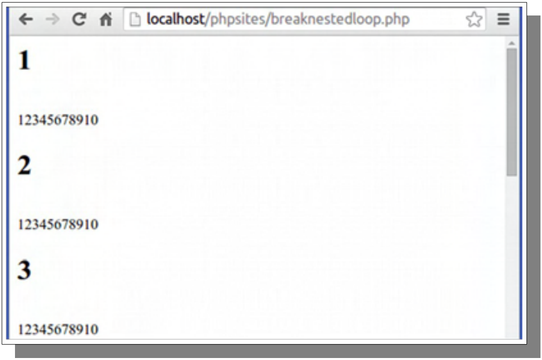
வெளியீடு:



Breaking out of Nested Loops

```
<?php
for ( $i = 1; $i < 20; $i++)
{
echo “<h1>$i</h1><br>”;
for ( $j = 1; $j < 50; $j++)
{
echo “$j<br>”;
if ($j == 5) {
break;
}
}
}
?>
```

வெளியீடு:


localhost/phpsites/breaknestedloop.php
1
12345678910
2
12345678910
3
12345678910

11 Functions

Functions (செயல்கூறு)

நிரல் எழுதுவதில் முறைகள் உள்ளது ஒன்று நீளமாக எழுதுவது மற்றொன்று சிறு சிறு துண்டுகளாக பிரித்து எழுதிப் பிறகு தேவையான இடத்தில் சிறிய பகுதிகளை பயன்படுத்திக் கொள்வது அல்லது சிறிய பகுதிகள் அனைத்தையும் சேர்த்து பெரிய நிரலாக மாற்றிக் கொள்வது.

11.1 Function (செய்லகூறு) என்றால் என்ன?

PHP யின் உண்மையான பலமே அதனுடைய செயல்கூறில்தான் இருக்கிறது. PHP யில் 1000 build-in functions மேலும் உள்ளது.

> செயல்கூறு(function) என்பது கூற்றுகளின்(statements) தொகுதி ஆகும்.

செயல்கூறுகள் வலைப்பக்கம் load ஆகும் போதே தனாக இயங்காது.

செயல்கூறுகளை அழைத்தால் மட்டுமே இயங்கும்.

11.2 செயல்கூறை(*function*) எப்படி எழுதுவது?

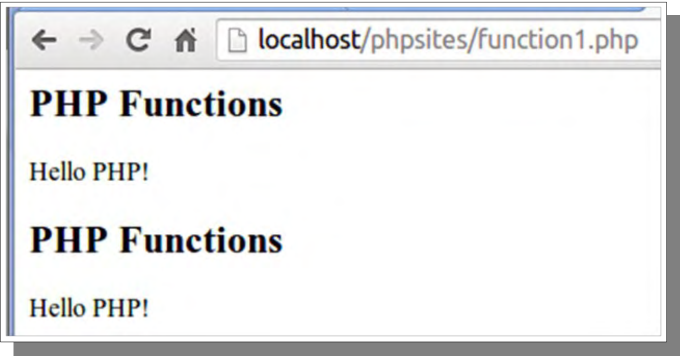
செயல்கூறை எழுதுவதில் முதல் படி என்னவென்றால், செயல்கூறை நிரலில் அழைப்பதற்காக அல்லது பயன்படுத்திக் கொள்வதற்காக அதற்கு பெயரிடுவதுதான். மாறிகளுக்கு (variable) பெயரிடும் முறைப்படியே செயல்கூறுக்கும் (function) பெயரிட வேண்டும். மாறிகளுக்கு பெயரிடும் முறையில் உள்ள கட்டுப்பாடுகள் அனைத்தும் செயல்கூறுக்கு பெயரிடுவதற்கும் பொருந்தும்.

function என்ற முதன்மைச் சொல்லைக் (key word) கொண்டு செயல்கூறு(function) உருவாக்கப்படுகிறது. Function என்ற முதன்மைச் சொல்லைத் தொடர்ந்து செயல்கூறின் பெயர் இருக்கும். இருதியில் ஒரு ஜோடி பிறை வளை (a set of parentheses) இருக்கும். செயல்கூறின் உடல் (body of function) opening and closing braces – க்குள் இருக்கும்.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php

//function creation
function myFunction() {
echo “<h2>PHP
Functions</h2>”;
echo “Hello PHP!”;
}

//function calling
echo myFunction();
myFunction();

?>
```

நிரலின் வெளியீடு



11.3 செயல்கூறில் இருந்து மதிப்புகள் திரும்புதல்

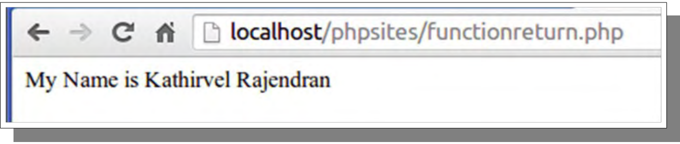
(*Returning a Value from a function*)

செயல்கூறினை நாம் அழைக்கும் போது அதிலிருந்து ஒற்றை மதிப்பு திருப்பி தரப்படலாம். திருப்பி தரப்படும் மதிப்புகள் எந்தவொரு மாறியினுடைய மதிப்பாகவும் இருக்க முடியும். அது எந்த மதிப்பு நம் விருப்பத்தைப் பொருத்தது.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php

function myReturnFunction()
{
$myFirstName = “Kathirvel”;
$myLastName = “Rajendran”;
$myFullName = $myFirstName.”
“.$myLastName;
return $myFullName;
}

echo “My Name is
“.myReturnFunction();
?>
```

நிரலின் வெளியீடு



11.4 செயல்கூறுக்கு அளபுருக்களை செலுத்துதல் (passing parameters to a function)

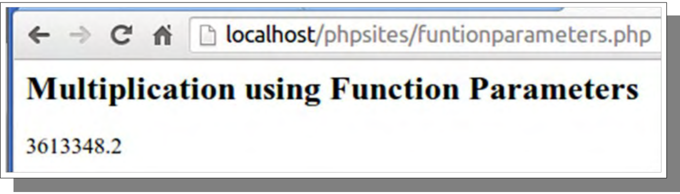
செயல்கூறுக்குள் அளபுருக்களைச் செலுத்த முடியும். இத்தனை அளபுருக்களைத்தான் செலுத்த வேண்டும் என்ற கட்டாயமில்லை. எவ்வளவு வேண்டுமானாலும் செலுத்திக் கொள்ளலாம். நாம் செயல்கூறை வடிவமைக்கும்போதே அளபுருக்களை ஏற்றுக் கொள்வதற்கு ஏற்ப வடிவமைக்க வேண்டும். அளபுருக்களின் பெயர்களை (parameters names) பிறை வளைக்குள் (parentheses) கொடுக்க வேண்டும். நாம் பொதுவாக மாறிகளுக்கு பெயரிடுவோமா இல்லையா அதுபோலவே அளபுருக்களுக்கும் பெயரிட்டுக் கொள்ள வேண்டும். இதற்கென தனியாக எந்த வரைமுறைகளும் கிடையாது.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php

function
myMultiplication($firstNumbe
r, $secondNumber) {
```

```
return $firstNumber *
$secondNumber;
}

echo “<h2>Multiplication
using Function
Parameters</h2>”;
echo
myMultiplication(1540,2346.3
3);
?>
```

நிரலின் வெளியீடு



மேலே நாம் பார்த்த நிரலில் myMultiplication() function இரண்டு மதிப்புகளை ஏற்றுக் கொண்டு அந்த மதிப்புகளை firstNumber and secondNumber ஆகிய மாறிகளுக்கு அளிக்கிறது பின்பு அந்த மதிப்புகள்

பெருக்கப்பட்டு திருப்பி அளிக்கப்படுகிறது.

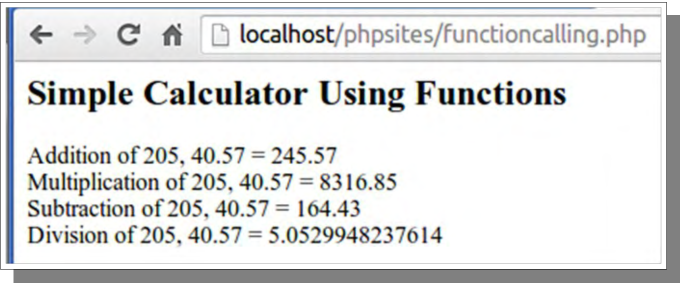
11.5 செயல்கூறை அழைத்தல் (*calling functions*)

செயல்கூறை உருவாக்கும் போது நாம் அதற்கு கொடுத்த பெயரைக் கொண்டு செயல்கூறை நாம் எங்கு வேண்டுமானாலும் அழைத்துக் கொள்ளலாம். கீழே உள்ள நிரலைப் பார்த்தால் உங்களுக்கு நன்கு புரியும்.

```
<?php

function
addNumbers($firstNumber,
$secondNumber) {
return $firstNumber +
$secondNumber;
}

function
mulNumbers($firstNumber,
$secondNumber) {
return $firstNumber *
$secondNumber;
}
```

```
function
subNumbers($firstNumber,
$secondNumber) {
return $firstNumber -
$secondNumber;
}

function
divNumbers($firstNumber,
$secondNumber) {
return $firstNumber /
$secondNumber;
}

$first = 205;
$second = 40.57;
echo “<h2>Simple Calculator
Using Functions</h2>”;
echo “Addition of $first,
$second =
“.addNumbers($first,
$second);
echo “<br>”;
echo “Multiplication of
$first, $second =
“.mulNumbers($first,
$second);
echo “<br>”;
echo “Subtraction of $first,
$second =
```

```
“.subNumbers($first,
$second);
echo “<br>”;
echo “Division of $first,
$second =
“.divNumbers($first,
$second);

?>
```



வெளியீடு

11.6 *Passing Parameters by Reference*

இந்த நிரலைப் பாருங்கள்

```
<?php

function
myAddition($firstNumber ,
$secondNumber) {
$firstNumber += 100;
$secondNumber += 200;
return $firstNumber +
$secondNumber;
}

$setFirstNumber = 100;
$setSecondNumber = 200;

echo
"<h4>Before</h4>setFirstNumb
er = $setFirstNumber,
setSecondNumber =
$setSecondNumber <br>";
myAddition($setFirstNumber ,
$setSecondNumber);
echo
"<h4>After</h4>setFirstNumbe
r = $setFirstNumber,
```

```
setSecondNumber =
$setSecondNumber<br>”;

?>
```

வெளியீடு

நாம் மேலே உள்ள நிரலில் $setFirstNumber *மற்றும்* $setSecondnumber *ஆகிய இரண்டு மாறிகளின் மதிப்புகளையும்* myAddition() *எனும் செயல்கூறுக்குள்*(function) *அனுப்புகிறோம். நாம் அனுப்பிய மாறிகளின் மதிப்புகளைப் பெற்றுக் கொண்டு, அந்த மதிப்புகளில் சில மாற்றங்கள்* myAddition()

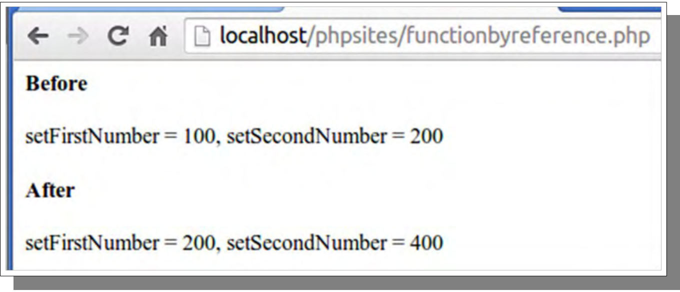
function க்குள் நடக்கிறது. அவ்வாறு நடந்த போதிலும் myAddition() function க்குள் மதிப்புகளைச் செலுத்துவதற்காக நாம் பயன்படுத்திய $setFirstNumber மற்றும் $setSecondNumber ஆகிய மாறிகளின் உண்மையான மதிப்புகளில் எந்த மாற்றமும் நடக்கவில்லை.

function க்குள்ளே மதிப்புகளில் நடைபெறும் மாற்றங்கள், function-க்கு மதிப்புகளை அனுப்புவதற்காக பயன்படுத்தப்படும் மாறிகளிலும் நடைபெற வேண்டுமென்றால் அதற்குத்தான் இந்த reference பயன்படுகிறது. இதை நாம் passing by reference என்று அழைக்கலாம். இதை நாம் ஒரு சிறிய மாற்றத்தின் மூலமாக செய்யலாம். அது எப்படியென்றால் function parameters variable க்கு முன் (&) குறியீட்டை இணைத்துவிட வேண்டும். மேலே நாம் நிரல் கீழே (&) குறீயீடு முன்— இணைப்பாக இணைத்து கொடுக்கப்பட்டுள்ளது. நிரலையும் அதற்கான வெளியீட்டையும் கீழே பாருங்கள்

```
<?php
function
myAddition(&$firstNumber ,
```

```
&$secondNumber) {
$firstNumber += 100;
$secondNumber += 200;
return $firstNumber +
$secondNumber;
}
$setFirstNumber = 100;
$setSecondNumber = 200;

echo
“<h4>Before</h4>setFirstNumb
er = $setFirstNumber,
setSecondNumber =
$setSecondNumber <br>”;

myAddition($setFirstNumber ,
$setSecondNumber);
echo
“<h4>After</h4>setFirstNumbe
r = $setFirstNumber,
setSecondNumber =
$setSecondNumber<br>”;

?>
```

வெளியீடு



11.7 *Functions and Variable Scope*

Two Scopes

1. global scope
2. local scope

Global Scope

function – க்கு வெளியே variable declare செய்யப்பட்டால் அது global scope என்று அழைக்கப்படுகிறது. Gobal scope உடைய variable ஐ நிரலில் எங்கு வேண்டுமானாலும் பயன்படுத்திக் கொள்ளலாம்.

Local Scope

function–க்கு உள்ளே variable declare செய்யப்பட்டால் அது local scope எனப்படும். local scope உடைய variable ஐ எங்கு declare செய்யப்பட்டதோ அந்த function-க்குள் மட்டும்தான் பயன்படுத்த முடியும். function-க்கு வெளியே பயன்படுத்த முடியாது.

12 Arrays

PHP Arrays பல மாறிகளை (variable) ஒன்றாக

இணைத்து ஒரு குழுவாக மாற்றி அதை ஒற்றை மாறியின் (variable) மூலமாக அணுகுவதற்கு வழி எற்படுத்தி தருகிறது. Array யானது ஒருமுறை உருவாக்கப்பட்டுவிட்டால் அதன் பிறகு அதில் நம்மால் உருப்படிகளைச் (items) சேர்க்க, நீக்க, மாற்ற, வரிசைப்படுத்த முடியும். ஒரு Array இருக்கும் உருப்படிகள் எந்த மாறி வகையினைச் சேர்ந்ததாக இருக்கலாம். Array யில் உருப்படிகள் அனைத்தும் ஒரே வகையினைச் சேர்ந்ததாகத்தான் இருக்க வேண்டும் என்று எந்த கட்டாயமும் இல்லை.

Array – யினைடைய உறுப்புகளை key யைக் கொண்டு அணுக முடியும். இரண்டு வகையான Array –க்கள் இருக்கின்றன. Array – யினுடைய உறுப்புகளை எந்தவகையான key யைக் கொண்டு அணுகிறோம் என்பதைப் பொறுத்து அவை வகைப்படுத்தப்படுகிறது.

1. Numerical Array
2. Associative Array

12.1 Numerical Array

Numerical Key Array யில் Array – யின் உறுப்புகள் உருப்படியினுடைய numerical position -க் கொண்டு அணுகப்படுகிறது. Array – யின் முதல் உருப்படி element 0, இரண்டாவது உருப்படி element 1 … and so on.

12.2 Associative Array

Associative Array யில் Array – யின் உறுப்புகள் ஒவ்வொரு உருப்படிக்கும் கொடுக்கப்பட்டுள்ள பெயரைக் கொண்டு அணுகப்படும்.

12.3 Array உருவாக்குதல் (Creating a Array)

array() function ஐக் கொண்டு Array – க்கள் உருவாக்கப்படுகிறது. Array() function 0 அல்லது அதற்கு அதிகமான argument களை எடுத்துக் கொண்டு ஒரு புதிய array – யை நமக்கு திருப்பி அளிக்கிறது. Assignment Operator (=) ஐக் கொண்டு புதிய array யானது இடதுபுறமாக இருக்கும் மாறிக்கு

கொடுக்கப்படுகிறது.

Array – யானது உருப்படிகள் சேர்க்கப்பட்டால் வளரும் (grow), உருப்படிகள் நீக்கப்பட்டால் சுருங்கும் (shrink). இவை dynamic ஆக நடைபெறும். ஆகையால் மற்ற நிரல் மொழிகளில் உள்ளதைப் போன்று array – யை உருவாக்கும் போதே அதனுடைய அளவையும் கொடுக்க வேண்டும் என்ற கட்டாயமில்லை.

12.4 Empty Array உருவாக்கம் (empty array creation)

```
<?php
$emptyArray = array();
?>
```

இதற்கு மாற்றாக, array க்கு மதிப்புகளை arguments களாக கொடுப்பதன் மூலமாக முன்-தொடக்கம் செய்யப்பட்ட array யையும் உருவாக்க முடியும்.

```
<?php
$linuxDistros =
array("Redhat" , "Debian" ,
"Slackware" , "Ubuntu" ,
"Fedora");
?>
```

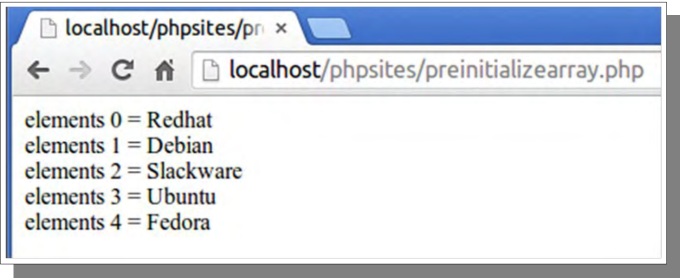
12.5 *Array* - யின் உறுப்புகளை அணுகுதல்

numerical key array *வகையில் உள்ள உறுப்புகள் மாறியின் பெயரைத் தொடர்ந்து வரக்கூடிய* square brackets ([]) *-க்குள் கொடுக்கப்படும் சுட்டைக்* (index) *கொண்டு அணுகப்படுகிறது. முதல் உறுப்பு* 0 *விலிருந்து தொடங்கும் என்பதை ஞாபகத்தில் வைத்துக் கொள்ளுங்கள். நான் மேலே பார்த்த நிரலில் உள்ள உறுப்புகளை அணுகுவது எப்படி என்று பார்ப்போமா?*

```
<?php
$linuxDistros =
```

```
array("Redhat" , "Debian" ,
"Slackware" , "Ubuntu" ,
"Fedora");
echo "elements 0 = ".
$linuxDistros[0];
echo "<br>";
echo "elements 1 = ".
$linuxDistros[1];
echo "<br>";
echo "elements 2 = ".
$linuxDistros[2];
echo "<br>";
echo "elements 3 = ".
$linuxDistros[3];
echo "<br>";
echo "elements 4 = ".
$linuxDistros[4];
?>
```

வெளியீடு



12.6 Associative Array யை உருவாக்குதல் (Creating an Associative Array)

Associative Array – யில் numerical position – க்கு பதிலாக பெயர்கள் கொடுக்கப்படுகிறது. இந்த முறையானது associative array – யின் உறுப்புகளை அணுகும் முறையை எளிமையாக்குகிறது. Associative Array – ஐ உருவாக்க array() function பயன்படுகிறது. Key => value எனும் முறைப்படி associative array – க்கு arguments களை கொடுக்க வேண்டும். இங்கு key என்பது value – ஐ அணுகுவதற்காக கொடுக்கப்படும் பெயர், value என்பது value -ஐ சேமித்து வைப்பதற்காக கொடுக்கப்படுவது.

உங்களுடைய விபரங்களை சேமிப்பதற்கு ஒரு associative array – ஐ உருவாக்கினால் எப்படி இருக்கும் என்பதை கீழே உள்ள நிரலின் மூலம் காணலாம்.

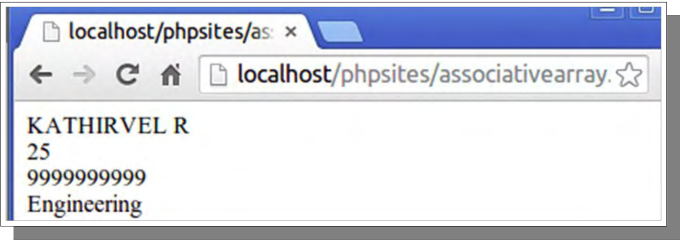
```
<?php
$myDetails =
array('name'=>'KATHIRVEL
R' , 'age'=>25 ,
'mobile'=>'9999999999′ ,'qu
alification'=>'Engineering')
;
?>
```

12.7 Associative Array — யின் உறுப்புகளை அணுகுதல் (Accessing Elements of an Associative Array)

Associative Array – ஐ உருவாக்குவது எப்படி என்று முந்தைய பகுதியில் நாம் பார்த்தோம். இப்பொழுது அதிலுள்ள உறுப்புகளை அணுகுவது எப்படி? என்று பார்ப்போம். மேலே உள்ள $myDetails என்பதையே உதாரணத்திற்கு எடுத்துக் கொள்வோம்.

```
<?php
$myDetails =
array('name'=>'KATHIRVEL
R' , 'age'=>25 ,
'mobile'=>'9999999999′ ,'qu
alification'=>'Engineering')
;
echo $myDetails['name'];
echo "<br>";
echo $myDetails['age'];
echo "<br>";
echo $myDetails['mobile'];
echo "<br>";
echo
$myDetails['qualification'];
echo "<br>";
?>
```

வெளியீடு



12.8 *Array சுட்டியைப் பயன்படுத்துதல்(Using Array Pointers)*

Array யானது உறுப்புகளை அணுகுவதற்காக உள்ளுக்குள்ளேயே ஒரு சுட்டியை(pointer) பராமரித்து வருகிறது. Next, previous, reset மற்றும் end ஆகிய function களைக் கொண்டு அந்த சுட்டியை நம்மால் மாற்ற முடியும். இந்த reset மற்றும் end செயல்கூறுகள்(functions) array -யினுடைய முதல் மற்றும் கடைசி உறுப்புகளுக்கு சுட்டியை நகர்த்துகிறது. Prev function தற்போதைய உறுப்புக்கு முன்னதாக உள்ள உறுப்புக்கு சுட்டியை நகர்த்துகிறது. Pre மற்றும் next functions சுட்டியை முன்பு அல்லது பின்பு சுட்டியை நகர்த்த முடியாத பட்சத்தில் false எனும் மதிப்பை நமக்கு

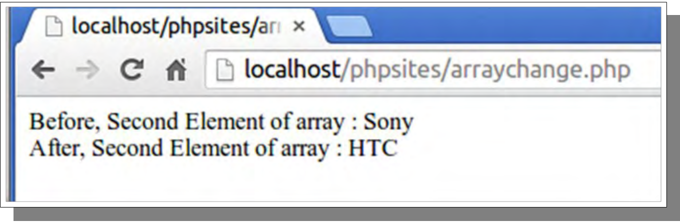
திரும்ப அளிக்கிறது. நாம் மேலே பார்த்த நான்கு செயல்கூறுகளும் எந்த array – யினுடைய சுட்டியை நகர்த்த வேண்டுமோ அந்த array – ஐ உள்ளீடாக எடுத்துக் கொள்கிறது.

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$linuxDistros =
array("Redhat" , "Debian" ,
"Slackware" , "Ubuntu" ,
"Fedora");
echo "elements 0 = ".
$linuxDistros[0];
echo "<br>";
echo "elements 1 = ".
$linuxDistros[1];
echo "<br>";
echo "elements 2 = ".
$linuxDistros[2];
echo "<br>";
echo "elements 3 = ".
$linuxDistros[3];
echo "<br>";
echo "elements 4 = ".
$linuxDistros[4];
echo "<br>";
```

```
echo “<b>Using Array
Pointers</b>”;
echo “<br>”;
echo “The Last element of
array is ” .
end($linuxDistros);
echo “<br>”;
echo “The Previous element
is ” . prev($linuxDistros);
echo “<br>”;
echo “The Previous element
is ” . prev($linuxDistros);
echo “<br>”;
echo “The First element of
array is ” .
reset($linuxDistros);
echo “<br>”;
echo “The Next element is
” . next($linuxDistros);
?>
```

வெளியீடு



12.9 *Array* யின் உறுப்புகளை மாற்றுதல், சேர்த்தல் மற்றும் நீக்குதல்(*Changing, Adding and Removing Array Elements*)

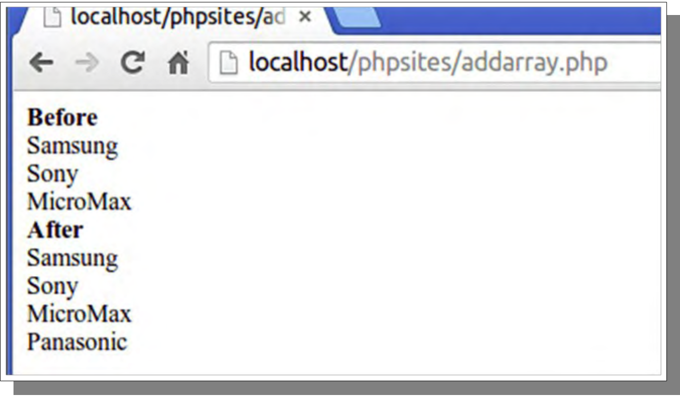
Array – யின் உறுப்பை மாற்றுதல் (change the element)

எந்த உறுப்பின் மதிப்பை நாம் மாற்றம் செய்ய வேண்டுமோ அந்த உறுப்பினுடைய சரியாக சுட்டியைக் கொண்டு அதற்கு புதிய மதிப்பை கொடுப்பதன் மூலம் அந்த உறுப்பின் மதிப்பை மாற்றலாம். இதற்கென தனியாக எந்த function கிடையாது.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$mobileBrands =
array(‘Samsung’ , ‘Sony’ ,
‘MicroMax’);
echo “Before, Second Element
of array : “.
$mobileBrands[1];
echo “<br>”;
$mobileBrands[1] = ‘HTC’;
echo “After, Second Element
of array : “.
$mobileBrands[1];
?>
```

வெளியீடு



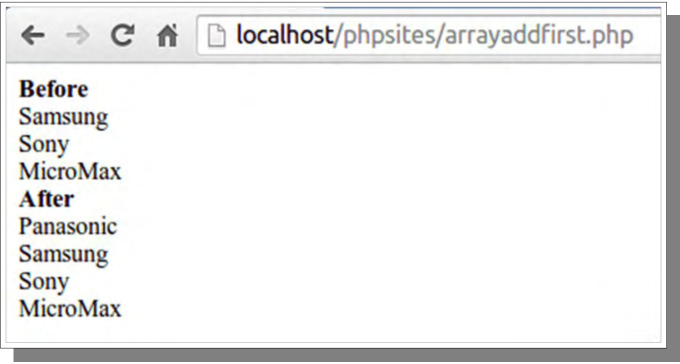
புதிய உறுப்பைச் சேர்த்தல் (Add a new element)

array_push() எனும் function ஐக் கொண்டு நாம் ஏற்கனவே இருக்கும் array யில் ஒரு புதிய உறுப்பைச் சேர்க்கலாம். array_push() function இரண்டு உள்ளீடுகளைப் பெற்றுக் கொள்கிறது. ஒன்று array யின் பெயர், மற்றொன்று புதிதாக இணைக்க வேண்டிய உறுப்பின் மதிப்பு.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$mobileBrands =
array('Samsung' , 'Sony' ,
'MicroMax');
echo "<b>Before</b>";
echo "<br>";
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo "<b>After</b>";
echo "<br>";
array_push($mobileBrands,
'Panasonic');
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo $mobileBrands[3];
echo "<br>";
?>
```

