

PHP

অটোমেটিক স্ক্রলের মাধ্যমে ই-বুক পড়া / রিডের জন্যঃ

আপনার ই-বুক বা pdf রিডারের Menu Bar এর **View** অপশনটি তে ক্লিক করে Auto /Automatically Scroll অপশনটি সিলেক্ট করুন (অথবা সরাসরি যেতে => **Ctrl + Shift + H**)। এবার **↑ up Arrow** বা **↓ down Arrow** তে ক্লিক করে আপনার পড়ার সুবিধা অনুসারে স্ক্রল স্পীড ঠিক করে নিন।

পি এইচ পি

PHP এর আগে সার্ভার সাইড স্ক্রিপ্টিং সম্পর্কে একটু জানুন: । আসলে স্ক্রিপ্টিং হচ্ছে প্রোগ্রামের আরেকটা সমার্থক শব্দ। এটা হচ্ছে কিছু instruction এর সেট যেটা run করলে স্বয়ংক্রিয় ভাবে কিছু কাজ হয়। “সার্ভার সাইড” বলতে বুঝানো হচ্ছে এই স্ক্রিপ্ট গুলোকে ইউজারের কম্পিউটার থেকে নিয়ন্ত্রনের বদলে সার্ভার থেকে handle করা। যখন কেউ PHP ওয়েব পেজ ভিজিট করবে তখন ওয়েব সার্ভার PHP কোডগুলিকে কিছু Process করবে যেমন:যেটা দেখানো দরকার (Picture,Content etc) সেটা দেখাবে আর যেটাকে লুকানো দরকার(math calculation,file operation etc)তা লুকাবে এবং শেষে HTML এ রূপান্তর করে ইউজারের ওয়েব ব্রাউজারে পাঠাবে।

PHP কি?:

PHP(Hypertext Preprocessor) একটা সার্ভার সাইড,ক্রস প্ল্যাটফর্ম,HTML-embedded স্ক্রিপ্টিং ল্যাংগুয়েজ। PHP এর বেশিরভাগ syntax গুলো C,Perl,Java থেকে ধার করা। এই ল্যাংগুয়েজটির উদ্দেশ্য হল ওয়েব পেজ দ্রুত তৈরী করা ডাইনামিকালি।

PHP শেখার আগে কি জানা থাকা দরকার:

১.HTML .বিশেষ করে HTML Form.

২.কোনো প্রোগ্রামিং ল্যাংগুয়েজ????:না। তবে যদি কোনো ল্যাংগুয়েজ যেমন C ইত্যাদি জানা থাকে তাহলে সুবিধা আছে।

কি কি সফটওয়্যার প্রয়োজন ?:

যেসব ওয়েব সার্ভার PHP সাপোর্ট করে সেখানে hosting(জায়গা) নিতে হবে। এ জন্য টাকা গুনতে হবে। এ বিষয়টি নিয়ে পরে বিস্তারিত আলোচনা করা হবে।

আপাতত শেখার জন্য আমরা নিজের কম্পিউটারেই ওয়েব সার্ভার ইনস্টল করে নেব। এ জন্য নিম্নোক্ত সফটওয়্যারগুলি ইনস্টল থাকতে হবে আপনার কম্পিউটারে-

*সার্ভার সফটওয়্যার

১.একটা PHP -compatible ওয়েব সার্ভার যেমন: apache

২.PHP

*ক্লাইন্ট সফটওয়্যার

১.ওয়েব ব্রাউজার যেমন মজিলা ফায়ারফক্স(এটাতো সবার ইনস্টল দেয়াই আছে)

২.একটা টেক্সট এডিটর যেমন:নোটপ্যাড।আপনি PHP এর জন্য Specialized এডিটরও ব্যবহার করতে পারেন।পরে বিস্তারিত আসছি।

ক্লাইন্ট সফটওয়্যারগুলিতো সবার আছেই এখন শুধু ইনস্টল দিতে হবে ওয়েব সার্ভার যেমন apache এবং অবশ্যই PHP আর একটা ডেটাবেস সফটওয়্যারও ইনস্টল দিয়ে নিন যেমন:MySQL,কি কাজে লাগবে তা পরে বলছি।এ সফটওয়্যারগুলি সব ফ্রি পাওয়া যায়।চালাকরা নিশ্চয় এতক্ষনে গুগলে সার্চ দিয়ে সফটওয়্যারগুলি বের করে ফেলেছেন।আর খোজার দরকার নেই এবার একটা এমন সফটওয়্যার এর কথা বলব যেটা ইনস্টল দিলে সবগুলি একবারেই ইনস্টল হয়ে যাবে।আর আলাদা আলাদা ভাবে ইনস্টল দিতে হবেনা।সফটওয়্যারটি হচ্ছে XAMPP. এই যে এখান থেকে ইনস্টল করুন

<http://www.apachefriends.org/en/xampp.html>. .exe

ফাইলটি ডাউনলোড করে অন্যান্য সফটওয়্যার এর মত ইনস্টল দিন।ওকে এখন যাই,ডাউনলোড+ইনস্টল শেষ করেন,আবার আসছি।

ইনস্টলেশন পর্ব

আমরা যারা মাইক্রোসফট উইন্ডোজে ডেভেলপমেন্ট করতে আগ্রহী তাদের বেশির ভাগই WAMP বা XAMPP ব্যবহার করি। সাধারণ ভাবে এগুলো ব্যবহার করলেই যথেষ্ট কিন্তু আপনাকে যদি কোথাও নিজের মত করে সার্ভারগুলোকে সেটাপ করতে বলা হয় তাহলে কী করবেন? সুতরাং নিজের মত করে AMP (Apache, MySQL, PHP) সেটাপ ও কনফিগার করতে জানাটা জরুরী। তাহলে শুরু করা যাক-

পুরো বিষয়টাকে আমরা তিনটি ভাগে ভাগ করে নিচ্ছি যাতে বুঝতে সুবিধা হয়।

ক) ওয়েব সার্ভার সেটাপ

খ) পিএইচপি সেটাপ

গ) ডাটাবেস সার্ভার সেটাপ

ক) ওয়েব সার্ভার সেটাপ

পিএইচপি চালানোর জন্য প্রথমেই আমাদের একটি ওয়েব সার্ভার সেটাপ করে নিতে হবে। এই ক্ষেত্রে আমরা অ্যাপাচি ওয়েব সার্ভার ব্যবহার করবো।

১) অ্যাপাচি ওয়েব সার্ভার সেটাপ করার জন্য আমাদেরকে প্রথমে এটি সংগ্রহ করে নিতে হবে। অনলাইনে এটি ফ্রি ডাউনলোড করতে পারবেন এই লিঙ্কে- <http://www.apache.org/dist/httpd/binaries/win32/> . এখান থেকে MSI Installer Package টি (যেমন, win32-x86-no_ssl.msi) ফাইলটি নামিয়ে নিতে হবে।

২) এবার সাধারণ সফটওয়্যার সেটাপ করার মত করে এটিকে সেটাপ করে নিতে হবে। সেটাপ উইজার্ড আপনার কাছে কিছু তথ্য জানতে চাইবে। নিচের মত করে তথ্যগুলো পূরণ করে নিন-

Server Information

Network Domain: localhost

Server Name: localhost

Admin Email: (your any email)





এবার, For All Users, on Port 80, as a Service সিলেক্ট করে Next এ ক্লিক করুন।

৩) পরবর্তী ধাপগুলোতে Next করে এগিয়ে যান।

8) Destination Folder হিসাবে যদি C:\ ড্রাইভ ব্যবহার করে থাকেন তাহলে আপনার অ্যাপাচি সার্ভার লোকেশন হবে C:/Apache2.2

৫) Finish চেপে ইনস্টলেশন শেষ করুন। লক্ষ্য করুন সিস্টেম ট্রে-তে আপনার সার্ভার আইকন যুক্ত হয়ে গিয়েছে।

- আইকন  মানে হচ্ছে সার্ভার চলছে।

- আইকন  মানে সার্ভার বন্ধ আছে।

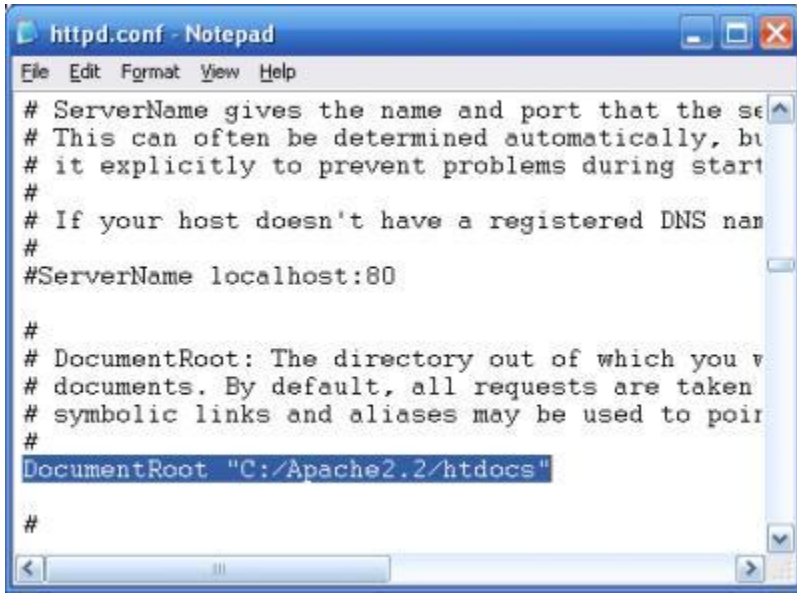
এই আইকনে ক্লিক করে সার্ভারটি চালু বা বন্ধ করা যাবে।

সার্ভার যথাযথ ভাবে কাজ করছে কিনা যাচাই করতে চাইলে যেকোন ব্রাউজার খুলে এড্রেসবারে <http://127.0.0.1/> বা <http://localhost/> লিখে এন্টার করুন। সবকিছু ঠিকঠাক থাকলে ব্রাউজারে 'It works!' লেখা একটি পেজ দেখাবে।

ডকুমেন্ট রুট(DocumentRoot):

ওয়েব সার্ভার সেটাপ সম্পন্ন হওয়ার পর তাতে সাইট রাখার জন্য যে ফোল্ডারটি ব্যবহৃত হয় তাকে বলে DocumentRoot।

সার্ভারটি C:/Apache2.2 ইনস্টল করা হলে বাই ডিফল্ট DocumentRoot হবে C:/Apache2.2/htdocs। আপনি চাইলে httpd.conf ফাইল থেকে পরিবর্তন করে যেকোন ড্রাইভ বা লোকেশনে এই DocumentRoot রাখতে পারবেন। এরজন্য স্টার্ট মেনুতে গিয়ে "Apache HTTP Server 2.2 > Configure Apache Server > httpd.conf" খুলে DocumentRoot পরিবর্তন করে নিন।



পরের ধাপে আমরা PHP সেটাপ ও কনফিগার করা শিখবো।

ইনস্টলেশন পর্ব-২

প্রথম ধাপে অ্যাপাচি ওয়েব সার্ভার সেটাপ করার পর আজকে আমরা PHP সেটাপ ও কনফিগার করা শিখতে যাচ্ছি। পূর্ব প্রস্তুতি হিসেবে <http://www.php.net/downloads.php> থেকে Windows Binaries এর zip package টা ডাউনলোড করে ফেলুন। তারপর নিচের সহজ ধাপগুলো অনুসরণ করুন-

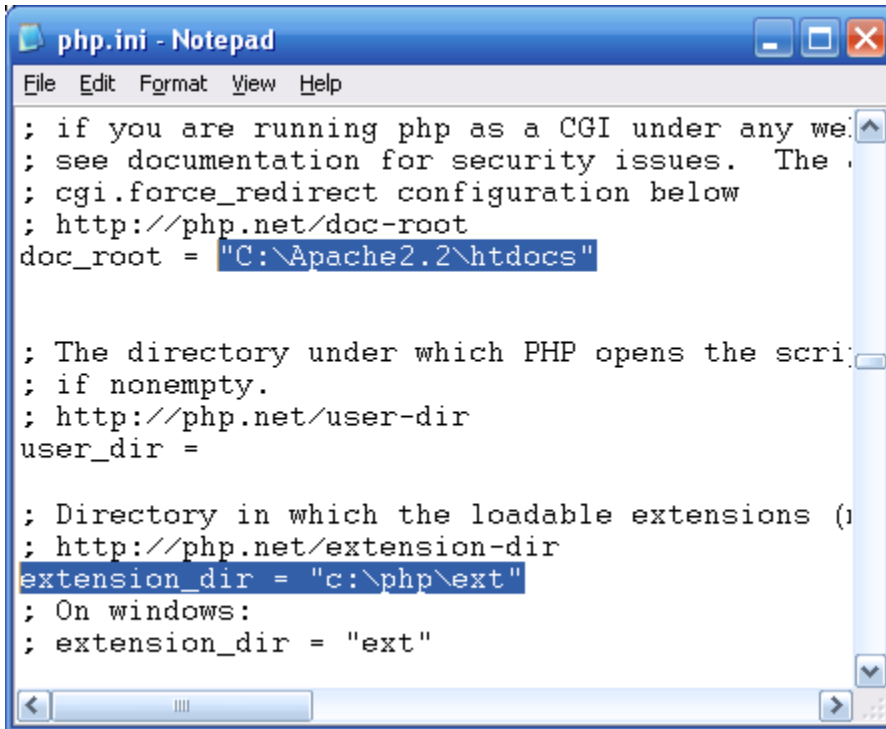
১) zip package টা C:\ ড্রাইভে আনজিপ করুন। তাহলে আমাদের php ফোল্ডারের লোকেশন হবে- C:\php\

২) C:\php\ ফোল্ডারে ঢুকে php.ini-dist ফাইলটির নাম পরিবর্তন করে php.ini করুন। PHP কনফিগারেশন পরিবর্তন করতে এই ফাইলটি এডিট করতে হবে।

৩) যেকোন ভাল টেক্সট এডিটর(যেমন Notepad বা [Notepad++](#)) দিয়ে php.ini ফাইলটি খুলে নিচের মত পরিবর্তন কর নিন-
doc_root = "C:\Apache2.2\htdocs" (উল্লেখ্য আমরা অ্যাপাচির কনফিগ ফাইলে DocumentRoot হিসাবে যা সেট

করেছিলাম এখানে doc_root হিসাবে তাই করতে হবে)

extension_dir = "c:\php\ext" (আপনার পিএইচপি ফোল্ডার লোকেশন ভিন্ন হলে এটিও পরিবর্তন করতে হবে)



```
php.ini - Notepad
File Edit Format View Help
; if you are running php as a CGI under any we
; see documentation for security issues. The
; cgi.force_redirect configuration below
; http://php.net/doc-root
doc_root = "C:\Apache2.2\htdocs"

; The directory under which PHP opens the scri
; if nonempty.
; http://php.net/user-dir
user_dir =

; Directory in which the loadable extensions (
; http://php.net/extension-dir
extension_dir = "c:\php\ext"
; On windows:
; extension_dir = "ext"
```

ব্যাস, আমাদের PHP কনফিগার করা আপাতত শেষ। এবার ওয়েব সার্ভারে কিছু পরিবর্তন করতে হবে। নিচে সহজ ধাপে তা বর্ণনা করা হলো-

১) স্টার্ট মেনুতে গিয়ে "Apache HTTP Server 2.2 > Configure Apache Server > httpd.conf" খুলতে হবে। এখানেই আমরা সার্ভার সম্পর্কিত সকল বিষয়াদী পরিবর্তন করতে পারবো। এটা নিশ্চয় বুঝতে পেরেছেন যে ড্রাইভে ফোল্ডারে গেলেই আমরা httpd.conf ফাইলটি পেয়ে যাবো। সুতরাং স্টার্ট মেনুতে না থাকলেও যেকোন সময় যেকোন টেক্সট এডিটর দিয়ে এটি খুলে কাজ করা যাবে।

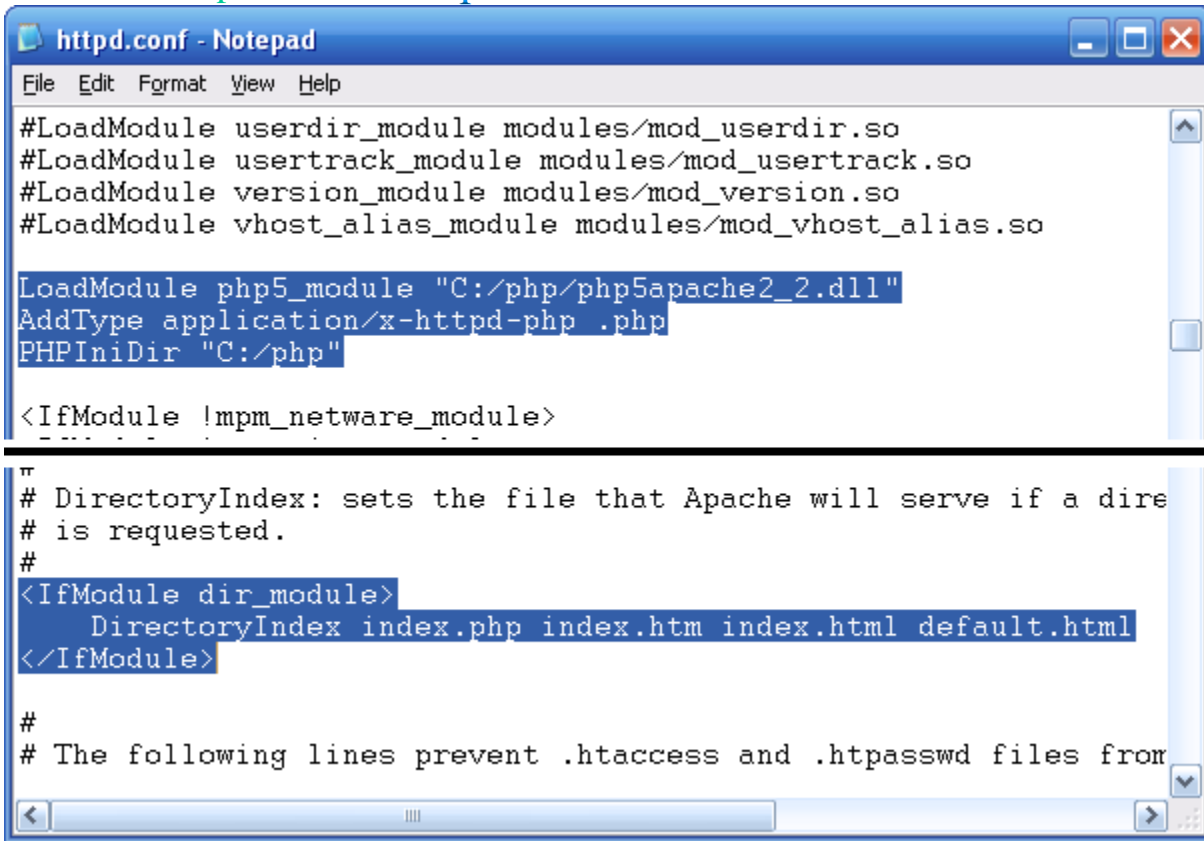
২) httpd.conf ফাইলে নিচের লাইনগুলো যুক্ত করে দিন-

1 LoadModule php5_module "C:/php/php5apache2_2.dll"

2 AddType application/x-httpd-php .php

3 PHPIniDir "C:/php"

লক্ষ্য রাখতে হবে যেন php5apache2_2.dll ফাইলটি আপনার php ফোল্ডারে থাকে।



```
httpd.conf - Notepad
File Edit Format View Help
#LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so

LoadModule php5_module "C:/php/php5apache2_2.dll"
AddType application/x-httpd-php .php
PHPIniDir "C:/php"

<IfModule !mpm_netware_module>
#
# DirectoryIndex: sets the file that Apache will serve if a dire
# is requested.
#
#
<IfModule dir_module>
DirectoryIndex index.php index.htm index.html default.html
</IfModule>
#
# The following lines prevent .htaccess and .htpasswd files from
```

৩) অ্যাপাচি ওয়েব সার্ভার সাধারণত index.htm ফাইলটি প্রথমে চালু করে থাকে। এটি পরিবর্তন করে index.php করতে চাইলে httpd.conf ফাইলে DirectoryIndex অংশটি খুঁজে নিচের মত পরিবর্তন করে নিন-

```
DirectoryIndex index.php index.htm index.html default.html
```

৪) httpd.conf ফাইলরে পরিবর্তনটুকু Save করে অ্যাপাচি সার্ভারটি রিস্টার্ট করুন। কোন রকম সমস্যা ব্যতিত সার্ভার পুনরায় চালু হলে বুঝতে পারবেন সবকিছু ঠিকঠাক মতই হয়েছে।

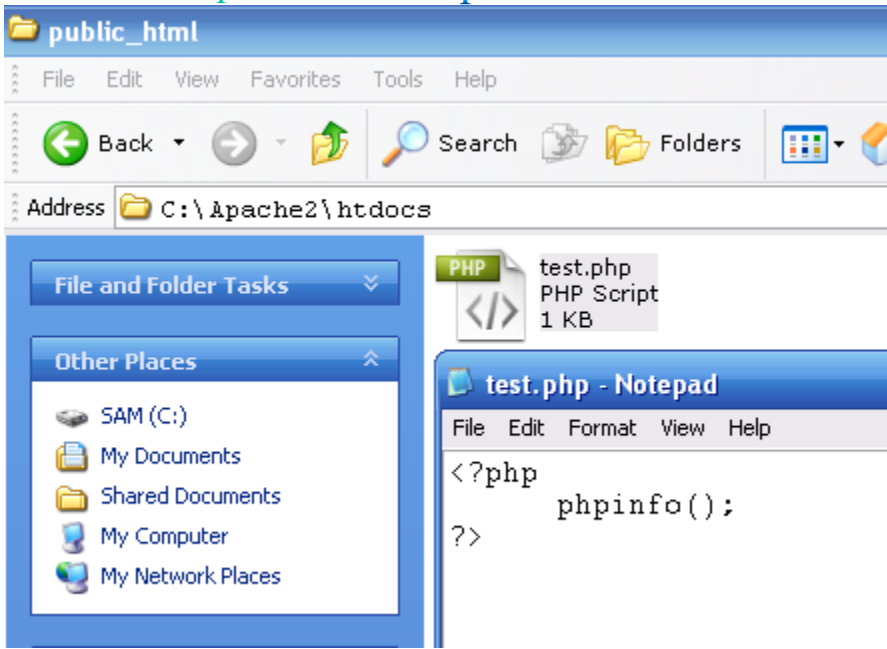
PHP ঠিকঠাক মত সংযুক্ত হলো কিনা পরিষ্কা করে দেখতে চলুন একটু কোড করে ফেলি। নবীনদের জন্য PHP কোডিং এর বেসিক কিছু ধারণাও এতে হয়ে যাবে-

১) DocumentRoot (C:/Apache2.2/htdocs) এ গিয়ে যেকোন টেক্সট এডিটর দিয়ে test.php নামে একটি ফাইল তৈরি করে তাতে নিচের কোডটুকু লিখে সেভ করুন-

```
1 <?php
```

```
2 phpinfo();
```

```
3 ?>
```



২) যেকোন ব্রাউজারে <http://localhost/test.php> খুলে নিচের মত একটি পেজ দেখতে পেলেন বুঝতে পারবেন সবকিছু ঠিকঠাকমতই হয়েছে-



System	Windows NT TNET 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
Build Date	Jun 29 2009 21:52:31
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86
Configure Command	cscrip/nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--with-pdo-oci=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8=D:\php-sdk\oracle\instantclient10\sdk,shared" "--with-oci8-11g=D:\php-sdk\oracle\instantclient11\sdk,shared" "--with-enchant=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\php\php.ini

এই পাতায় আপনার সার্ভার সম্পর্কিত অনেক গুরুত্বপূর্ণ তথ্য পেয়ে যাবেন।

পরবর্তী ধাপে আমরা ডাটাবেস সার্ভার (MySQL) সেটাপ ও কনফিগার করা শিখবো।

ইনস্টলেশন পর্ব-৩



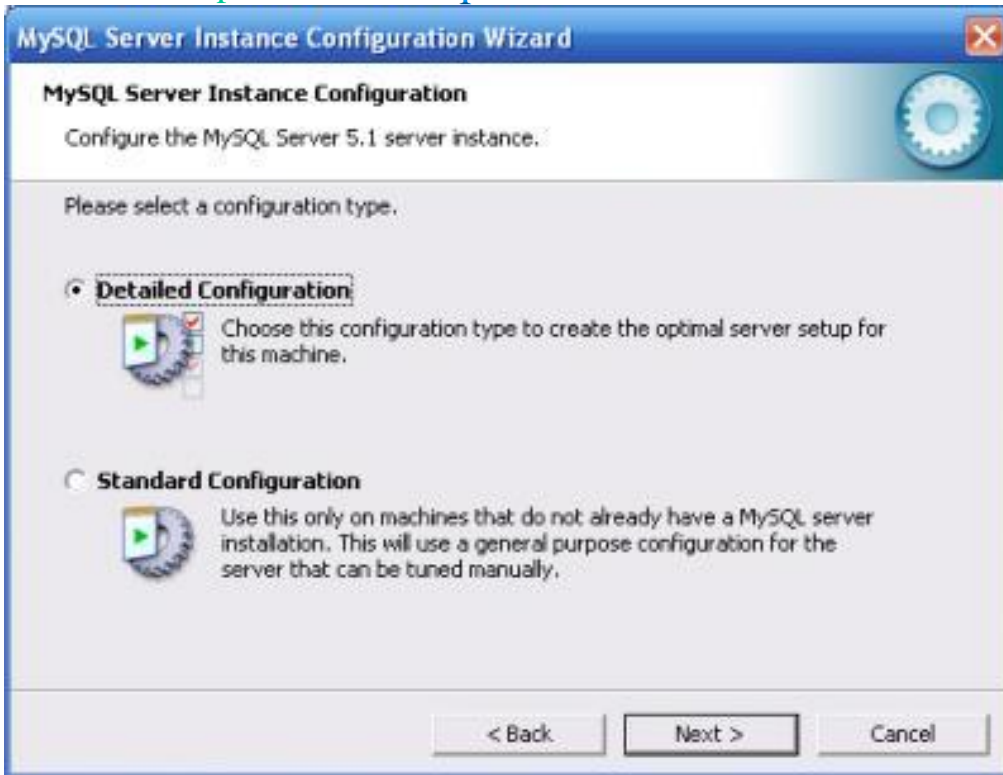
ওয়েব সার্ভার সেটআপ করার পর আমরা PHP সেটআপ ও কনফিগার করা শিখেছি। এবার ডাটাবেস সেটআপ করে নিলেই কাজ করার জন্য মোটামুটি আমরা প্রস্তুত হয়ে যাবো। Apache ও PHP এর সাথে ডাটাবেস হিসেবে ব্যবহার করবো MySQL. ডাটাবেস সম্পর্কিত সাধারণ আলোচনা পরবর্তীতে করা যাবে আপাতত সেটআপ শুরু করা যাক-

১) প্রথমেই অনলাইন থেকে MySQL এর ফ্রি [ইনস্টলারটি ডাউনলোড](#) করে সাধারণ সফটওয়্যারের মত সেটআপ করে নিতে হবে।

২) সেটআপের শেষ ধাপে ইনস্টলেশন উইজার্ডে Configure the MySQL Server now চেক করে দিলে MySQL কনফিগারেশন উইজার্ড শুরু হবে। অথবা স্টার্ট মেনু হতে MySQL > MySQL Server Instance Config Wizard এ ক্লিক করে কনফিগারিং শুরু করা যেতে পারে।



৩) কনফিগার টাইপ Detailed Configuration করে Next করুন।



- ৪) সার্ভার টাইপ এ Developer Machine নির্বাচন করে Next।
- ৫) ডেটাবেস ইউজেস Multifunctional Database নির্বাচন করুন।
- ৬) InnoDB Tablespace Settings এ \Data\ নির্বাচন করুন।
- ৭) নাম্বার অব কনকারেন্ট কানেকশনে রাখুন Decision Support (DSS)/OLAP।
- ৮) এরপরের ধাপে Enable TCP/IP Networking চেক করতে হবে যার পোর্ট হবে 3306।

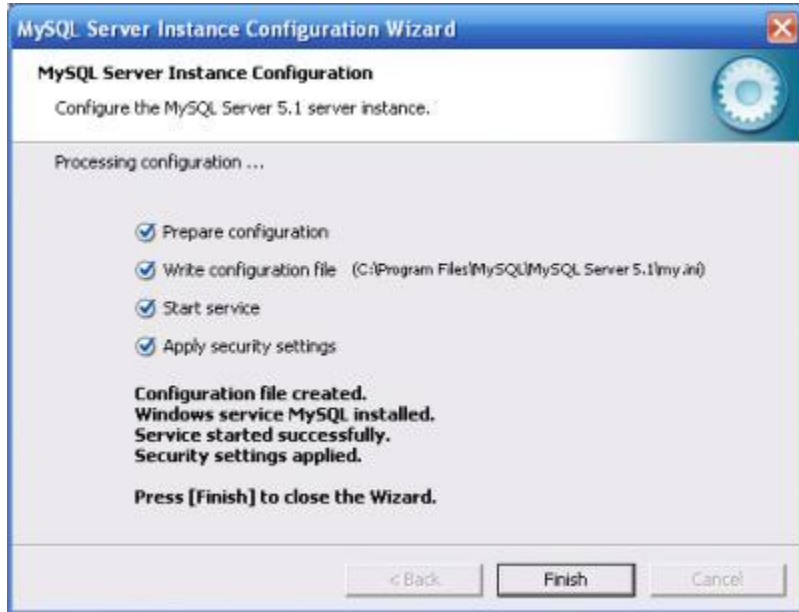


- ৯) ক্যারেক্টার সেটে Standard Character Set নির্বাচন করুন।
- ১০) উইন্ডোজ অপশনে Install As Windows Service চেক করলে MySQL সার্ভিস হিসেবে ইনস্টল হবে। এবং Launch the MySQL Server automatically চেক করা থাকলে প্রতিবার উইন্ডোজ ওপেন হওয়ার সময় নিজে থেকে MySQL চালু হয়ে

যাবে। দু'টো অপশনই চেক করে Next করতে পারেন।

১১) এরপর সিকিউরিটি অপশনে অবশ্যই রুট পাসওয়ার্ড নির্ধারন করবেন। সার্ভারের নিরাপত্তার জন্য Enable root access from remote machines আনচেক থাকা ভাল। তবে অন্য মেশিন থেকে সরাসরি MySQL এডমিন একসেস করতে চাইলে চেক করে Next করুন।

১২) সব শেষে Execute বাটনে ক্লিক করলে কনফিগার হয়ে যাবে। সব ঠিকঠাক মত হলে নিচের মত দেখাবে।



এটুকু হয়ে যাওয়ার পর PHP5 কে আমাদের MySQL এর সাথে সমন্বয় করতে আরো কিছু কাজ বাকী। এরজন্য PHP যেখানে সেটাপ করা হয়েছে ওখান থেকে libmysql.dll বা php_mysql.dll ফাইলটিকে উইন্ডোজে(Windows\System32\) কপি করতে হবে। সংক্ষেপে-

C:\php\ext\php_mysql.dll কপি হবে C:\Windows\System32\php_mysql.dll এ।

অতঃপর php ফোল্ডারে ঢুকে (C:\php\) php.ini ফাইলটি খুলে ;**extension=php_mysql.dll** লাইনটি এডিট করে **extension=php_mysql.dll** করুন। মানে লাইনের প্রথমে থাকা সিমিকোলনটি (;) সরিয়ে দিতে হবে।

সব ধাপ মোটামুটি শেষ। এবার আমরা আমাদের ওয়েব সার্ভারটি (Apache) রিস্টার্ট করে কাজ শেষ করবো।

চলুন এবার যাচাই করি MySQL ঠিক মত চলছে কিনা। এর জন্য নিচের ধাপগুলো অনুসরণ করুন-

১) স্টার্ট মেনু থেকে MySQL > MySQL Command Line Client-টি খুলে কনফিগার করার সময় যে root পাসওয়ার্ডটা ব্যবহার করা হয়েছিলো সেটি টাইপ করে লগইন করুন।

২) কনসোলো Status কমান্ড দিলে MySQL সম্পর্কিত কিছু তথ্য দেখাবে যেখানে সার্ভার Uptime: দেখে বুঝতে পারবেন সার্ভারটি

ঠিক ভাবে চলছে কিনা।

```
MySQL Command Line Client
mysql> status
-----
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
for Win32 (ia32)

Connection id:          3
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:       ;
Server version:         5.1.36-community MySQL Communi
Protocol version:       10
Connection:             localhost via TCP/IP
Server characteraset:   latin1
Db characteraset:       latin1
Client characteraset:   latin1
Conn. characteraset:    latin1
TCP port:               3306
Uptime:                 1 hour 44 min 19 sec

Threads: 1  Questions: 21  Slow queries: 0  Opens: 15
les: 0  Queries per second avg: 0.3

mysql>
```