வெளியீடு



array_push() function *புதிய உறுப்பை* array *யில் கடைசியாக சேர்க்கும். முதலில் சேர்க்க வேண்டுமென்றால்* array_unshift() *எனும்* function *ஐப் பயன்படுத்த வேண்டும்.*

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$mobileBrands =
```

```
array('Samsung' , 'Sony' ,
'MicroMax');
echo "<b>Before</b>";
echo "<br>";
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo "<b>After</b>";
echo "<br>";
array_unshift($mobileBrands,
"Panasonic");
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo $mobileBrands[3];
echo "<br>";
?>
```

வெளியீடு

Array – யின் உறுப்பை நீக்குதல் (remove the array element)

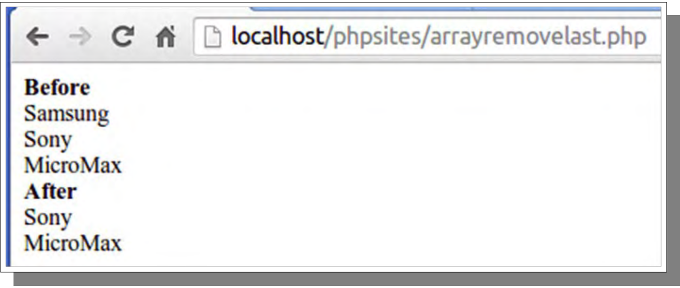
array_pop() function – *ஐப் பயன்படுத்தி* array – *யில் கடைசியாக இருக்கும் உருப்படியை நீக்கிவிடலாம்.*

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$mobileBrands =
array('Samsung' , 'Sony' ,
'MicroMax');
echo "<b>Before</b>";
echo "<br>";
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo "<b>After</b>";
echo "<br>";
array_pop($mobileBrands);
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
```

```
?>
```

வெளியீடு



array_shift() function -ஐக் கொண்டு array – யில் முதலாவதாக இருக்கும் உறுப்பை நீக்கிவிடலாம்.

```
<?php
$mobileBrands =
array('Samsung' , 'Sony' ,
'MicroMax');
echo "<b>Before</b>";
echo "<br>";
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
echo "<br>";
echo "<b>After</b>";
echo "<br>";
array_shift($mobileBrands);
echo $mobileBrands[0];
echo "<br>";
echo $mobileBrands[1];
echo "<br>";
echo $mobileBrands[2];
?>
```

12.10 *Looping* மூலம் *array* — யின் உறுப்புகளை அணுகுதல்(*Looping through array Elements*)

array – யின் உறுப்புகளை அணுகி படிப்பதற்கும், அதன் மதிப்புகளில் மாற்றங்கள் செய்வதற்கும் loop மூலமாக அணுகுவது அடிக்கடி அவசியமாகிறது. இதற்காக

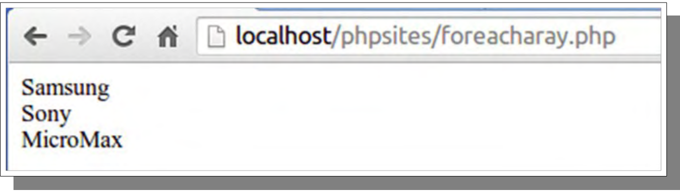
பயன்படுத்தப்படும் ஒன்றுதான் foreach loop. Foreach loop -ம் array – யின் உறுப்புகளை திரும்பத் திரும்ப அணுகுவதற்கு for அல்லது while loop ஐப் போன்றுதான் செயல்படுகிறது.

Foreach loop – ஐப் பயன்படுத்துவதற்கு இரண்டு வழிகள் இருக்கின்றது. முதலில் array – யின் தற்போதைய உறுப்பை ஒரு குறிப்பிட்ட variable (மாறி) -க்கு நிர்ணயித்து விட்டு அதன்பிறகு அதை loop – இன் body க்குள் பயன்படுத்திக் கொள்வது.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$mobileBrands =
array('Samsung' , 'Sony' ,
'MicroMax');
foreach( $mobileBrands as
$mobileBrandArrayValues ) {
echo
"$mobileBrandArrayValues
<br>";
}
?>
```

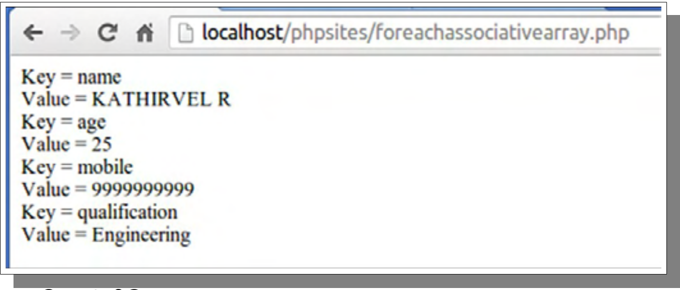
இதன் வெளியீடு கீழ்காண்பதைப் போன்று இருக்கும்



associative array – யின் உறுப்புகளை அணுகுவதற்கும் நாம் மேலே பார்த்த அதே முறைதான். சிறிய வித்தியாசம் என்னவென்றால். associative array – யில் key, value என்ற இரண்டு இருக்கும் ஆகையல் இங்கு key, value இரண்டிற்கும் variable – களை foreach loop – இல் அமைக்க வேண்டும்.

```
<?php
$myDetails =
array('name'=>'KATHIRVEL
R' , 'age'=>25 ,
'mobile'=>'9999999999′ ,'qu
alification'=>'Engineering')
;

foreach ( $myDetails as
$myDetailsKey =>
```

```
$myDetailsValue) {
echo “Key = $myDetailsKey
<br>”;
echo “Value =
$myDetailsValue <br>”;
}

?>
```



வெளியீடு

12.11 Replacing Sections of an Array

array_splice() function ஐப் பயன்படுத்தி array – யினுடைய மொத்த தொகுதியையும் மாற்ற முடியும். array_splice() function இரண்டு அத்தியாவசியமான உள்ளீடுகளையும், விரும்பினால் கொடுக்கக்கூடிய இரண்டு உள்ளீடுகளையும் பெற்றுக் கொள்கிறது. Array – யினுடைய பெயரை முதல் உள்ளீடாகவும், எந்த சுட்டியிலிருந்து தொடங்கி எந்த சுட்டி வரை முடிக்க வேண்டும் என்பதை இரண்டாவது மற்றும் மூன்றாவது உள்ளீடாக பெற்றுக் கொள்கிறது.

12.12 Array - யை வரிசைப்படுத்துதல்.

இரண்டு வரிசையில் வரிசைப்படுத்தலாம். ஒன்று ஏறுவரிசை மற்றொன்று இறங்கு வரிசை

எறுவரிசைக்கு sort() function – னும், இறங்கு வரிசைக்கு rsort() function -னும் பயன்படுத்தப்படுகிறது.

இரண்டு function -களுமே இரண்டு உள்ளீடுகளைப்

பெற்றுக் கொள்கின்றன. ஒன்று array யின் பெயர், மற்றொன்று எந்த நெறிமுறையில் (algorithm) வரிசைப்படுத்த வேண்டும் என்பது. மூன்று வகையான நெறிமுறைகள் உள்ளன. அவை

SORT_NUMERIC

SORT_STRING

SORT_REGULAR

எந்த நெறிமுறை என்று குறிப்பிடாதப் பட்சத்தில் SORT_REGULAR முறை பயன்படுத்தப்படும்.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$myArray =
array('KATHIRVEL', 9500,
'KARTHIK', 'ARIVAZHAGAN',
4598);
echo "<h2>Ascending
```

```
Order</h2>";
echo
"<b>SORT_NUMERIC</b><br>";
sort($myArray,
SORT_NUMERIC);

foreach ( $myArray as
$mySortArray) {
echo "$mySortArray<br>";
}

echo "<br>";
echo
"<b>SORT_STRING</b><br>";
sort($myArray, SORT_STRING);

foreach ( $myArray as
$mySortArray) {
echo "$mySortArray<br>";
}
echo "<br>";
echo
"<b>SORT_REGULAR</b><br>";
sort($myArray,
SORT_REGULAR);

foreach ( $myArray as
$mySortArray) {
echo "$mySortArray<br>";
}
```

```
echo “<h2>Descending
Order</h2>”;
echo
“<b>SORT_NUMERIC</b><br>”;
rsort($myArray,
SORT_NUMERIC);

foreach ( $myArray as
$mySortArray) {
echo “$mySortArray<br>”;
}

echo “<br>”;
echo
“<b>SORT_STRING</b><br>”;
rsort($myArray,
SORT_STRING);

foreach ( $myArray as
$mySortArray) {
echo “$mySortArray<br>”;
}

echo “<br>”;
echo
“<b>SORT_REGULAR</b><br>”;
rsort($myArray,
SORT_REGULAR);

foreach ( $myArray as
$mySortArray) {
```

```
echo “$mySortArray<br>”;
}
?>
```

வெளியீடு

localhost/phpsites/arraysorting.php
Ascending Order
SORT_NUMERIC
ARIVAZHAGAN
KATHIRVEL
KARTHIK
4598
9500
SORT_STRING
4598
9500
ARIVAZHAGAN
KARTHIK
KATHIRVEL
SORT_REGULAR
ARIVAZHAGAN
KARTHIK
KATHIRVEL
4598
9500

Descending Order

SORT_NUMERIC
9500
4598
KATHIRVEL
KARTHIK
ARIVAZHAGAN

SORT_STRING
KATHIRVEL
KARTHIK
ARIVAZHAGAN
9500
4598

SORT_REGULAR
9500
4598
KATHIRVEL
KARTHIK
ARIVAZHAGAN

12.13 *Associative Array - யை வரிசைப்படுத்துதல்*

இரண்டு வழிகளில் Associative Array – *யை வரிசைப்படுத்தலாம்*

1.key -*ஐக் கொண்டு வரிசைப்படுத்துதல்*

2.value -*ஐக் கொண்டு வரிசைப்படுத்துதல்*

Key – ஐக் கொண்டு வரிசைப்படுத்துதல்

ஏறுவரிசைக்கு ksort() function – *னும்*, *இறங்கு வரிசைக்கு* krsort() function – *னும் பயன்படுத்தப்படுகிறது*.

Value – ஐக் கொண்டு வரிசைப்படுத்துதல்

ஏறுவரிசைக்கு asort() function – *னும்* , *இறங்கு வரிசைக்கு* arsort() function – *னும்*

பயனபடுத்தப்படுகிறது. Sort மற்றும் rsort – இல் உள்ள syntax and options தான் இதற்கும், இதற்கென்று தனியாக எதுவுமில்லை.

12.14 *Array* – யைப் பற்றிய தகவல்களைப் பெறுதல் மற்றும் இதர *array* செயல்கூறுகள்(*functions*)

array – யைப் பற்றிய தகவல்களைப் பெறுவதற்கு பயனுள்ள பல function -கள் PHP யில் இருக்கின்றது. கீழே உள்ள அட்டவணையில் அவைகள் விளக்கங்களுடன் பட்டியலிடப்பட்டுள்ளது.

Function(செயல்கூறு)	**Description(விளக்கம்)**
Print_r	Array – யின் உறுப்புகளை வெளியிடுகிறது
array_keys	Associative array – யில் இருக்கும் key கள் அனைத்தையும் தருகிறது
array_search	நாம் தேடுவதற்காக கொடுக்கக்கூடிய மதிப்பு இருக்கும்பட்சத்தில், அந்த மதிப்புக்குரிய key – யை திருப்பித் தருகிறது.
array_values	Array – யில் இருக்கும் மதிப்புகள் அனைத்தையும் திருப்பித் தருகிறது.

in_array	குறிப்பிட்ட மதிப்பு array – யில் இருந்தால் true என்றும் இல்லையென்றால் false எனவும் திருப்பித் தருகிறது.
array_merge	இரண்டு அல்லது அதற்கும் மேற்பட்ட array – களை ஒரே array மாற்றுகிறது.
array_reverse	Array – யின் உறுப்புகளை தலைகீழாக மாற்றுகிறது.
Shuffle	Random வரிசையில் array உறுப்புகளை வரிசைப்படுத்துகிறது.

13 Working with Strings and Text in PHP

PHP என்ற நிரல் மொழி உருவாக்கப்பட்டதன் முக்கிய நோக்கமே web content களை திறம்பட கையாள்வதற்குத்தான். web content என்பது உரைகளை (text) அடிப்படையாகக் கொண்டது. ஆகையால் உரைகளைத் திறம்பட, எளிமையாக கையாள்வதற்காக பலதரப்பட்ட வசதிகளை (features) PHP கொண்டிருப்பதில் ஆச்சர்யப்படுவதற்கு ஒன்றுமில்லை.

உரைகளைக் கையாள்வதற்காக PHP வழங்கியுள்ள

பலதரப்பட்ட நுட்பங்களை இந்தப் பகுதியில் நாம் விரிவாக பார்க்க இருக்கிறோம். Web developer ஆக பணிபுரியும் ஒருவருக்கு உரைகளைக் கையாள்வதில் நிறைய வேலைகள் இருக்கும். அவற்றில் எழுத்துக்களை மாற்றுதல், உரையினுடைய ஒரு பகுதியை வேறொரு பகுதியைக் கொண்டு நிரப்புதல், உரைகளில் தேடுதல் ஆகியவைகள். இதையும் தாண்டி இன்னும் நிறைய வேலைகள் உரைகளைக் கையாள்வதில் இருக்கிறது. அதற்கான வழிகளையும் PHP ஏற்படுத்தி தருகிறது.

13.1 எழுத்துக்களை மாற்றுதல் (*Changing the Case of a PHP String*)

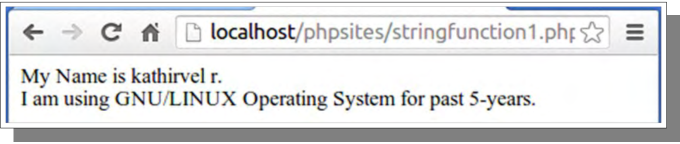
சரத்தில்(string) இருக்கக்கூடிய எழுத்துக்களில் மாற்றங்களை செய்தவதற்காக நிறைய செயல்கூறுகளை (function) PHP நமக்கு வழங்குகிறது. இந்த செயல்கூறுகள் மாற்றம் செய்யப்பட வேண்டிய சரத்தினை (string) உள்ளீடாகப் பெற்றுக் கொண்டு, மாற்றங்கள் செய்யப்பட்ட புதிய சரத்தினை நமக்கு வெளியீடாக தருகிறது. இதில் கவனிக்க வேண்டிய செய்தி என்னவென்றால் இதற்காகப் பயன்படுத்தப்படும் செயல்கூறுகள் அனைத்தும் nondestructive (சிதைவுறா), அதாவது உள்ளீடாகப் பெறும் அசல் சரத்தில் (Original String) எந்தவித மாற்றத்தையும்

செய்யாது. மாற்றம் செய்யப்பட்ட சரத்தினை ஒரு புதிய மாறியில் (variable) சேமித்து வைத்துக் கொண்டு தேவையான இடங்களில் பயன்படுத்திக் கொள்ளலாம்.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$myName = ‘KATHIRVEL R’;
$myOS = ‘GNU/Linux’;


$myNameInSmall =
strtolower($myName);
$myOsInUpper =
strtoupper($myOS);

echo “My Name is
$myNameInSmall.<br>”;
echo “I am using
$myOsInUpper Operating
System for past 5-
years.<br>”;
?>
```

வெளியீடு



மாற்றம் செய்யப்பட்ட சரம் (string) அசல் சரத்திலேயே சேமிக்க வேண்டுமென்றால். மாற்றத்தை புதிதாக ஒரு மாறியில் சேமிக்காமல் அசல் மாறியிலேயே சேமித்துவிட வேண்டியதுதான்.

```
<?php
$myName = ‘KATHIRVEL R’;
$myOS = ‘GNU/Linux’;

$myName =
strtolower($myName);
$myOS = strtoupper($myOS);

echo “My Name is
$myName.<br>”;
echo “I am using $myOS
Operating System for past 5-
years.<br>”;
?>
```

எழுத்துகளில் மாற்றங்களை ஏற்படுத்துவதற்காக PHP வழங்கியுள்ள செயல்கூறுகளும், அதன் செயல்களும் கீழே பட்டியலிடப்பட்டுள்ளது.

Strtolower() – முழு சரத்தினையும் சிற்றெழுத்தாக(lower case) மாற்றி தருகிறது.

Strtoupper() – முழு சரத்தினையும் பேரெழுத்தாக(upper case) மாற்றி தருகிறது.

Ucfirst() – வாக்கியத்தில் உள்ள முதல் எழுத்தை மட்டும் பேரெழுத்தாக மாற்றி தருகிறது.

Ucwords() – ஒவ்வொரு வார்த்தையிலும் இருக்கும் முதல் எழுத்தை மட்டும் பேரெழுத்தாக மாற்றி தருகிறது.

13.2 ASCII மதிப்புக்கு மாற்றுதல் மற்றும் ASCII மதிப்புகளிலிருந்து மாற்றுதல்

ASCII (American Standard Code for Information Interchange) மதிப்புகளுடன் வேலை செய்வதற்கான

வசதியையும் PHP நமக்கு வழங்கியிருக்கிறது. மொத்தம் 127 ASCII எழுத்துக்கள் உள்ளன (சிறப்புக் குறியீடுகளும் இதில் அடக்கம்).

ASCII யிலிருந்து மற்றும் ASCII க்கு மாற்றுவதற்காக இரண்டு செயல்கூறுகளை PHP வழங்கியுள்ளது. அவைகள்

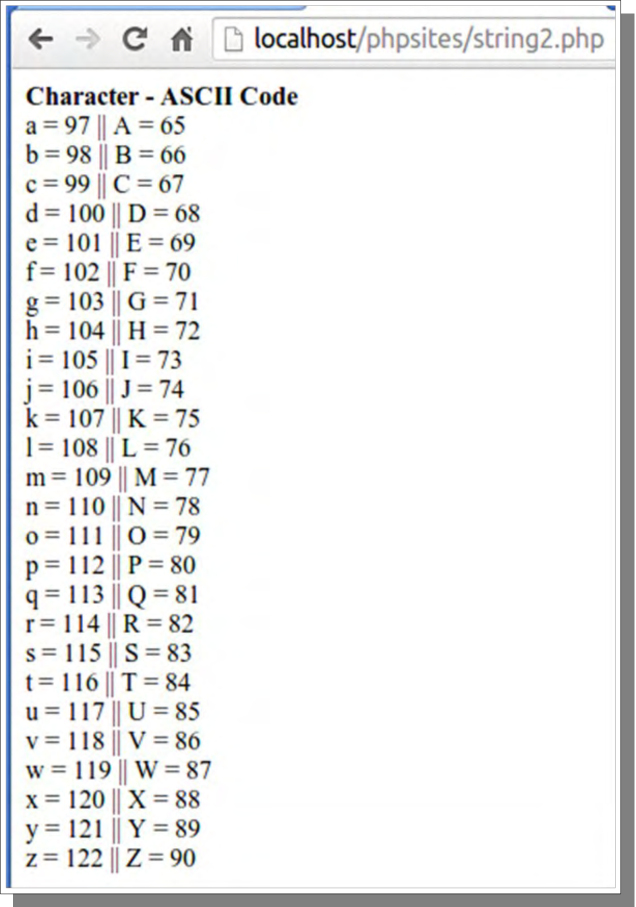
ord() – ஒரு character -ஐ உள்ளீடாக பெற்றுக் கொண்டு அதற்குச் சமமான ASCII code ஐ வெளியீடாக தருகின்றது.

Chr() – ஓர் ASCII character – ஐ உள்ளீடாக பெற்றுக் கொண்டு அதற்குச் சமமான charater -ஐ வெளியீடாக தருக்கிறது.

```
<?php
$smallCase =
'abcdefghijklmnopqrstuvwxyz'
;
$upperCase =
strtoupper($smallCase);
echo "<b>Character - ASCII
Code</b><br>";
```

```
for ( $i=0 ;
$i<strlen($smallCase) ; $i++
) {
echo "$smallCase[$i] = " .
ord($smallCase[$i]);
echo " || ";
echo "$upperCase[$i] = " .
ord($upperCase[$i]) .
"<br>";
}

?>
```

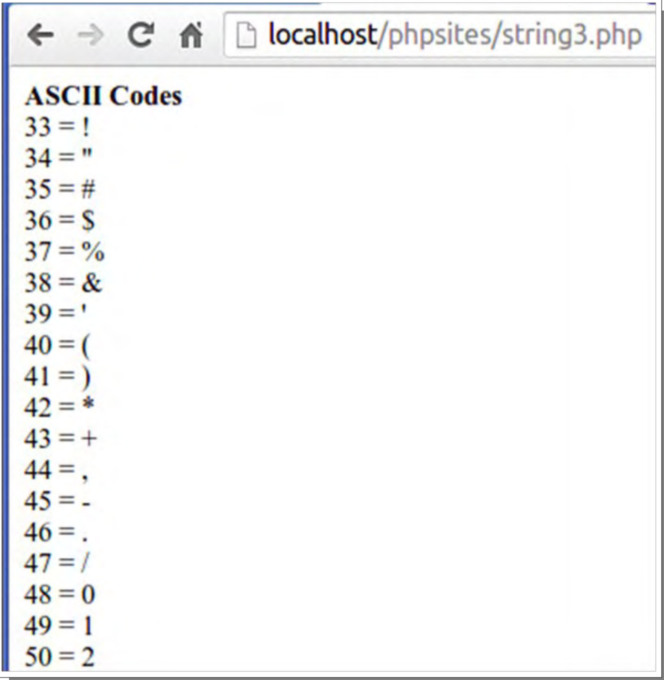
வெளியீடு


localhost/phpsites/string2.php
Character - ASCII Code
a = 97 || A = 65
b = 98 || B = 66
c = 99 || C = 67
d = 100 || D = 68
e = 101 || E = 69
f = 102 || F = 70
g = 103 || G = 71
h = 104 || H = 72
i = 105 || I = 73
j = 106 || J = 74
k = 107 || K = 75
l = 108 || L = 76
m = 109 || M = 77
n = 110 || N = 78
o = 111 || O = 79
p = 112 || P = 80
q = 113 || Q = 81
r = 114 || R = 82
s = 115 || S = 83
t = 116 || T = 84
u = 117 || U = 85
v = 118 || V = 86
w = 119 || W = 87
x = 120 || X = 88
y = 121 || Y = 89
z = 122 || Z = 90

ASCII to Character

```
<?php
echo “<b>ASCII
Codes</b><br>”;
for ( $i = 33 ; $i < 127 ;
$i++ ) {
echo “$i = “.chr($i).”<br>”;
}
?>
```

வெளியீடு


localhost/phpsites/string3.php
ASCII Codes
33 = !
34 = "
35 = #
36 = $
37 = %
38 = &
39 = '
40 = (
41 =)
42 = *
43 = +
44 = ,
45 = -
46 = .
47 = /
48 = 0
49 = 1
50 = 2

13.3 வடிவுறு சரங்களை அச்சிடுதல் (Printing Formatted Strings)

fprintf() function வடிவுறு(formatted) சரங்களை அச்சிட பயன்படுகிறது. இரண்டு அல்லது அதற்கு மேற்பட்ட உள்ளீடுகளைப் எடுத்துக் கொள்கிறது. கீழ்காணும் வடிவத்தில் இருக்கும்.

fprintf(“String”, variable1, variable2);

வடிவுறு செய்யப்பட்ட சரத்தை string கொடுக்கும், formatting specifiers இருக்கும் இடத்தில் அதற்கேற்றாற்போல் variable களின் மதிப்பு அளிக்கப்படும்.

printf Formatting Specifiers

formatting specifiers ‘%’ குறியீட்டுடன் ஆரம்பமாகும். அதனைத் தொடர்ந்து எந்த வகையான மதிப்புகள் அச்சிடப்பட வேண்டுமோ அதற்கான specifier இருக்கும். உதாரணமாக ஒரு decimal number ஐ அச்சிட வேண்டுமென்றால் அந்த இடத்தில் %d என இருக்கும்.

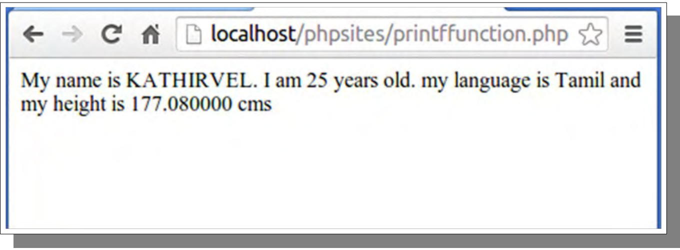
கீழ்காணும் அட்டவணையில் specifier அதற்கான விளக்கங்களும் கொடுக்கப்பட்டுள்ளது.

Specifier	Description
%%	சதவீத குறியீட்1 டை அச்சிடுகிறது.
%b	இரும எண்ணை544 அச்சிடுகிறது.
$c	ASCII மதிப்புக்குறிய character – ஐ அச்சிடுகி
%d	முழு எண்ணை அச்சிடுகிறது.
%e	Scientific notation (ex. 1.2e+5)
%u	Unsigned decimal number
%f	Floating point number
%F	Floating point number
%o	Octal number
%s	String
%x	Hexadecimal number
%X	Hexadecimal number

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
$myName = ‘KATHIRVEL’;
$myAge = 25;
$myLang = ‘Tamil’;
$myHeight = 177.08;
printf(“My name is %s. I am
%d years old. my language is
%s and my height is %f cms”,
$myName,$myAge,$myLang,
$myHeight);
?>
```

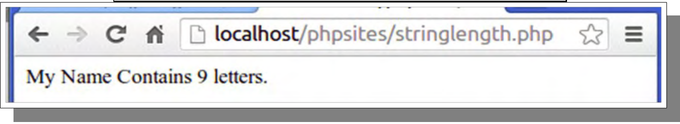
வெளியீடு



13.4 சரத்தின் நீளத்தை கண்டுபிடித்தல் (*Finding the Length of a String*)

ஒரு சரத்தின் நீளத்தைக் கண்டுபிடிக்க strlen() function பயன்படுகிறது. Strlen() function ஒரு சரத்தினை உள்ளீடாகப் பெற்றுக் கொண்டு அதனுடைய நீளத்தை வெளியீடாக தருகிறது.