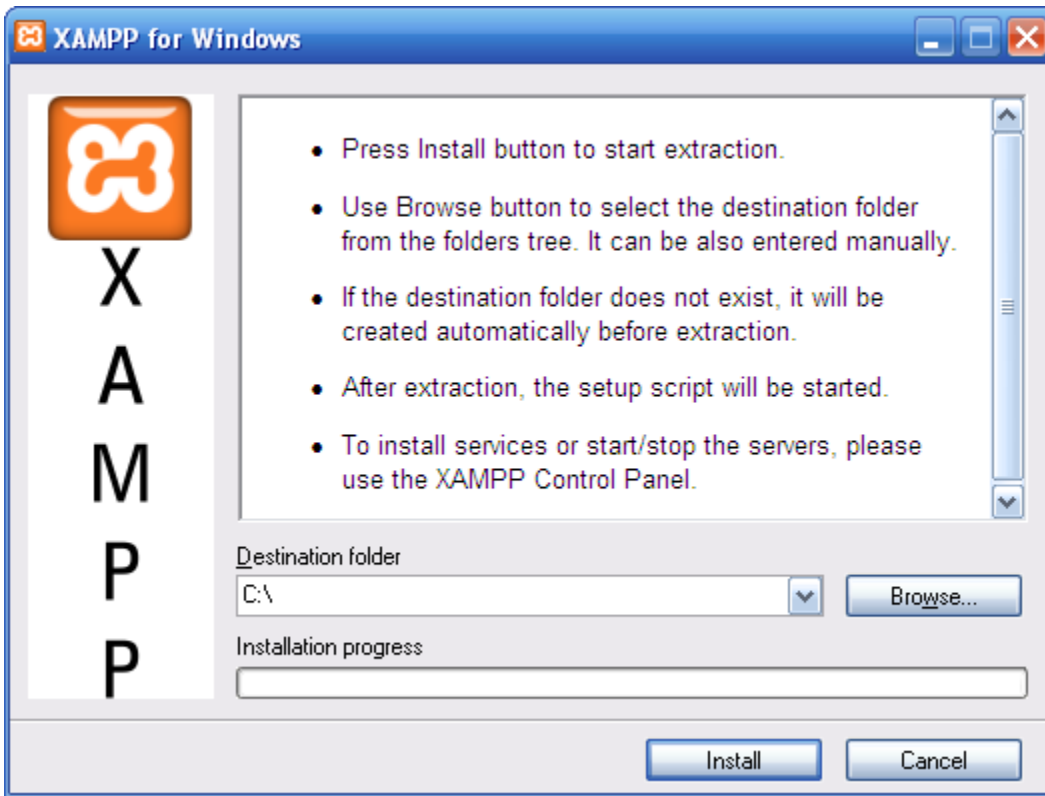
উইন্ডোজ পরিবেশে আমাদের কাজের পরিবেশ মোটামুটি প্রস্তুত এখন। এরপর আরো এডভান্সড ও বিভিন্ন কার্যকরী টুল নিয়ে আলোচনা করা যেতে পারে। তবে আপাতত আমরা কোড করার জন্য প্রস্তুত। সুতরাং, কোডিং শুরু করা যেতে পারে-

ইনস্টলেশন শর্টকাট ও শুরু (উইন্ডোজ)

উইন্ডোজ পরিবেশে ম্যানুয়ালি সবকিছু সেটাপ শিখতে গিয়ে যেসকল নবীন ডেভেলপাররা হোচট খেয়েছেন তাদের জন্য খুব আনন্দের খবর হচ্ছে কোন রকম ঝামেলা ছাড়াই উইন্ডোজ পরিবেশে ওয়েব ডেভেলপ করার পরিবেশ তৈরি করা সম্ভব। এর জন্য খুব জনপ্রিয় দু'টি প্যাকেজ (XAMPP, WAMP) আছে যাতে Apache, PHP, MySQL সহ আরো বেশ কিছু প্রয়োজনীয় টুল একত্রে কনফিগার করে দেয়া থাকে। এদের যেকোন একটি সংগ্রহ করে সাধারণ যেকোন সফটওয়্যারের মত সেটাপ করে নিলেই ওয়েব ডেভেলপ করার প্রয়োজনীয় পরিবেশ তৈরি হয়ে যাবে। XAMPP এবং WAMP নামের এই প্যাকেগুলো থেকে আমরা অধিক জনপ্রিয় XAMPP-কে বেছে নিচ্ছি। চলুন খুব সংক্ষেপে এটি সংগ্রহ, ইনস্টলেশন ও সাধারণ ব্যবহারবিধি জেনে নেই-

১) প্রথমেই আপনাকে XAMPP প্যাকেজটি ডাউনলোড করে নিতে হবে। যারা গুগল ব্যবহার জানেন তারা এতক্ষনে নিশ্চয় DOWNLOAD XAMPP লিখে সার্চ দিয়ে যায়গামত পৌঁছে গিয়েছেন। যারা এখনো পাচ্ছেন না তারা apachefriends.org খুলে [Windows](#) প্যাকেজটি খুঁজে বের করে [ডাউনলোড](#) দিন।

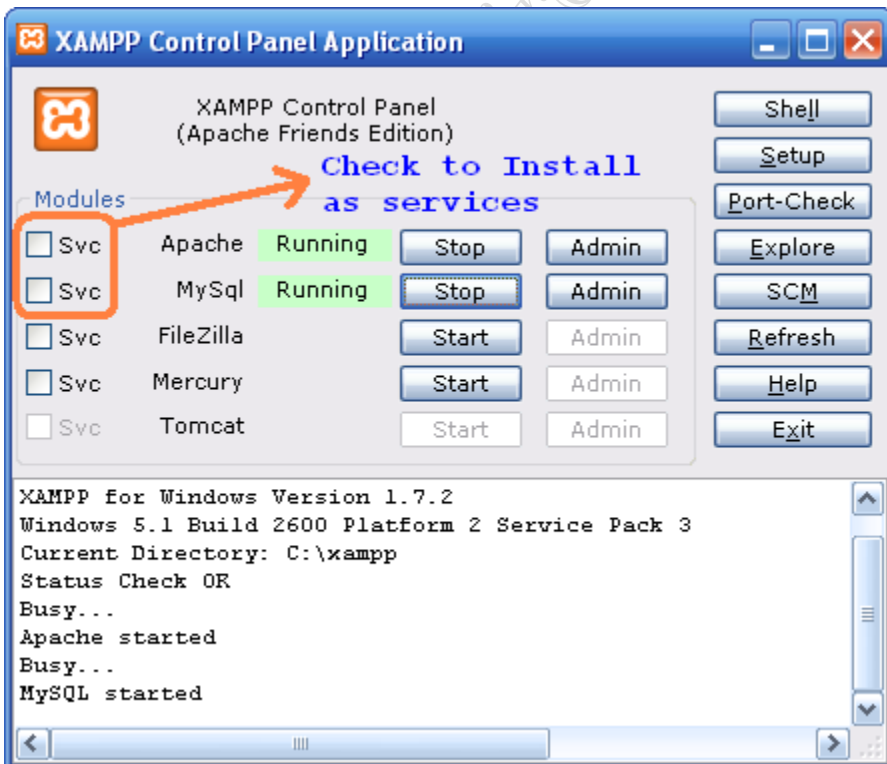
২) ডাউনলোড হয়ে যাওয়ার পর অন্য যেকোন সফটওয়্যারের মত করে সেটাপ শুরু করুন। এক্ষেত্রে Destination Folder এ C:/ বা অন্য যেকোন ড্রাইন নির্বাচন করতে পারেন।



৩) ইনস্টলেশনের শেষ পর্যায়ে কিছু ডেস কনসোল আসবে যেখানে বিভিন্ন সেটিংস নির্বাচন করতে বলবে। এন্টার করে যেতে থাকুন (বাই ডিফল্ট যা থাকে সেভাবে সব হয়ে যাবে)।

৪) সবশেষে XAMPP কন্ট্রোল প্যানেলটি খোলার অপশন পাবেন (বা স্টার্ট মেনু থেকে খুলে নিন)।

৫) XAMPP Control Panel থেকে সরাসরি Apache, MySQL স্টার্ট বা স্টপ করতে পারবেন। বাঁদিকে Svc লেখা চেকবক্সগুলোতে টিক দিয়ে দিলে সার্ভিস হিসেবে সেটাপ করে রাখতে পারেন যাতে প্রতিবার উইন্ডোজ চালু হওয়ার সাথে সাথে নিজে থেকেই এগুলো চলতে শুরু করে।



অনেক সময় স্টার্ট দিতে গেলে আপনার স্টার্ট নাও নিতে পারে। সাধারণত Apache'র জন্য প্রয়োজনীয় পোর্টগুলো অন্য কেউ দখল করে রাখলে এটি বেশি ঘটে থাকে। যেমন IIS বা অন্য কোন সফটওয়্যার যা পোর্ট ৮০ বা ৮০৮০ ব্যবহার করে। Skype যারা ব্যবহার করে থাকেন তাদের Apache কোনভাবেই চালু হবে না যতক্ষন না Skype'র সেটিংস এ ঢুকে পোর্ট ৮০ ব্যবহার করতে বারণ করছেন।

৬) সবকিছু ঠিকঠাক মত হলে ব্রাউজার খুলে এড্রেসবারে <http://localhost> বা <http://127.0.0.1> টাইপ করে এন্টার করুন, XAMPP এর ডিফল্ট হোমপেজ খুলে যাবে যেখানে XAMPP এর সাথে দেয়া প্রয়োজনীয় কিছু টুলের ওয়েব ইন্টারফেস পেয়ে যাবেন (যেমন MySQL ব্যবস্থাপনা টুল PHPMyAdmin)।

ব্যাস, আমাদের উইন্ডোজ পিসি এবার Apache, PHP, MySQL সহ প্রয়োজনীয় বেশ কিছু বস্তু সহ ওয়েব ডেভেলপমেন্ট করার জন্য প্রস্তুত।

XAMPP for Windows

English /



The screenshot shows the XAMPP 1.7.2.1 installation success page. The left sidebar contains navigation links: XAMPP [PHP: 5.3.0], Welcome, Status, Security, Documentation, Components, phpinfo(), pearinfo(), perlinfo(), aspinfo(), Demos (CD Collection, Biorhythm, Instant Art, Flash Art, Phone Book, ADOdb, Guest Book), and Tools (phpMyAdmin, Webalizer, Mercury Mail, FileZilla FTP). The main content area displays a congratulatory message: 'Congratulations: You have successfully installed XAMPP on this system!' followed by instructions on how to start using Apache and Co., and a list of contributors: Kay Vogelgesang, Kai 'Oswald' Seidler, and Carsten Wiedmann.

আপনি যদি C:/ ড্রাইভে XAMPP সেটাপ করে থাকেন তাহলে আপনার root directory হবে C:\xampp\htdocs .. এই ফোল্ডারের ভেতরে আপনার পছন্দ মত ফোল্ডার তৈরি করে ফাইলগুলো রাখতে পারবেন। C:\xampp\htdocs লোকেশনে example নামের একটি ফোল্ডার তৈরি কিছু রাখলে ব্রাউজারে তা একসেস করতে <http://localhost/example> লিখতে হবে। আপাতত এটুকুই। পরের পর্বে আমরা ছোট একটি এপ্লিকেশন ডেভেলপমেন্ট শুরু করবো। সেই পর্যন্ত PHP বেসিকগুলো শিখতে থাকুন।

টিপস: কিছু প্রাকটিস করুন Apache ও MySQL চালুর পর যেমন: ব্রাউজারের এড্রেসবারে লিখুন <http://localhost> ফলে নিচের মত একটা পেজ খুলবে। আসেনাই?? না আসলে English বাটনে ক্লিক করুন।



কিন্তু আপনার web content গুলো কোথায় থাকা দরকার?

সকল www ডকুমেন্টের মূল ডাইরেক্টরি হচ্ছে “C:\xampp\htdocs” (তবে যদি অন্য ড্রাইভে ইনস্টল দিয়ে থাকেন যেমন:D drive তখন এটা হবে “D:\xampp\htdocs”). এখন যদি এই ডাইরেক্টরিতে “mytest.html” নামে কোন ফাইল রাখেন তাহলে আপনি এটাতে অ্যাকসেস পেতে পারেন এভাবে-ব্রাউজারের এড্রেসবারে লিখুন <http://localhost/mytest.html>.
আচ্ছা এবার কোড লেখা শুরু করি চলুন, তার আগে একটা কথা কোড কোথায় লিখবেন? নোটপ্যাডে? লিখতে পারেন তবে PHP কোডলেখার জন্য কিছু স্পেশালাইজড সফটওয়্যার আছে যেমন: **Net Beans, Dreamweaver** ইত্যাদি এগুলোতে কোড লিখলে অনেক সুবিধা পাবেন। এগুলো আর বললাম না লিখতে ধরলেই টের পাবেন, যে সুবিধাগুলো নোটপ্যাডে পাবেন না। এগুলোকে বলে IDE (Integrated Development Environment). আপনি যেটাতে সাচ্ছন্দবোধ করেন সেটা ব্যবহার করুন। এখানে টিউটোরিয়ালগুলিতে নেটবিনস এ লেখা কোডের উদাহরণ দেয়া হবে। (কারণ আমি নিজে এটা ব্যবহার করি)

ব্যাসিক সংকেত(syntax):

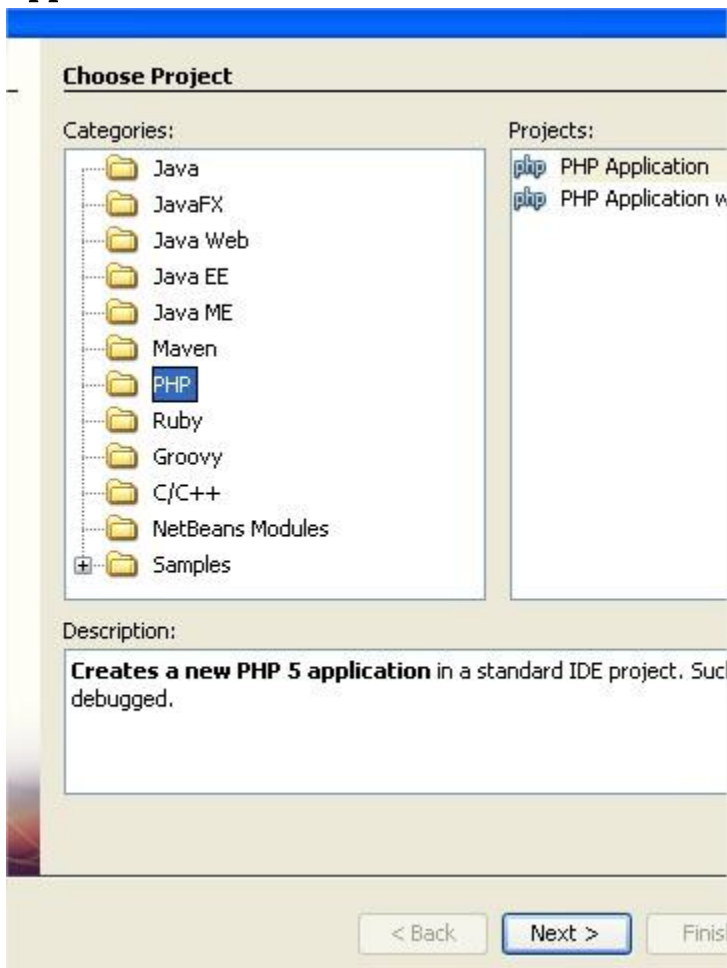
PHP Code কে কাজ করাতে অবশ্যই ফাইল extension **.php** থাকতে হবে। যদি **.html** থাকে তাহলে PHP Code execute হবেনা।

* PHP Code এর প্রতিটি অংশ **<?php** চিহ্ন দিয়ে শুরু এবং **?>** চিহ্ন দিয়ে শেষ হবে।

* একটা PHP Scripting Block(**<?php** and **?>** এর ভিতরে যা লেখা হয় সেসব নিয়ে একটা ব্লক) ডকুমেন্টের ভিতর থাকতে পারে।

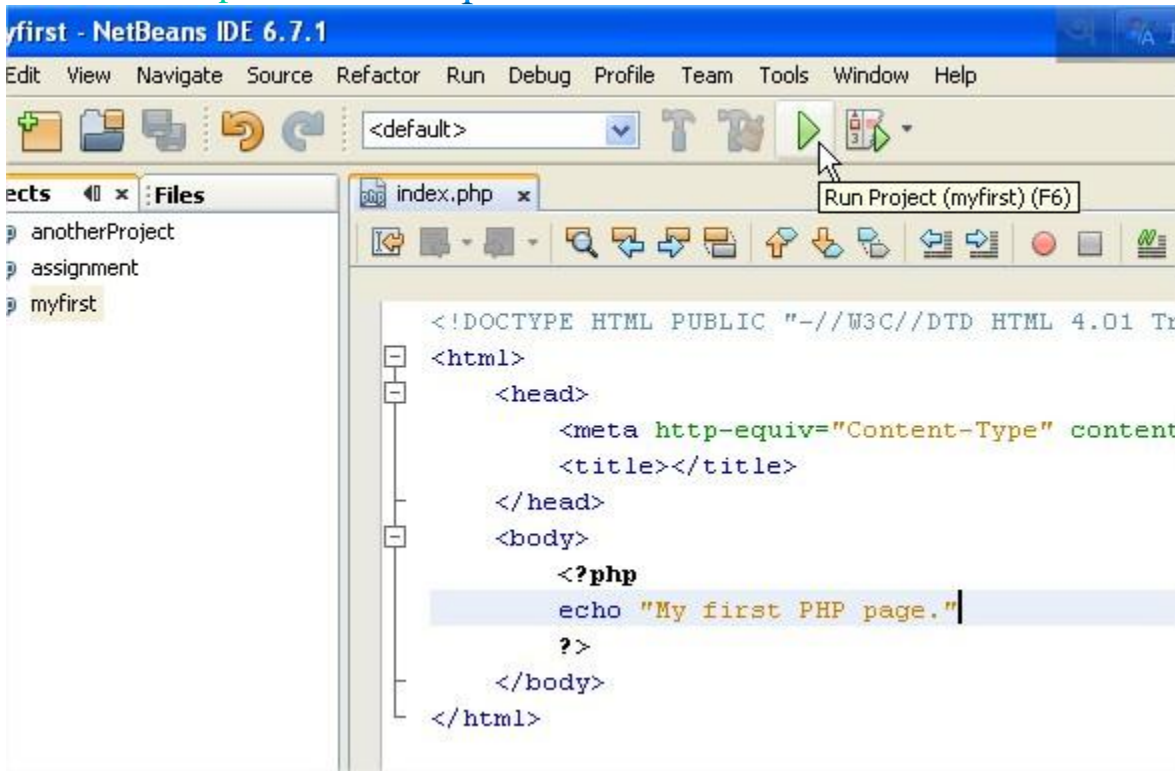
* প্রতিটি আলাদা instruction(code line) সেমিকোলন দ্বারা শেষ হবে।

আচ্ছা এবার নেটবিনস ওপেন করুন। **File>New Project** এবং **Categories** থেকে **PHP** সিলেক্ট করে ডান প্যানে **PHP Application** তারপর **Next**.

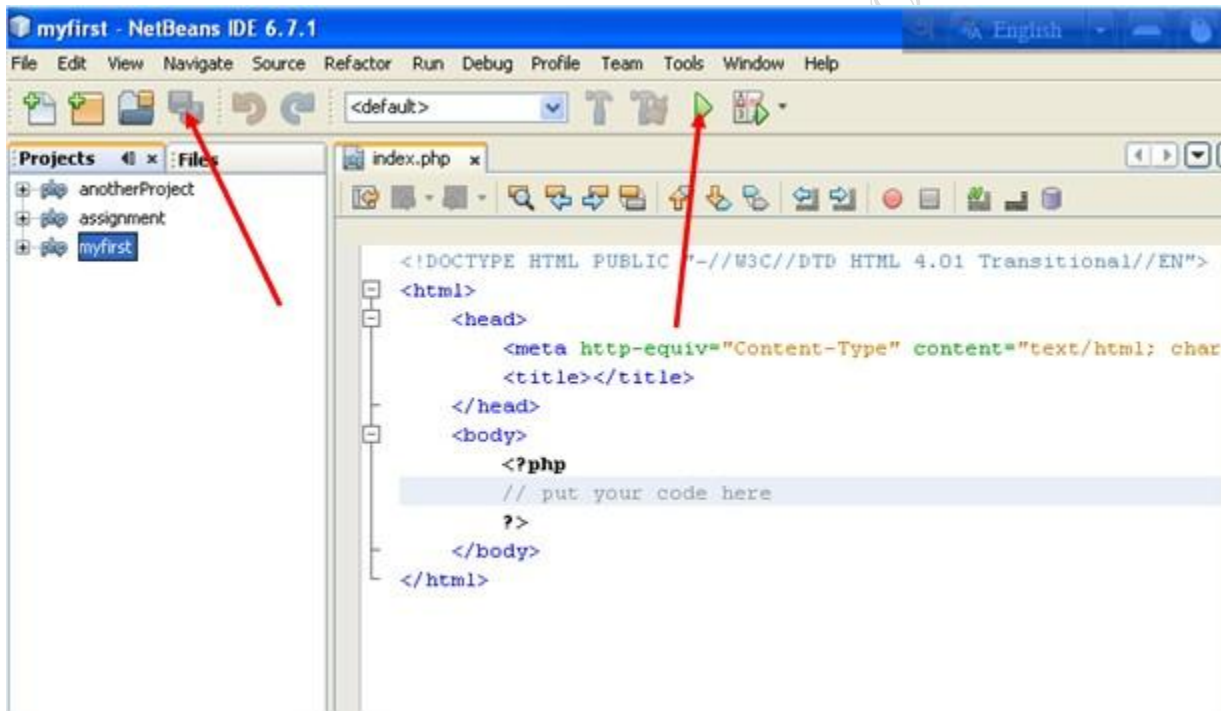


এবার প্রজেক্টের নাম দিতে পারেন, ধরুন নাম দিলাম **myfirst** এখন আবার **Next** এবং শেষে **Finish** button এ ক্লিক করুন। নিচের মত আসছেন? ওকে এবার **<?php** ও **?>** এর মাঝখানে `//put your code here` সিলেক্ট করে ওখানে লিখুন `echo "My first PHP page."`

তাহলে এবার সেভ করে রান করুন। (XAMPP চালু থাকতে হবে কিন্তু। না করলে করুন ওই যে সবার উপরে দেখুন Start button এ ক্লিক করে running করতে হবে। Apache & MySql)



লাল অ্যারো দেয়া বামেরটি দিয়ে সেভ এবং ডানেরটি দিয়ে রান করাতে হয়।



PHP এর টেক্সট আউটপুটের জন্য দুটি স্টেটমেন্ট ব্যবহৃত হয় **echo** ও **print**.

PHP কমেন্ট:কয়েকটা চিহ্ন আছে যদি কোন PHP কোডের সামনে এগুলো দিয়ে রাখেন তাহলে এ কোডগুলি আর execute হবেনা।তবে কোডগুলি এডিটরে থাকবে, অনেক লম্বা সময় পর যদি খোলেন তাহলে এসব কমেন্ট দেখে বুঝতে পারবেন আসলে কি করতে চেয়েছিলেন।একটা লাইনকে কমেন্ট করে রাখতে চাইলে // বা # আর বহু লাইনকে কমেন্ট করে রাখতে চাইলে কোডের আগে /* এবং শেষে */ চিহ্ন ব্যবহার করতে হবে।

পরামর্শ: অনেক প্রাকটিস করুন,যেটুকু শিখেছেন সেটুকুই।যেমন <?php এবং ?> কোডের ভিতর


```
echo "Hello World! ";echo "Hello World! ";echo "Hello World! ";echo "Hello World! ";echo "Hello World! ";এসব লিখতে থাকুন।  
<?php echo "Hello World!";
```

```
echo "Hello World!";
```

?>আর রান করান দেখুন PHP কোডের ভিতর স্পেস কাজ করেনা,এভাবে প্রাকটিস করলে নতুন নতুন জিনিস দেখতে পাবেন।

ভ্যারিয়েবল

ভ্যারিয়েবলের সবচেয়ে সহজ সংজ্ঞা বলা যায় এটা হল একটা পাত্র (Container) যাতে আমরা বিভিন্ন ধরনের ডেটা রাখতে পারি। পিএইচপি একটি লুজলি টাইপড ([Loosely Typed](#)) ল্যাংগুয়েজ আর তাই এখানে আলাদা ভাবে উল্লেখ করে দেয়া লাগে না যে এটা কি ধরনের ভ্যারিয়েবল হবে। সংরক্ষিত ডেটা দেখে পিএইচপি নিজেই বুঝে নেয় যে ভ্যারিয়েবল টি কি টাইপের। এই প্রসঙ্গে আরেকটি টার্মস চলে আসে আর তা হল অ্যারে। অ্যারে নিয়ে পরবর্তীতে আমরা বিস্তারিত দেখব, এখানে শুধু এটুকু জেনে রাখি যে সহজ কথায় অ্যারে হল অনেকগুলো ভ্যারিয়েবলের একটা কনটেইনার 😊 - একটা অ্যারের মাঝে অনেক ভ্যারিয়েবল রাখা যায়। যাকগে, সে পরের কাহিনী।

পিএইচপিতে ভ্যারিয়েবল এর নামের আগে \$ সাইন লাগাতে হয়, তখন পিএইচপি বুঝতে পারে যে এটা হল একটা ভ্যারিয়েবল। একটি ভ্যারিয়েবলে আমরা যদি সংখ্যা স্টোর করি তাহলে সেটা হবে ইন্টিজার ভ্যারিয়েবল (দশমিক এর পরে কিছু না থাকলে) অথবা ডেসিমাল ভ্যারিয়েবল (দশমিক সহ), আর প্রিন্ট করার মত যেকোন কিছু স্টোর করলে তা হল স্ট্রিং। কিন্তু মজার ব্যাপার হল যে এর জন্য আলাদা ভাবে কোন ইনস্ট্রাকশন দেয়ার প্রয়োজন নেই 😊

ভ্যারিয়েবলে মূলত এইটুকুই মনে রাখার বিষয়। চলুন দেখি পিএইচপিতে আমরা কিভাবে ভ্যারিয়েবল নিয়ে কাজ করতে পারি

```
1 <?php  
2 //filename: /home/hasin/www/variable1.php  
3 $var1 = 1;  
4 $var2 = "World's";  
5 echo $var2." Number ".$var1;
```

6 ?>

এবার আমরা যদি ব্রাউজারে <http://localhost/variable1.php> এই ইউআরএল ব্রাউজ করি তাহলে দেখব লেখা আছে

World's Number 1

কিভাবে কি হল? চলুন এবার একটু কোড বোঝা যাক

line 1: পিএইচপি স্ক্রিপ্ট শুরু করতে হয় এভাবে

line 2: এটা একটা কমেন্ট। যেকোন লাইনের আগে # বা // দিলে পিএইচপি সেই লাইন কে কমেন্ট হিসেবে দেখে। নিজেদের বোঝার সুবিধার্থে আমরা বিভিন্ন সময় বিভিন্ন অংশ কমেন্ট করে রাখতে পারি। যেমন এই কমেন্টের মাধ্যমে আমি বুঝিয়েছি যে এই ফাইলটি ডকুমেন্ট রুটে variable1.php নামে সেভ করা হয়েছে।

line 3: আমরা \$var1 নামে একটা ভ্যারিয়েবলে একটা সংখ্যা স্টোর করেছি।

line 4: আমরা \$var2 নামে একটা ভ্যারিয়েবলে “world’s” একটা স্ট্রিং স্টোর করেছি। স্ট্রিং ভ্যালু সংরক্ষণ বা স্টোর করতে হলে তা অবশ্যই সিঙ্গেল কোটেশন ‘ বা ডাবল কোটেশন ” দিয়ে ঘিরে দেয়া লাগবে। পিএইচপিতে সিঙ্গেল কোটেশন এবং ডাবল কোটেশনের আলাদা তাৎপর্য রয়েছে যা আমরা পরে আলোচনা করব।

line 5: এই লাইনে আমরা বেশকয়েকটি ভ্যারিয়েবল এবং একটি স্ট্রিং জোড়া দিয়ে আউটপুট হিসেবে প্রিন্ট করেছি। পিএইচপি তে ” . ” কে বলা হয় কনক্যাটেনেশন অপারেটর বা জোড়াতালি অপারেটর 😊। এর কাজই হল এর দুই পাশের যা কিছু থাকবে তাদেরকে জোড়া তালি দিয়ে একটা বানাবে 😊। আর echo স্টেটমেন্ট দিয়ে তা আউটপুট হিসেবে প্রিন্ট করা হয়েছে।

line 6: পিএইচপি স্ক্রিপ্ট শেষ করার জন্য এভাবে ?> লিখতে হয়

এখানে একটা জিনিস মনে রাখতে হবে। কনক্যাটেনেশন অপারেটর যদি আমরা খুব বেশি ব্যবহার করি তাহলে কোডের রিডেবিলিটি নষ্ট হয়ে যায়। পড়ে বোঝা কঠিন হয়ে যায় যে আসলে কি বলা হয়েছে। তাই লাইন # ৫ কে আমরা নিচের মত করেও লিখতে পারি

```
1 echo "{$var2} Number {$var1}";
```

এখন কোডটুকু দেখতে আগের চেয়ে অনেক সুন্দর লাগছে না? পরবর্তী আর্টিকলে আমরা দেখব কিভাবে

ভ্যারিয়েবল গুলো দিয়ে ম্যাথমেটিকাল অপারেশন (যেমন যোগ, বিয়োগ ইত্যাদি) করতে পারি

শেষ করার আগে আরেকটি বিষয় উল্লেখ করা দরকার আর তা হল ভ্যারিয়েবলের নামকরণের ক্ষেত্রে শুধু মাত্র আলফা নিউমেরিক ক্যারেক্টার গুলো এবং আন্ডারস্কোর (_) ব্যবহার করা যাবে, এবং শুরু অবশ্যই হতে হবে আলফাবেট অথবা আন্ডারস্কোর দিয়ে।

যেমন \$var1, \$var_1 , \$_var এবং \$Var_1_var_2 ভ্যারিয়েবলের নাম হিসেবে সঠিক

কিন্তু \$1var ভ্যারিয়েবলের নাম হিসেবে সঠিক নয়

পড়ার জন্য সবাইকে ধন্যবাদ। কারো কোন কনফিউশন থাকলে কমেন্ট লিখে জানাবেন। আমরা সাধ্যমত চেষ্টা করব সকল

কনফিউশন দূর করার। প্রোগ্রামিং হল খুবই মজার একটা জব 😊ভয় পাওয়ার কোন কারনই নেই 😊

Variables:

Variables হচ্ছে একটা পাত্রের মত(Container)যেখানে আমরা অনেক তথ্য রাখতে পারি।যেমন একটা টেক্সট String “Hello Bangladesh” অথবা একটা integer value 100. কোন একটা Variable এ একবার তথ্য রেখে (কোন Variable এ কিছু রাখা এটাকে বলে Variable declare বা ঘোষণা)সেটা পুরো কোডজুরে বারবার ব্যবহার করতে পারেন,মূল তথ্য(value)টি বারবার রাখার পরিবর্তে।PHP তে Variable “\$” এই চিহ্নটি দিয়ে অবশ্যই শুরু করতে হবে নাহলে কাজ করবেনা।

*Variable নাম case sensitive.যেমন \$a_number and \$A_number দুটি আলাদা Variable PHP এর দৃষ্টিতে।

*নিম্নোক্ত ভাবে PHP তে Variable লেখা হয়

```
$variable_name = Value;
```

উপরে আলোচিতগুলোতে যদি কোন তথ্য রাখতে চাই তাহলে-

```
<?php
$hello = "Hello World!";
$a_number = 4;
$A_Number = 8;
?>
```

*PHP একটা “Loosely Typed” ল্যাংগুয়েজ তাই Variable declare করার সময় Variable এর টাইপ(ধরন) উল্লেখ না করলেও PHP নিজে থেকে Variable কে সঠিক ডেটা টাইপে রূপান্তর করে নেবে।

Variable নামকরণ পদ্ধতি:

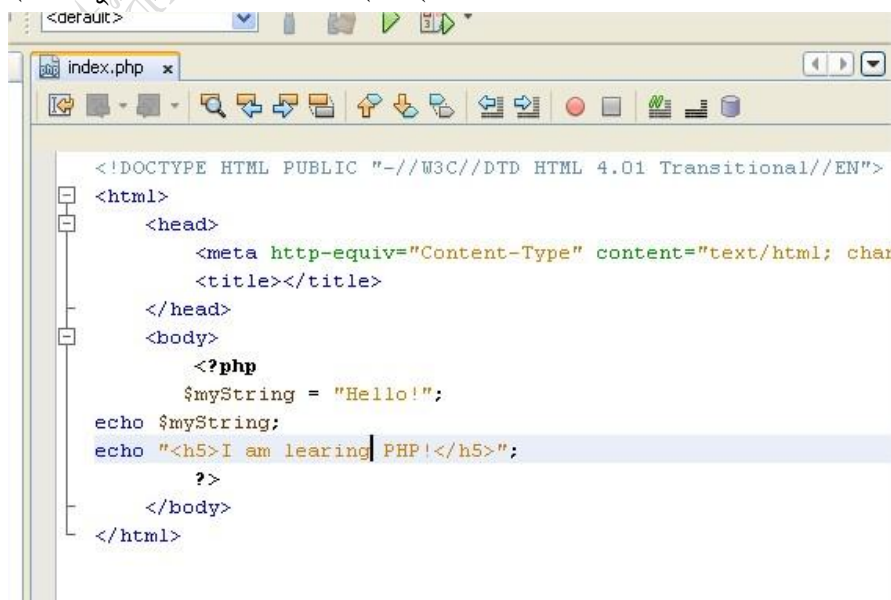
১.অবশ্যই কোন letter or “_”(under score) দিয়ে শুরু করতে হবে।

২. নামের মধ্যে alpha-numeric characters ও underscores. a-z, A-Z, 0-9, or _ . থাকতে পারে।

৩. Variable নামে স্পেস থাকা যাবেনা।যদি নাম একের অধিক হয় তাহলে “_”underscore (\$my_string) অথবা বড় হাতের অক্ষরে(\$myString)লিখতে হবে।

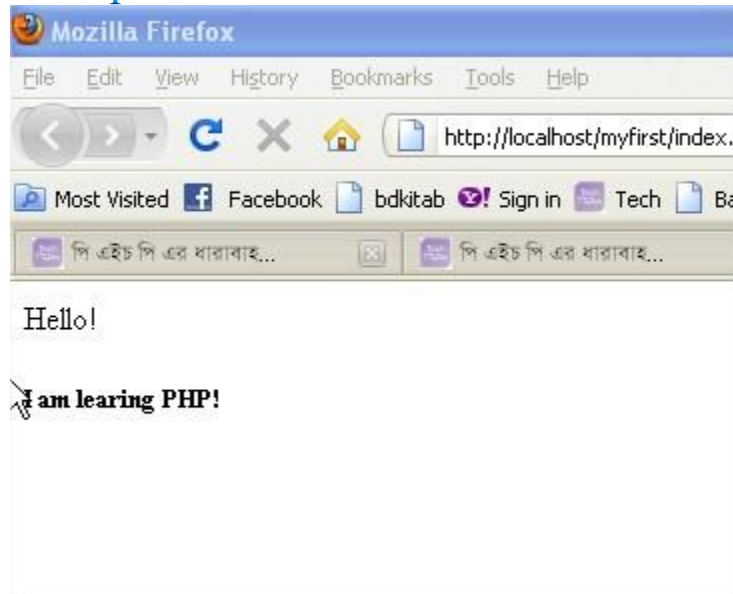
PHP echo:

আগেই বলেছি ব্রাউজারে টেক্সট আউটপুট আনার জন্য PHP তে echo কমান্ডটি ব্যবহৃত হয়।কোনো ভ্যারিয়ারবলের আউটপুট যদি ব্রাউজারে দেখতে চাই তাহলে দেখুন নিচে কিভাবে লেখা হয়েছে।



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; char
  </head>
  <body>
    <?php
      $myString = "Hello!";
    echo $myString;
    echo "<h5>I am learning PHP!</h5>";
    ?>
  </body>
</html>
```

এভাবে নেটবিনস এ লিখে সেভ করে রান করান।নিচের মত আউটপুট পাবেন



এভাবে আরও প্রাকটিস করুন।যেমন-

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title></title>
  </head>
  <body>
    <?php
    $my_string = "Hello Rezwan. My name is: ";
    $my_number = 4;
    $my_letter = a;
    echo $my_string;
    echo $my_number;
    echo $my_letter;
    ?>
  </body>
</html>
```

ভ্যারিয়েবল এক দৃষ্টিতে

ভ্যারিয়েবল টেম্পোরারী ডাটা সেভ করে রাখে । পিএইচপিতে ভ্যারিয়েবল যে কোন প্রকারের ডাটা সেভ করে রাখতে পারে । যেমন : স্ট্রিং(string) , ফ্লোটিং নাম্বার(floating) , অবজেক্ট (object), অ্যারে (Array) । পিএইচপিতে ভ্যারিয়েবল চিহ্নিত করার জন্য শুরুতে ডলার সাইন (\$) দিয়ে তারপর letter (a-zA-Z) বা (_) underscore পর number (0-9) দিতে হয় ।

যেমন:

```
$var = 'valid'; //ঠিক আছে ।  
$var1= 'valid'; //ঠিক আছে ।  
$_var= 'valid'; //ঠিক আছে ।  
$1var = 'valid'; //ভুল কারণ number শুরুতে ।
```

পিএইচপিতে ভ্যারিয়েবল এর বেলায় আরেকটা মজার জিনিস হচ্ছে , এখানে ভ্যারিয়েবল এর ডাটা টাইপ দিতে হয় না । কারণ পিএইচপি হচ্ছে loosely coupling , প্রয়োজনমতো এটা ডাটা টাইপ পরিবর্তন করতে পারে ।

যেমন:

```
$var = 'valid'; //এখন string ।  
$var = 1; //এখন number ।
```

ভ্যারিয়েবল ভ্যারিয়েবলস:

পিএইচপিতে একটা ভ্যারিয়েবলএর ভেলু আরেকটা ভ্যারিয়েবলকে রেফারেন্স করতে পারে ।

যেমন:

```
$var = 'mango';  
$$var = 'apple';  
<?php echo $mango;?> // দেখাবে apple
```

কনস্ট্যান্ট

ভ্যারিয়েবল এর বিপরীত ই হচ্ছে কনস্ট্যান্ট । কনস্ট্যান্ট এর ভেলু পরিবর্তন হয় না।...

পিএইচপি ডাটা টাইপ

ডাটা টাইপ সম্পর্কে বলতে গেলে আগে ভেরিয়েবল নিয়ে বলতে হয়। হাসিন ভাই ভেরিয়েবল নিয়ে বলেছেন। এখানে ডাটা টাইপ নিয়ে সংক্ষেপে কিছু আলোচনা করা যাক।

ভেরিয়েবল আসলে কিছু তথ্য রাখার জন্য পাত্র সরূপ। দৈনন্দিন জীবনে আমরা বিভিন্ন জিনিস রাখার জন্য যেমন বিভিন্ন ধরনের পাত্র ব্যবহার করি। ভেরিয়েবল আসলে সেরকম পাত্রের মতই। বিভিন্ন জিনিস রাখার জন্য পাত্রগুলো বিভিন্ন রকম হয়। যেমন, তরল পদার্থ রাখার জন্য এক ধরনের পাত্র, কঠিন পদার্থ রাখার জন্য এক ধরনের পাত্র, বায়বীয় পদার্থ রাখার জন্য আরেক ধরনের পাত্র। বস্তুর ধরন অনুযায়ী পাত্রের ধরন নির্বাচন করতে হয়।

প্রোগ্রামিংএর ক্ষেত্রে তথ্যের ধরনের উপর ভিত্তি করে ভেরিয়েবল ভিন্ন হয়। PHP একটি শিথিল ধরনের প্রোগ্রামিং ল্যাংগুয়েজ। ভেরিয়েবল বিভিন্ন রকম হলেও ভেরিয়েবল ঘোষণা (declare) করার সময় বলে দিতে হয় না এটা কোন ধরনের পাত্র হবে, মানে কোন ধরনের ডাটা টাইপ হবে। পাঠক, এতক্ষণের আলোচনায় ডাটা টাইপ (data type) সম্পর্কে কিছু ধারণা পেয়েছেন আশা করি।

এবার দেখা যাক কত ধরনের ডাটা টাইপ আছে।

PHP তে আট ধরনের ডাটা টাইপ আছে। সেগুলো হচ্ছে-

[integer](#) ইন্টিজার বা পূর্ণ সংখ্যা, যেমন- 10, 5, 6

[float](#) ফ্লটিং পয়েন্ট বা দশমিক সংখ্যা, যেমন- 5.25, 3.0

[string](#) স্ট্রিং বা বর্ণগুচ্ছ, যেমন- "This is a string."

[boolean](#) বুলিয়ান বা সত্য মিথ্যা, TRUE এবং FALSE

[array](#) এ্যারে

[object](#) অবজেক্ট

[resource](#) রিসোর্স

[NULL](#) নাল

PHP 6 এ Unicode এবং Binary নামে আরো দুই ধরনের ডাটা টাইপ সংযুক্ত করা হয়েছে। (চলবে...)

বর্ণগুচ্ছের(String)

ওয়েবে এখন অনেক উপাদান যেমন ভিডিও , এনিমেশন, সাউন্ড, ছবি অনেক কিছুই পাওয়া যাচ্ছে, তারপরও ওয়েবের অনেকাংশ জুড়ে আছে টেক্সট-মানে অক্ষরের পর অক্ষর!

এই টেক্সট-কে ওয়েবে প্রকাশ করার জন্য পিএইচপি-তে যে ডাটাটাইপ ব্যবহার করা হয় সেটি হলো স্ট্রিং।

এই স্ট্রিং হ্যান্ডলিং আজকের অনলাইন সফটওয়্যার ডেভেলপমেন্টে অনেক গুরুত্বপূর্ণ ভূমিকা রাখে , ডাটা ভ্যালিডেশন থেকে শুরু করে অনেক সিকিউরিটি ইস্যুতে স্ট্রিং হ্যান্ডলিং জরুরী, কোন ডাটা আমার স্টোরেজে জমা হচ্ছে , কোন ডাটা আমি user-কে দেখাবো এইসব ব্যাপার control করতে চাইলে স্ট্রিং হ্যান্ডলিং সম্পর্কে জানতে হবে, সুতরাং ভালো ডেভেলপার হতে হলে পিএইচপিতে স্ট্রিং ও এর ফাংশন সমূহের ব্যবহার ভালোভাবে জানাটা জরুরী ।

স্ট্রিং কি?

প্রথমে জেনে নেই স্ট্রিং কি জিনিস, খায় না মাথায় দেয় 😊।

স্ট্রিং আসলে কতগুলো অক্ষরের সমাহার, অনেকগুলো বর্ণের সমষ্টি, সেজন্যই বলছি বর্ণগুচ্ছ 😊।

এই স্ট্রিং হতে পারে একটি বাক্য, নাম্বার , অথবা একটি শব্দ অথবা একটি মাত্র অক্ষর, একটিমাত্র নাম্বার, অথবা কোন ডেটও হতে পারে।

স্ট্রিং এর মধ্যে অক্ষর, নাম্বার, স্পেস, পাস্কচুয়েশন যে কোন কিছুই থাকতে পারে।

যেমন নিচের সবগুলোই স্ট্রিং -

- 'phpbook'
- 'December 4, 2009'
- "I"
- "5"
- "Hi!"

পিএইচপিতে স্ট্রিং

পিএইচপিতে স্ট্রিং নিয়ে কাজ করতে অন্যান্য ল্যাঙ্গুয়েজের মতো আলাদা কোন বাক্সি পোহাতে হয় না। কারণ পিএইচপি যেহেতু loosely typed তাই কোন ভ্যারিয়েবলে কোন ভ্যালু এসাইন করার সাথে সাথে সেই ভ্যারিয়েবল typecast করে নেয়-স্ট্রিং নিয়ে কাজ করা আর কখনো এতো সহজ ছিলো না 😊

পিএইচপিতে একটি ক্যারেকটার মিন করে এক বাইটকে- তার মানে হচ্ছে আট বিট। তার মানে এই এক বাইট দিয়ে আলাদা আলাদা ২৫৬-টি ক্যারেকটার প্রকাশ করা সম্ভব। এবং বর্তমানে পিএইচপিতে ইউনিকোডের কোন নেটিভ সাপোর্ট নেই!

তবে একটা সুখবর শোনাতে পারি যে পিএইচপিতে স্ট্রিং -এর সাইজ নিয়ে কোন বাধ্যবাধকতা নেই, সাইজ নির্ভর করে কম্পিউটারের মেমরীর উপর।

ও, আরেকটি সুখবর হচ্ছে যে, [পিএইচপি-৬](#) থেকে আমরা নেটিভ ইউনিকোড স্ট্রিং টাইপ পেতে যাচ্ছি, শুধু তাই না অতিরিক্ত একটি বাইনারী স্ট্রিং টাইপও আসছে।

অর্থাৎ বর্তমানে পিএইচপিতে ব্যবহৃত ASCII encoding-এর পরিবর্তে আমরা Unicode encoding পাবো যেখানে আমরা এখন ১ ক্যারেকটার = ১ বাইট পাচ্ছি, সেখানে পাবো ১ ক্যারেকটার >= ১ বাইট!

পরবর্তী পর্বে আমরা দেখবো কিভাবে পিএইচপিতে স্ট্রিং ডিক্লেয়ার করতে হয়।

(স্ট্রিং)

PHP String এতক্ষন ব্যবহার করলেও গভীর আলোচনা করা হয়নি। PHP Career এ এই String একটা গুরুত্বপূর্ণ ভূমিকা পালন করে। তাই এ ব্যাপারে পরিষ্কার ধারণা থাকা আবশ্যিক।

String ব্যবহারের আগে এটাকে তৈরী করে নিতে হবে। একটা String সরাসরি একটা ফাংশনে ব্যবহার হতে পারে অথবা একটা ভ্যারিয়েবলে store থাকতে পারে। নিচে দেখুন একই string দুবার তৈরী করেছি, একবার ভ্যারিয়েবলে store করা হয়েছে আরেকবার সরাসরি echo করা হয়েছে।

```
$my_string = "o merciful make me bold and brave!";
```

```
echo " o merciful make me bold and brave!";
```

```
echo $my_string;
```

উপরের উদাহরণে প্রথম string কে \$my_string ভ্যারিয়েবলে ঢুকিয়ে দেয়া হয়েছে আর দ্বিতীয়বার আরেকটা string কে echo করা হয়েছে, কোনো ভ্যারিয়েবলে না store করেই। একটা জিনিস মাথায় রাখতে হবে যে যখনই কোন string কে আমরা একাধিকবার ব্যবহারের প্লান করব শুধু তখনই এটাকে কোনো ভ্যারিয়েবলে ঢুকিয়ে store করে রাখব।

আচ্ছা এবার উপরের কোডটুকু নিশ্চয় এতক্ষনে নেটবিনসে লিখে ফেলেছেন(এর মধ্যে কোড কিন্তু লিখতে হবে) এখন সেভ করে রান করান। ব্রাউজারে নিচের মত আউটপুট পাবেন।

```
o merciful make me bold and brave! o merciful make me bold and brave!
```

এতক্ষনতো Double quotes দ্বারা string তৈরী করা হয়েছে এখন Single quotes দ্বারা string তৈরী করতে পারেন বরং এটাই ঠিক, তা নাহলে আসলেতো ওটা apostrophes নামে পরিচিত।

```
$my_string = 'o merciful make me bold and brave!';
```

```
echo 'o merciful make me bold and brave!';
```

```
echo $my_string;
```

যদি string এর ভিতর single quotes ব্যবহার দরকার হয় তাহলে এভাবে করুন-

```
Echo 'PHP it\'s neat'
```


আমরা এখানে আপাতত double quotes ব্যবহার করব এতে কিছু সুবিধা আছে যেটা single quotes এ নাই।

String লেখার যে পদ্ধতিদুটি আলোচনা করা হল এ দুটি সাধারনত সব প্রোগ্রামিং ল্যাংগুয়েজ এর ক্ষেত্রে ব্যবহৃত হয়,কিন্তু PHP তে একটা পাওয়ারফুল টুল আছে যেটা দিয়ে বহুলাইনের string লেখা যায় কোনো quotation ব্যবহার করা ছাড়াই।সেটা হল heredoc,একটু সতর্কতার সাথে string কোডিং করতে হবে নাহলে ঝামেলা হবে।নিচে দেখুন কিভাবে এটা করতে হয়-

```
$my_string = <<<
```

```
He will be succeeded here
```

```
and here after!
```

```
best;
```

```
echo $my_string;
```

এভাবে যদি string লেখেন তাহলে কয়েকটি জিনিস অবশ্যই খেয়াল রাখতে হবে-

*<<< বা কিছু identifier আছে যা আপনাকে ব্যবহার করতে হবে heredoc শুরু করার আগে যেমন আমি best ব্যবহার করেছি।

*শেষেও এটি ব্যবহার করেছি এবং সেমিকোলন দিয়ে শেষ হবে।

*এটা নিজেই একটা লাইন হবে,(indent) ফাকা রেখে লাইনটি শুরু করা যাবেনা।

আউটপুট নিচের মত আসবে যেহেতু আমরা

(লাইনের ব্রেক দেয়ার জন্য ব্যবহৃত হয়)ট্যাগ string এর ভিতর ব্যবহার করিনি।

```
He will be succeeded here and here after!
```

স্ট্রিং এর বিস্তারিত

স্ট্রিং হচ্ছে ক্যারেক্টারের অ্যারে | পাশাপাশি অনেকগুলো ক্যারেক্টারকে একটি কোটেশন এর মাঝে রাখলে, সেটি একটি স্ট্রিং।

পিএইচপি এবং অন্যান্য ওয়েব প্রোগ্রামিং "স্ট্রিং" একটি ভেরিএবল যেটি তার ভেলু ধারণ করে দুটি quote এর মধ্যে।

যেমন -

```
<?php
```

```
$str = "This is a string";
```

```
$number = "5";
```

```
?>
```

এখানে \$number একটি স্ট্রিং শুধু মাত্র quote দুটির জন্য।

স্ট্রিং ক্রিয়াশন বেসিক টেকনিক

পিএইচপি তিন উপায়ে স্ট্রিং লিখা হয়।

1. Single Quote এর মধ্যে স্ট্রিং । আমরা যদি স্ট্রিং এর মধ্যে Single Quote ব্যবহার করতে চাই , তাহলে তার আগে একটা Backslash '\' বসাতে হবে।

যেমন -

```
<?php
```

```
$str1 = 'This is a simple string';
```

```
$str2 = 'This is a simple string using single quote(\')';
```

```
echo $str1;
```

```
echo $str2;
```

```
?>
```

আউপুটঃ This is a simple string

 This is a simple string using single quote(')

2. Double Quote এর মধ্যে স্ট্রিং । Double Quote এর মধ্যে আমরা বিভিন্ন escape character ব্যবহার করতে পারব।

যেমন -

```
<?php
```

```
$newline = "A newline is \n";
```

```
$tab = "A tab is \t";
```

```
$dollar = "A dollar sign is \$";
```

```
$doublequote = "A double-quote is \"";
```

```
?>
```

আমরা যদি স্ট্রিং এর মধ্যে escape character দেখাতে চাই , তাহলে তার আগে একটা Backslash '\' বসাতে হবে।

3. Heredoc ব্যবহার করে quote ব্যতীত-ই muliti line স্ট্রিং তৈরী করতে পারে।

যেমন -

```
<?php
```

```
    $str = <<<DEMO
```

```
    This is a string using heredoc identifier
```

```
    DEMO; // এখানে এই DEMO; কথাটির আগে কোনো স্পেস দেয়া যাবেনা।
```

```
    echo $str
```

```
?>
```

Heredoc ব্যবহার করার সময়ক খেয়াল রাখতে হবে যে, এটা <<< এবং যে একটা Identifier (এখানে DEMO) দিয়ে শুরু হবে এবং শেষ হবে নতুন একটা লাইন দিয়ে যার শুরু হবে সেই Identifier দিয়ে।

স্ট্রিং এ ভেরিএবাল ব্যবহার

Library Function নিয়ে আলোচনাঃ

অধ্যায়ের এই অংশে Library Function নিয়ে বিস্তারিতভাবে আলোচনা করা হবে। বিস্তারিতভাবে আলোচনা করার জন্য যেই Library Function -গুলোকে প্রাথমিকভাবে নির্বাচিত করা হয়েছে, তাদের নাম নিম্নে দেওয়া হলোঃ