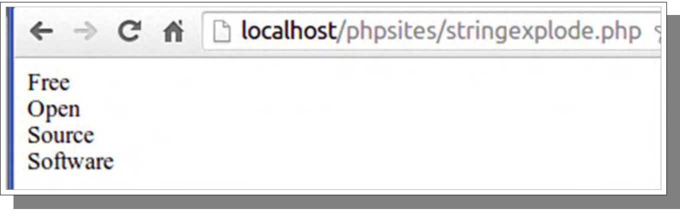
```
<?php
$myName = ‘KATHIRVEL’;
echo “My Name Contains ” .
strlen($myName) . ”
letters.”;
?>
```



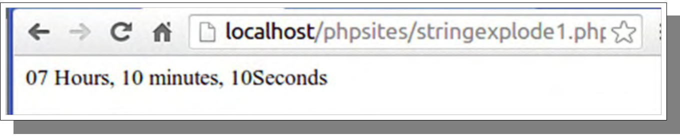
13.5 சரத்தை Array யாக மாற்றுதல் (*Converting a String into a Array*)

explode() function ஒரு சரத்தை array மாற்றுகிறது. Explode() function மூன்று உள்ளீடுகளை எடுத்துக் கொள்கிறது. **Delimeter –** எதை வைத்து array -யாக பிரிக்க வேண்டும் என்பதைக் குறிக்கிறது. உதாரணமாக, space or comman வைத்து பிரிப்பது. **String –** array யாக மாற்றப்பட வேண்டி சரம் (string). **divisions(விருப்பத்துக்குரியது) –** அதிகபட்சம் எத்தனை உறுப்புகளாக சரத்தை பிரிக்க வேண்டும் என்பதைக் குறிக்கிறது.

```
<?php
$foss = “Free Open Source
Software”;
$fossArray = explode(” “,
$foss);
foreach ($fossArray as $i) {
echo $i;
echo “<br>”;
}
?>
```




```
<?php
$timeNow = “07:10:55”;
$timeArray = explode(“:”,
$timeNow);
echo $timeArray[0] . ”
Hours, ” . $timeArray[1] . ”
minutes, ” . $timeArray[1] .
“Seconds”;
?>
```



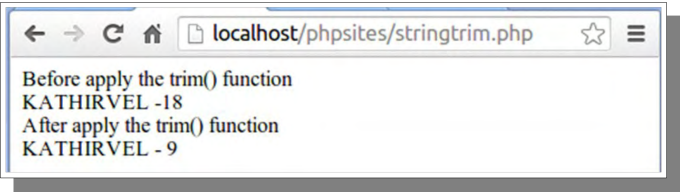
13.6 சரத்தின் முன்னும் பின்னும் இருக்கக்கூடிய *whitespace* ஐ நீக்குதல் (*Removing Leading and Trailing Whitespace from a String*)

ஒரு சரத்தின் முன்னும், பின்னும் இருக்கக்கூடிய வெற்றிடத்தை நீக்குவதற்கு trim() function பயன்படுகிறது. வெற்றிடமானது tab, space, newline, carriage return, NULL and vertical tab என எதுவாக வேண்டுமானாலும் இருக்கலாம். trim() function ஆனது string ஐ உள்ளீடாகப் பெற்று whitespace -ஐ நீக்கி அதனை வெளியீடாக தருகிறது.

```
<?php
$myName = ”    KATHIRVEL
“;
```

```
echo “Before apply the
trim() function<br>”;
echo “$myName
-“.strlen($myName).”<br>”;
echo “After apply the trim()
function<br>”;
echo trim($myName).” - ” .
strlen(trim($myName));
?>
```

வெளியீடு



13.7 சரங்களை ஒப்பிடுதல் (*Comparing Strings*)

web developing இல் இரண்டு சரங்களை ஒப்பிட்டுப் பார்ப்பது என்பது அடிக்கடி நடைபெறக்கூடிய ஒன்று.

சரங்களை ஒப்பிடுவதற்காக பல்வேறு வகையான செயல்கூறுகளை(functions) PHP வழங்கியிருக்கிறது. அவைகளின் பட்டியல் விளக்கங்களுடன் கீழே கொடுக்கப்படுள்ளது.

Strcmp() – இரண்டு சரங்களை உள்ளீடாகப் பெற்று case-sensitive ஒப்பிடுதலைச் செய்கிறது. பொருந்துவதைப் பொருத்து மதிப்புகளைத் திருப்பித் தருகிறது.

Strcasecmp() – இரண்டு சரங்களை உள்ளீடாகப் பெற்று case-insensitive ஒப்பிடுதலைச் செய்கிறது மற்றும் பொருந்துவதைப் பொறுத்து மதிப்புகளைத் திருப்பித் தருகிறது.

Strncmp() – மூன்று சரங்களை உள்ளீடாக ஏற்றுக்கொண்டு, அதில் இரண்டு ஒப்பிடுவதற்கான சரங்கள், மற்றொன்று எத்தனை character களை ஒப்பிட வேண்டும் என்ற எண்ணிக்கை. case-sensitive ஒப்பிடுதலைச் செய்து, ஒப்பிடுதலைப் பொறுத்த மதிப்புகளைத் திருப்பித் தருகிறது.

Strncasecmp() – மூன்று சரங்களை உள்ளீடாக ஏற்றுக்கொண்டு, அதில் இரண்டு ஒப்பிடுவதற்கான

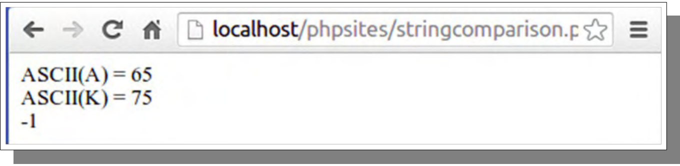
சரங்கள், மற்றொன்று எத்தனை character களை ஒப்பிட வேண்டும் என்ற எண்ணிக்கை. case-insensitive ஒப்பிடுதலைச் செய்து, ஒப்பிடுதலைப் பொறுத்து மதிப்புகளைத் திருப்பித் தருகிறது.

சரங்களை ஒப்பீடு செய்தலும் மதிப்புகளை திரும்ப பெறுதலும் (String Comparison Functions Return Value)

ASCII அடிப்படையிலான ஒப்பீடுகளையே சர ஒப்பீடு செயல்கூறுகள் (string comparison functions) மேற்கொள்கிறது. ஒவ்வொரு character -ஐயும் ASCII அடிப்படையிலேயே ஒப்பிடுகிறது. ஒப்பிடக்கூடிய இரண்டு strings களும் ASCII அடிப்படையில் பொருந்தினால் 0 எனும் மதிப்பை திருப்பி அளிக்கிறது. முதல் சரத்தின் ASCII மதிப்பு , இரண்டாவது சரத்தின் ASCII மதிப்பை விட குறைவாக இருந்தால் negative number -ஐ திருப்பி அளிக்கிறது. அதிகமாக இருந்தால் positive number – ஐ திருப்பி அளிக்கிறது.

```
<?php
$string1 = 'A';
$string2 = 'K';
```

```
echo “ASCII($string1) =
“.ord($string1);
echo “<br>”;

echo “ASCII($string2) =
“.ord($string2);
echo “<br>”;

echo strcmp($string1,
$string2);
?>
```



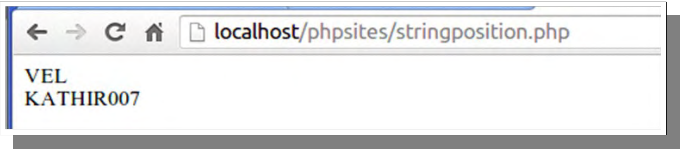
13.8 சரத்தை அணுகுதல் மற்றும் மாற்றுதல் (*Accessing and Modifying Characters in String*)

ஒரு சரத்தில் உள்ள ஒரு குறிப்பிட்ட உருவை அதனுடைய இருப்பநிலையைக் கொண்டு அணுக மற்றும்

மாற்ற முடியும். இதை செய்வதற்கு string variable – ஐத் தொடர்ந்து {} க்குள் தேவையான உருவின் இருப்புநிலையைக் கொடுக்க வேண்டும். இருப்பு நிலை 0 – விலிருந்தே ஆரம்பிக்கும் என்பதை மனதில் வைத்துக்கொள்ளவும். 1 – லிருந்து ஆரம்பிக்காது.

```
<?php
$myName = 'KATHIRVEL';
$myNewName = $myName{6}.
$myName{7}.$myName{8};

echo $myNewName;
echo "<br>";

$myName{6} = 0;
$myName{7} = 0;
$myName{8} = 7;

echo $myName;
?>
```

வெளியீடு



13.9 சரத்திற்குள் உருவை தேடுதலும் , பகுதிச்சரமாக பிரித்தலும் (*Searching for Characters and Substrings in a String*)

சரத்திற்குள் ஒரு குறிப்பிட்ட உருவைத் தேடும் வசதியை PHP நமக்கு வழங்கியிருக்கிறது. இதை substring என்று சொல்லுவோம். இதைச் செய்வதற்கு strpos() மற்றும் strrpos() ஆகிய செயல்கூறுகள் பயன்படுகிறது.

Strpos() செயல்கூறு மூன்று உள்ளீடுகளைப் பெற்றுக் கொள்கிறது. அதில் இரண்டு கட்டாயமானதாகவும், ஒன்று விருப்பத்துக்கு உரியதாகவும் இருக்கிறது. நாம் எந்த சரத்திற்குள் தேடுதலைச் செய்ய வேண்டுமோ அந்த சரத்தை முதல் உள்ளீட்டிலும், தேடவேண்டிய சரத்தை இரண்டாவது உள்ளீட்டிலும் கொடுக்க வேண்டும். தேடுதலை சரத்தினுடைய எந்த நிலையிலிருந்து தொடங்க வேண்டும் என்பதை மூன்றாவது உள்ளீட்டிலும்

கொடுக்க நாம் விரும்பினால் கொடுக்கலாம்.

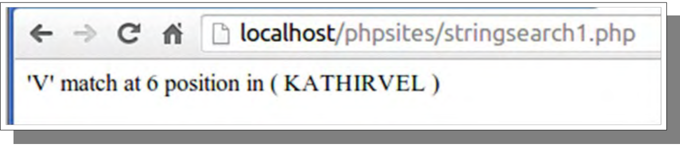
தேடுதல் பொருந்தினால் எந்த நிலையில் பொருந்தியதோ அந்த நிலையையும், பொருந்தாவிட்டால் 0 எனும் பூலியன் மதிப்பையும் திரும்பத்தரும். முதல் இருப்புநிலையிலேயே பொருந்தி விட்டால் 0(Numeric) என்பதை வெளியீடாகத் தரும், பொருந்தாவிட்டால் பூலியன் 0 வைத் (Boolean 0) திரும்பத் தரும் இரண்டும் ஒன்றல்ல. இந்த பிரச்சினையை சரி செய்ய நாம் === (Identically equal) மற்றும் !== (Identically not equal) வினைக்குறிகளைப் பயன்படுத்திக் கொள்ளலாம். Operator எனும் தலைப்பில் இந்த வினைக்குறிகளைப் பற்றி பார்த்திருக்கிறோம். ஞாபகம் வருகிறதா?

நிரல்

```
<?php
$myName = “KATHIRVEL”;
$searchStr = “V”;
```

```
if ( strpos($myName,
$searchStr) !== false ) {
echo "'$searchStr' match at
" . strpos($myName,
$searchStr) . " position in
( $myName )";
}

else {
echo "Match Not Found";
}

?>
```

வெளியீடு



13.10 Extracting and Replacing Substrings

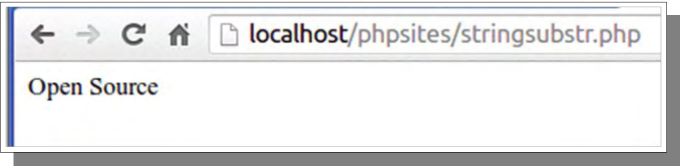
substr() மற்றும் substr_replace() செயல்கூறுகளைப் பயன்படுத்தி சரத்தினுடை உருக்களைப் பிரித்து எடுக்கலாம், அல்லது மாற்றி அமைக்கலாம்.

substr() செயல்கூறு இரண்டு உள்ளீடுகளைப் பெறுகிறது. ஒன்று ஆதாரச் சரம்(source string), மற்றொன்று எந்த சுட்டியிலிருந்து சரத்தை பிரிக்க வேண்டும் என்பது. நீங்கள் விரும்பினால் எவ்வளவு நீளத்துக்கு பிரிக்க வேண்டும் என்பதைக் கொடுத்துக் கொள்ளலாம்.

நிரல்

```
<?php
$foos = “Free Open Source
Software”;
$fossSub = substr($foos, 5,
11);
echo $fossSub;
?>
```

வெளியீடு

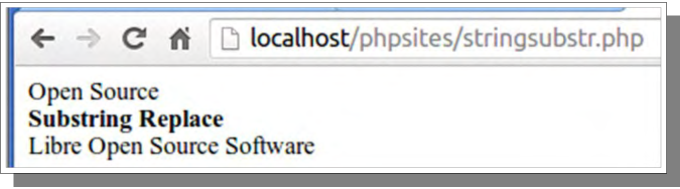


substr_replace() function *நான்கு உள்ளீடுகளைப் பெற்றுக் கொள்கிறது. முதலாவது மூலச்சரம், இரண்டவது மாற்ற வேண்டியச் சரம், மூன்றாவதாக மூலச்சரத்தில் எந்த நிலையிலிருந்து மாற்ற வேண்டும் என்ற விபரம், நான்காவதாக எவ்வளவு நீளத்துக்கு மூலச்சரத்தை எடுத்துவிட்டு மாற்ற வேண்டும் என்பது.*

```
<?php
$foos = “Free Open Source
Software”;
$fossSub = substr($foos, 5,
11);

echo $fossSub;
echo “<br>”;
echo “<b>Substring
Replace</b>”;
```

```
$fossRep = “Libre”;
echo “<br>”;
echo substr_replace($foos,
$fossRep,0,4);
?>
```

வெளியீடு

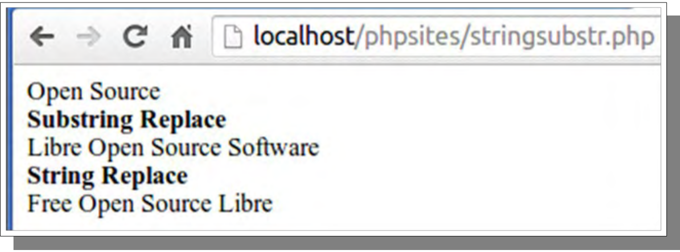


13.11 *Replacing All Instances of a Word in a String*

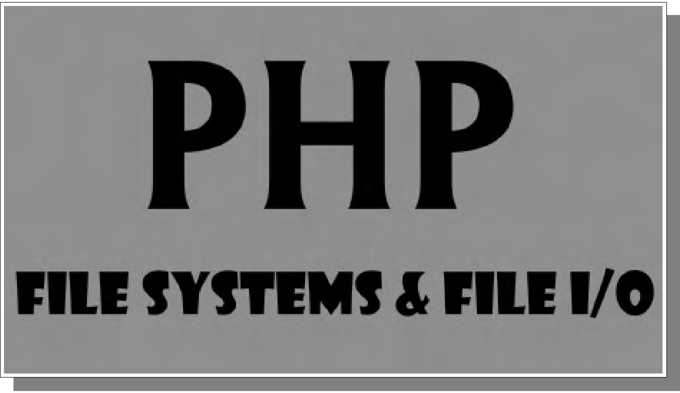
சரத்தில் இருக்கும் வார்த்தையை முழுமையாக Replace செய்தல். இந்த வேலையைச் செய்ய str_replace() function பயன்படுகிறது. மூன்று கட்டாய உள்ளீடுகளையும், ஒரு விருப்ப உள்ளீடையும் எடுத்துக்

கொள்கிறது. முதல் உள்ளீட்டில் மாற்றப்பட வேண்டிய சரத்தையும், இரண்டாவது உள்ளீட்டில் புதிதாக மாற்ற வேண்டிய சரத்தையும், மூன்றாவது உள்ளீட்டில் மூலச்சரத்தையும் கொடுக்க வேண்டும்.

```
<?php
$foss = “Free Open Source
Software”;
$fossSub = substr($foss, 5,
11);

echo $fossSub;
echo “<br>”;
echo “<b>Substring
Replace</b>”;
$fossRep = “Libre”;

echo “<br>”;
echo substr_replace($foss,
$fossRep,0,4);
echo “<br>”;
echo “<b>String
Replace</b>”;
$fossRep = “Libre”;

echo “<br>”;
echo str_replace(“Software”,
$fossRep, $foss);
?>
```


localhost/phpsites/stringsubstr.php
Open Source
Substring Replace
Libre Open Source Software
String Replace
Free Open Source Libre

14 கோப்பு முறைமையும், கோப்புகள் உள்ளீடும் / வெளியீடும் (*File systems and File I/O*)



PHP server side scripting ஆக இருப்பதில் என்ன பலனென்றால், web developer சேவையகத்தினுடைய (server) கோப்பு முறைமையை எளிமையாக அணுகுவதற்கான வசதிகளை ஏற்படுத்தித் தருகிறது. கோப்புகளை உருவாக்குவது, திறப்பது, நீக்குவது மற்றும் கோப்புகளில் எழுவது போன்ற வசதிகளை நமக்கு PHP உருவாக்கித் தருகிறது. மேலும், அடைவுகளுக்குள் பயணிப்பது, அடைவுகளை பட்டியலிடுவது, புதிய அடைவுகளை உருவாக்குவது போன்ற வேலைகளையும் செய்ய முடியும்.

14.1 கோப்புகளை திறத்தலும் உருவாக்குதலும் (*Opening and Creating Files*)

ஏற்கனவே இருக்கக்கூடிய ஒரு கோப்பை திறப்பதற்கும், புதிதாக ஒரு கோப்பை உருவாக்குவதற்கும் fopen() function பயன்படுகிறது. Fopen() function கோப்புகளை கையாள்வதற்கு இரண்டு உள்ளீடுகளைப் பெற்றுக்கொள்கிறது. முதலாவது உள்ளீட்டில் திறக்க வேண்டிய கோப்பின் பெயரை கொடுக்க வேண்டும். கோப்பின் முழு பாதையையும் உள்ளீடாக கொடுக்க

வேண்டும். கோப்பின் பாதையானது சேவையகத்தின் கோப்பு முறைமையோடு தொடர்புடையது. இணைய வழங்கியின்(web server) root -டோடு தொடர்புடையதல்ல. இரண்டாவது உள்ளீட்டில் எந்த பண்புடன்(create, read only, write only etc) கோப்பைத் திறக்க வேண்டும் என்பதை கொடுக்க வேண்டும்.

கீழே உள்ள அட்டவணையில் கோப்பினுடைய பண்புகள் முழு விபரங்களுடன் கொடுக்கப்பட்டுள்ளது.

Mode (முறைமை)	**Desc**
R	Read only access. கோப்பினுடைய தொடக்கத்
R+	Read and Write access. கோப்பினுடைய தொ
W	Write only access. கோப்பினுடைய தொடக்க உருவாக்கப்படவில்லையென்றால் , புதிதாக உ
W+	Read and Write access. கோப்பினுடைய தொ உருவாக்கப்படவில்லையென்றால் , புதிதாக உ
A	Write only access. கோப்பினுடைய இறுதியி உருவாக்கப்படவில்லையென்றால் , புதிதாக உ
A+	Read and write access. கோப்பினுடைய இறு உருவாக்கப்படவில்லையென்றால் , புதிதாக உ
X	Create and open for write only. கோப்பினு இல்லையென்றால் false எனும் மதிப்பை திரும்
X+	Create and open for read and write. கோப்பி

	இல்லையென்றால் false எனும் மதிப்பை திரும்

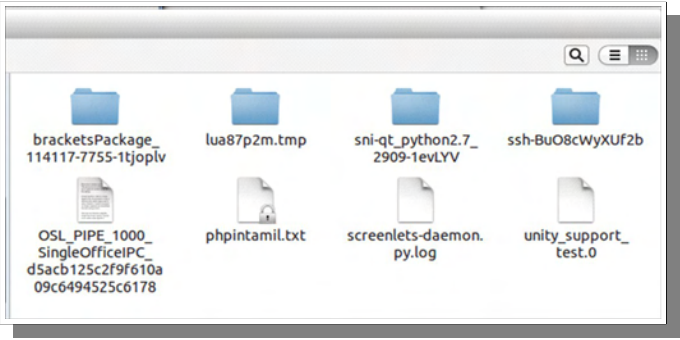
14.2 கோப்புகளை மூடுதல் (Closing Files)

கோப்பு ஒருமுறை திறக்கப்பட்டுவிட்டால் அந்த கோப்பை fclose() function -ஐ பயன்படுத்தி மூட முடியும். fclose() function ஒரே ஒரு உள்ளீடை மட்டும் பெற்றுக்கொள்கிறது.

மேலே நாம் பார்த்த தகவல்களைக் கொண்டு ஒரு நிரலை உதாரணமாகப் பார்ப்போம்.

```
<?php
$fileHandle =
fopen('/tmp/phpintamil.txt',
'w+') or die("Can't open the
file");
fclose($fileHandle);
?>
```

வெளியீடு



மேலே உள்ள நிரல் /tmp/ அடைவிற்குள் phpintamil.txt எனும் கோப்பை உருவாக்குகிறது. இங்கு நாம் w+ எனும் பண்பைப் பயன்படுத்தியிருக்கிறோம். w+ பண்பு கோப்பு ஏற்கனவே உருவாக்கப்படவில்லையென்றால், புதிதாக ஒரு கோப்பை உருவாக்குகிறது. படித்தல் மற்றும் எழுதுதல் அனுமதியையும் அளிக்கிறது. Fclose() function கோப்பை மூடுகிறது.

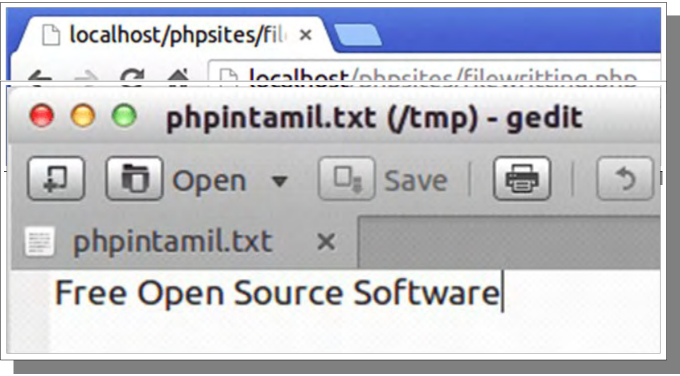
14.3 கோப்பில் எழுதுதல் (*Writing to a File*)

கோப்பு உருவாக்கப்பட்டு, திறக்கப்பட்டவுடன் அடுத்த வேலை என்னவென்றால் அந்த கோப்பில் தகவல்களை எழுதுவது. Fwrite() மற்றும் fputs() funtions இந்த வேலையைச் செய்ய உதவுகிறது. Fwrite() இரண்டு உள்ளீடுகளைப் பெற்று கொள்கிறது. முதலாவதாக Fopen() function க்கான variable – ஐயும், இரண்டாவதாக கோப்பில் எழுதுவதற்குண்டான தகவல் சரத்தையும் எடுத்துக் கொள்கிறது.

```
<?php
$myFile =
fopen('/tmp/phpintamil.txt'
, 'w+') or die("Can't Open
the file.");
$myFileWrite = fwrite
( $myFile, "Free Open Source
Software" );



if ( $myFileWrite ) {
echo "Data Written
Successfully.<br>";
}
```

```
else {
echo “Data Write
Failed.<br>”;
}

fclose($myFile);
?>
```

வெளியீடு



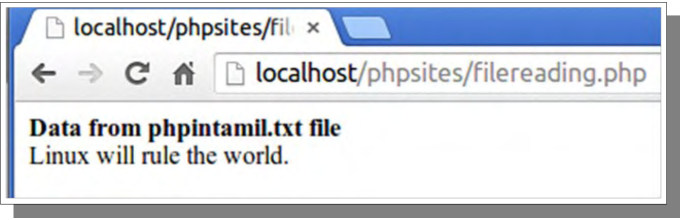
14.4 கோப்பிலிருந்து தகவல்களைப் படித்தல் (*Reading From a File*)

fread() function ஐ பயன்படுத்தி கோப்பிலிருந்து தகவல்களை படிக்க முடியும். fread() function இரண்டு உள்ளீடுகளை பெற்றுக் கொள்கிறது. முதலாவதாக கோப்பைத் திறப்பதற்கான variable – ஐயும், இரண்டாவதாக எத்தனை byte – களை கோப்பிலிருந்து படிக்க வேண்டும் என்பதையும் பெற்றுக் கொள்கிறது.

```
<?php
$fileOpen =
fopen('/tmp/phpintamil.txt'
, 'w+') or die ("Can't Open
the File");
fwrite ($fileOpen, "Linux
will rule the world.");
fclose($fileOpen);
$fileOpen =
fopen('/tmp/phpintamil.txt'
, 'r') or die ("Can't open
the file.");
$fileRead = fread
($fileOpen, 1024);
echo "<b>Data from
phpintamil.txt file</b><br>"
```

```
    . $fileRead;
    ?>
```

வெளியீடு



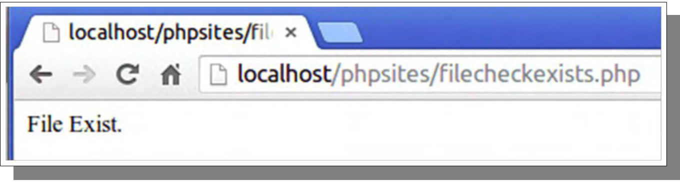
இங்கு die() function *எதற்கு பயன்படுத்தப்படுகிறதென்றால், ஒருவேளை கோப்பு திறக்கப்பட முடியவில்லையென்றால் அதில் கொடுக்கப்பட்டுள்ள செய்தியை வெளியிடும். இது மற்ற* function – *கள் கோப்பைத் திறப்பதற்கு முற்படுவதைத் தடுக்கிறது.*

14.5 கோப்பு இருக்கிறதா என சோதித்தல் (*Checking Whether a File Exists*)

கோப்பு முறைமையில் கோப்பு இருக்கிறதா இல்லையா என்பதை சோதிப்பதற்கு file_exists() function பயன்படுகிறது. கோப்பினுடைய path – ஐ மட்டும் file_exists() function பெற்றுக்கொள்கிறது. கோப்பு இல்லையென்றால் false என்பதையும் , கோப்பு இருந்தால் true என்பதையும் வெளியீடாக தருகிறது.

```
<?php

if
( file_exists('/tmp/phpintam
il.txt') ) {
echo "File Exist.";
}

else {
echo "File Doen't Exist.";
}

?>
```

வெளியீடு



14.6 கோப்புகளை பிரதியெடுத்தல், நகர்த்துதல் மற்றும் அழித்தல் (*Moving, Copying and Deleting Files*)

copy() function *கோப்புகளை பிரதியெடுக்கவும்*, rename() function *பெயரை மாற்றவும்*, unlink() function *கோப்பை நீக்கவும் பயன்படுகிறது*.

Copy

```
<?php
if
( copy(‘/tmp/practice.old’ ,
‘/tmp/practice’) ) {
echo “Copy
```

```
Successfully<br>”;
}
?>
```

Rename

```
<?php
if
( rename( ‘/tmp/practice.txt
’ , ‘/tmp/practice.old’) ) {
echo “Renamed
Successfully<br>”;
}
?>
```

Delete

```
<?php
if
( unlink(‘/tmp/practice.txt’
) ) {
echo “Delete
Successfully<br>”;
}
```

```
?>
```

14.7 கோப்புகளின் பண்புகளை அணுகுதல்(*Accessing File Attributes*)

கோப்பு எப்பொழுது உருவாக்கப்பட்டது, கோப்பின் அளவு, கோப்பு படிக்கக்கூடியதாக இருக்கிறதா அல்லது இல்லையா என்பவைகளைப் போன்று கோப்பின் பல்வேறு பண்புகளைப் அணுகுவதற்கு PHP வழிவகை செய்கிறது.

கோப்புகளைப் பற்றிய முழு விபரங்களையும் PHP யினுடைய stat() மற்றும் fstat() செய்லகூறுகள்(functions) நமக்கு அளிக்கின்றன. கோப்புகளைப் பற்றிய நிறைய விபரங்களை அளிப்பதால், அந்த தகவல்கள் ஒரு associative array -யில் சேமிக்கப்படுகிறது. அந்த array யிலிருந்து நாம் நமக்கு தேவையான தகவல்களை மட்டும் பெற்றுக்கொள்ளலாம்.

stat() மற்றும் fstat() ஆகிய இரண்டு function களும் ஒற்றை உள்ளீட்டையே பெற்றுக்கொள்கின்றன. Stat()

function -க்கு கோப்பினுடைய முழு பாதையையும்(full path of file), fstat() function -க்கு fopen() மூலம் ஒரு மாறியில் மதிப்பை கொடுத்துவிட்டு அதன்பின் அந்த மாறியின் மதிப்பை உள்ளீடாக கொடுக்க வேண்டும்.

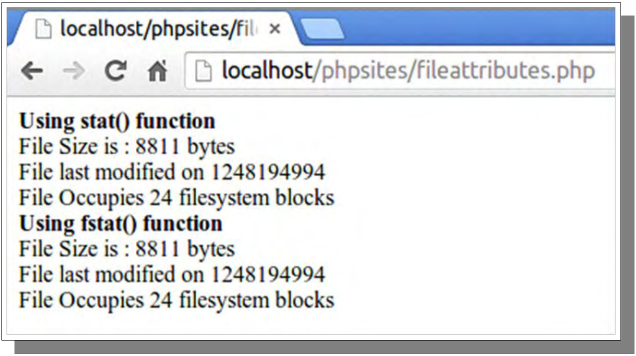
Key	**Description**
Dev	Device Number
Ino	Inode number
Mode	Inode protection mode
Nlink	Number of links
Uid	User ID of owner
Gid	Group ID of owner
Rdev	Inode device type
Size	Size in bytes
Atime	Last access (Unix timestamp)
Mtime	Last modified (Unix timestamp)
Ctime	Last inode change (Unix timestamp)
Blksize	Blocksize of filesystem IO (platform dependent)
Blocks	Number of blocks allocated

கீழ்காணும் நிரலைப் பாருங்கள்

```
<?php
$results = stat
("/tmp/phpintamil.txt");
$fileNew =
fopen("/tmp/phpintamil.txt",
'r');
$fileDetails =
fstat($fileNew);
echo "<b>Using stat()
function</b><br>";
echo "File Size is :
$results[size] bytes<br>";
echo "File last modified on
$results[mtime]<br>";
echo "File Occupies
$results[blocks] filesystem
blocks<br>";
echo "<b>Using fstat()
function</b><br>";
echo "File Size is :
$fileDetails[size]
bytes<br>";
echo "File last modified on
$fileDetails[mtime]<br>";
echo "File Occupies
```

```
$fileDetails[blocks]
filesystem blocks<br>";
fclose($fileDetails);
?>
```

நிரலின் வெளியீடு



மேலும், கோப்புகளின் அணுகுதல் அனுமதிகளையும் (access rights) நாம் தெரிந்து தெரிந்து கொள்ள முடியும். is_readable() மற்றும் is_writable() ஆகிய இரண்டு function களும் இதற்கு பயன்படுகின்றன. கோப்பினுடைய

பாதையை உள்ளீடாகப் பெற்றுக்கொண்டு true or false ஆகிய மதிப்புகளில் ஏதேனும் ஒன்றை வெளியீடாக தருகிறது.

14.8 வெளியீட்டு வைப்பகம் (*Output Buffering*)

தகவல்தளத்திலிருந்து தகவல்களைப் பெறுவதற்கு தாமதமாகும் நேரங்களில் பயனருக்கு தகவலை தெரிவிக்கவும் நேரடியாக உள்ளடக்கங்களை output stream -க்கு அனுப்பவும் output buffering mechanism பயன்படுகிறது.

Output Buffering ஐத் தொடங்க ob_start() function பயன்படுத்தப்படுகிறது. ob_start() function -க்கு எந்தவொரு உள்ளீட்டை அளிக்காமலும் நாம் பயன்படுத்தலாம். ஆனாலும் மூன்று optional உள்ளீடுகளைப் கொடுக்கலாம்.

1.callback funtion

2.bytes

3.delete buffer

Buffer-னுடைய தகவல்கள் ob_flush() function -ஐப் பயன்படுத்தி வெளித்தள்ளப்படுகிறது. இதற்கு ob_end_flush() function ஐயும் இதற்கு பயன்படுத்திக் கொள்ளலாம்.

ob_clean() function ஐப் பயன்படுத்தி buffer இன் தகவல்களை நம்மால் அழிக்க முடியும். ob_get_contents() function -ஐப் பயன்படுத்தி buffer -இல் இருக்கும் தகவல்களை பெற்றுக்கொள்ளலாம்.

கீழே இருக்கும் நிரலைப் பாருங்கள்

```
<?php
echo "<b>Before Using
ob_start()
function</b><br>";

ob_start(); //start
```

```
buffering
echo “This content will be
buffered<br>”; //write to
the buffer

echo “<b>Display buffered
content using
ob_get_contents()
function</b></br?”;
echo “<br>” .
ob_get_contents();
echo “<br>”;
ob_end_flush(); //flush the
output from the buffer

echo “<b>After Using
ob_end_flush()
function</b><br>”;
echo ob_get_contents();
?>
```

இதன் வெளியீடு

localhost/phpsites/fil
localhost/phpsites/filebuffering.php
Before Using ob_start() function
This content will be buffered
Display buffered content using ob_get_contents() functionThis content will be buffered
Display buffered content using ob_get_contents() functionAfter Using ob_end_flush() function
Before Using ob_start() function
This content will be buffered
Display buffered content using ob_get_contents() functionThis content will be buffered
Display buffered content using ob_get_contents() functionAfter Using ob_end_flush() function

15 அடைவுகளுடன் பணியாற்றுதல் (*Working with Directories*)

கோப்புகளைப் கையாளுவது எப்படி? என்று முந்தைய பகுதியில் பார்த்தோம். இந்த பகுதியில் PHP யில் அடைவுகளை கையாளுவது எப்படி? என்று பார்ப்போம். புதிதாக ஒரு அடைவை உருவாக்குதல், ஏற்கனவே இருக்கும் ஒரு அடைவை நீக்குதல், அடைவுகளுக்குள் இருக்கும் கோப்புகளை பார்வையிடுதல் என நிறைய function கள் PHP யில் இருக்கின்றன.

15.1 புதிதாக அடைவுகளை உருவாக்குதல் (Creating Directories)

mkdir() function ஐப் பயன்படுத்தி நாம் புதிதாக ஒரு அடைவை உருவாக்கலாம். தற்போது இருக்கும் அடைவுக்குள்ளே புதிதாக ஒரு அடைவை உருவாக்க வேண்டுமானால் நேரடியாக புதிய அடைவை பெயரை mkdir() function க்கு உள்ளீடாக கொடுத்துவிடலாம். வேறொரு அடைவிற்குள் புதிதாக ஒரு அடைவை உருவாக்க வேண்டுமானால் எங்கு புதிய அடைவு உருவாக்கப்பட வேண்டுமோ அதனுடைய முழு பாதையையும் (full path) கொடுக்க வேண்டும்.

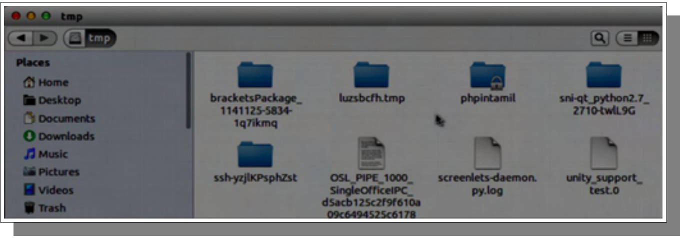
நீங்கள் விரும்பினால் அடைவிற்கான அனுமதியையும் இரண்டாவது உள்ளீடாக கொடுக்கலாம்.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
//create a new directory
using PHP
$yourDirectoryName =
“/tmp/phpintamil”;
```

```
if
( mkdir($yourDirectoryName)
) {
echo “$yourDirectoryName is
successfully created.<br>”;
}
else {
echo “Directory creation
failed.<br>”;
}
?>
```

வெளியீடு



15.2 அடைவை நீக்குதல் (Deleting Directory)

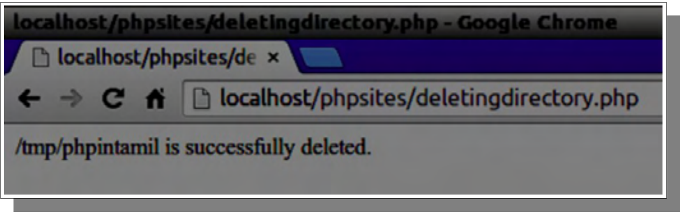
rmdir() function ஐ பயன்படுத்தி அடைவுகள் அழிக்கப்படுகின்றது. எந்த அடைவை நாம் அழிக்க வேண்டுமோ அந்த அடைவின் பெயரை உள்ளீடாக கொடுக்க வேண்டும். அடைவு காலியாக இருந்தால் மட்டுமே அடைவு அழிக்கப்படும். அடைவிற்குள் ஏதேனும் கோப்புகளோ அல்லது துணை அடைவுகளோ இருந்தால் அடைவானது அழிக்கப்படமாட்டாது. அடைவிற்குள் இருப்பவைகள் அழிக்கப்பட்டு காலியாகிய பின்புதான் அடைவை அழிக்க முடியும்.

கீழே உள்ள நிரலைப் பாருங்கள்

```
<?php
//create a new directory
using PHP
$yourDirectoryName =
"/tmp/phpintamil";
if
```

```
( rmdir($yourDirectoryName)
) {
echo “$yourDirectoryName is
successfully deleted.<br>”;
}
else {
echo “Can’t delete the
directory.<br>”;
}
?>
```

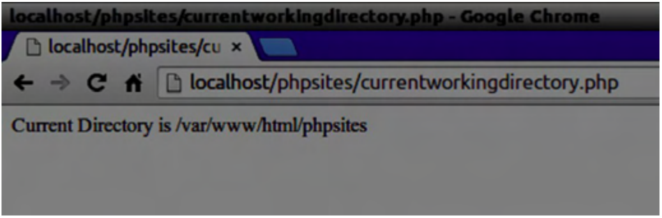
வெளியீடு



15.3 Finding and Changing the Current Working Directory

தற்போது நாம் இருக்கும் அடைவின் பாதையை getCwd() function ஐ பயன்படுத்தி கண்டுபிடிக்கலாம்.

```
<?php
$currentDirectroy =
getCwd();
echo “Current Directory is
$currentDirectroy”;
?>
```

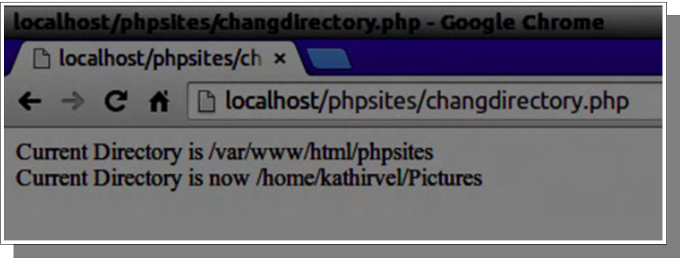


வெளியீடு

chdir() funtion ஐ பயன்படுத்தி நாம் விரும்பிய அடைவிற்குள் மாற்றிக் கொள்ளலாம். அடைவின் பாதையை மட்டும் உள்ளீடாக கொடுக்க வேண்டும்.

```
<?php
$currentDirectroy =
getCwd();
echo “Current Directory is
$currentDirectroy<br>”;
$changeDirectory =
“/home/kathirvel/Pictures”;
chdir($changeDirectory);
$currentDirectroy =
getCwd();
echo “Current Directory is
now $currentDirectroy”;
?>
```

வெளியீடு

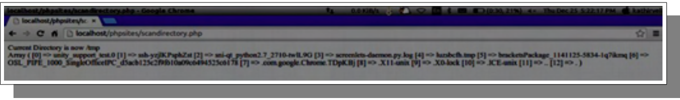


15.4 அடைவிற்குள் இருக்கும் கோப்புகளை பட்டியலிடுதல் (*Listing Files in a Directory*)

அடைவுகளுக்குள் இருக்கும் கோப்புகளை scandir() function ஐப் பயன்படுத்தி பட்டியலிடலாம். scandir() இரண்டு உள்ளீடுகளைப் பெற்றுக் கொள்கிறது. அடைவினுடைய பாதையை முதலாவது உள்ளீடாகவும், எந்த வரிசையில் கோப்பு பட்டியலிடப்பட வேண்டும் என்பதை இரண்டாவது உள்ளீடாகவும் பெற்றுக் கொள்கிறது. 0 என்றால் alphabetical முறையிலும், 1 என்றால் reverse-alphabetical முறையிலும் வரிசைப்படுத்துகிறது.

```php
<?php
chdir("/tmp");
$currentDirectory =
getCwd();
echo "Current Directory is
now $currentDirectory<br>";
$dirArray = scandir(".",
1 );
print_r($dirArray);
?>
```

வெளியீடு



வலை அடிப்படையிலான(web based) பயன்பாட்டில்(application) பெரும்பகுதி இணைய உலாவியின் மூலமாக பயனருடன் தொடர்பு கொள்வதற்காகவே செலவிடப்படுகிறது. இவ்வாறு உருவாக்கப்படும் web based application -இல் அதிகமாகவும், அடிக்கடியும் செய்யும் வேலை என்னவென்றால், பயனரிடமிருந்து தகவல்களை பெறுவதற்காக படிவங்களை(forms) காண்பிப்பதும், அந்த படிவம் மூலமாக பெறப்படும் தகவல்களை செயல்படுத்துவதும்தான்.

HTML <form> tag ஐப் பயன்படுத்தி வலைபடிவங்கள்(web forms) உருவாக்கப்படுகிறது. PHP மற்றும் HTML form களுக்கிடையே தகவல்களை பரிமாறுவதைப் பற்றி பார்ப்பதற்கு முன், HTML form ஐப் பற்றிய அடிப்படைகளைத் தெரிந்து கொள்வது அவசியம். ஆகையால் இந்தப் பகுதியில் நாம் HTML form களைப் பற்றி பார்க்க இருக்கிறோம். உங்களுக்கு ஏற்கனவே HTML form இல் பரிச்சயம் இருக்கிறதென்றால் இந்தப் பகுதியை விட்டு விட்டு அடுத்தப் பகுதிக்குச் செல்லலாம்.

16.1 HTML படிவங்கள் உருவாக்குதல் (Creating HTML Forms)

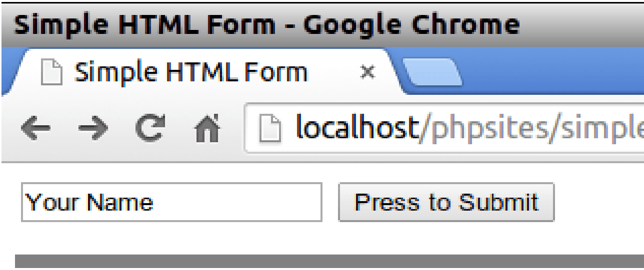
பயனர்களிடமிருந்து தகவல்களைச் சேகரிக்க HTML forms கள் பயன்படுகிறது. படிவங்களில் இருக்கும் உருப்படிகளின் மூலமாக பயனர் தன்னுடைய தகவல்களை உள்ளிட்டப் பிறகு அந்த தகவல்கள் இணைய சேவையகத்துக்கு(web server) அனுப்பி வைக்கப்படுகிறது. அங்கு அந்த தகவல்கள் செயலாக்கம்(process) செய்யப்படுகிறது.

<form> tag ஐப் பயன்படுத்தி HTML form கள் அடையாளப்படுத்தப்படுகிறது. GET அல்லது POST ஆகிய இரண் முறைகளில் ஏதாவது ஒரு முறையைப் பயன்படுத்தி பயனரினுடைய தகவல்கள் இணைய சேவையகத்துக்கு அனுப்பி வைக்கப்படுகிறது. GET முறையில் அனைத்து தகவல்களும் URL -க்குள் பொதிந்து அனுப்பி வைக்கப்படுகிறது. GET முறையின் மூலமாக அதிக அளவிலான தகவல்களை இணைய சேவையகத்துக்கு அனுப்பி வைக்கமுடியாது. அதே சமயத்தில் அதிக அளவிலான தகவல்களை POST முறையின் மூலமாக அனுப்பி வைக்க முடியும். பாதுகாப்பான முறையும் கூட.

ஒரு சிறிய HTML form ஐ உருவாக்குவது எப்படி என்று

பார்ப்போமோ?

```
<html>
<head>

<title>Simple HTML
Form</title>

</head>

<body>

<form action="submit.php"
method="post">

<input type="text"
name="customerName"
value="Your Name" />

<input type="submit"
name="submit_button"
value="Press to Submit" />

</form>

</body>

</html>
```



16.2 HTML Text Object (உரை பொருள்)

HTML Form -இல் அதிகமாக பயன்படுத்தக்கூடிய பொதுவான ஒன்று என்னவென்றால் அது Text Obect தான். படிவத்தில் எங்கு பயனர் ஒற்றைவரியில் தகவலை உள்ளிட வேண்டுமோ அங்கு இந்த Text Object வைக்கப்படுகிறது.

ஒரு படிவத்தினுள்ளே Text Object -ஐ உருவாக்குவதற்கான Syntax பின்வருமாறு

<input type="text" name="objectname" id="objectid" value="currentvalue" size="30" event_handling>

type – text வகையிலான object என்பதைக் குறிக்கிறது.

Name – text object -னுடைய பெயரைக் குறிக்கிறது. இந்தப் பெயர்தான் JavaScript, PHP போன்ற நிரல்களில் text object -னுடைய மதிப்புகளைப் பெறுவதற்காக பின்பு பயன்படுத்தப்படுகிறது.

Id – getElementsById() method -ஐப் பயன்படுத்தி text object -ஐ அணுகும் போது இந்த id பயன்படுகிறது.

Value – Text Object -இன் தொடக்க மதிப்பை குறிக்கிறது.

Size – Text Field க்குள்ளே அதிகபட்சமாக எத்தை உள்ளீடுகளைக் கொடுக்க வேண்டும் என்பதைக் குறிக்கிறது.

Text Object -இன் மீது ஒரு குறிப்பிட்ட event நடக்கும் போது எந்தவிதமான JavaScript Action நடைபெற வேண்டும் என்பதை குறிக்க Event handling

பயன்படுத்தப்படுகிறது.

ஒரு Text Object -இன் மீது கீழ்காணும் Event கள் Trigger செய்யப்படலாம்.

onFocus

onBlur

onChange

onSelect

16.3 HTML TextArea Object (உரைப்பகுதி பொருள்)

<textarea> எனும் tag ஐக் கொண்டு TextArea Object உருவாக்கப்படுகிறது. நான் மேலே பார்த்த text object இல் ஒற்றைவரியில்தான் உள்ளீட்டினைக் கொடுக்க முடியும். ஆனால், textarea பகுதியில் பயனரால்

பலவரியிலான உள்ளீடுகளைக் கொடுக்க முடியும். TextArea Object -இன் அளவினை கூடுதலான attributes களைக் கொண்டு கடுப்படுத்தலாம். உதாரணமாக எத்தனை row and columns(வரிசை மற்றும் நெடுவரிசை) இருக்க வேண்டும், படிக்க மட்டும் கூடியதாக இருக்க வேண்டுமா அல்லது தகவல்களை உள்ளீடக்கூடியதாக இருக்க வேண்டுமா போன்றவைகளைக் குறிப்பிடலாம். wrap(மடிப்ப்) attribute ஐப் பொறுத்தமட்டிலே இரண்டுவிதமான தெரிவுகள் உள்ளன. அவை virtual(மாயத்தோற்றம்) and physical(பருநிலை).

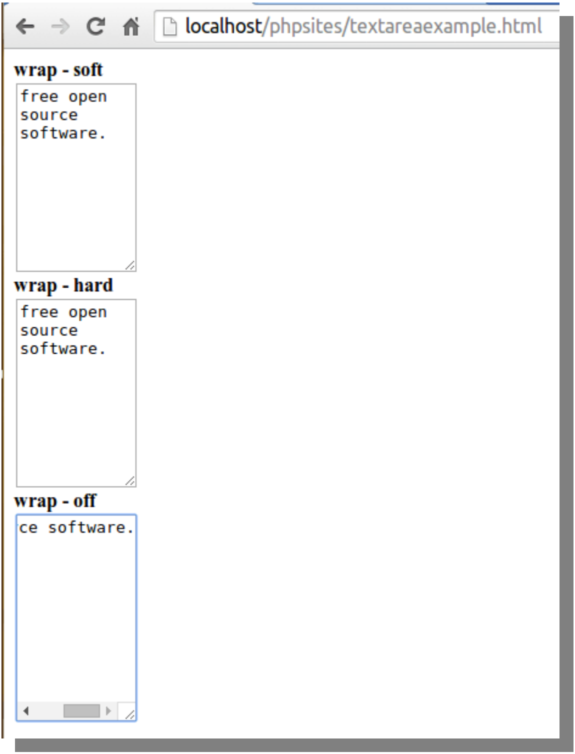
virtual(soft) – ஒவ்வொரு வரியினுடைய இறுதியிலும் carriage return ஐக் கொண்டிருக்காது.

physical(hard) – ஒவ்வொரு வரியினுடைய இறுதியிலும் carriage return ஐக் கொண்டிருக்கும்.

off – நாம் உள்ளிடும் தகவல்கள் அப்படியே தட்டச்சு ஆகிக்கொண்டிருக்கும். புதிய வரியாக தட்டச்சு செய்ய வேண்டுமென்றால், Enter Key ஐ அழுத்தி தட்டச்சு செய்து கொள்ள வேண்டும்.

உதாரண நிரலைப் பாருங்கள்.

```
<html>
<head>
<title>TextArea
Example</title>
</head>
<body>
<b>wrap - soft</b><br>
<textarea rows="10"
cols="10"
wrap="soft"></textarea><br>
<b>wrap - hard</b><br>
<textarea rows="10"
cols="10"
wrap="hard"></textarea><br>
<b>wrap - off</b><br>
<textarea rows="10"
cols="10"
wrap="off"></textarea><br>
</body>
</html>
```


localhost/phpsites/textareaexample.html
wrap - soft
free open
source
software.
wrap - hard
free open
source
software.
wrap - off
ce software.

16.4 The HTML Button Object (பொத்தான் பொருள்)

HTML *படிவத்தில்*(form) text object *அடுத்ததாக அதிகமாக பயன்படுத்தப்படுவது*, button(*பொத்தான்*) object *தான்*. *மூன்று வகையான பொத்தான்கள் உள்ளன. அவைகள்*

type="button" – *இது ஒரு அடிப்படையான பொத்தான். எந்தவிதமான செயலையும் இந்த வகை பொத்தான் செய்யாது. இதை அழுத்தும் போது ஏதாவது செயல் நடைபெற வேண்டுமென்றால் அதற்கான நிரலை நாம்தான் எழுத வேண்டும்.*

type="submit" – *படிவத்திற்குள்ளே நாம் உள்ளீடு செய்த தகவலை சேவையகத்துக்கு*(server) *அனுப்பி வைக்கிறது.* <form> tag *உள்ளே* Onsubmit Attribute *கொடுக்கப்பட்டிருந்தால் சேவையகத்துக்கு தகவல்களை அனுப்புதற்கு முன்னால்* onsubmit attribute *இல் கொடுக்கப்பட்டிருக்கும்* function *இயக்கப்படும்.* JavaScript *ஐக் கொண்டு படிவத்தில் இருக்கும் தகவல்கள் செல்லுபடியாக்கூடியதாக இருக்கிறதா*(validation) *என்று சோதனை செய்ய* onsubmit attribute *பயனுள்ளதாக இருக்கும்.*

type="reset" – *படிவத்தில் இருக்கும் தகவல்களை* clear

செய்யும் அல்லது default value(கொடாநிலை மதிப்பு) ஐ கொண்டு வந்து வைக்கும்.

<input> tag யே button object பயன்படுத்திக்கொள்கிறது. Type என்பதில் நாம் எந்த வகையான பொத்தான் என்பதைக் குறிப்பிட வேண்டும்.

உதாரண நிரலைப் பாருங்கள்

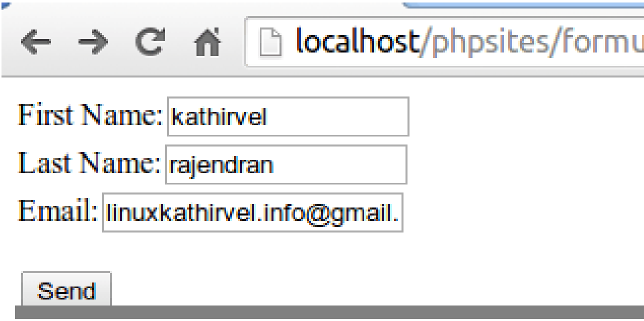
formuserinput.html

```
<html>
<head>
<title>User Input
Form</title>
</head>
<body>
<form onsubmit=""
method="post"
action="formgetuserdata.php"
>
<p>First Name:<input
type="text" name="firstname"
size="15"><br>
Last Name:<input type="text"
```

```
name="lastname"
size="15"><br>
Email:<input type="email"
name="emailid"><br></p>
<input type="submit"
value="Send">
</form>
</body>
</html>
```

formgetuserdata.php