ফাংশন নামঃ **substr**

লিখার নিয়মঃ string substr (string string, int start [, int length])

কাজঃ এই ফাংশন দিয়ে string-কে খন্ড করা হয়। কোন কাংখিত string-কে খন্ড করার জন্য এই ফাংশনে ৩টি parameter-এর সাহায্য নেওয়া

হয়েছে।

parameter ৩টি হলো string, start এবং length।

"string" হলো - আমরা যেই স্ট্রিং নিয়ে কাজ করবো

"start" হলো - স্ট্রিংটাকে আমরা কোথা থেকে খন্ড করা শুরু করবো

"length" হলো - স্ট্রিংটা থেকে আমরা কতটুকু অংশ বা কয় charecter নিবো

উদাহরণঃ

```
<?php
```

```
    echo substr("abcdefgh", 2, 3); // return "cde"
```

```
?>
```

উপরের উদাহরণে মূল স্ট্রিং হলো "abcdefgh" এবং "cde" হলো ঐ স্ট্রিংয়ের একটি খন্ডাংশ।

পেরামিটার বিশ্লেষণ:

string

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

start

যেখান থেকে string-টাকে খন্ড (কেটে নেওয়া) করা হবে। মনে রাখতে হবে যে, বর্ণগুলোর গননা শুরু হবে 0 থেকে।

অর্থাৎ, যদি স্ট্রিংটা হয় "abcdefgh" তাহলে 0->a,1->b,2->c,...,7->h এভাবে।

যদি start-এর মান ধনাত্মক হয়,

তাহলে গননার দিক হবে বামদিক থেকে ডানদিকে এবং

গননা শুরু হবে 0 থেকে এবং গননার মান ক্রমাগত বাড়বে(0,1,2,3,...)।

আর যদি start-এর মান ঋনাত্মক হয়,

তাহলে গননার দিক হবে ডানদিক থেকে বামদিকে এবং

গননা শুরু হবে -1 থেকে এবং গননার মান ক্রমাগত কমবে(.....-3,-2,-1)।

মনে রাখতে হবে যে, start-এর মান যদি সম্পূর্ণ string length-এর সমান অথবা বড় হয়,

তাহলে ফাংশনটি FALSE রিটার্ন করবে।

উদাহরণঃ

```
<?php
```

```
echo substr("abcdefgh", 0); // return "abcdefgh"
echo substr("abcdefgh", 0, 2); // return "ab"
echo substr("abcdefgh", 5, 3); // return "fgh"
echo substr("abcdefgh", -1); // return "h"
echo substr("abcdefgh", -4); // return "efgh"
echo substr("abcdefgh", -3, 2); // return "fg"
echo substr("abcdefgh", 8); // return ""
```

```
?>
```

length

যদি length-এর মান দেওয়া থাকে এবং তা ধনাত্মক হয়,

তাহলে substr ফাংশন length-এর সমান বর্ণ রিটার্ন করবে।

যদি length-এর মান ঋনাত্মক হয়,

তাহলে length-এর সমান সংখ্যক বর্ণ ডানদিক থেকে বাদ দেওয়া হবে। এমনভাবেই, যদি start-এর মান length-

এর অবস্থানকে অতিক্রম করে তাহলে ফাংশনটি খালি-স্ট্রিং রিটার্ন করবে।

উদাহরণঃ

```
<?php
echo substr("abcdefgh", 1, 4); // return "bcde"
echo substr("abcdefgh", -4, 2); // return "ef"
echo substr("abcdefgh", 2, -3); // return "cde"
echo substr("abcdefgh", 4, -4); // return ""

?>
```

রিটার্ন করবেঃ

স্ট্রিংয়ের খন্ডাংশ।

ফাংশন নামঃ **explode**

লিখার নিয়মঃ `array explode (string delimiter, string string [, int limit])`

কাজঃ কোন string-কে তারই নির্দিষ্ট sub-string-দিয়ে একাধিক অংশে ভাগ করার জন্য এই ফাংশনটি ব্যবহৃত হয়।
যেই sub-string দিয়ে ভাগ করা হবে, তাকে আমরা "প্রান্তসীমা বা delimiter" বলবো।
বুঝতে পারেননি, তাই না? আসুন একটি উদাহরণ দেখি।

উদাহরণঃ

মনে করুন, আপনাকে কাছে কিছু নাম আছে। নামগুলো পৃথক করা আছে "," দিয়ে।
কিন্তু কাজের সুবিধার জন্য নামগুলো আপনার আলাদাভাবে প্রয়োজন। কি করবেন?

```
<?php

$var_names = "mukul,shohel,ashiq,emon,jewel";
$arr_name = explode(",", $var_names);

echo $arr_name[0]; // mukul
echo $arr_name[1]; // shohel
echo $arr_name[2]; // ashiq
```

```
echo $arr_name[3]; // emon
echo $arr_name[4]; // jewel

?>
```

পেরামিটার বিশ্লেষণ:

delimiter

এইখানে delimiter-কে আমরা প্রান্তসীমা বলছি। উপরের উদাহরণে "," হচ্ছে delimiter।
আমরা আরেকটা উদাহরণ দেখি, তাহলে বিষয়টা স্বচ্ছ হবে।

উদাহরণঃ

```
<?php

$var_id = "temp1.com temp2.com temp3.com temp4.com temp5.com";
$arr_id = explode(" ", $var_id);
print_r($arr_id);

/*
Output:
Array
(
    [0] => temp1.com
    [1] => temp2.com
    [2] => temp3.com
    [3] => temp4.com
    [4] => temp5.com
)
*/

?>
```

এই উদাহরণটাকে প্রান্তসীমা(delimiter) হচ্ছে " "(blank space)।

string

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

limit

যদি limit-এর মান দেওয়া থাকে এবং তা ধনাত্মক হয়,

তাহলে limit-এর সমমানের size-এর Array রিটার্ন করবে। এবং string-এর অতিরিক্ত অংশ শেষ element হিসেবে থাকবে।

আর যদি limit-এর মান ঋনাত্মক হয়,

তাহলে limit-এর সমমানের data ডানদিক থেকে বাদ যাবে।

উদাহরণঃ

```
<?php

$var_names = "mukul|shohel|ashiq|emon|jewel";

// positive limit
$arr_name_pos = explode("|", $var_names, 2);
print_r($arr_name_pos);

/*
  Output:
  Array
  (
    [0] => mukul
    [1] => shohel|ashiq|emon|jewel
  )
*/

// negative limit
$arr_name_neg = explode("|", $var_names, -2);
print_r($arr_name_neg);

/*
```

```
Output:
Array
(
    [0] => mukul
    [1] => shohel
    [2] => ashiq
)
*/
?>
```

রিটার্ন করবেঃ

যদি প্রান্তসীমা খালি স্ট্রিং("") হয়, তাহলে এই ফাংশন FALSE রিটার্ন করবে।

আর যদি প্রান্তসীমা খালি না হয় এবং string-এর ভিতর থাকে, তাহলে এই ফাংশন string-এর Array রিটার্ন করবে।

ফাংশন নামঃ **implode**

লিখার নিয়মঃ `string implode (string glue, array pieces)`

কাজঃ glue string-কে নিয়ে Array-এর প্রত্যেকটা ভিন্ন element-কে একটা string-এ রূপান্তর করা হয়, এই ফাংশনের সাহায্যে।

উদাহরণঃ

ধরুন আপনার কিছু data আছে Array-তে। data-গুলো আপনার একসাথে প্রয়োজন-

```
<?php
```

```
$arr_division = array("Dhaka","Chittagong","Rajshahi","Khulna","Silhet","Barishal");
```

```
$comma_separated = implode(", ", $arr_division);
```

```
echo $comma_separated;
```

```
// Dhaka,Chittagong,Rajshahi,Khulna,Silhet,Barishal
```


?>

এই উদাহরনে comma(,) হলো glue string।

পেরামিটার বিশ্লেষণ:

glue

এই ফাংশনে glue string না দিলেও হয়। কেননা ইহা optional।

উদাহরণঃ

উপরের উদাহরনটাই যদি glue string ছাড়া implode ফাংশনে দেওয়া হয়...

```
<?php
```

```
$arr_division = array("Dhaka","Chittagong","Rajshahi","Khulna","Silhet","Barishal");  
$without_glue = implode($arr_division);  
  
echo $without_glue;  
// DhakaChittagongRajshahiKhulnaSilhetBarishal
```

```
?>
```

pieces

যেই Array নিয়ে কাজ করা হবে।

রিটার্ন করবেঃ

সকল Array element নিয়ে গঠিত string রিটার্ন করবে, যেখানে প্রত্যেক element-এর মাঝে glue string বিদ্যমান।

ফাংশন নামঃ **list**

লিখার নিয়মঃ void list (mixed varname, mixed ...)

কাজঃ list কোন ফাংশন না। ইহা array'র মতোই আকটি language construct.

একসাথে আকাধিক variable-এর মান assign করার জন্য এই ফাংশন ব্যবহার করা হয়।

উদাহরণঃ

```
<?php

$info = array('coffee', 'brown', 'caffeine');

// Listing all the variables
list($drink, $color, $power) = $info;
echo "$drink is $color and $power makes it special.\n";
// coffee is brown and caffeine makes it special.

// Listing some of them
list($drink, , $power) = $info;
echo "$drink has $power.\n";
// coffee has caffeine.

// Or let's skip to only the third one
list( , , $power) = $info;
echo "I need $power!\n";
// I need caffeine!

// list() doesn't work with strings
list($bar) = "abcde";
var_dump($bar); // NULL

?>
```

উপরের উদাহরণগুলো দেখলে বুঝা যাবে যে, list-কে আমরা কিভাবে কাজে লাগাতে পারবো।

পেরামিটার বিশ্লেষণঃ

mixed varname

list শুধুমাত্র numeric array'র ক্ষেত্রে কাজ করে, যাদের ইন্ডেক্স 0 থেকে শুরু।

উদাহরণঃ

```
<?php

// array with 0 index
$info = array('0'=>'Ami', '1'=>'Tumi', '2'=>'She');
list($var1, $var2, $var3) = $info;
echo "$var1, $var2, $var3 Amra.\n";
// Ami, Tumi, She Amra

// array without 0 index
$info = array('1'=>'Ami', '2'=>'Tumi', '3'=>'She');
list($var1, $var2, $var3) = $info;
echo "$var1, $var2, $var3 Amra.\n";
// "" (Nothing)

?>
```

শেষ আরেকটা উদাহরণ দেখি, যেখানে date ফাংশন থেকে তারিখ নিয়ে আলাদা করে দেখানো হয়েছে।

উদাহরণঃ

```
<?php

list($day, $month, $year)=explode(",", date("d,m,Y"));

echo "Day : ".$day."\n";
echo "Month : ".$month."\n";
echo "Year : ".$year;

?>
```

রিটার্ন করবেঃ

ফাংশন নামঃ `str_split`

লিখার নিয়মঃ `array str_split (string string [, int split_length])`

কাজঃ `string`-কে `array`-তে রূপান্তর করার জন্য এই ফাংশন ব্যবহার করা হয়।

এই ফাংশনে `array`-এর প্রত্যেকটা অংশের সর্বোচ্চ দৈর্ঘ্য (maximum length) কত হবে তা বলে দেওয়া যায়।

উদাহরণঃ

```
<?php

$str = "Hello Friend";

$arr1 = str_split($str);
$arr2 = str_split($str, 3);

print_r($arr1);
print_r($arr2);

/*
Array
(
    [0] => H
    [1] => e
    [2] => l
    [3] => l
    [4] => o
    [5] =>
    [6] => F
    [7] => r
    [8] => i
    [9] => e
)
*/
```

```
[10] => n
[11] => d
)

Array
(
    [0] => Hel
    [1] => lo
    [2] => Fri
    [3] => end
)
*/

?>
```

পেরামিটার বিশ্লেষণঃ

string

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

split_length

আউটপুট array-এর প্রত্যেকটা অংশের সর্বোচ্চ দৈর্ঘ্য (maximum length) কত হবে তাই হলো split_length।

default ভাবে split_length-এর মান থাকে ১। মানে হলো, যদি split_length-এর মান দেওয়া না থাকে, তবে তা ১।

রিটার্ন করবেঃ

যদি split_length-এর মান উল্লেখ করা থাকে,

তাহলে আউটপুট array-এর প্রত্যেকটা অংশের সর্বোচ্চ দৈর্ঘ্য (maximum length) হবে split_length-এর মানের সমান।

আর যদি split_length-এর মান উল্লেখ করা না থাকে,

তাহলে আউটপুট array-এর প্রত্যেকটা অংশের সর্বোচ্চ দৈর্ঘ্য (maximum length) হবে ১।

মনে রাখতে হবে যে, split_length-এর মান ঋনাত্মক হলে এই ফাংশন FALSE রিটার্ন করবে।

ফাংশন নামঃ **str_repeat**

লিখার নিয়মঃ `string str_repeat (string input, int multiplier)`

কাজঃ নির্দিষ্ট string-কে নির্দিষ্ট সংখ্যকবার পুনরাবৃত্তি করার জন্য এই ফাংশন ব্যবহৃত হয়ে থাকে।

উদাহরণঃ

```
<?php  
  
echo str_repeat("--*--", 5);  
  
// --*--*--*--*--*--*--  
  
?>
```

পেরামিটার বিশ্লেষণঃ

input

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)। অথবা যেই স্ট্রিংয়ের পুনরাবৃত্তি হবে।

multiplier

ইনপুট স্ট্রিংকে কতো সংখ্যকবার পুনরাবৃত্তি করা হবে, multiplier-দিয়ে তাই নির্ধারণ করা হয়।

রিটার্ন করবেঃ

পুনরাবৃত্তি হওয়া স্ট্রিংটাকে রিটার্ন করবে।

ফাংশন নামঃ **strcmp**

লিখার নিয়মঃ `int strcmp (string str1, string str2)`

কাজঃ দুইটা স্ট্রিং সমান কিনা তা যাচাই করার জন্য এই ফাংশন ব্যবহার করা হয়। মনে রাখতে হবে যে, এই ফাংশনটি

case sensitive। অর্থাৎ, এই ফাংশনটির কাছে ছোট হাতের লেখা এবং বড় হাতের লেখা ভিন্ন। যেমনঃ

phpbook এবং phpBook এই দুইটি শব্দ আমাদের কাছে একই অর্থ বহন করে, কিন্তু strcmp-এর কাছে এই শব্দ

দুটি ভিন্ন। কেননা, এই শব্দ দুটিতে 'b' বর্ণটি এক জায়গায় ছোট হাতের এবং এক জায়গায় বড় হাতের আছে।

উদাহরণঃ

```
<?php

$str1 = 'phpbook';
$str2 = 'phpBook';

if(strcmp($str1, $str2)>0)
    echo "str1 is greater than str2.";
if(strcmp($str1, $str2)<0)
    echo "str1 is less than str2.";
if(strcmp($str1, $str2)==0)
    echo "str1 and str2 is equal.";

// str1 is greater than str2.

/*
    str1      str2      output
    -----
    phpbook   phpBook   str1 is greater than str2.
    phpBook   phpbook   str1 is less than str2.
    phpbook   phpbook   str1 and str2 is equal.
*/

?>
```

পেরামিটার বিশ্লেষণঃ

str1

প্রথম ইনপুট স্ট্রিং।

str2

দ্বিতীয় ইনপুট স্ট্রিং।

রিটার্ন করবেঃ

যদি $str1 > str2$ অর্থাৎ str1, str2 থেকে বড় হয়,

তাহলে ধনাত্মক মান রিটার্ন করবে।

যদি `str1 == str2` অর্থাৎ `srt1` এবং `str2` সমান হয়,

তাহলে 0 রিটার্ন করবে।

যদি `str1 < str2` অর্থাৎ `srt1`, `str2` থেকে ছোট হয়,

তাহলে ঋনাত্মক মান রিটার্ন করবে।

ফাংশন নামঃ **strlen**

লিখার নিয়মঃ `int strlen (string string)`

কাজঃ কোন string-এর দৈর্ঘ্য(length)-এর মান জানার জন্য এই ফাংশন ব্যবহার করা হয়।

উদাহরণঃ

```
<?php  
  
echo strlen("phpbook");           // 7  
echo strlen("phpbook group");     // 13  
echo strlen(" phpbook group ");   // 15  
  
?>
```

পেরামিটার বিশ্লেষণঃ

string

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

রিটার্ন করবেঃ

স্ট্রিংয়ের দৈর্ঘ্য রিটার্ন করবে। যদি স্ট্রিংটি খালি(empty string) হয়, তাহলে 0 রিটার্ন করবে।

ফাংশন নামঃ **strtolower**

লিখার নিয়মঃ `string strtolower (string str)`

কাজঃ বড় হাতের বর্ণকে ছোট হাতের বর্ণতে রূপান্তর করার জন্য এই ফাংশন ব্যবহৃত হয়।

মনে রাখতে হবে যে, এই ফাংশন সাধারণ বর্ণ "A থেকে Z"-কে রূপান্তর করে।

umlaut-A(Ä) বা উপরে দুই ফোটাওয়ালা বর্ণগুলোকে এই ফাংশন রূপান্তর করে না।

উদাহরণঃ

```
<?php  
  
$str = "BANGLADESH IS MY COUNTRY.";  
$str = strtolower($str);  
echo $str;  
  
// bangladesh is my country.  
  
?>
```