```
<?php
$name =
$_POST["firstname"]." ".
$_POST["lastname"];
$email = $_POST["emailid"];
echo "<p>Welcome <b>$name!
</b><br>You can reach $name
via <i>$email</i></p>";
?>
```



16.5 HTML check Boxes

சிறிய சதுரம் வடிவிலான உருவத்தை check box object *உருவாக்குகிறது. பயனரை அதை* click *செய்யும் போது* checked *அல்லது* unchecked *நிலையை அடைகிறது. ஒன்றிற்கு மேற்பட்ட விருப்பங்களை பயனர் தேர்வு*

செய்ய வேண்டுமென்றால், நாம் checkbox object *ஐ பயன்படுத்திக்கொள்ளலாம்.*

<input> tag *ஐப் பயன்படுத்தி* check box object *உருவாக்கப்படுகிறது. எப்படி உருவாக்குவது என்று கீழ்காணும் நிரலைப் பார்த்து நீங்கள் தெரிந்து கொள்ளலாம்.*

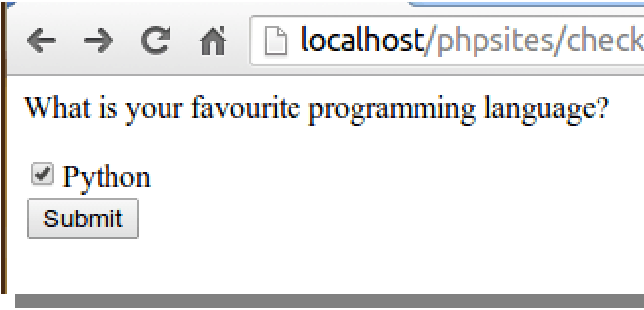
checkbox.html

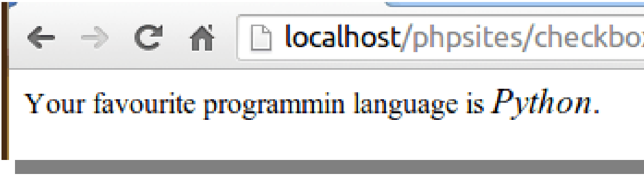
```
<html>
<head>
<title>Check Box</title>
</head>
<body>
<form method="post"
name="orderform"
action="checkboxgetdata.php"
>
<p>What is your favourite
programming language?</p>
<input type="checkbox"
name="language"
value="Python">Python<br>
<input type="submit">
</form>
```

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

checkboxgetdata.php

```
<?php
$name = $_POST["language"];
echo "Your favorite
programming language is
<i><big>$name</big></i>.";
?>
```





16.6 HTML Radio Button

பல விருப்பங்கள் இருந்து அதில் ஏதாவது ஒன்றைத்தான் தேர்வு செய்ய வேண்டும் என்ற நிலை வரும்போது நாம் Radio Button Object ஐ பயன்படுத்திக்கொள்ளலாம்.

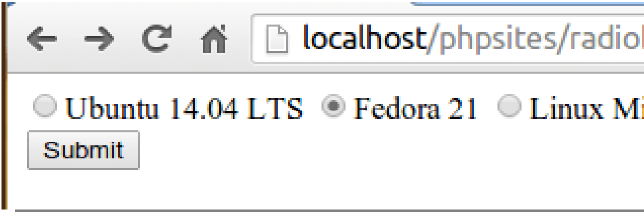
Radion Button ஐ உருவாக்குவது எப்படி என்று பார்ப்போமா?

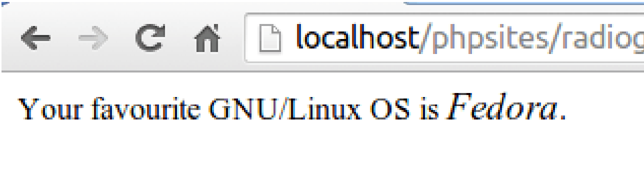
```
<html>
<head>
<title>Radion Button
Example</title>
</head>
<body>
<form method="post"
action="radiogetdata.php"
```

```
name="userchoice">
<input type="radio"
name="myLinux"
value="Ubuntu"
checked>Ubuntu 14.04 LTS
<input type="radio"
name="myLinux"
value="Fedora">Fedora 21
<input type="radio"
name="myLinux"
value="LinuxMint">Linux Mint
17
<br><input type="submit">
</form>
</body>
</html>
```

radiogetdata.php

```
<?php
$name = $_POST["myLinux"];
echo "Your favourite
GNU/Linux OS is
<i><big>$name</big></i>.";
?>
```

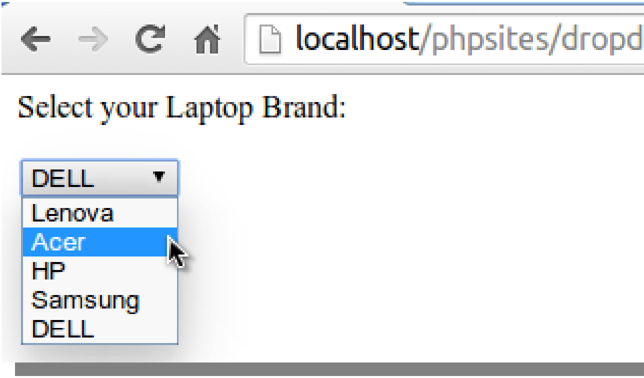




16.7 HTML Drop-down / Select Object

பயனரினுடைய விருப்பங்களை select object ஆனது drop down list முறையில் காண்பிக்கிறது. பயனர் தன்னுடைய விருப்பத்தை அந்த பட்டியலிலிருந்து தேர்ந்தெடுக்கலாம்.

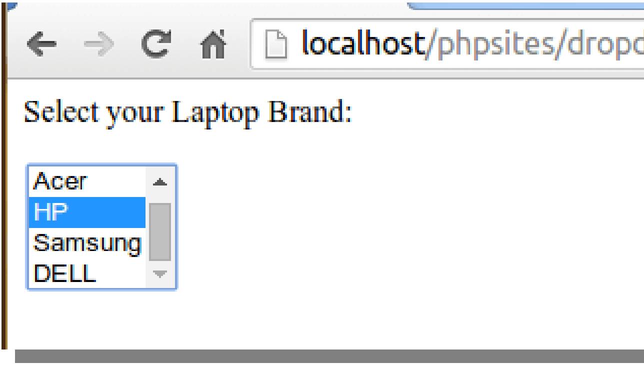
```
<html>
<head>
<title>Drop Down
List</title>
</head>
<body>
<p>Select your Laptop
Brand:</p>
<select
name="myLaptopBrand">
<option
value="Lenova">Lenova</optio
n>
<option
value="Acer">Acer</option>
<option
value="HP">HP</option>
<option
value="Samsung">Samsung</opt
ion>
<option value="DELL"
selected>DELL</option>
</select>
</body>
</html>
```



size attribute இல் ஒன்றைவிட மேலான மதிப்பு இருந்தால், scrolled list ஆக காண்பிக்கும். பயனர் scrolling செய்து விருப்பங்களை தேர்ந்தெடுக்கலாம்.

```
<html>
<head>
<title>Drop Down
List</title>
</head>
<body>
<p>Select your Laptop
```

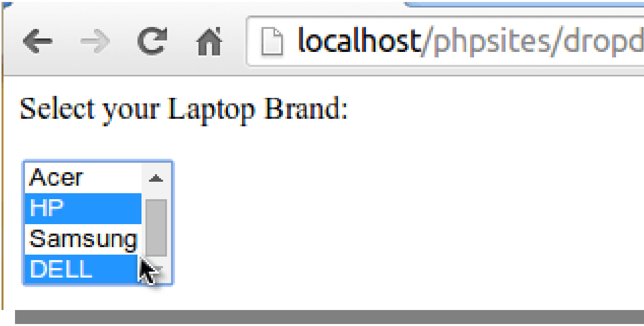
```
Brand:</p>
<select name="myLaptopBrand"
size="2">
<option
value="Lenova">Lenova</optio
n>
<option
value="Acer">Acer</option>
<option
value="HP">HP</option>
<option
value="Samsung">Samsung</opt
ion>
<option value="DELL"
selected>DELL</option>
</select>
</body>
</html>
```



multiple attribute ஐ கொடுப்பதன் மூலமாக dropdown list ஒன்றிற்கு மேற்பட்டவைகளை தேர்ந்தெடுக்கலாம்.

```
<html>
<head>
<title>Drop Down
List</title>
</head>
<body>
<p>Select your Laptop
Brand:</p>
<select
```

```
name="myLaptopBrand[]"
size="2" multiple>
<option
value="Lenova">Lenova</optio
n>
<option
value="Acer">Acer</option>
<option
value="HP">HP</option>
<option
value="Samsung">Samsung</opt
ion>
<option value="DELL"
selected>DELL</option>
</select>
</body>
</html>
```

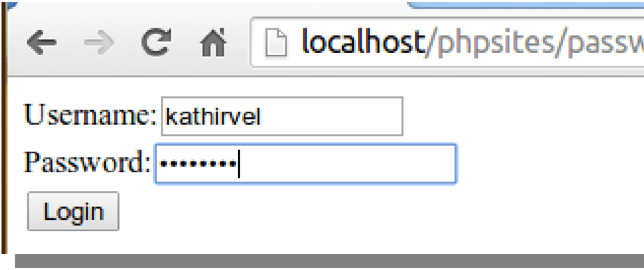


16.8 HTML Password Object

பயனர் என்ன உள்ளிடுகிறாரோ அந்த தகவல்கள் திரையில் தெரியக்கூடாது என்றால் நாம் password object *ஐ பயன்படுத்திக்கொள்ளலாம். பயனர் உள்ளிடும் ஒவ்வொரு* character *-ம் '*' போன்று காட்சியளிக்கும். பயனர் கடவுச்சொல்,* PIN *போன்ற தகவலை உள்ளிடும் போது இதை பயன்படுத்தலாம்.*

```
<html>
<head>
```

```
<title>Passwor
Object</title>
</head>
<body>
Username:<input type="text"
size="15"><br>
Password:<input
type="password"
size="20"><br>
<input type="submit"
value="Login">
</body>
</html>
```



17 PHP and HTML Forms

இந்த பகுதியில் நாம், பயனரிடமிருந்து தகவலை பெறுவதற்காக ஒரு சிறிய HTML படிவத்தையும், அந்த தகவல் சேவையகத்துக்கு அனுப்பி வைக்கப்பட்டபின் அதை செயல்படுத்த ஒரு PHP Script ஐயும் உருவாக்க இருக்கிறோம். இந்த பகுதி உங்களுக்கு முழுமையாக புரிய வேண்டுமென்றால் இதற்கு முந்தைய பகுதியான Overview of HTML Forms பகுதியை ஒரு முறை படித்து விடவும்.

17.1 படிவம் உருவாக்குதல் (Creating the Form)

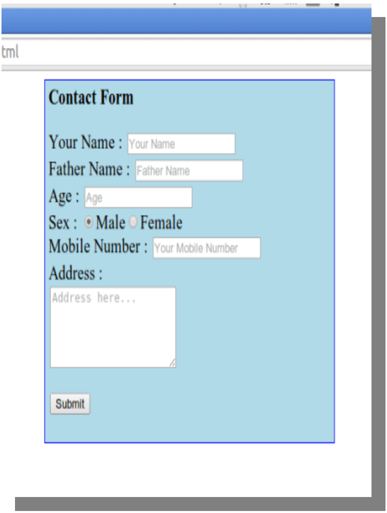
ஒரு பயனரிடமிருந்து அவரை தொடர்பு கொள்வதற்குண்ட தகவலை பெறுவதற்காக ஒரு படிவத்தை உருவாக்குவது எப்படி என்பதை இங்கு பயிற்சிக்காக எடுத்துக்கொள்வோம்.

ஒரு பயனரை தொடர்புகொள்ள அவரிடமிருந்து என்னென்ன தகவலை நாம் பெற வேண்டும் என்பதை முதலில் நாம் முடிவு செய்து கொள்ள வேண்டும். உதாரணமாக பயனர் பெயர், அவர் தந்தை பெயர், வயது, பாலினம், கைப்பேசி எண், மின்னஞ்சல் முகவரி, முழு

முகவரி இவைகளை பெறுவதாக வைத்துக்கொள்வோம். இந்த தகவலை பெறுவதற்கு ஒரு HTML படிவத்தை முதலில் உருவாக்குவோம்.

```
<html>
<head>
<title>Contact Form</title>

<style>
#contactform {
background-color: lightblue;
width: 400px;
margin: auto;
border: 1px solid blue;
padding: 5px;
font-size: 20px;
}
</style>

</head>

<body>
<div id="contactform">
<b>Contact Form</b>
<form method="post"
action="contactdetails.php">
Your Name : <input
type="text" name="username"
placeholder="Your Name">
```

```
Father Name : <input
type="text"
name="fathername"
placeholder="Father Name">
Age : <input type="text"
name="age"
placeholder="Age">
Sex : <input type="radio"
name="sex" value="Male"
checked>Male<input
type="radio" name="sex"
value="Female">Female
Mobile Number : <input
type="text"
name="mobilenumber"
placeholder="Your Mobile
Number">
Address :
<textarea name="address"
rows="5" cols="20"
placeholder="Address
here..."></textarea>
<input type="submit">
</form>
</div>
</body>
</html>
```


tml
Contact Form
Your Name :
Your Name
Father Name :
Father Name
Age :
Age
Sex :
Male
Female
Mobile Number :
Your Mobile Number
Address :
Address here...
Submit

17.2 PHP ஐ பயன்படுத்தி படிவத்தின் தகவலை Process செய்தல் (Processing Form Data Using PHP)

படிவத்தை உருவாக்குவது எப்படி என்று மேலே பார்த்தோம். இப்பொழுது அந்த தகவல்களை PHP ஐக் கொண்டு செயல்படுத்துவது எப்படி என்று பார்ப்போம். HTML Form -லிருந்து சேவையகத்துக்கு(server) தகவலை அனுப்ப இரண்டுவிதமான Mechanisms இருக்கிறது. ஒன்று GET மற்றொன்று POST. மேலே நாம் பார்த்த படிவம் உருவாக்கும் நிரலிலும், அதற்கு முன்னர் பார்த்த நிரல்களிலும் நாம் POST method ஐத் தான் பயன்படுத்தியிருகிறோம்.

படிவத்திலிருந்து கிடைக்கும் தகவல்களை PHP ஒரு associative array யில்தான் சேமித்து வைக்கிறது. அந்த array யைக் கொண்டுதான் நாம் தகவல்களை process செய்ய வேண்டும். நாம் HTML form ஐ உருவாக்கும் போது method attribute இல் POST என கொடுத்திருந்தால் அது PHP யில் $_POST எனும் associative array யிலும், GET என கொடுத்திருந்தால் அது PHP யில் $_GET எனும் associative array யிலும் சேமிக்கப்பட்டு இருக்கும்.

நாம் மேலே பார்த்த HTML படிவத்தினை process

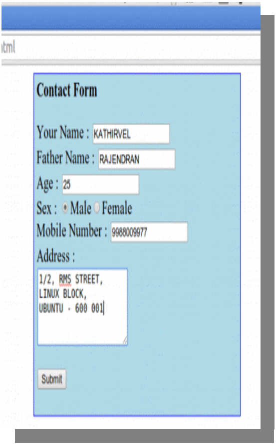
செய்வதற்காக contactdetails.php எனும் script ஐ $_POST பயன்படுத்தி உருவாக்கியுள்ளேன்.

படிவத்தில் தகவல்கள் உள்ளிடப்பட்டு Submit Button ஐ அழுத்தியவுடன், அதற்கான வெளியீடு contactdetails.php எனும் script ஐக் கொண்டு வெளியிடப்படும்.

contactdetails.php

```
<?php
$userName =
$_POST["username"];
$fatherName =
$_POST["fathername"];
$age = $_POST["age"];
$sex = $_POST["sex"];
$mobileNumber =
$_POST["mobilenumber"];
$address =
$_POST["address"];
echo "<b>Your Data is
Successfully Received.
Thanks.</b>";
//echo "$userName,
$fatherName, $age, $sex,
$mobileNumber, $address";
```

```
print_r($_POST);
?>
```

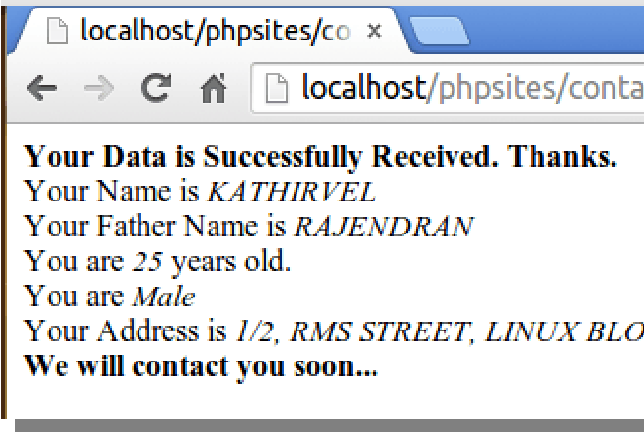


tml
Contact Form
Your Name : KATHIRVEL
Father Name : RAJENDRAN
Age : 25
Sex : Male Female
Mobile Number : 9988009977
Address :
1/2, RMS STREET,
LINUX BLOCK,
UBUNTU - 600 001
Submit

contactdetails.php script இல் கொஞ்சம் மாற்றம் செய்தால் கீழ்காணுமாறு வெளியீடு இருக்கும்.

```
<?php
$userName =
$_POST["username"];
$fatherName =
$_POST["fathername"];
$age = $_POST["age"];
$sex = $_POST["sex"];
$mobileNumber =
$_POST["mobilenumber"];
$address =
$_POST["address"];
echo "<b>Your Data is
Successfully Received.
Thanks.</b>";
```

```
echo "Your Name is
<i>$userName</i>";
echo "Your Father Name is
<i>$fatherName</i>";
echo "You are <i>$age</i>
years old.";
echo "You are <i>$sex</i>";
echo "Your Address is
<i>$address</i>";
echo "<b>We will contact you
soon...<b>";
?>
```



17.3 Processing Multiple Selections with PHP(பல தேர்வுகளை செயல்படுத்துதல்)

ஒன்றிற்கு மேற்பட்ட தேர்வினை கையாள்வது எப்படி? என்பதைப் பற்றி இந்த பகுதியில் பார்ப்போம். Drop-down list எனும் பகுதியில் இதைப் பற்றி நாம் ஏற்கனவே பார்த்துள்ளோம். அதை ஒருமுறை ஞாபகபடுத்திக்கொள்வோம்.

நாம் இதற்கு முன்னர் பார்த்த அனைத்துமே ஒரு மதிப்பை மட்டும் தருவதால் மிக எளிதாக எந்தவித குழப்பமும் இல்லாமல் நிரலில் கையாள முடிந்தது. ஆனால் இப்பொழுது நாம் பார்க்கும் சூழலே வேறு, ஒரே உள்ளீடுதான் ஆனால் அதிலிருந்து கிடைக்கும் மதிப்புகள் ஒன்றுக்கு மேற்பட்டவை. ஆகையால் HTML Form அளவிலும், PHP Script அளவிலும் சிறிய மாற்றங்களை ஏற்படுத்த வேண்டியுள்ளது.

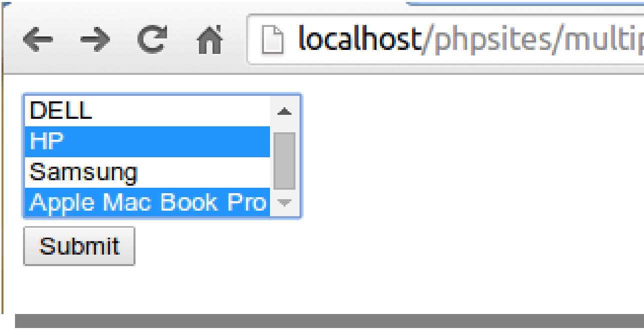
HTML Form ஐ பொறுத்தமட்டிலே

<select name="laptop[]" size="2″ multiple> எனும் வரியில் name இல் அதன்பெயருக்கு பின்னால் [] சேர்த்துள்ளோம். ஒன்றிற்கு மேற்பட்ட தேர்வுகளை தேர்தெடுக்க வேண்டி இருப்பதால் multiple எனும் attribute ஐ இறுதியில் சேர்த்துள்ளோம்.

Multiple Selections கள் செய்வதற்கான படிவம் கீழே கொடுக்கப்பட்டுள்ளது.

```
<html>
```

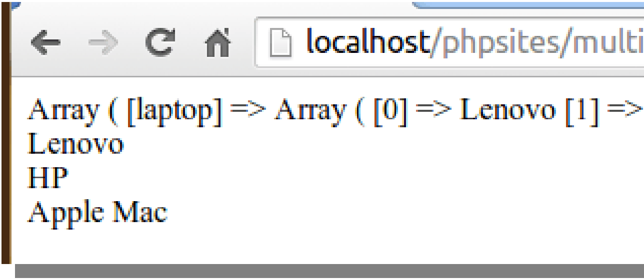
```
<head>
<title>Multiple
Selections</title>
</head>
<body>
<form
action="multipleselections.p
hp" method="post">
<select name="laptop[]"
size="2" multiple>
<option
value="Acer">Acer</option>
<option
value="Lenovo">Lenovao</opti
on>
<option
value="DELL">DELL</option>
<option
value="HP">HP</option>
<option
value="Samsung">Samsung</opt
ion>
<option value="Apple
Mac">Apple Mac Book
Pro</option>
</select>
<input type="submit">
</form>
</body>
</html>
```



PHP நிரல்

```
<?php
print_r($_POST);
echo "";

echo $_POST["laptop"][0];
echo "";

echo $_POST["laptop"][1];
echo "";

echo $_POST["laptop"][2];
echo "";
```

```
echo $_POST["laptop"][4];
echo "";

echo $_POST["laptop"][5];
echo "";
?>
```



18 PHP and Cookies — Creating, Reading and Writing (குக்கீஸ் உருவாக்குதல், படித்தல் மற்றும் எழுதுதல்)

வலைப்பக்கம் வேண்டுமென்று யார் வேண்டுகோள் கொடுத்தாலும் எதைப்பற்றியும் கண்டுகொள்ளாமல் வலை சேவையங்கள், வலைப்பக்கங்களை கேட்டவர்களுக்கு அனுப்பி வைக்கும். வலைப்பக்கத்தைக் கேட்கும் நபர் இதற்கு முன்னர் வலைப்பக்கம் வேண்டி வேண்டுகோள் கொடுத்துள்ளாரா என்பதைப் பற்றிய எந்த விஷயத்தையும் கவனத்தில் எடுத்துக்கொள்ளாது. ஒவ்வொரு முறை வேண்டுகோள் வரும்போதும் அதை புதிய வேண்டுகோளாகவே கருதி வலைப்பக்கத்தை வேண்டுகோள் விடுத்தவருக்கு வலை சேவையகம் அனுப்பி வைக்கும்.

இதனால் பார்வையாளர் வலைப்பக்கத்துக்கு புதியவரா அல்லது ஏற்கனவே வலைப்பக்கத்தை பயன்படுத்திக் கொண்டு இருப்பவரா என்பதை கண்டுபிடிப்பதோ அல்லது பார்வையாளரை பின்தொடர்வதோ கடினமான ஒன்றாக ஆகி விடுகிறது.

இந்த பிரச்சனையை தீர்க்கும் விதமாகவும், வலையின்(web) நிலையற்ற தன்மையினைக் கண்காணிக்கவும், பார்வையாளரினை பின்தொடர்வதற்காகவும் உருவாக்கப்பட்ட ஒரு இயந்திரம்தான் Cookies.

18.1 குக்கீஸ்

நமது வலைப்பக்கத்துக்கு வருகை தரும் பார்வையாளர் பற்றிய விபரங்களை, பார்வையாளர்களின் கணினியிலேயே, சிறிதளவில் சேமித்து வைக்கும் வசதியினை Cookies வழங்குகிறது. இதனால் நமது வலைப்பக்கத்துக்கு வருகை தரும் பார்வையாளரினுடைய அனைத்து நிலைகளையும் வலைப்பக்கத்தின் மூலமாக பராமரித்து வர முடியும். அதோடு பார்வையாளர் ஒரு வலைப்பக்கத்தில் உள்ளிட்ட பயனரினுடைய பெயர், முகவரி, மின்னஞ்சல் முகவரி ஆகியவைகளை வேறொரு பக்கத்தில் உள்ளிட வேண்டிய நிலை வரும்போது மறுபடியும் உள்ளிட்டுக்கொண்டிருக்க வேண்டி அவசியமில்லை.

நமது வலைப்பக்கத்துக்கு Cookies வசதியினை ஏற்படுத்துவதற்கு முன்னர் சில விஷயங்களைக் கவனத்தில் கொள்ள வேண்டும். பயனரினுடைய இணைய

உலாவியில் cookies வசதியினை பயனரால் நிறுத்தி வைக்க முடியும். இதனால் நம்முடைய cookies தொடர்பான செயல்பாடுகள் இயங்காமல் போக வாய்ப்பிருக்கிறது. இந்த காரணத்தினால் நம்முடைய வலைப்பக்கம் முழுவதும் cookies ஐயே நம்பி இருக்கும் விதத்தில் வடிவமைப்பதை தவிர்க்க வேண்டும்.

Cookies க்கு மாற்றாக வலைப்பக்கத்துக்கு வரும் பார்வையாளர்களின் நிலையை நிர்வகிக்க PHP யில் sessions இருக்கிறது. இதைப்பற்றி வரும் பகுதிகளில் நாம் விரிவாக பார்க்க இருக்கிறோம். Cookies மற்றும் Sessions களைப் வலைப்பக்கத்தில் பயன்படுத்துவது என்பது நம்முடைய தேவைகளைப் பொறுத்தது. தேவைகள் இல்லாத பட்சத்தில் இதை நாம் தவிர்த்துக்கொள்ளலாம்.

18.2 The Difference Between Cookies and Sessions (Cookies and Sessions இரண்டிற்குமான வேறுபாடு)

cookies மற்றும் sessions ஆகிய இரண்டுமே தகவல்களை சேமித்து வைத்து நம்முடை வலைதளத்தின் வெவ்வேறு பக்கங்கள் அந்த தகவல்களை அணுகுவதற்கு உதவுகிறது. அதே நேரத்தில் இரண்டினுடைய அணுகுமுறையிலும் வேறுபாடுகள் இருக்கிறது.

குக்கீஸ்

நம்முடைய வலைதளம் எந்த கணினியில் பார்க்கப்படுகிறதோ அந்த கணினியினுடைய வன் வட்டிலேயே(Hard Disk) Cookies *கள் சேமிக்கப்படுகின்றன. நம்முடைய வலைதளத்தை பார்வையிட்டு முடிவிட்ட பின்பும்* Cookies *வன் வட்டிலேயே சேமிக்கப்பட்டு இருக்கும். ஒரு* Domain *க்கு அதிகபட்சமாக* 20 *குக்கீஸ்கள் வரை அனுமதியுண்டு. ஒவ்வொரு குக்கீஸும்* 4Kb *அளவு கொண்டதாக இருக்கலாம்.*

Sessions

Sessions *கள் வலை சேவையகத்தில்*(Web Server) *சேமிக்கப்படுகின்றன. வலை சேவையகத்தில் சேமிக்கப்பட்டிருந்தாலும் அதே சேவையகத்தில் இருக்கும் மற்ற* Domain *கள் நம்முடைய தளத்திற்கான* sessions *உடைய தகவல்களை அணுக முடியாது. எவ்வளவு தகவல்களை வேண்டுமானாலும்* session *மூலமாக சேமித்துக்கொள்ளலாம். அதே நேரத்தில் சேமிக்கப்பட்ட தகவல்கள் பாதுகாப்பாகவும் இருக்கும். குக்கீஸைப் போன்று பயனரினுடைய உலாவிக்கு தகவல்கள் அனுப்பி வைக்கப்பட மாட்டாது.*

18.3 குக்கீயினுடைய அமைப்பு (The Structure of Cookie)

name/value(பெயர்/மதிப்பு) எனும் வடிவத்தில் தகவல்களை சேமித்து வைக்க cookies அனுமதிக்கிறது. Name/value ஆகிய இரண்டையுமே நம்முடைய விருப்பத்திற்கு ஏற்றாற்போல கொடுத்துக்கொள்ளலாம். உதாரணமாக பயனரினுடைய பெயரை userName = Kathirvel Rajendran எனும் வடிவத்தில் சேமித்து வைப்பதை எடுத்து கொள்ளலாம். குக்கீஸ் எவ்வளவு நேரம் கணினியில் சேமித்து வைக்கப்பட்டிருக்க வேண்டும் அதாவது expiration date போன்ற கூடுதலான தகவல்களையும் cookies வைத்திருக்கும்.

குக்கீஸின் வடிவம் கீழ்காணும் வடிவத்தில் இருக்கும்:

name = Value; expires = expirationDateGMT; path=URLpath; domain=siteDomain

18.4 குக்கீஸ் காலாவதியாகும் நேரத்தை அமைத்தல்(Cookie Expiration Setting)

expires= எனும் விருப்பத் தேர்வு ஒரு குறிப்பிட்ட குக்கீ எப்பொழுது காலாவதியாக வேண்டும் என்பதை

குறிப்பிடுகிறது. காலாவதியாகும் தேதியினை பெறுவதற்கு PHP யின் time() செயல்கூறு(function) பயன்படுத்தப்படுகிறது. இதைப் பற்றி வரும் பகுதிகளில் பார்க்க இருக்கிறோம்.

18.5 குக்கீயின் பாதை அமைப்பு(Cookie path Setting)

path= அமைப்பு குக்கீ எந்த URL க்குள் சேமிக்கப்பட வேண்டும் என்பதை முடிவு செய்கிறது. இயல்பாகவே, வலைப்பக்கம் எந்த அடைவிற்குள் இருக்கிறதோ அந்த அடைவிற்குள்தான் குக்கீஸ் சேமிக்கப்படும். உதாரணமாக, www.kaniyam/php/phpintamil.html எனும் பக்கத்திற்கான குக்கீ /php எனும் அடைவிற்குள் சேமித்து வைக்கப்பட்டிருக்கும்.

18.6 குக்கீ domain அமைப்பு(Cookie domain Setting)

path setting இல் உள்ளதைப் போன்று வலை சேவையகத்தில் இருக்கும் எந்த வலைதளம் குக்கீயை உருவாக்கியதோ, அந்த குக்கீயை அந்த தளம் மட்டுமே அணுக முடியும். மற்ற பக்கத்தினால் அதை அணுக முடியாது. அதே நேரத்தில் domain=domain name என்பதில் நாம் வேறொரு தளத்தினைய முகவரியைக்

கொடுப்பதன் மூலம் மற்ற தளங்களும் குக்கீயை அணுகும் வகையில் செய்ய முடியும்.

உதாரணமாக, www.kaniyam.com ஒரு குக்கீயை உருவாக்கியிருந்தால், domain=www.gnutamil.blogspot.in என்று கொடுப்பதன் மூலம் அந்த குக்கீயை www.gnutamil.blogspot.in தளமும் அணுக முடியும்.

18.7 குக்கீயின் பாதுகாப்பு அமைப்பு(Cookie Security Setting)

குக்கீகள் பாதுகாப்பான HTTPS(Hyper Text Transfer Protocol Secure) பயன்படுத்தி அனுப்ப வேண்டுமா அல்லது பாதுகாப்பு இல்லாத HTTP யைப் பயன்படுத்தி அனுப்ப வேண்டுமா என்பதை முடிவு செய்கிறது.

18.8 குக்கீ உருவாக்குதல்(Creating a Cookie in PHP)

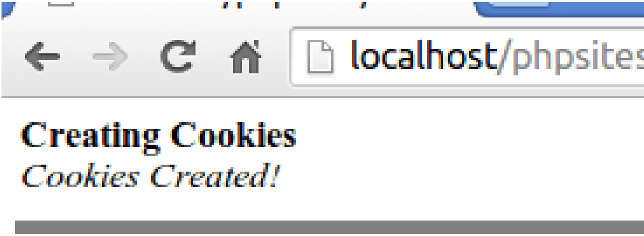
setcookie() செயல்கூறு(funtion) ஐப் பயன்படுத்தி குக்கீகள் உருவாக்கப்படுகின்றது. ஒரு குறிப்பிட்ட எண்ணிக்கையிலான arguments களை setcookie() function எடுத்துக்கொள்கிறது. முதல் argument குக்கீயின் பெயர், இரண்டாவது argument அந்த பெயருக்கான

மதிப்பு. மூன்றாவது argument குக்கீ காலாவதியாகும் தேதி. நான்காவது argument குக்கீயினுடைய active path, ஐந்தாவது argument domain setting மற்றும் ஆறாவது argument security setting(0 என்பது HTTP என்பதையும், 1 என்பது HTTPS என்பதையும் குறிக்கிறது.)

குக்கீயை உருவாக்குதல் உதாரண நிரல்

```
<?php
//Creating Cookies
echo "<b>Creating
Cookies</b><br />";

setcookie('userName','Kathir
vel Rajendran', time() +
15);
setcookie('emailid','<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>',
time() + 15);

echo "<i>Cookies Created!
</i><br />";
```

```
?>
```



மேற்கண்ட நிரல் userName=Kathirvel Rajendran ஜோடி முறையில் குக்கீயை உருவாக்குகிறது. மேற்கண்ட குக்கீ உருவாக்கப்பட்ட நேரத்திலிருந்து 15 வினாடிகள் கழித்து காலாவதியாகிறது.

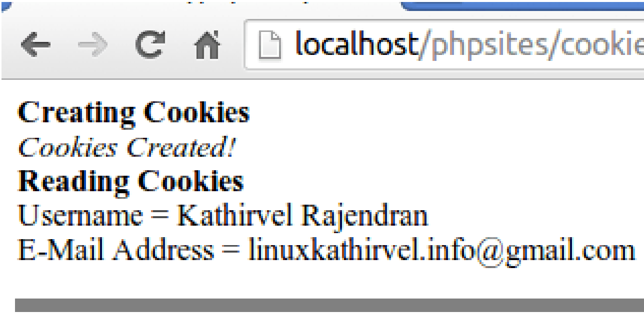
18.9 குக்கீயினைப் படித்தல்(Reading a Cookie in PHP)

மேலே நாம் உருவாக்கிய குக்கீயினை $_COOKIE array யின் மூலமாக அணுக முடியும். $_COOKIE array யானது

ஒரு associative array ஆகும். $_COOKIE array யினுடைய index மதிப்பாக குக்கீயின் பெயரை கொடுப்பதன் மூலமாக அதன் மதிப்பை அணுக முடியும்.

```
<?php
//Creating Cookies
echo "<b>Creating
Cookies</b><br />";

setcookie('userName','Kathir
vel Rajendran', time() +
15);

setcookie('emailid','<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>',
time() + 15);

echo "<i>Cookies Created!
</i><br />";

//Reading Cookies
echo "<b>Reading
Cookies</b><br />";
echo "Username = ".
```

```
$_COOKIE['userName']."<br
/>";
echo "E-Mail Address = ".
$_COOKIE['emailid']."<br
/>";
?>
```



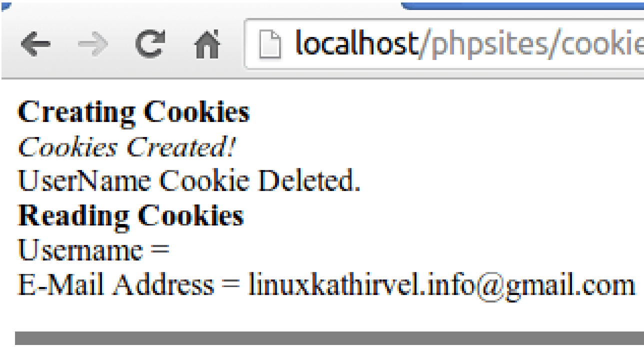
18.10 குக்கீயை அழித்தல்(Deleting a Cookie)

setcookie() *செயல்கூறு*(function) *மூலமாக* cookie *ஐ அழிக்க முடியும். மறுபடியும் இங்கு* time() function *குக்கீ காலாவதியாகும் தேதியை கணக்கிட*

பயன்படுத்தப்படுகிறது. குக்கீயை உருவாக்குவதற்கும் setcookie() function ஐத் தான் பயன்படுத்தினோம். அழிக்கவும் setcookie() function ஐத் தான் பயன்படுத்துகிறோம். வி ் த்தியாசம் என்னவென்றால் காலாவதியாகும் நேரத்தினை time() + seconds எனும் முறையில் கொடுத்தோம். இங்கு time() – seconds எனும் முறையில் கொடுத்துள்ளோம். – (கழித்தல் குறியீடு) கடந்த காலத்தை குறிக்கிறது.

```
<?php

//Creating Cookies
echo "<b>Creating
Cookies</b><br />";

setcookie('userName','Kathir
vel Rajendran', time() +
15);
setcookie('emailid','<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>',
time() + 15);
```

```
echo "<i>Cookies Created!
</i><br />";

//Deleting Cookie
setcookie('userName', '' ,
time() - 15);
echo "UserName Cookie
Deleted.<br />";

//Reading Cookies
echo "<b>Reading
Cookies</b><br />";
echo "Username = ".
$_COOKIE['userName']."<br
/>";
echo "E-Mail Address = ".
$_COOKIE['emailid']."<br
/>";
?>
```


localhost/phpsites/cookie
Creating Cookies
Cookies Created!
UserName Cookie Deleted.
Reading Cookies
Username =
E-Mail Address = linuxkathirvel.info@gmail.com

19 அமர்வு (Understanding PHP Sessions)

இதற்கு முந்தைய பகுதியில் குக்கீஸைப் பற்றி பார்த்தோம். இந்த பகுதியில் குக்கீஸுக்கு மாற்றாக இருக்கும் sessions ஐப் பற்றி பார்க்க இருக்கிறோம். இந்த பகுதியில் sessions ஐப் பற்றி உதாரணங்களுடன் மேலும் விரிவாக பார்க்க இருக்கிறோம். sessions ஐ உருவாக்குதல் மற்றும் sessions ஐப் பயன்படுத்துவது போன்றவைகளைப் பற்றியும் பார்க்க இருக்கிறோம்.

19.1 Session என்றால் என்ன?

PHP session ஆனது வலைப்பக்கங்களை ஒரு குழுவாக(group) பார்க்கிறது. அவ்வாறு குழுவாக இருக்கும் பக்கங்களுக்கு இடையில் ஒரு மாறியின் மதிப்பை பகிர்ந்து கொள்ள அனுமதிக்கிறது. குக்கீஸின் பலவீனம் என்னவென்றால், குக்கியின் மதிப்பு பயனருடைய(வலைமுகவரியை பார்வையிடுபவர்) கணினியில் சேமிக்கப்படுகிறது. இதனால் குக்கீயின் மதிப்புகளை பயனரால் படிக்கவும், திருத்தங்கள் செய்யவும் முடியும். அதே சமயத்தில் sessions பொறுத்தமட்டிலே ID குக்கீ மட்டும் பயனருடைய கணினியில் சேமிக்கப்படுகிறது. இந்த ID குக்கீயானது

சேவையகத்தில்(server) இருக்கும் session file ஐ அணுக பயன்படுகிறது. இதனால் பயனரால் நேரடியாக session file இன் content ஐ அணுகமுடியாது. இதன் மூலம் குக்கீயை விட பாதுகாப்பான வழியை session ஏற்படுத்தி தருகிறது. உலாவியில் cookie support ஐ பயனர் நிறுத்தி வைத்தாலும் session வேலை செய்யும். ஒருவேளை பயனர் cookie support ஐ உலாவியில் நிறுத்தி வைத்திருந்தால் வலை முகவரியில் session ID சேமித்து வைக்கப்படுகிறது.

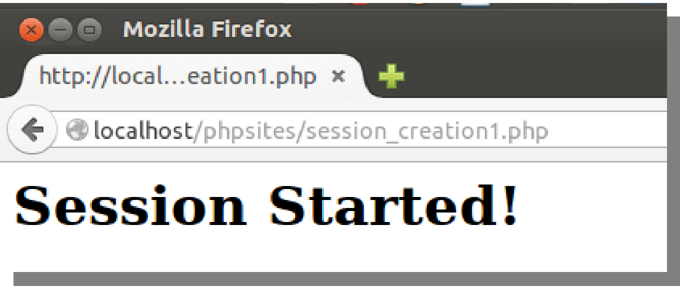
19.2 PHP Session உருவாக்குதல் (Creation a PHP Session)

session_start() எனும் Function ஐப் பயன்படுத்தி sessions உருவாக்கப்படுகிறது. session_start() function ஆனது வலைப்பக்கத்தின் first function call ஆக இருக்க வேண்டும்.

நிரல் :

```
<?php

//session creation
if(session_start())
```

```
{
echo "<h1>Session Started!
</h1>";
}
else {
echo "<h1>Session Not
Create!</h1>";
}
?>
```



19.3 Session மாறிகளை உருவாக்குதல் மற்றும் படித்தல்(Creating and Reading Session Variables)

$_SESSION array மூலமாக session variable களை உருவாக்கலாம் மற்றும் அதற்கு மதிப்புகள் கொடுக்கலாம். $_SESSION ஆனது ஒரு Global Array ஆகும். ஆகையால் ஒரு இணையதளத்தின் அனைத்து பக்கங்களிலும் session variable களை பயன்படுத்தலாம். மேலும் இது ஒரு associative array ஆகும். Array ஐப் பற்றி மேலும் தெரிந்து கொள்ள PHP Array எனும் பகுதியைப் பார்க்கவும்.

Session மாறியின் மதிப்பு strings, numbers, arrays and objects என எந்த வகையினைச் சேர்ந்ததாக வேண்டுமானாலும் இருக்கலாம்.

variable name மற்றும் assignment operator ஆகியவைகளைப் பயன்படுத்தி நேரடியாக $_SESSION array யில் variable ஐ உருவாக்குவதுடன் அதற்கான மதிப்பையும் அளிக்கலாம்.

```
<?php

$_SESSION['userName'] = 'Kathirvel Rajendran';
```

?>

நிரல்:

```
<?php

//session creation
if(session_start())
{
echo "<h1>Session Started!
</h1>";
$_SESSION['userName'] =
'Kathirvel Rajendran';
}

else {
echo "<h1>Session Not
Create!</h1>";
}

//session accessing
if(isset($_SESSION['userName
'])) {
echo "<b>User Name :
</b><i>".
$_SESSION['userName']."</i>"
;
}
```

```
else {
echo "Session Accessing
Failed!";
}
?>
```

19.4 Session தகவல்களை கோப்பில் எழுதுதல்(Writing Session Data to a File)

session காலவதியாகும் வரையோ அல்லது அழிக்கப்படும் வரையோதான் session இன் தகவல்கள் சேவையகத்தில் உயிருடன் இருக்கும். ஒருமுறை அழிக்கப்பட்டு விட்டால் session உடன் தொடர்புடைய அனைத்து தகவல்களும் அழிக்கப்பட்டுவிடும்.

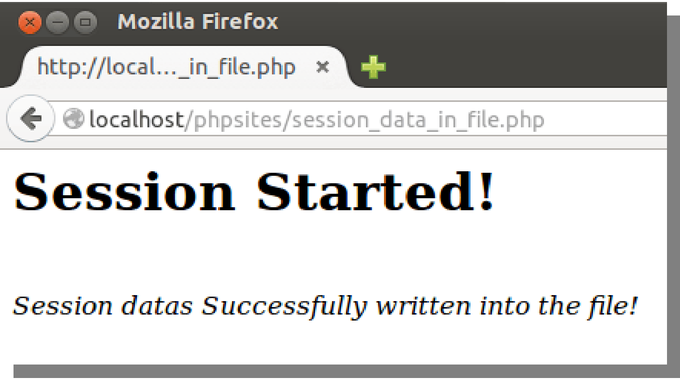
session இன் தகவல்களை ஒரு கோப்பில் எழுதி வைத்து விட்டால் தேவைப்படும் போது நாம் எடுத்து பயன்படுத்திக்கொள்ளலாம்.

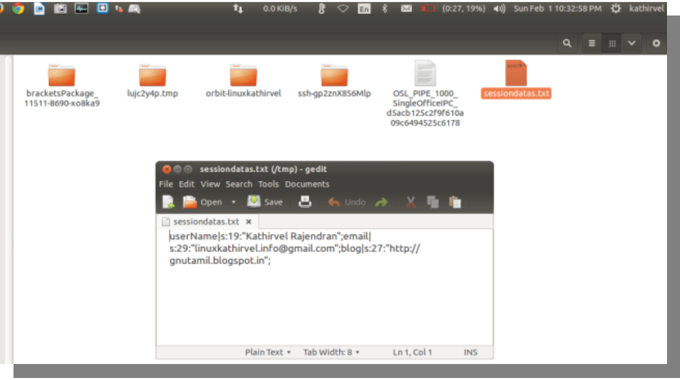
session இன் அனைத்து தகவல்களும் session_encode() எனும் function மூலமாக பெறப்படுகிறது. அவ்வாறு பெறப்படும் தகவல்கள் file function களுடன் சேர்த்து பயன்படுத்தப்படுகிறது.

உதாரண நிரல்:

```
<?php

//session creation
if(session_start())
{
echo "<h1>Session Started!
</h1>";
$_SESSION['userName'] =
'Kathirvel Rajendran';
$_SESSION['email'] = '<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>';
$_SESSION['blog'] =
'http://gnutamil.blogspot.in
';
}

else {
echo "<h1>Session Not
Create!</h1>";
}

//session data writer into
the file
if(isset($_SESSION['userName
']) &&
isset($_SESSION['email']) &&
```

```
isset($_SESSION['blog'])) {
//open a file for to save
session datas
$fileopen =
fopen('/tmp/sessiondatas.txt
','w+');
//get the session datas
$session_data =
session_encode();
//write the session datas
into the file

if(fwrite($fileopen,
$session_data)) {
echo "<i>Session Write
Successfully!</i>";
}

//close the file
fclose($fileopen);
}

else {
echo "Session Accessing
Failed!";
}
?>
```


Mozilla Firefox
http://local..._in_file.php
localhost/phpsites/session_data_in_file.php
Session Started!
Session datas Successfully written into the file!

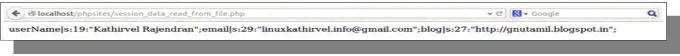


0.0 KiB/s
(0:27, 19%)
Sun Feb 1 10:32:58 PM
kathirvel
bracketsPackage_ 11511-8690-xo8ka9
lujc2y4p.tmp
orbit-linuxkathirvel
ssh-gp2znX856Mlp
OSL_PIPE_1000_ SingleOfficeIPC_ d5acb125c2f9f610a 09c6494525c6178
sessiondatas.txt
sessiondatas.txt (/tmp) - gedit
File Edit View Search Tools Documents
Open Save Undo
sessiondatas.txt
userName|s:19:"Kathirvel Rajendran";email|
s:29:"linuxkathirvel.info@gmail.com";blog|s:27:"http://
gnutamil.blogspot.in";
Plain Text Tab Width: 8 Ln 1, Col 1 INS

19.5 கோப்பில் சேமிக்கப்பட்ட *session* தகவல்களை படித்தல் (*Reading Saved Session*)

session_decode() function மூலமாக கோப்பில் சேமிக்கப்பட்ட தகவல்களை decode செய்யலாம்.

```
<?php
$fileopen =
fopen('/tmp/sessiondatas.txt
','r');
//read the session datas
$session_data =
fread($fileopen, 4096);
//close the file
fclose($fileopen);
session_decode($session_data
);
print_r($session_data);
?>
```


localhost/phpsites/session_data_read_from_file.php
Google
userName|s:19:"Kathirvel Rajendran";email|s:29:"linuxkathirvel.info@gmail.com";blog|s:27:"http://gnutamil.blogspot.in";

20 பொருள் நோக்கு நிரலாக்கம் (Object Oriented Programming)

பொருள்நோக்கு நிரலாக்கத்திற்கு PHP நன்கு ஆதரவு தருகிறது. பொருள்நோக்கு நிரலாக்கம் என்பது ஒரு பெரிய பகுதி இந்த தொடரில் மட்டுமே அதை பார்த்து விட முடியாது. இதற்கென தனியாக ஒரு புத்தகமே எழுதினாலும் போதாது அந்தளவிற்கு நிறைய செய்திகள் பொருள்நோக்கு நிரலாக்கத்தில் உள்ளது. PHP -யில் அனைத்தும் இருக்கிறது என்ற வகையில் OOP தொடர்பானவற்றையும் பார்த்து விடவேண்டும் என்ற நோக்கில் இதை எழுதியுள்ளேன். PHP யில் பொருள் நோக்கு நிரலாக்கம் எப்படி செய்வது? என்பது தொடர்பான அடிப்படை செய்திகளை இங்கு காண்போம்.

20.1 Object என்றால் என்ன?

Object என்பது செயல்கூறுகளின் பகுதிகளை சுயமாக கொண்டுள்ள ஒன்றாகும். இதை நாம் எளிமையாக பயன்படுத்திக்கொள்ளலாம் மற்றும் மறுசுழற்சியும் செய்துகொள்ளலாம்.

Objects தகவல் மாறிலிகள் மற்றும் செயல்கூறுகளைக்

கொண்டிருக்கும். Object ஐ பொறுத்தமட்டிலே செயல்கூறுகள்(functions) methods என அழைக்கப்படுகிறது. இவைகளை நமது பணிகளை முடிப்பதற்காக Object மூலமாக அழைத்துக்கொள்ளலாம். இவைகளனைத்தும் மொத்தமாக உறுப்பினர்கள்(members) என்று அழைக்கப்படுகிறது.

20.2 *Class* என்றால் என்ன?

Class என்பது, ஒரு கட்டிடத்தை கட்டுவதற்கு முன் அந்த கட்டிடத்திற்கான வரைபடத்தை தயாரிப்பதைப் போன்றது. கட்டிடம் கட்டப்படும் போது ஒவ்வொரு பொருளும் எப்படி தோற்றமளிக்க வேண்டும்? என வரைபடத்தில் சொல்லப்படுகிறது இல்லையா? அதுபோல Object உருவாக்கும் போது அது எப்படி தோற்றமளிக்க வேண்டும் என்பதை Class வரையறை செய்கிறது. உதாரணமாக methods கள் என்ன செய்ய வேண்டும், உறுப்பினர்கள்(members) எப்படி இருக்க வேண்டும் போன்றவற்றை வரையறை செய்கிறது.

20.3 *Class* —லிருந்து *Object* ஐ உருவாக்குவது எப்படி?

வரைபடத்திலிருந்து கட்டிடம் கட்ட ஆரம்பிப்பதைப்

போன்றதுதான் object உருவாக்குவதும். வரைபடத்தை வைத்துக்கொண்டு என்ன செய்வது, அதை பயன்பாட்டிற்கு கொண்டுவரவேண்டுமல்லவா, அது போலவே Class லிருந்து Object உருவாக்குவதும். வெறும் Class வைத்துக்கொண்டு என்ன செய்வது, அதை பயன்பாட்டிற்கு கொண்டு வர Object தேவை. ஒரு Class லிருந்து எத்தனை Object ஐ வேண்டுமானாலும் உருவாக்கிக்கொள்ளலாம். ஒருவரைபடத்தைக் கொண்டு எத்தனைக் கட்டிடத்தை வேண்டுமானாலும் கட்டலாம் அல்லவா அதுபோலத்தான்.

Object ஐக் கொண்டு Class இல் இருக்கும் methods மற்றும் மாறிலிகளை நான் அணுக முடியும் மற்றும் பயன்படுத்திக்கொள்ள முடியும். உதாரணமாக bankAccount எனும் class இன் Object ஐ கீழ்காணுமாறு உருவாக்க முடியும்.

```
$accountObject = new bankAcccount();
```

20.4 sub-classing என்றால் என்ன?

ஒரு class லிருந்து இன்னொரு class ஐ உருவாக்குவதை sub-class என்கிறோம். இதன்மூலமாக ஏற்கனவே உள்ள

ஒரு class இன் variables மற்றும் methods நான் புதிதாக உருவாக்கும் ஒரு class க்கும் கொண்டுவர முடியும். புதிதாக உருவாக்கப் போகும் class க்குத் தேவையானவைகள் 50% ஏற்கனவே உள்ள class இல் இருக்கும் போது, ஏற்கனவே உள்ளவற்றையே நாம் பயன்படுத்திக்கொண்டால் என்ன? 50% வேலைகள் மிச்சப்படுமில்லையா.

car எனும் class ஐ நீங்கள் உருவாக்குவதாக வைத்துகொள்வோம், அதற்கென்று சிறப்பாக உள்ளவற்றைத் தவிர்த்து மற்றவையெல்லாம் vechile எனும் class இல் ஏற்கனவே உள்ளதென்றால் அதை நாம் பயன்படுத்திக்கொள்ளலாம். இதுதான் sub-class. இங்கு vechile class ஆனது Parent class எனவும், car class ஆனது child class அல்லது sub-class எனவும் அழைக்கப்படுகிறது.

20.5 PHP class ஐ வரையறை செய்தல்

கட்டிடம் கட்டுவதெல்லாம் சரி அதற்கு வரைபடம் வேண்டுமல்லவா? Object ஐ உருவாக்குவதற்கு முன்பு Class வேண்டுமல்லவா? PHP யில் class ஐ உருவாக்க **class** எனும் முதன்மைச்சொல்(keyword) பயன்படுத்தப்படுகிறது. Class இன் உடலை(body)

வரையறை செய்ய curly braces ({}) பயன்படுத்தப்படுகிறது.