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

রিটার্ন করবেঃ

রূপান্তরিত ছোট হাতের বর্ণসহ স্ট্রিংকে রিটার্ন করবে।

ফাংশন নামঃ **strtoupper**

লিখার নিয়মঃ string strtoupper (string str)

কাজঃ ছোট হাতের বর্ণকে বড় হাতের বর্ণতে রূপান্তর করার জন্য এই ফাংশন ব্যবহৃত হয়।

মনে রাখতে হবে যে, এই ফাংশন সাধারণ বর্ণ "a থেকে z"-কে রূপান্তর করে।

umlaut-a(ä) বা উপরে দুই ফোটাওয়ালা বর্ণগুলোকে এই ফাংশন রূপান্তর করে না।

উদাহরণঃ

```
<?php  
  
$str = "bangladesh is my country.";  
$str = strtoupper($str);  
echo $str;
```

```
// BANGLADESH IS MY COUNTRY.
```

```
?>
```

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

রিটার্ন করবেঃ

রূপান্তরিত বড় হাতের বর্ণসহ স্ট্রিংকে রিটার্ন করবে।

ফাংশন নামঃ **ucfirst**

লিখার নিয়মঃ string ucfirst (string str)

কাজঃ

এই ফাংশন ব্যবহার করলে, ইনপুট স্ট্রিংয়ের প্রথম অক্ষর বড় হাতের লেখা হবে।

মনে রাখতে হবে যে, স্ট্রিংয়ের প্রথম অক্ষর যদি বর্ণ হয় তবেই এই ফাংশন কাজ করবে। সংখ্যা হলে করবে না।

উদাহরণঃ

```
<?php

$str1 = "bangladesh is my country.";
$str1temp = ucfirst($str1);
echo $str1temp;
// Bangladesh is my country.

$str2 = "BANGLADESH IS MY COUNTRY.";
$str2temp = ucfirst($str2);
echo $str2temp;
// BANGLADESH IS MY COUNTRY.

$str3temp = ucfirst(strtolower($str2));
echo $str3temp;
```

```
// Bangladesh is my country.
```

```
?>
```

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

রিটার্ন করবেঃ

রূপান্তরিত স্ট্রিংকে রিটার্ন করবে।

ফাংশন নামঃ **ucwords**

লিখার নিয়মঃ string ucwords (string str)

কাজঃ

এই ফাংশন ব্যবহার করলে, ইনপুট স্ট্রিংয়ের প্রত্যেক শব্দের প্রথম অক্ষর বড় হাতের লেখা হবে। মনে রাখতে হবে যে, শব্দের প্রথম অক্ষর যদি বর্ণ হয় তবেই এই ফাংশন কাজ করবে। সংখ্যা হলে করবে না।

উদাহরণঃ

```
<?php

$str1 = "bangladesh is my country.";
$str1temp = ucwords($str1);
echo $str1temp;
// Bangladesh Is My Country.

$str2 = "BANGLADESH IS MY COUNTRY.";
$str2temp = ucwords($str2);
echo $str2temp;
// BANGLADESH IS MY COUNTRY.

$str3temp = ucwords(strtolower($str2));
echo $str3temp;
// Bangladesh Is My Country.
```

?>

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

রিটার্ন করবেঃ

রূপান্তরিত স্ট্রিংকে রিটার্ন করবে।

ফাংশন নামঃ **trim**

লিখার নিয়মঃ `string trim (string str [, string charlist])`

কাজঃ এই ফাংশন দিয়ে ইনপুট স্ট্রিংয়ের সামনের ও পিছনের দিক হতে সকল শূন্য স্থান মুছে ফেলা হয় (যদি না দ্বিতীয় পেরামিটার charlist দেওয়া থাকে)। শূন্য স্থান বলতে বুঝায়ঃ

- " " (ASCII 32 (0x20)), an ordinary space.[সাধারণ শূন্য স্থান]
- "\t" (ASCII 9 (0x09)), a tab.[আনুভূমিক ট্যাব]
- "\n" (ASCII 10 (0x0A)), a new line (line feed).[নতুন লাইন]
- "\r" (ASCII 13 (0x0D)), a carriage return.[কারিজ রিটার্ন (কারসর প্রথমে আনা)]
- "\0" (ASCII 0 (0x00)), the NULL-byte.[ন্যাল ক্যারেঞ্জার]
- "\x0B" (ASCII 11 (0x0B)), a vertical tab.[লাম্বিক টেব]

উদাহরণঃ

```
<?php

$str1 = "\t\tThese are a few words :) ... ";
echo trim($str1);
// "These are a few words :) ..."

$str2 = "\x09Example string\x0A";
echo trim($str2);
// "Example string"
```

```
$str3 = "Hello World";  
echo trim($str3);  
// "Hello World"  
  
?>
```

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

charlist

এই পেরামিটার দিয়ে নির্দিষ্ট করে দেওয়া যায় যে কোন কোন বর্ণগুলো মুছে ফেলতে চাই।

মুছতে ফেলতে চাওয়া বর্ণগুলোকে দিয়ে একটি স্ট্রিং তৈরি করে দিলেই হবে অথবা বর্ণগুলোর একটি রেঞ্জ দিলেও হবে।

(রেঞ্জ দেওয়ার নিয়ম হলো, শুরুর বর্ণ, শেষের বর্ণ এবং তাদের মাঝে ".."।

যেমনঃ আমরা যদি রেঞ্জ দিতে চাই a থেকে z, তাহলে লিখতে হবে "a..z")

উদাহরণঃ

```
<?php
```

```
$str1 = "php is a widely-used Open Source general-purpose scripting language";  
echo trim($str1, "aeguph")."\n";  
// " is a widely-used Open Source general-purpose scripting lan"
```

```
$str1 = "php is a widely-used Open Source general-purpose scripting language";  
echo trim($str1, "aeguph ")."\n";  
// "is a widely-used Open Source general-purpose scripting lan"
```

```
$str1 = "php is a widely-used Open Source general-purpose scripting language";  
echo trim($str1, "a..z")."\n";  
// " is a widely-used Open Source general-purpose scripting"
```

```
$str1 = "php is a widely-used Open Source general-purpose scripting language";
```

```
echo trim($str1, "a..z ")."\n";  
// "-used Open Source general-"
```

?>

রিটার্ন করবেঃ

রূপান্তরিত স্ট্রিংকে রিটার্ন করবে।

ফাংশন নামঃ **strip_tags**

লিখার নিয়মঃ `string strip_tags (string str [, string allowable_tags])`

কাজঃ কোন স্ট্রিংয়ের মধ্য থেকে **HTML** এবং **php** ট্যাগ মুছে ফেলার জন্য এই ফাংশন ব্যবহৃত হয়।

উদাহরণঃ

<?php

```
$text = '<p>Test paragraph.</p><!-- Comment --> <a href="#fragment">Other text</a>';  
echo strip_tags($text);  
// Test paragraph. Other text  
  
// Allow <p> and <a>  
echo strip_tags($text, '<p><a>');  
// <p>Test paragraph.</p> <a href="#fragment">Other text</a>
```

?>

পেরামিটার বিশ্লেষণঃ

str

যেই স্ট্রিং নিয়ে কাজ করা হবে(ইনপুট স্ট্রিং)।

allowable_tags

কোন ট্যাগ যদি আমরা মুছে ফেলতে না চাই, তাও আমরা বলে দিতে পারি এই allowable_tags পেরামিটারের মাধ্যমে।

রিটার্ন করবেঃ

কাজিত ট্যাগবিহীন স্ট্রিংকে রিটার্ন করবে।

বর্ণগুচ্ছের সহজ পাঠ- ডিক্লেয়ারেশন

আমরা জেনেছিলাম স্ট্রিং কি। আজ আমরা দেখবো কিভাবে স্ট্রিং ডিক্লেয়ার করতে হয়।

পিএইচপিতে তিন উপায়ে স্ট্রিং লিখা যায়-

- সিঙ্গেল কোট (single quote) ব্যবহার করে,
- ডাবল কোট (double quote) ব্যবহার করে,
- হেয়ারডক (heredoc) ব্যবহার করে।

সিঙ্গেল কোট (single quote) এর ব্যবহারঃ

স্ট্রিং-কে সিঙ্গেল কোট(")-এর মধ্যে লেখা যায়। কে যেন মনে হয় বলেছিলো-

An example is worth a thousand words 😊

একটা উদাহরণ দিলেই পরিষ্কার হবে-

```
1 <?php
2 //filename: /home/samiul/www/string1.php
3 echo 'php book project';
4 ?>
```

এবার আউটপুট-টি দেখি-

```
php book project
```

output

খয়াল করে দেখেন যে সিঙ্গেল কোট(")-এর ভিতরের অংশটুকুই শুধু ব্রাউজারে রেন্ডারড হয়েছে। সিঙ্গেল কোট(")-এর ভিতরের শব্দগুলোর মাঝের স্পেসগুলোও বাদ যায়নি।

আরো কিছু উদাহরণ -

```
1 <?php
```

```
2 //filename: /home/samiul/www/string2.php
3 echo '1024';//OUTPUT: 1024
4 echo '14th November 2009';//OUTPUT: 14th November 2009
5 $var= 'string variable';
6 echo $var;//OUTPUT: string variable
7 ?>
```

এখন যদি আমরা কোন টেক্সটে কোন সিঙ্গেল কোট দিতে চাই তাহলে এটাকে এক্সেপ করতে হবে ব্যাকস্ল্যাশ(\) দিয়ে। যেমন-

```
1 <?php
2 //filename: /home/samiul/www/string3.php
3 echo 'It\'s me';//OUTPUT: It's me
4 ?>
```

```
1 <?php
2 //filename: /home/samiul/www/string2.php
3 echo '1024';//OUTPUT: 1024
4 echo '14th November 2009';//OUTPUT: 14th November 2009
5 $var= 'string variable';
6 echo $var;//OUTPUT: string variable
7 ?>
```

এখন যদি আমরা কোন টেক্সটে কোন সিঙ্গেল কোট দিতে চাই তাহলে এটাকে এক্সেপ করতে হবে ব্যাকস্ল্যাশ(\) দিয়ে।।

```
1 <?php
2 //filename: /home/samiul/www/string3.php
3 echo 'It\'s me';//OUTPUT: It's me
4 ?>
```

এবং কোন টেক্সটে

কোন ব্যাকস্ল্যাশ(\) দিতে চাইলেও সেটাকে এক্সেপ করতে হবে ব্যাকস্ল্যাশ(\) দিয়ে। যেমন-

```
1 <?php
2 //filename: /home/samiul/www/string4.php
3 echo 'C:\\windows';// OUTPUT: C:\windows
4 ?>
```


কিন্তু অন্য আর কোন ক্যারেক্টার এক্সেপ করা যাবে না, এক্সেপ করলে

ব্যাকস্ল্যাশ(\) সহ আউটপুট আসবে ।

যেমন —

```
1 <?php
```

```
2 //filename: /home/samiul/www/string5.php
```

```
3 echo 'hi\n'; //OUTPUT: hi\n
```

```
4 ?>
```

ম্যাথমেটিকাল অপারেটর সমূহ

এইযে আমরা যোগ, বিয়োগ, গুন, ভাগ করার জন্য যেই সাইন গুলো ব্যবহার করি তাদেরকেই বলে ম্যাথমেটিকাল অপারেটর ।

চলুন দেখে তাদের নাম এবং কাজ

+ এইটা হল সবচেয়ে কঠিন (থুক্ক সহজ) অপারেটর 😊 এর কাজ হল যোগ করা

- বিয়োগ করার জন্য

* এটা দিয়ে গুন করা হয়, মাল্টিপ্লিকেশন অপারেটর

/ ডিভিশন অপারেটর, ভাগ করার জন্য

% মডুলাস অপারেটর, মনে আছে আমরা ছোটবেলায় পাটিগনিতের অংকে দুটি সংখ্যার ভাগফলের অবশিষ্ট বের করতাম । সেই কাজ করা হয় মডুলাস অপারেটর দিয়ে

অপারেটর প্রিসিডেন্স

এর মানে হল কোন অপারেটর কার আগে কাজ করবে এইটা বলা যায় এইভাবে যে *, / এবং % এর একই রকম প্রিসিডেন্স এবং

+ ও - আসবে এর পরে । ফলে কোনো স্টেটমেন্ট যদি এমন হয়

```
১+৫*৩
```

তাহলে তার ফলাফল হবে ১৬, কারন * কাজ করবে সবার আগে ।

মনে আছে ছোটবেলার সেই BODMAS এর কথা, সরল করার সময় বার বার মনে করা লাগত - ওই যে

Bracket Or Division Multiplication Addition Subtraction = BODMAS

এখানেও সেই একই ব্যপার ।

ভ্যারিয়েবল পঞ্চবিংশতি

বেতাল পঞ্চবিংশতির মত আমার ভ্যারিয়েবল নিয়েও ২৫টা আর্টিকেল লিখে ফেলতে ইচ্ছা করছে । কিন্তু আসলে ভ্যারিয়েবল নিয়ে

আপাতত অত কিছু লেখার নাই । কিন্তু নিরাশ হওয়ারও কোন কারন নাই, হেহ হেহ, একদিন না একদিন লিখে ফেলবই

ভ্যারিয়েবল পর্ব শেষ করার আগে আরো কিছু কিছু জিনিস জেনে নেয়া যাক । যেমন যেকোন ভ্যারিয়েবল সম্বন্ধে বিস্তারিত জানার

জন্য (এটাকে বলা হয় ইন্সপেকটিং) আমরা যে দুটি ফাংশন ব্যবহার করি তা হল print_r() এবং var_dump() । print_r()

আমরা ব্যবহার করব প্রধানত অ্যারের ক্ষেত্রে। সেজন্য আপাতত আমরা এখানে আলোচনা করব না, বরং আমরা দেখব কিভাবে var_dump() ব্যবহার করা যায়

```
01 <?php
```

```
02 //filename: /home/hasin/www/dump1.php
```

```
03 $number1 = 123;
```

```
04 $number2 = 12.3;
```

```
05 $string1 = "Hello Readers";
```

```
06 $string2 = "112233";
```

```
07 echo var_dump($number1)."<br/>";
```

```
08 echo var_dump($number2)."<br/>";
```

```
09 echo var_dump($string1)."<br/>";
```

```
10 echo var_dump($string2)."<br/>";
```

```
11                                     ?>
```

```
01 <?php
02 //filename: /home/hasin/www/dump1.php
03 $number1 = 123;
04 $number2 = 12.3;
05 $string1 = "Hello Readers";
06 $string2 = "112233";
07 echo var_dump($number1)."<br/>";
08 echo var_dump($number2)."<br/>";
09 echo var_dump($string1)."<br/>";
10 echo var_dump($string2)."<br/>";
11 ?>
```

আমরা যদি উপরের কোড রান করাই (ব্রাউজারে <http://localhost/dump1.php>) তাহলে আউটপুট হিসেবে দেখব নিচের টেক্সট টুকু

```
1 int(123)
```

```
2 float(12.3)
```

```
3 string(13) "Hello Readers"
```

```
4 string(6) "112233"
```

আমরা দেখতে পাচ্ছি যে পিএইচপিতে কোন সংখ্যা যদি আমরা ডাবল কোটেশন বা সিংগল কোটেশনের মাঝে রাখি, তাহলে পিএইচপি তাকে স্ট্রিং হিসেবে ব্যবহার করে। যেমনটি করেছে \$string2 এর ক্ষেত্রে। কিন্তু আপনি যদি কোন এরিথমেটিক অপারেশন যেমন যোগ, বিয়োগ করেন তাহলে পিএইচপি কিন্তু তাকে স্বয়ংক্রিয় ভাবে সঠিক টাইপে পরিবর্তন করে নেবে। এটাকে বলা হয় অটো টাইপ কাস্টিং। নিচের কোড টি দেখুন

```
01 <?php
```

```
02 //filename: /home/hasin/www/dump2.php
```

```
03 $string1 = "11";
```

```
04 $string2 = "2nd Dec";
```

```
05 $string3 = "Dec 2nd";
```

```
06 echo $string1*2 . "\n"; //output = 22
```

```
07 echo $string1+2 . "\n"; //output = 13
```

```
08 echo $string1-2 . "\n"; //output = 9
```

```
09 echo $string1/2 . "\n"; //output = 5.5
```

```
10
```

```
11 echo $string2*2 . "\n"; //output = 4
```

```
12 echo $string2+2 . "\n"; //output = 4
```

```
13 echo $string2-2 . "\n"; //output = 0
```

```
14 echo $string2/2 . "\n"; //output = 1
```

```
15
```

```
16 echo $string3*2 . "\n"; //output = 0
```

```
17 echo $string3+2 . "\n"; //output = 2
```

```
18 echo $string3-2 . "\n"; //output = -2
```

```
19 echo $string3/2 . "\n"; //output = 0
```

```
20 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/dump2.php
03 $string1 = "11";
04 $string2 = "2nd Dec";
05 $string3 = "Dec 2nd";
06 echo $string1*2 . "\n"; //output = 22
07 echo $string1+2 . "\n"; //output = 13
08 echo $string1-2 . "\n"; //output = 9
09 echo $string1/2 . "\n"; //output = 5.5
10
11 echo $string2*2 . "\n"; //output = 4
12 echo $string2+2 . "\n"; //output = 4
13 echo $string2-2 . "\n"; //output = 0
14 echo $string2/2 . "\n"; //output = 1
15
16 echo $string3*2 . "\n"; //output = 0
17 echo $string3+2 . "\n"; //output = 2
18 echo $string3-2 . "\n"; //output = -2
19 echo $string3/2 . "\n"; //output = 0
20 ?>
```

বলতে পারবেন লাইন নম্বর ১১র আউটপুট কেন ৪ হবে? কারন স্ট্রিং থেকে ইন্টিজার বা ফ্লোট টাইপ কাস্টিং এর সময় পিএইচপি যতক্ষন পর্যন্ত একটি ভ্যালিড সংখ্যা পাবে সে পর্যন্ত নিবে। এখানে 2nd Dec এ ২ পর্যন্ত একটি ভ্যালিড সংখ্যা। যদি এটা হত 16th Dec তাহলে পিএইচপি ১৬ পর্যন্ত নিত। এখন লক্ষ্য করুন, যেহেতু টাইপ কাস্টিং এর সময় পিএইচপি ২ পেয়েছে, তাই সবগুলো এরিথমটিক অপারেশন হয়েছে ২ এর সাথে। ফলে আসলে লাইন নম্বর ১১ তে ২*২ এর ফলাফল ৪ পেয়েছি আমরা একই কারনে লাইন নম্বর ১৬র আউটপুট ০। কারন Dec 2nd এর প্রথমেই আলফাবেট থাকায় পিএইচপি একে ০ হিসেবে নিয়েছে, আর তাই সবগুলো এরিথমটিক অপারেশন হয়েছে ০ এর সাথে মজার ব্যপার না?