```
<?php

class bankAccount {

}

?>
```

இப்பொழுது நாம் புதிதாக ஒரு class ஐ உருவாக்கிவிட்டோம். அடுத்து அதில் சில செயல்கூறுகளை சேர்ப்பது பற்றிப் பார்ப்போம்.

20.6 PHP class உருவாக்குதல் மற்றும் சிதைத்தல் (class constructors and destructors)

அடுத்தபடி என்னவென்றால், Object ஐ உருவாக்கும் போது என்ன நடக்க வேண்டும் என்பதை வரையறை

செய்வது. இறுதியில் Object ஐ சிதைப்பது. இந்த இரண்டு செயல்களும் constructor மற்றும் destructor method களை பயன்படுத்தி வரையறை செய்யப்படுகிறது.

Constructor மற்றும் destructor இரண்டும் செயல்கூறுகள்தான்(functions) function எனும் முதன்மைச் சொல்லை பயன்படுத்தி அழைக்கப்படுகிறது. Function முதன்மைச் சொல்லுக்கு முன் public எனும் qualifier பயன்படுத்த வேண்டும். public qualifier பயன்படுத்துவதால் object க்கு வெளியேயும் function ஐ நாம் பயன்படுத்த முடியும்.

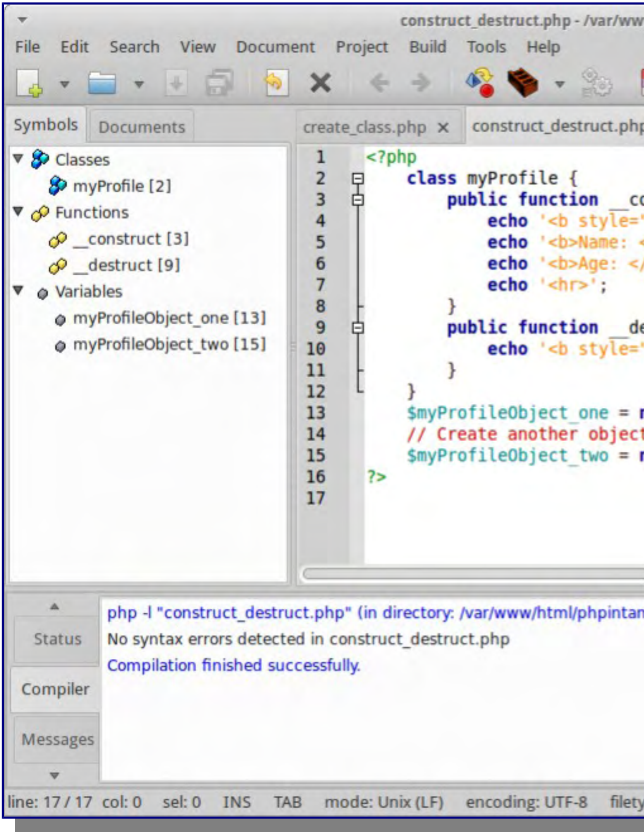
constructor மற்றும் destructor ஆகிய இரண்டுக்கும் இருப்பியல்பான பெயர்கள் __construct மற்றும் __destruct ஆகும். function -னில் argument கொடுப்பதுபோல __construct and __destruct இரண்டுக்கும் கொடுத்துக்கொள்ளலாம்.

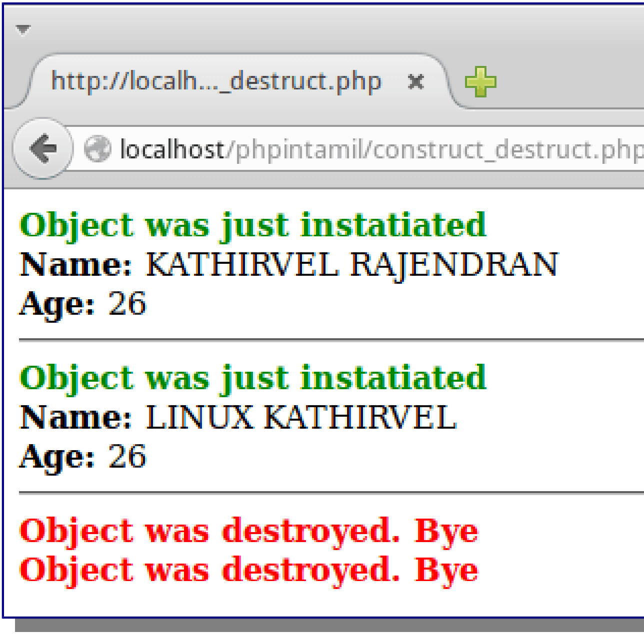
உதாரண நிரலைப் பாருங்கள்

```
<?php
```

```
class myProfile {

public function
__construct($myName, $myAge)
{
echo '<b
style="color:green;">Object
was just
instatiated</b><br>';
echo '<b>Name: </b>'.
$myName.'<br>';
echo '<b>Age: </b>'.
$myAge.'<br>';
echo '<hr>';
}

public function __destruct()
{
echo '<b
style="color:red;">Object
was destroyed. Bye</b><br>';
}

}

$myProfileObject_one = new
myProfile('KATHIRVEL
RAJENDRAN', 26);

// Create another object
```

```
$myProfileObject_two = new
myProfile('LINUX KATHIRVEL',
26);
?>
```


construct_destruct.php - /var/ww
File Edit Search View Document Project Build Tools Help
Symbols
Documents
create_class.php
construct_destruct.php
Classes
myProfile [2]
Functions
__construct [3]
__destruct [9]
Variables
myProfileObject_one [13]
myProfileObject_two [15]
<?php
class myProfile {
public function __co
echo '<b style='
echo '<b>Name: <
echo '<b>Age: </
echo '<hr>';
}
public function __de
echo '<b style='
}
}
$myProfileObject_one =
// Create another object
$myProfileObject_two =
?>
Status
Compiler
Messages
php -l "construct_destruct.php" (in directory: /var/www/html/phpintan
No syntax errors detected in construct_destruct.php
Compilation finished successfully.
line: 17 / 17 col: 0 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filety



20.7 PHP class இல் உறுப்பினர்கள்(members) உருவாக்குதல்

class -க்குள் *இருக்கும் அத்தியாவசியமான மாறிலிகள் மற்றும் செயல்கூறுகள்* class members *எனப்படுகின்றன.*

Members public அல்லது private ஆக இருக்கலாம், static அல்லது variable ஆக இருக்கலாம்.

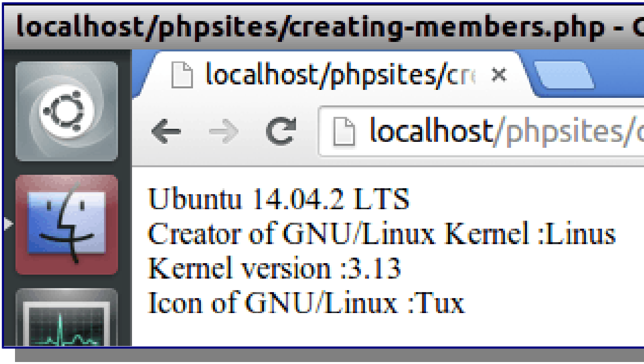
public members ஐ object க்கு வெளியிலும் பயன்படுத்த முடியும். private members ஐ அதனுடைய class க்குள் மட்டுமே பயன்படுத்த முடியும். இதுவே data encapsulation என்று அழைக்கப்படுகிறது.

Static member இன் மதிப்பை மாற்றமுடியாது, வரையறை மட்டும்தான் செய்ய முடியும். Members மற்றும் செயல்கூறுகளை(functions) class உள்ளே வரையறை செய்யும்போது, இவைகள் public, private மற்றும் static ஆகிய முதன்மைச்சொல்லினை முன்இணைப்பாக கொண்டு இருக்கும். Default ஆக public என்று இருக்கும்.

உதாரண நிரலைப் பாருங்கள்:

```
<?php
class LinuxProfile {
public $creator;
public $kernel;
public $icon;

// This variable is private.
private $os;

public function
__construct($creator,
$kernel, $icon, $os) {
$this->creator = $creator;
$this->kernel = $kernel;
$this->icon = $icon;
$this->os = $os;
echo $this->os;
echo '<br>';
}

}

// Create object for
LinuxProfile class
$linux = new
LinuxProfile('Linus',
'3.13', 'Tux', 'Ubuntu
14.04.2 LTS');

// Access members using
objects
```

```
echo 'Creator of GNU/Linux
Kernel :'.$linux->creator;
echo '<br>';
echo 'Kernel version :'.
$linux->kernel;
echo '<br>';
echo 'Icon of GNU/Linux :'.
$linux->icon;

// This variable will not
print due to private.
echo $linux->os;
?>
```



மேலே உள்ள நிரலில் நாம் public எனும் முதன்மைச்சொல்லை பயன்படுத்தியுள்ளேன். அதனால் class க்கு வெளியிலும் $creator, $kernel, $icon variable களை அணுக முடிந்துள்ளது. Private ஆக வரையறுத்துள்ள $os ஐ class -க்கு வெளியில் அணுக முடியவில்லை.

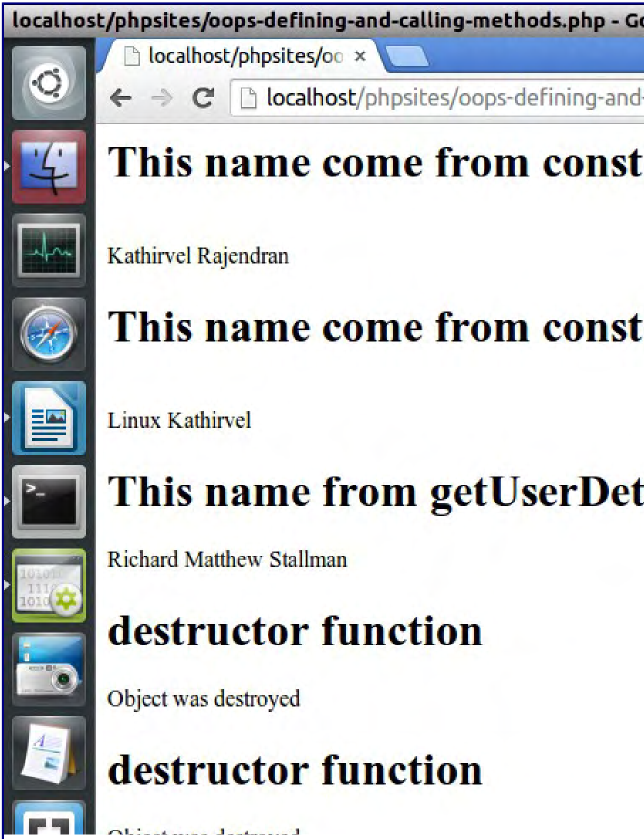
20.8 Methods ஐ வரையறை செய்தல் மற்றும் அழைத்தல்(Defining and Calling Methods)

நாம் ஏற்கனவே constructor மற்றும் destructor method களை உருவாக்கியதுபோல, நாம் நம்முடைய சொந்த methods களை அதே வழியில் உருவாக்கிக்கொள்ளலாம். நாம் விரும்பிய பெயரையும் methods களுக்கு கொடுத்துக்கொள்ளலாம். உதாரண நிரலை பாருங்கள்.

```
<?php

class User {
private $firstName;
private $lastName;

public function
__construct($fname, $lname)
```

```
{
$this->firstName = $fname;
$this->lastName = $lname;
echo '<h1>This name come
from constructor
function.</h1><br>';
echo $this->firstName.' '.
$this->lastName;
echo '<br>';
}

public function __destruct()
{
echo '<h1>destructor
function</h1>';
echo 'Object was destroyed
<br>';
}

public function
setUserDetails($userName,
$fatherName) {
$this->firstName =
$userName;
$this->lastName =
$fatherName;
}

public function
getUserDetails() {
return $this->firstName.' '.
```

```
$this->lastName;
}

}




// Create object to access
methods
$user1 = new
User('Kathirvel',
'Rajendran');
$user2 = new User('Linux',
'Kathirvel');
$user1-
>setUserDetails('Richard
Matthew', 'Stallman');

echo '<h1>This name from
getUserDetails
function.</h1>';
echo $user1-
>getUserDetails();
?>
```


localhost/phpsites/oops-defining-and-calling-methods.php - Go
localhost/phpsites/oo ×
localhost/phpsites/oops-defining-and-
This name come from const
Kathirvel Rajendran
This name come from const
Linux Kathirvel
This name from getUserDet
Richard Matthew Stallman
destructor function
Object was destroyed
destructor function

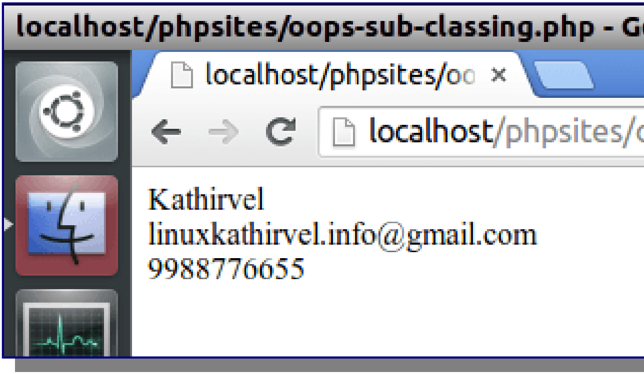
Object ஐத் தொடர்ந்து '->' ஐப் பயன்படுத்தி getUserDetails method ஐ அழைத்துள்ளோம்.

20.9 Subclassing in PHP

ஒரு class லிருந்து இன்னொரு class ஐ உருவாக்குதல் subclass எனப்படும். ஒரு class ஐ உருவாக்கிவிட்டாலே, அதிலிருந்து நம்மால் புதியதாக ஒரு class ஐ உருவாக்கிக்கொள்ள முடியும். இந்த முறைக்கு Inheritence என்று பெயர். Extends எனும் முதன்மைச்சொல்லைப் பயன்படுத்தி நாம் subclass ஐ உருவாக்கிக்கொள்ளலாம்.

கீழே உள்ள நிரலில் ParentClass இல் உள்ள name, email விபரங்களை, childclass object ஐ பயன்படுத்தி print செய்துள்ளோம். அதாவது ParentClass variables களை ChildClass பயன்படுத்தியுள்ளது. Methods களையும் இதுபோல பயன்படுத்திக்கொள்ளலாம்.

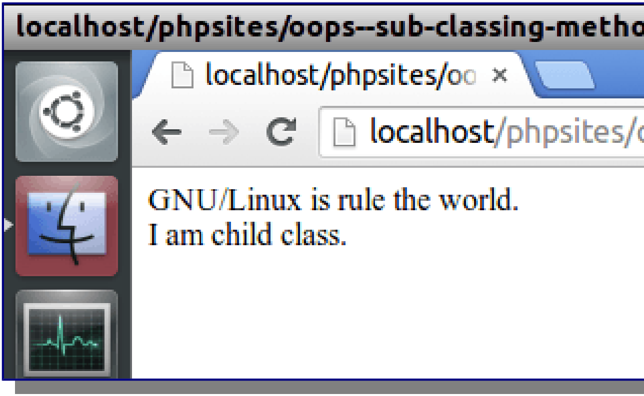
```
<?php

class ParentClass {
public $name = 'Kathirvel';
public $email = '<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>';
}

class ChildClass extends
ParentClass {
public $mobile =
'9988776655';
}

$childclass = new
ChildClass();

// print name, email using
child class
echo $childclass->name;
echo '<br>';
echo $childclass->email;
echo '<br>';
echo $childclass->mobile;
?>
```



20.10 *ChildClass* மூலமாக *ParentClass* இன் *method* ஐ பயன்படுத்திக்கொள்ளுதல்

```
<?php

class ParentClass {
public $message1 =
'GNU/Linux is rule the
world.<br>';

public function
```

```
printMessage() {
echo $this->message1;
}

}

class ChildClass extends
ParentClass {
public $message2 = 'I am
ChildClass.<br>';
}

// Create ChildClass object
$child = new ChildClass();

// call printMessage method
through ChildClass
$child->printMessage();
echo $child->message2;
?>
```



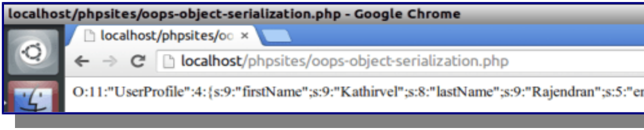
20.11 PHP Object Serialization

object ஐ serialize செய்வதற்கு serialize() எனும் செயல்கூறு பயன்படுத்தப்படுகிறது. Unserialize செய்ய unserialize() செயல்கூறு பயன்படுத்தப்படுகிறது.

```
<?php
class UserProfile {
```

```
public $firstName;
public $lastName;
public $email;
public $mobile;

public function
__construct($name,
$fatherName, $email, $mob) {
$this->firstName = $name;
$this->lastName =
$fatherName;
$this->email = $email;
$this->mobile = $mob;
}

}

// Create object
$profile = new
UserProfile('Kathirvel',
'Rajendran', '<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>',
'9988776655');

// Serialize using
serialize() function
```

```
$ser = serialize($profile);

// print serialized values
echo $ser;
?>
```



20.12 PHP Object பற்றிய தகவல்களைப் பெறுதல்

get_declared_classes() – Declare செய்துள்ள class பெயர்களை array வடிவில் கொடுக்கும்.

class_exists() – Class பெயரை உள்ளீடாக கொடுத்தால், இருக்கிறதென்றால் 1 என்று வெளியீடு வரும். இல்லையென்றால் எதுவும் வெளியீடாக வராது.

get_class_methods() – பயன்படுத்தியுள்ள methods களின் பெயர் பட்டியலைக் கொடுக்கும்.

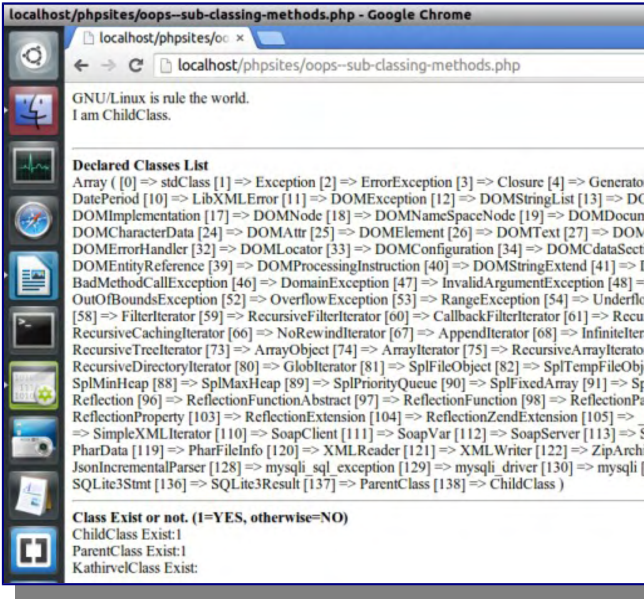
get_parent_class() – Parent class களின் பட்டியல் கொடுக்கும் இல்லையென்றால் empty string ஐ வெளியிடும்.

method_exists() – method பெயரைக் argument ஆக கொடுத்தால் இருக்கிறதென்றால் true எனவும், இல்லையென்றால் false எனவும் வெளியீட்டைக்கொடுக்கும்.

```
<?php

class ParentClass {
public $message1 =
'GNU/Linux is rule the
world.<br>';

public function
printMessage() {
echo $this->message1;
}

}
```

```
class ChildClass extends
ParentClass {
public $message2 = 'I am
ChildClass.<br>';
}

// Create ChildClass object
$child = new ChildClass();

// call printMessage method
through ChildClass
$child->printMessage();
echo $child->message2;

// Get declared class names,
It will print as Array.
echo '<br><hr>';
echo '<b>Declared Classes
List<br></b>';
print_r(get_declared_classes
());

// Find Class Exist or not
echo '<br><hr>';
echo '<b>Class Exist or not.
(1=YES,
otherwise=NO)</b><br>';
echo "ChildClass
Exist:".class_exists('ChildC
```

```
lass').'<br>';
echo "ParentClass
Exist:".class_exists('Parent
Class').'<br>';
echo "KathirvelClass
Exist:".class_exists('Kathir
velClass').'<br>';
?>
```


localhost/phpsites/oops--sub-classing-methods.php - Google Chrome
localhost/phpsites/oo
localhost/phpsites/oops--sub-classing-methods.php
GNU/Linux is rule the world.
I am ChildClass.
Declared Classes List
Array ([0] => stdClass [1] => Exception [2] => ErrorException [3] => Closure [4] => Generato
DatePeriod [10] => LibXMLError [11] => DOMException [12] => DOMStringList [13] => DO
DOMImplementation [17] => DOMNode [18] => DOMNameSpaceNode [19] => DOMDocum
DOMCharacterData [24] => DOMAttr [25] => DOMElement [26] => DOMText [27] => DOM
DOMErrorHandler [32] => DOMLocator [33] => DOMConfiguration [34] => DOMCdataSecti
DOMEntityReference [39] => DOMProcessingInstruction [40] => DOMStringExtend [41] =>
BadMethodCallException [46] => DomainException [47] => InvalidArgumentException [48] =
OutOfBoundsException [52] => OverflowException [53] => RangeException [54] => Underflo
[58] => FilterIterator [59] => RecursiveFilterIterator [60] => CallbackFilterIterator [61] => Recu
RecursiveCachingIterator [66] => NoRewindIterator [67] => AppendIterator [68] => InfiniteIter
RecursiveTreeIterator [73] => ArrayObject [74] => ArrayIterator [75] => RecursiveArrayIterato
RecursiveDirectoryIterator [80] => GlobIterator [81] => SplFileObject [82] => SplTempFileObj
SplMinHeap [88] => SplMaxHeap [89] => SplPriorityQueue [90] => SplFixedArray [91] => Sp
Reflection [96] => ReflectionFunctionAbstract [97] => ReflectionFunction [98] => ReflectionPa
ReflectionProperty [103] => ReflectionExtension [104] => ReflectionZendExtension [105] =>
=> SimpleXMLIterator [110] => SoapClient [111] => SoapVar [112] => SoapServer [113] =>
PharData [119] => PharFileInfo [120] => XMLReader [121] => XMLWriter [122] => ZipArchi
JsonIncrementalParser [128] => mysqli_sql_exception [129] => mysqli_driver [130] => mysqli
SQLite3Stmt [136] => SQLite3Result [137] => ParentClass [138] => ChildClass)
Class Exist or not. (1=YES, otherwise=NO)
ChildClass Exist:1
ParentClass Exist:1
KathirvelClass Exist:

21 PHP யும் தரவுத்தளமும் (Using PHP with MySQL)

நாம் அன்றாடம் பயன்படுத்தும் தொலைபேசிகள், நாற்காலிகள், கணினிகள் போன்றவைகளை உருவாக்க பிளாஸ்டிக் என்பது எப்படி அவசியமானதோ அதே போன்றுதான் இணைய உலகில் தரவுத்தளமும்(Database). பிளாஸ்டிக் இல்லாத உலகை நாம் முடிவு செய்தால், இந்த உலகத்தில் பாதிக்கு மேலான பொருட்களை நம்மால் பயன்படுத்த முடியாது. அதுபோலவே தரவுத்தளம் இல்லையென்றால் பாதிக்கு மேலான இணையதளங்கள் பயனில்லாததாகிவிடும். இன்னும் சிறப்பாக சொல்ல வேண்டுமென்றால் தரவுதளம்தான் இணையம் மற்றும் வலை ஆகியவைகளின் இதயம் என்று கூடச் சொல்லலாம். தகவல்களை சேமிக்கவும், சேமித்த தகவல்களை திரும்ப எடுக்கவும் வழியில்லையென்றால் இணையம் என்பது பயனற்ற ஒன்றாக ஆகிவிடும்.

MySQL உடன் PHP யை எளிமையாக பயன்படுத்தலாம். இது PHP யின் முக்கியமான அம்சங்களில் ஒன்றாகும். MySQL ஐப் பற்றி விரிவாகவும் தெளிவாகவும் தெரிந்து கொள்ள தமிழில் த.நித்யா அவர்கள் எழுதிய புத்தகத்தைப்

படிக்கவும். இந்த புத்தகத்தில் MySQL ஐப் பற்றிய தகவல்கள் அனைத்தும் விரிவாக அனைவருக்கும் புரியும் படி விளக்கப்பட்டுள்ளது. ஆகையால் நாம் நேரடியாக PHP உடன் MySQL ஐப் இணைப்பதைப் பற்றி பார்க்கலாம்.

21.1 PHP உடன் MySQL ஐ இணைத்தல் (Connect with PHP to a MySQL Server)

PHP உடம் MySQL தரவுத்தளத்தை இணைப்பதற்கு mysql_connect() எனும் function பயன்படுத்தப்படுகிறது. mysql_connect function ஆனது தரவுத்தளத்துடன் ஒரு இணைப்பை ஏற்படுத்தி தரவுத்தளத்தில் இருக்கும் தகவல்களை நாம் அணுகுவதற்கான வசதியை ஏற்படுத்தித் தருகிறது. mysql_connect function ஐந்து arguments களை பெற்றுக்கொண்டு நமக்கு இணைப்பை ஏற்படுத்தித் தருகிறது. இதில் முதல் மூன்று arguments கள் அவசியமானவைகள்.

முதல் argument தரவுத்தளம் இருக்கும் சேவையகத்தின்(server) முகவரி, இது default ஆக localhost:3306 என இருக்கும். இரண்டாவது argument தரவுத்தளத்திற்குள் நுழைவதற்கான பயனரின் பெயர், மூன்றாதவது argument பயனருக்குண்டான

கடவுச்சொல்(password).

தரவுத்தளத்துடனான இணைப்பைத் துண்டிப்பதற்கு mysql_close() எனும் function பயன்படுத்தப்படுகிறது. எந்த இணைப்பை நாம் துண்டிக்க வேண்டுமோ அந்த இணைப்பின் பெயரை இதற்கு argument ஆக கொடுக்க வேண்டும்.

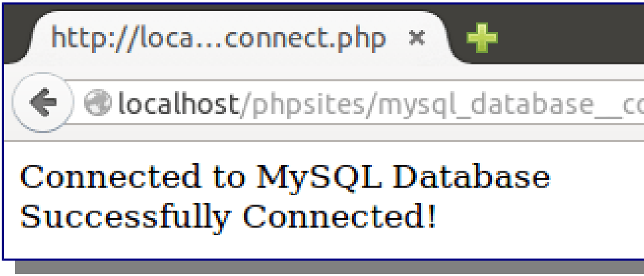
இப்பொழுது PHP யைக் கொண்டு MySQL தரவுத்தளத்துடன் இணைப்பை ஏற்படுத்துவதற்குண்டான நிரலைப் பார்ப்போமா? இந்த நிரலில் நான் பயன்படுத்தியிருக்கும் பயனர் பெயர், கடவுச்சொல், தரவுத்தளத்தின்(Database) பெயர் ஆகியவைகள் என்னுடைய கணினியில் நான் அமைத்து வைத்திருப்பது. உங்களுடைய கணினியில் இருப்பதற்கு ஏற்ப மேற்கண்டவைகளின் மதிப்புகளைக் நீங்கள் கொடுத்துக்கொள்ளுங்கள்.

நிரல்:

```
<?php
```

```
$dbhandle =
mysql_connect('localhost','r
oot', 'password');

//servername, username,
password of user

if($dbhandle) {
echo "Connected to MySQL
Database<br>";
echo "Successfully
Connected!";
mysql_close($dbhandle);
}

else {
echo "Unable to connect to
MySQL Database.<br>";
}

?>
```

வெளியீடு:



21.2 MySQL தரவுதளத்திலிருந்து PHP மூலமாக பதிவேடுகளை(Record) தேர்வு செய்தல் (Selecting Records from a MySQL Database Using PHP):

நாம் தற்போது MySQL தரவுத்தளத்துடன் வெற்றிகரமாக இணைப்பை ஏற்படுத்தி விட்டோம். அடுத்ததாக தரவுத்தளத்தில் இருக்கும் தகவல்களை அணுக வேண்டும். அதற்கு முதலில் நாம் தகவல்கள் சேமித்து வைக்கப்பட்டிருக்கும் தரவுத்தளத்தினை(Database) தேர்வு செய்ய வேண்டும். அதன்பிறகுதான் நம்மால் தரவுத்தளத்திற்குள் இருக்கும் அட்டவணைகளில்(Tables) இருந்து தகவல்களை பெற முடியும். ஆகையால் நாம் முதலில் தரவுத்தளத்தினை தேர்வு செய்வது எப்படி என்று பார்ப்போம். தரவுத்தளத்தினை தேர்வு செய்வதற்கு mysql_select_db() எனும் function பயன்படுகிறது.

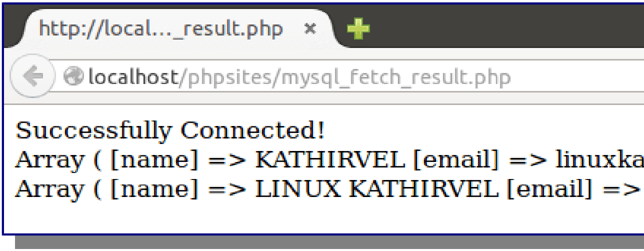
அதன்பிறகு நம்முடைய SQL Query களை mysql_query() function க்கு argument ஆக கொடுப்பதன் மூலமாக தகவல்களை அணுக முடியும்.

mysql_query() function மூலமாக கிடைக்கும் முடிவுகள்(results) array -யில் சேமிக்கப்படுகிறது. அவ்வாறு array -யில் சேமிக்கப்படும் முடிவுகளை mysql_fetch_array() funtion மூலமாக நாம் பெற்றுக்கொள்ளலாம்.

சரி மேல சொல்லப்பட்ட கருத்துக்களுக்கான நிரலைப் பார்ப்போமா?

```
<?php

$dbhandle =
mysql_connect('localhost','r
oot', 'password');
//servername, username,
password of user

if($dbhandle) {
echo "Successfully
Connected!<br />";
```

```
$db =
mysql_select_db('phptest');
$query = 'SELECT * FROM
customer';
$query_result =
mysql_query($query,
$dbhandle);

if(!$query_result) {
echo "Unable to perform
query!<br />";
}

else {
while( $result_row =
mysql_fetch_array($query_res
ult, MYSQL_ASSOC)) {
print_r($result_row);
echo "<br />";
}
}
}

mysql_close($dbhandle);
?>
```



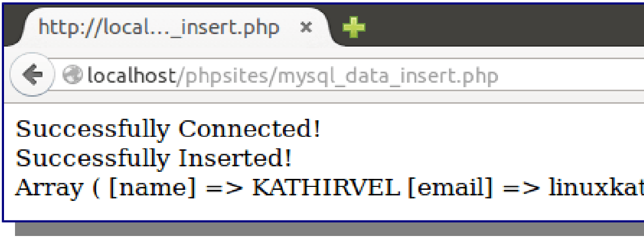
21.3 பதிவேட்டில் தகவல்களை சேர்த்தல் Adding Records to MySQL Database using PHP

தரவுத்தளத்தில் பதிவேட்டில் தகவல்களை சேர்ப்பது மிகவும் எளிமையானது. இதற்கும் mysql_query() function *பயன்படுத்திக்கொள்ளலாம்.* SQL Query *யை மட்டும்* Insert Query *யாக மாற்ற வேண்டியதுதான்.*

```
<?php

$dbhandle =
mysql_connect('localhost','r
```

```
oot', 'password');
//servername, username,
password of user

if($dbhandle) {
echo "Successfully
Connected!<br />";
$db =
mysql_select_db('phptest');
$insert_query = "insert into
customer(name, email,
mobileno)
values( 'KATHIRVEL', '<a
class="autohyperlink"
href="mailto:linuxkathirvel.
info@gmail.com"
title="mailto:linuxkathirvel
.info@gmail.com">linuxkathir
vel.info@gmail.com</a>',
'9900990099')";
echo $query;
$select_query = "select *
from customer";
$query_insert =
mysql_query($insert_query,
$dbhandle);
$query_select =
mysql_query($select_query,
$dbhandle);

if($query_insert) {
```

```
echo "Successfully Inserted!
<br />";
}

else {
echo "Unable to perform
inset query!<br />";
}

if(!$query_select) {
echo "Unable to perform
query!<br />";
}

else {
while( $result_row =
mysql_fetch_array($query_sel
ect, MYSQL_ASSOC)) {
print_r($result_row);
echo "<br />";
}
}
}

mysql_close($dbhandle);
?>
```



21.4 *Using PHP to get Information about a MySQL Database*

MySQL தரவுத்தளத்தினுடைய தகவல்களைப் பெறுவதற்கென PHP பல்வேறு பயனுள்ள functions களைக் கொண்டுள்ளது. ஒரு அட்டவணையில் இருக்கும் fields களின் பட்டியல்களைப் பெறுவதற்கு mysql_list_fields() function பயன்படுகிறது. தரவுத்தளத்தின் பெயர், அட்டவணையின் பெயர், mysql_connect() function -னிலிருந்து கிடைத்த தகவல் ஆகிய மூன்று arguments களை உள்ளீடாக இந்த function பெற்றுக்கொள்கிறது.

அட்டவணையில் இருக்கும் fields களின் எண்ணிக்கையைப் பெறுவதற்கு mysql_num_fields() funciton பயன்படுகிறது. இந்த function mysql_list_field() function மூலமாக கிடைக்கும் resource identifier ஐ argument ஆக எடுத்துக்கொள்கிறது.

ஒருமுறை mysql_list_fields() funtion லிருந்து resource identifier ஐ நாம் பெற்றுவிட்டால் அதன்பின் அட்டவணையின் ஒவ்வொரு field ஐப் பற்றிய தகவல்களையும் பெறுவதற்கு mysql_field_name(), mysql_field_type(), mysql_field_len() போன்ற function களைப் பயன்படுத்திக்கொள்ளலாம்.

```
<?php

$dbhandle =
mysql_connect('localhost','r
oot', 'password');
//servername, username,
password of user

if($dbhandle) {
echo "Successfully
Connected!<br />";
$db =
```

```
mysql_select_db('phptest');
}

// Obtain the Database
Information
$table_fields_list =
mysql_list_fields('phptest',
'customer', $dbhandle);
$table_no_of_fields =
mysql_num_fields($table_fiel
ds_list);

for($i=0;
$i<$table_no_of_fields; $i+
+) {
echo '<b>Field Name:</b> '.
mysql_field_name($table_fiel
ds_list, $i). " - ";
echo '<b>Field Type:</b> '.
mysql_field_type($table_fiel
ds_list, $i). " - ";
echo '<b>Field Length:</b>
'.
mysql_field_len($table_field
s_list, $i);
echo "<br />";
}

mysql_close($dbhandle);
?>
```

http://local...n_fetch.php
localhost/phpsites/mysql_information_fetch.php
Successfully Connected!
Field Name: name - Field Type: string -
Field Name: email - Field Type: string -
Field Name: mobileno - Field Type: string -

22 PHP மற்றும் SQLite (PHP and SQLite)

22.1 PHP வழியாக SQLite Database உருவாக்குதல் (Creating an SQLite Database with PHP)

SQLite என்பது MySQL போல ஒரு Client, Server ஆக இல்லாமல், மொத்த தரவுதளமும் ஒரு கோப்பாகவே செயல்படும் ஒரு மென்பொருளாகும். இது PHP உடன் சேர்த்தே நிறுவப் படுகிறது.

குறைந்த அளவிலான தகவல்களை சேமிக்க, இதைப் பயன்படுத்தலாம்.

22.2 PDO (PHP Data Objects) மூலமாக SQLite DB ஐ உருவாக்குதல்

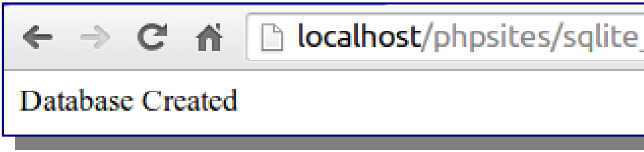
நாம் புதிதாக ஒரு SQLite Database ஐ உருவாக்குவதற்கும், ஏற்கனவே இருக்கும் SQLite Database ஐ திறந்து பயன்படுத்துவதற்கும் PDO பயன்படுகிறது. PDO க்கு sqlite:sqliteDBname.db எனும் முறையில் argument ஐ கொடுக்கவும். SQLite DB உருவாக்கப்படுவதில் ஏதேனும் பிரச்சனையென்றால் Catch இல் நாம் கொடுத்துள்ளபடி

பிழைச்செய்தி காண்பிக்கப்படும்.

கீழ் உள்ள உதாரண நிரலைப் பாருங்கள்:

```
<?php

try
{
$con = new
PDO('sqlite:customer2.db');

if ($con) {
echo "Database Created";
}
}

catch (PDOException $e)
{
echo "DB Connections
Failed!" . $e->getMessage();
}

$con = null;
?>
```

திறக்கப்பட்ட DB ஐ மூடுவதற்கு null மதிப்பை PDO வுக்கு கொடுக்க வேண்டும்.



22.3 PHP மூலமாக SQLite இல் Table உருவாக்குதல் (Using PHP to Create Table to an SQLite Database)

PDO மூலமாக Table ஐ உருவாக்குவது எளிதானதுதான், Table ஐ உருவாக்குவதற்கான query நாம் தயார் செய்த பின்பு அதனை exec எனும் function க்கு உள்ளீடாக கொடுக்க வேண்டியதுதான். query வெற்றிகரமாக execute செய்யப்பட்ட பின்பு Table உருவாக்கப்பட்டிருக்கும்.

மாதிரி நிரல்

```
<?php

try
{
$con = new
```

```
PDO('sqlite:customer2.db');

if ($con) {
echo "<p>Database
Created</p>";
}

echo "<p>DB Connected
Successfully!</p>";

// Table Creation
$create_table_query =
"create table
product(distribution
varchar(10), latest_version
varchar(10))";

// Queries Execution
$create = $con-
>exec($create_table_query);

catch (PDOException $e)
{
echo "DB Connections
Failed!" . $e->getMessage();
}

$con = null;
?>
```

22.4 Using PHP to Add Records to an SQLite Database

Table இல் Records களை சேர்ப்பதற்கு அதற்கான query ஐ தயார் செய்த பின்பு அதனை PDO Object மூலமாக exec() க்கு உள்ளீடாக கொடுத்து இயக்க வேண்டியதுதான்.

நிரல்

```
<?php

try
{
$con = new
PDO('sqlite:customer2.db');

if ($con) {
echo "<p>Database
Created</p>";
}

echo "<p>DB Connected
Successfully!</p>";

// Table Creation
$create_table_query =
"create table
```

```
product(distribution
varchar(10), latest_version
varchar(10))";

// Insert Data Into Table
$insert_data_query1 =
"insert into
product(distribution,
latest_version)
values('Ubuntu', '14.10')";
$insert_data_query2 =
"insert into
product(distribution,
latest_version)
values('Fedora', '21')";

// Queries Execution
$create = $con-
>exec($create_table_query);
$insert1 = $con-
>exec($insert_data_query1);
$insert2 = $con-
>exec($insert_data_query2);
}

catch (PDOException $e)
{
echo "DB Connections
Failed!" . $e->getMessage();
}
```

```
$con = null;
?>
```

22.5 PHP மூலமாக Records களை தேர்வு செய்தல் (Using PHP to Select Records from an SQLite Database)

Table இருக்கும் தகவல்களை Select செய்வதும் எளிமையானதுதான். நாம் select செய்ய வேண்டிய தகவல்களுக்கு ஏற்ப சரியான query ஐ தயார் செய்து விட்டு, அதை PDO மூலமாக query() function உள்ளீடாக கொடுக்க வேண்டியதுதான். query() function அதற்கான முடிவுகளை Associative Array யாக திருப்பிக்கொடுக்கும். அதன்பின் நாம் foreach loop ஐக் கொண்டு தகவல்களை பிரித்து எடுத்துக்கொள்ளலாம்.

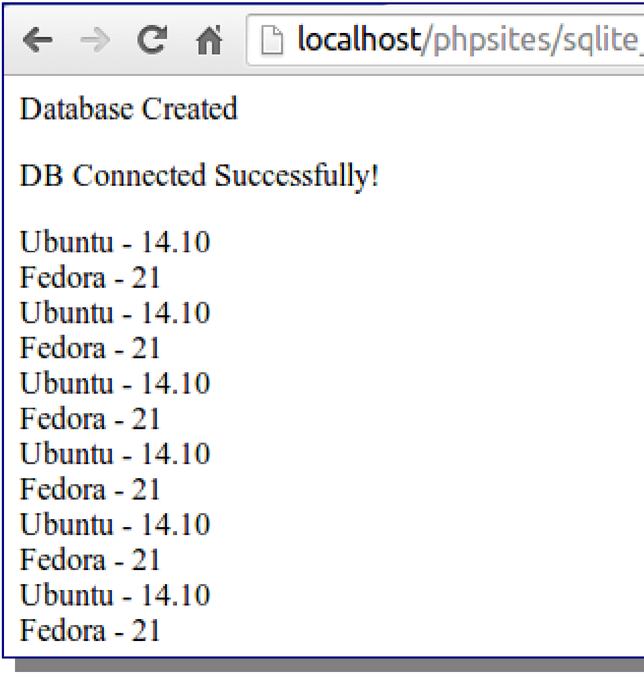
நிரல்:

```
<?php

try
```

```
{
$con = new
PDO('sqlite:customer2.db');

if ($con) {
echo "<p>Database
Created</p>";
}
echo "<p>DB Connected
Successfully!</p>";

// Table Creation
$create_table_query =
"create table
product(distribution
varchar(10), latest_version
varchar(10))";

// Insert Data Into Table
$insert_data_query1 =
"insert into
product(distribution,
latest_version)
values('Ubuntu', '14.10')";
$insert_data_query2 =
"insert into
product(distribution,
latest_version)
values('Fedora', '21')";

// Select Query
```

```
$query = "select * from
product";

// Queries Execution
$create = $con-
>exec($create_table_query);
$insert1 = $con-
>exec($insert_data_query1);
$insert2 = $con-
>exec($insert_data_query2);
$temp = $con->query($query);

foreach($temp as $details) {
echo
$details['distribution']." -
".
$details['latest_version'];
echo "<br />";
}
}

catch (PDOException $e)
{
echo "DB Connections
Failed!" . $e->getMessage();
}

$con = null;
?>
```

நிரலுக்கான வெளியீடு:


localhost/phpsites/sqlite_
Database Created
DB Connected Successfully!
Ubuntu - 14.10
Fedora - 21
Ubuntu - 14.10
Fedora - 21
Ubuntu - 14.10
Fedora - 21
Ubuntu - 14.10
Fedora - 21
Ubuntu - 14.10
Fedora - 21
Ubuntu - 14.10
Fedora - 21

23 முடிவுரை

PHP பற்றிய அடிப்படை செய்திகளை மட்டும் இங்கு பார்த்துள்ளோம். நல்ல கைதேர்ந்த PHP Developer ஆக ஆகவேண்டுமென்றால் PHP அடிப்படைகளைத் தாண்டி நாம் நிறைய கற்றுக்கொள்ள வேண்டும். அதுமட்டுமில்லாமல் HTML, CSS, JQUERY, JAVA SCRIPT, MY SQL போன்றவைகளையும் அவசியம் கற்க வேண்டும். jQuery, Java Script போன்றவைகளைத் தவிர HTML, CSS, MY SQL போன்ற தொழில்நுட்பங்களை நீங்கள் கணியம் மூலமாகவே கற்றுக்கொள்ளலாம். இவைகளனைத்தும் கணியம் தளத்திலே மின்னூலாகவே கிடைக்கின்றன. மிக எளிமையாக தமிழிலேயே நீங்கள் கற்றுக்கொள்ளலாம். Freshers ஆக வேலைக்குச் செல்பவர்களுக்கு இவைகள் போதுமானது. WordPress ஐ நீங்கள் கற்றிருந்தால் இன்னும் கூடுதல் மதிப்பு உங்களுக்கு கிடைக்கும். அடிப்படையை நன்கு புரிந்து கொண்டு வீட்டீர்களேயானால் அதன்பின் நீங்கள் சுயமாகவே இணையத்தின் மூலமாக மற்றவைகளை கற்றுக்கொண்டுவிட முடியும். அதற்கு கீழ்காணும் தளங்கள் பயனுள்ளதாக இருக்கும்.

www.phptpoint.com/

phppot.com/

www.w3resource.com/

www.w3programmers.com/

www.w3schools.com/php/default.asp

php.net/

www.dollarfry.com/how-to-get-a-job-as-web-developer-by-learning-php/

கணிணி மாணவர்கள் எப்படி வேலை தேடலாம்?

www.kaniyam.com/how-to-get-a-computer-science-job/

இரா.கதிர்வேல்

சென்னையில் Python Developer *ஆக பணிபுரிந்து கொண்டிருக்கிறேன். சுதந்திர மென்பொருள்*(Freedom Software) *தத்துவத்தின் மீது அசைக்கமுடியாத நம்பிக்கையும், தீராத காதலும் கொண்டுள்ளேன்.*

அனைத்து தொழில்நுட்பங்களும் தமிழில் மொழிபெயர்க்கப்பட வேண்டும். ஆங்கிலம் என்ற ஒன்றைக் காட்டி மாணவர்களை ஆசிரியர்கள் பயமுறுத்திக்கொண்டிருக்கின்றனர். காரணம் பட்டம், பட்டயம், பொறியியல், மருத்துவம், அறிவியல் என அனைத்தையும் ஆங்கிலத்திலேயே நாம் அனைவரும் படித்துக்கொண்டிருக்கிறோம். தமிழ்வழிக் கல்வியில்

பயின்று, கிராமப் புறத்திலிருந்து வரும் மாணவர்கள் இதனால் அதிகம் பாதிக்கப்பட்டு வருகின்றனர். அவர்களுக்கு ஆங்கிலம் என்பது எல்லா வகையிலும் பிரச்சனையாக இருக்கிறது. ஆங்கிலத்தை விட்டு விட்டு அப்படியே தமிழில் படிக்க விரும்பினாலும் அதற்கான வழிகள் இன்னும் அமைக்கப்படவில்லை. சீனா, ஜப்பான் போன்ற நாடுகள் ஆராய்ச்சி படிப்பு முதற்கொண்டு அனைத்தையும் தங்கள் தாய்மொழியிலேயே மேற்கொள்கின்றனர். அவ்வாறு இந்தியாவில் படிக்கமுடியுமா? என்பதற்கான பதில் கேள்விக்குறியே.

ஏங்க தமிழ்ல படிச்சு என்னத்துக்குங்க ஆகப்போகுது? எல்லாமே ஆங்கிலத்திலேயேத்தானே இருக்கு. ஆங்கிலம் தெரிந்தால் எதையும் நான் எளிமையாக கற்றுக்கொள்ளமுடியுமே? என்ற கேள்வியை படித்தவர்கள் முதல் பாமரர் வரைக்கும் கேட்க ஆரம்பித்து விட்டனர். இதே கேள்வியை சீனர்களோ, ஜப்பானியர்களோ கேட்க மாட்டார்கள். காரணம் அவர்கள் தங்கள் மொழியை காக்க வேண்டும், அனைத்தும் நம் தாய்மொழியிலேயே கிடைக்க வேண்டும் என்ற அக்கறையுடன் உள்ளனர். ஆனால் நாம்? உலகில் எந்த மொழியும் தமிழ் மொழியுடன் போட்டி போட முடியாது. அந்தளவிற்கு சுயமாக தனித்து இயங்கக்கூடிய மொழி நம் தாய்மொழித் தமிழ் மொழி. தமிழ்வழியில் படிப்பதற்கான பிரச்சனை தமிழ் அல்ல. அறிவியல், தொழில்நுட்பம், மருத்துவம் போன்று எந்த

துறையை எடுத்துக்கொண்டாலும் எல்லாமே ஆங்கிலத்தில் கிடைக்கிறது. ஆனால் இந்த துறையிலுள்ளவைகள் தமிழில் இன்னும் மொழிபெயர்க்கப்படவில்லை, அக்கறையும் இன்னும் காட்டப்படவில்லை. கலை, அறிவியல், வணிகம், மருத்துவம், பொறியியல், ஆராய்ச்சி என எந்தவொரு தொழில்நுட்பத்தைக் கற்றவராயினும் ஒவ்வொருவரும் தாம் கற்ற விஷயங்களை அவசியம் தமிழில் பகிர்ந்து கொள்ள வேண்டும். அப்படிச் செய்யும் பட்சத்தில் எந்த தகவலையும் நாம் தமிழில் பெற முடியும் என்ற நம்பிக்கை தமிழ்ச் சமுகத்தில் ஏற்படும்.

அந்த வகையில் நான் படித்தது கணினி அறிவியல். என்னுடைய ஆர்வம் FOSS, GNU/Linux. இந்த துறைகளில் நான் கற்றவைகளை http://gnutamil.blogspot.in தளத்தில் தமிழில் பகிர்ந்து கொண்டு வருகிறேன். நாம் கற்றவைகளை அவசியம் தமிழில் பகிர்ந்து கொள்ள வேண்டும் என்ற நோக்கத்துடன் நான் தொடர்ந்து செயல்பட்டுக்கொண்டிருக்கிறேன். அனைத்தும் தமிழில் கிடைக்கவேண்டும் என்ற உயரிய நோக்கத்தோடு கணியம்(http://www.kaniyam.com/) செயல்பட்டுக்கொண்டிருக்கின்றது. அதில் என்னுடைய பங்களிப்பும் இருக்கிறது என்பதில் பெரும் மகிழ்ச்சி அடைகிறேன்.

மின்னஞ்சல் : linuxkathirvel.info@gmail.com

வலைப்பதிவு : http://gnutamil.blogspot.in/

25 கணியம் பற்றி

இலக்குகள்

- கட்டற்ற கணிநுட்பத்தின் எளிய விஷயங்கள் தொடங்கி அதிநுட்பமான அம்சங்கள் வரை அறிந்திட விழையும் எவருக்கும் தேவையான தகவல்களை தொடர்ச்சியாகத் தரும் தளமாய் உருபெறுவது.
- உரை, ஒலி, ஒளி என பல்லூடக வகைகளிலும் விவரங்களை தருவது.
- இத்துறையின் நிகழ்வுகளை எடுத்துரைப்பது.
- எவரும் பங்களிக்க ஏதுவாய் யாவருக்குமான நெறியில் விவரங்களை வழங்குவது.
- அச்சு வடிவிலும், புத்தகங்களாகவும், வட்டுக்களாகவும் விவரங்களை வெளியிடுவது.

- *விருப்பமுள்ள எவரும் பங்களிக்கலாம்.*

- *கட்டற்ற கணிநுட்பம் சார்ந்த விஷயமாக இருத்தல் வேண்டும்.*

- *பங்களிக்கத் தொடங்கும் முன்னர் கணியத்திற்கு உங்களுடைய பதிப்புரிமத்தை அளிக்க எதிர்பார்க்கப்படுகிறீர்கள்.*

- *editor@kaniyam.com முகவரிக்கு கீழ்க்கண்ட விவரங்களடங்கிய மடலொன்றை உறுதிமொழியாய் அளித்துவிட்டு யாரும் பங்களிக்கத் தொடங்கலாம்.*

    - ***மடலின் பொருள்:*** *பதிப்புரிமம் அளிப்பு*

    - ***மடல் உள்ளடக்கம்***

        - *என்னால் கணியத்திற்காக*

அனுப்பப்படும் படைப்புகள் அனைத்தும் கணியத்திற்காக முதன்முதலாய் படைக்கப்பட்டதாக உறுதியளிக்கிறேன்.

- இதன்பொருட்டு எனக்கிருக்கக்கூடிய பதிப்புரிமத்தினை கணியத்திற்கு வழங்குகிறேன்.

- உங்களுடைய முழுப்பெயர், தேதி.

- தாங்கள் பங்களிக்க விரும்பும் ஒரு பகுதியில் வேறொருவர் ஏற்கனவே பங்களித்து வருகிறார் எனின் அவருடன் இணைந்து பணியாற்ற முனையவும்.

- கட்டுரைகள் மொழிபெயர்ப்புகளாகவும், விஷயமறிந்த ஒருவர் சொல்லக் கேட்டு கற்று இயற்றப்பட்டவையாகவும் இருக்கலாம்.

- படைப்புகள் தொடர்களாகவும் இருக்கலாம்.

- தொழில் நுட்பம், கொள்கை விளக்கம், பிரச்சாரம், கதை, கேலிச்சித்திரம், நையாண்டி எனப் பலசுவைகளிலும் இத்துறைக்கு பொருந்தும்படியான ஆக்கங்களாக இருக்கலாம்.

- தங்களுக்கு இயல்பான எந்தவொரு நடையிலும் எழுதலாம்.

- தங்களது படைப்புகளை எளியதொரு உரை ஆவணமாக editor@kaniyam.com முகவரிக்குஅனுப்பிவைக்கவும்.

- தள பராமரிப்பு, ஆதரவளித்தல் உள்ளிட்ட ஏனைய விதங்களிலும் பங்களிக்கலாம்.

- ஐயங்களிருப்பின் editor@kaniyam.com மடலியற்றவும்.

- *கணித் தொழில்நுட்பத்தை அறிய விழையும் மக்களுக்காக மேற்கொள்ளப்படும் முயற்சியாகும் இது.*

- *இதில் பங்களிக்க தாங்கள் அதிநுட்ப ஆற்றல் வாய்ந்தவராக இருக்க வேண்டும் என்ற கட்டாயமில்லை.*

- *தங்களுக்கு தெரிந்த விஷயத்தை இயன்ற எளிய முறையில் எடுத்துரைக்க ஆர்வம் இருந்தால் போதும்.*

- *இதன் வளர்ச்சி நம் ஒவ்வொருவரின் கையிலுமே உள்ளது.*

- *குறைகளிலிருப்பின் முறையாக தெரியப்படுத்தி முன்னேற்றத்திற்கு வழி வகுக்கவும்.*

வெளியீட்டு விவரம்

பதிப்புரிமம் © 2013 கணியம்.
கணியத்தில் வெளியிடப்படும் கட்டுரைகள் http://creativecommons.org/licenses/by-sa/3.0/ பக்கத்தில் உள்ள கிரியேடிவ் காமன்ஸ் நெறிகளையொத்து வழங்கப்படுகின்றன.

இதன்படி,
கணியத்தில் வெளிவரும் கட்டுரைகளை கணியத்திற்கும் படைத்த எழுத்தாளருக்கும் உரிய சான்றளித்து, நகலெடுக்க, விநியோகிக்க, பறைசாற்ற, ஏற்றபடி அமைத்துக் கொள்ள, தொழில் நோக்கில் பயன்படுத்த அனுமதி வழங்கப்படுகிறது.

ஆசிரியர்: த. சீனிவாசன் – editor@kaniyam.com +91 98417 95468

கட்டுரைகளில் வெளிப்படுத்தப்படும் கருத்துக்கள் கட்டுரையாசிரியருக்கே உரியன.

Creative Commons உரிமையில், யாவரும் இலவசமாகப் பகிரும் வகையில் தமது நூல்களை வெளியிடும் எழுத்தாளரை உங்கள் நன்கொடைகள் ஊக்குவிக்கும்.

வங்கி விவரங்கள்.

பெயர் : KATHIRVEL R
வங்கி : Karur Vysya Bank
கிளை : CHENNAI-MADIPAKKAM

கணக்கு எண் : 1614155000037991

IFSC எண் : KVBL0001614