লজিক

আমরা প্রতিদিনই কাজের মাঝে কিছু কথা এভাবে বলি, যে “যদি এটা হয়, তবে ওটা কর, নয়ত ঐটা করবে”, বলি না? পিএইচপি এবং অন্য যেকোন প্রোগ্রামিং ল্যাংগুয়েজেও কোনো একটা ঘটনা বা ফলাফলের উপরে ভিত্তি করে এরকম সিদ্ধান্ত নেয়া লাগে। একেই বলে লজিক। এবং যে ঘটনার প্রেক্ষিতে আমরা সিদ্ধান্ত নেই, তাকে বলে কন্ডিশন বা লজিক্যাল এক্সপ্রেশন। চলুন দেখা যাক আমরা কিভাবে পিএইচপিতে লজিক ব্যবহার করি পিএইচপি তে দুই ধরনের লজিক্যাল ব্লক লেখা যায়। একটা হল if-else দিয়ে, আরেকটা হল switch-case দিয়ে। আজকের আর্টিকলে আমরা এই if-else ব্লকের উদাহরণ দেখব, switch-case পরের আর্টিকলে। আমরা তো বেসিক এরিথমটিক অপারেশন শিখেছিই তাই না? ঐ যে, যোগ বিয়োগ গুন ভাগের কথা বলছিলাম। চলুন এবার লজিক ব্যবহার করে আমরা ঠিক করি কোন একটা সংখ্যা কি জোড় (even) না বেজোড় (odd)। তার আগে মনে মনে ভেবে নিন কিভাবে আসলে বোঝা যায় যে কোন একটি সংখ্যা জোড় না বেজোড়? যদি সংখ্যাটিকে ২ দিয়ে ভাগ করলে অবশিষ্ট ০ হয়, তাহলে জোড়। নয়ত বেজোড়।

```
01 <?php
02 //filename: /home/hasin/www/logic1.php
03 $number = 11;
04 if ( ($number%2) == 1)
05 {
```

```
06 echo "{$number} is odd number";
07 }
08 else
09 {
10 echo "{$number} is even number";
11 }
12 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/logic1.php
03 $number = 11;
04 if ( ($number%2) == 1)
05 {
06     echo "{$number} is odd number";
07 }
08 else
09 {
10     echo "{$number} is even number";
11 }
12 ?>
```

এবার আমরা যদি উপরের কোড রান করাই (ব্রাউজারে <http://localhost/logic1.php>) তাহলে আউটপুট হিসেবে দেখব “11 is odd number”.কিভাবে হল এটা? চলুন কিছুক্ষন কোড বুঝি

লাইন # ৪ এ আমরা আমাদের লজিক্যাল ব্লক শুরু করেছি। এখানে ($\$number\%2$)== 1এটা হল আমাদের কন্ডিশন বা লজিক্যাল এক্সপ্রেশন। আর == হল একটা লজিক্যাল অপারেটর। এই অপারেটর দিয়ে বাম পাশের অংশ ডান পাশের অংশের সমান কিনা তা চেক করা হয়। এধরন আরো কয়েকটি অপারেটর রয়েছে যা আমরা নিচে উল্লেখ করেছি। পিএইচপি তে পুরো লজিক্যাল এক্সপ্রেশন ফার্স্ট ব্র্যাকেটের মাঝে রাখতে হয়।

লাইন # ৫ এ আমরা একটি সেকেন্ড ব্র্যাকেট দিয়ে স্কোপ শুরু করি। একটি সেকেন্ড ব্র্যাকেট শুরু থেকে সেকেন্ড ব্র্যাকেট শেষ না হওয়া পর্যন্ত একই স্কোপে থাকে। অর্থাৎ লাইন # ৫ থেকে লাইন # ৭ পর্যন্ত একই স্কোপের মাঝে পড়ে। আমাদের এই কোডে লাইন নম্বর ৪ এর if এর কন্ডিশন যদি সত্য হয়, তাহলে লাইন # ৫ থেকে লাইন # ৭ পর্যন্ত চলবে (যেহেতু একই স্কোপের অধীনে)।

লাইন # ৬ এ আমরা echo স্টেটমেন্ট দিয়ে আউটপুট প্রিন্ট করেছি।

লাইন # ৭ এ if ব্লকের স্কোপ শেষ করা হয়েছে সেকেন্ড ব্র্যাকেট দিয়ে।

লাইন # ৮ এ আমরা else ব্লক লিখেছি। if ব্লকের কন্ডিশন যদি মিথ্যা হয় তাহলে এই ব্লকটি রান করবে।

লাইন # ৯ – লাইন # ১১ এখানেও আমরা আগের মত সেকেন্ড ব্র্যাকেট দিয়ে স্কোপ শুরু করেছি, ফলাফল প্রিন্ট করেছি এবং স্কোপ শেষ করেছি

বাস, আমাদের লজিক বোঝা শেষ 😊। আরে আরেকটু দাঁড়ান, আর অল্প একটু বাকি আছে। উপরের কোডে আমরা দেখেছি মাত্র একটি কন্ডিশন চেক করতে। কিন্তু আমাদের যদি অনেকগুলো কন্ডিশন চেক করতে হয়, তাহলে? নিচের কোড ব্লকটি দেখুন।

```
01 <?php
02 //filename: /home/hasin/www/logic2.php
```

```
03 $planet = "Earth";
04 if ( $planet == "Mars")
05 {
06     echo "Welcome to Earth, Mr Green";
07 }
08 else if($planet == "Jupiter")
09 {
10     echo "Wow, I've never seen anyone from Jupiter before. How is that?";
11 }
12 else if($planet == "Saturn")
13 {
14     echo "I really love the belt around your planet";
15 }
16 else
17 {
18     echo "It's nice to meet you";
19 }
20 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/logic2.php
03 $planet = "Earth";
04 if ( $planet == "Mars")
05 {
06     echo "Welcome to Earth, Mr Green";
07 }
08 else if($planet == "Jupiter")
09 {
10     echo "Wow, I've never seen anyone from Jupiter before. How is that?";
11 }
12 else if($planet == "Saturn")
13 {
14     echo "I really love the belt around your planet";
15 }
16 else
17 {
18     echo "It's nice to meet you";
19 }
20 ?>
```

সহজ না? চলুন এবার আমরা দেখে নেই অন্যান্য লজিক্যাল অপারেটর গুলোর কি কাজ

> এটা চেক করে যে বাম পাশের অংশ ডান পাশের অংশের চেয়ে বড় কিনা

< এটা চেক করে যে বাম পাশের অংশ ডান পাশের অংশের চেয়ে ছোট কিনা

>= এটা চেক করে যে বাম পাশের অংশ ডান পাশের অংশের সমান বা বড় কিনা

<= এটা চেক করে যে বাম পাশের অংশ ডান পাশের অংশের সমান বা ছোট কিনা

!= এটা চেক করে যে বাম পাশের অংশ এবং ডান পাশের অংশ অসমান কিনা

== এটা চেক করে যে বাম পাশের অংশ ডান পাশের অংশের সমান কিনা

এই হল মোটামুটি পিএইচপিতে লজিকের সাতকাহন 😊 পরের আর্টিকলে দেখাব কিভাবে switch-case ব্যবহার করতে হয়

লজিক, switch-case দিয়ে

এর আগের পর্বে আমরা দেখেছি কিভাবে if-elseif-else দিয়ে লজিকাল ব্লক লিখতে হয়। এই পর্বে চলুন দেখি সুইচ-কেস এর ব্যবহার। আমরা আগের জোড়-বেজোড় বের করার স্ক্রিপ্টটুকুই আবার নতুন করে switch-case দিয়ে লিখব।

```
01 <?php
02 //filename: /home/hasin/www/logic3.php
03 $number = 11;
04 switch ($number%2)
05 {
06     case 0:
07         echo "{$number} is even number";
08         break;
09     default:
10         echo "{$number} is odd number";
11 }
12 ?>
```

01 <?php

02 //filename: /home/hasin/www/logic3.php

03 \$number = 11;

04 switch (\$number%2)

05 {

06 case 0:

07 echo "{\$number} is even number";

08 break;

09 default:

10 echo "{\$number} is odd number";

11 }

12 ?>

এই কোডটুকু রান করলে আমরা দেখব যে আউটপুট আসবে “11 is odd number” । কিভাবে কাজ করছে তা বোঝার জন্য চলুন কোডে একবার নজর দেই

লাইন # ৪ এ আমরা সুইচ কেস শুরু করেছি। ইফ-এলস ব্লকের মত এখানেও লজিকাল এক্সপ্রেশন বা কন্ডিশন রাখা হয়েছে ফার্স্ট ব্র্যাকেটের মধ্যে। আর সুইচ লেখার পরেই স্কোপ শুরু করতে হয় সেকেন্ড ব্র্যাকেট দিয়ে (লাইন # ৫) এবং স্কোপ শেষ করতে হয় আবার সেকেন্ড ব্র্যাকেট দিয়ে (লাইন # ১১)

সুইচ কেসে প্রতিটি কন্ডিশন চেক করতে হয় case লিখে, এবং কন্ডিশনের শেষে কোলন দিয়ে। লাইন # ৬ এ আমরা চেক করেছি যে ভাগফলের অবশিষ্টের মান ০ কিনা।

লাইন # ৭ এ আমরা আউটপুট প্রিন্ট করেছি

লাইন # ৮ এ আমরা পিএইচপি কে বলছি যে যদি এই কেস টি সঠিক হয়, তাহলে আর কিছু চেক করার দরকার নাই, সুইচ ব্লক থেকে বের হয়ে যাও। আমরা যদি প্রতিটি কেসে break না লিখি তাহলে পিএইচপি একটি কেস সঠিক হওয়ার পরেও পরবর্তী কেস চেক করবে। তা না করার জন্যই আমরা break লিখেছি।

if ব্লকে যেমন কোন কন্ডিশন না মিললে else ব্লক রান করে, switch-case এ default ঠিক একই কাজ করে। ডিফল্ট কেসে break লেখার কোন দরকার নেই।

চলুন আমরা একটু কমপ্লেক্স একটা উদাহরণ দেখি সুইচ কেসের। নিচের স্ক্রিপ্টে আমরা কোন একটি বছর লিপ-ইয়ার কিনা তা চেক করা হবে। তার আগে জেনে নিন লিপ ইয়ার ক্যালকুলেট করার জন্য একটি অ্যালগোরিদম আছে। আর তা হল নিচের মত যদি কোন বছর ৪০০ দিয়ে ভাগ করলে অবশিষ্ট ০ হয়, তাহলে লিপ ইয়ার নয়ত যদি ১০০ দিয়ে ভাগ করলে অবশিষ্ট ০ হয়, তাহলে নন-লিপ-ইয়ার নয়ত যদি ৪ দিয়ে ভাগ করলে অবশিষ্ট ০ হয়, তাহলে লিপ ইয়ার নয়ত নন লিপ ইয়ার

তো চলুন, এবার আমরা পিএইচপিতে এটা লিখে ফেলি

```
01 <?php
```

```
02 //filename: /home/hasin/www/logic4.php
```

```
03 $year =  
2004;
```

```
04 switch($year)
```

```
05 {
```

```
06 case ($year % 400 == 0):
```

```
07     echo "{$year} is a leap Year";
```

```
08     break;
```

```
09 case ($year % 100 == 0):
```

```
10     echo "{$year} is not a leap Year";
```

```
11     break;
```

```
12 case ($year % 4 == 0):
```



```
13     echo "{$year} is a leap Year";
14     break;
15     default:
16     echo "{$year} is not a leap Year";
17 }
18 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/logic4.php
03 $year = 2004;
04 switch($year)
05 {
06     case ($year % 400 == 0):
07         echo "{$year} is a leap Year";
08         break;
09     case ($year % 100 == 0):
10         echo "{$year} is not a leap Year";
11         break;
12     case ($year % 4 == 0):
13         echo "{$year} is a leap Year";
14         break;
15     default:
16         echo "{$year} is not a leap Year";
17 }
18 ?>
```

একদম সহজ, তাই না? পরবর্তী আর্টিকলে আমরা দেখব কিভাবে পিএইচপিতে লুপ ব্যবহার করতে হয়।

লুপ, while এবং do-while দিয়ে

কোডের কোন একটি অংশ বার বার রান করানোর জন্য আমরা লুপ ব্যবহার করি। পিএইচপি তে বেশ কয়েকভাবে লুপ তৈরী করা যায়, যেমন while, do-while, for এবং foreach। এদের মাঝে foreach টা অ্যারের সাথে ব্যবহৃত হয়। বাকিগুলো যে যেভাবে ব্যবহার করে। চলুন আমরা দেখি পিএইচপিতে কিভাবে লুপ তৈরী করা হয়। এই পোস্টে আমরা এবং while এর do-while এর উদাহরণ দেখব। পরের পোস্টে দেখব for এর। প্রথমেই চলুন while দিয়ে একটি উদাহরণ দেখি

```
01 <?php
02 //filename: /home/hasin/www/loop1.php
03 $i=0;
04 while ($i<10)
05 {
06     echo $i."<br/>";
07     if($i>7) break;
```

```
08 $i = $i+1;
```

```
09 }
```

```
10 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/loop1.php
03 $i=0;
04 while ($i<10)
05 {
06     echo $i."<br/>";
07     if($i>7) break;
08     $i = $i+1;
09 }
10 ?>
```

আমরা যদি কোড দেখি, তাহলে দেখব যে এখানে while ব্লক শুরু হয়েছে লাইন নম্বর ৪ থেকে। while ব্লক কতক্ষন চলবে সেই কন্ডিশন বা এক্সপ্রেশন আমরা দিয়ে দিয়েছি ফাস্ট ব্র্যাকেটের মাঝে, যে যতক্ষন \$i এর মান ১০ এর নিচে থাকবে ততক্ষন আমাদের while লুপ রান করবে।

লাইন # ৫ এ আমরা লুপের স্কোপ শুরু করেছি সেকেন্ড ব্র্যাকেট দিয়ে যা শেষ হয়েছে লাইন # ৯ এ

লাইন # ৬ এ আমরা \$i এর মান প্রিন্ট করেছি

লাইন # ৭ এ আমরা চেক করেছি যে \$i এর মান ৭ এর উপরে গেছে কিনা, যদি গিয়ে থাকে তাহলে আমরা পিএইচপি কে বলছি যে লুপ থেকে বের হয়ে যাও। লুপের কোন অংশে break রান করলে পিএইচপি সেই লুপ থেকে বের হয়ে যায়।

লাইন # ৮ এ আমরা \$i এর মান ১ করে বাড়িয়েছি। এই লাইন টিকে নিচের মত করেও লেখা যেত

```
1 $i+=1;
```

অথবা

```
1 $i++;
```

আমরা যদি কোডটুকু রান করাই তাহলে আমরা স্ক্রিনে ০ থেকে ৮ পর্যন্ত লেখা দেখতে পাব।

একই ভাবে আমরা do-while লুপ লিখতে পারি, যেমন নিচের উদাহরণটি দেখুন

```
1 <?php
```

```
2 //filename: /home/hasin/www/loop2.php
```

```
3 $i=0;
```

```
4 do{
```

```
5     echo $i."<br/>";
```

```
6     if ($i > 7) break;
```

```
7     $i++;
```

```
8 } while($i < 10);
```

9 ?>

```
1 <?php
2 //filename: /home/hasin/www/loop2.php
3 $i=0;
4 do{
5     echo $i."<br />";
6     if ($i > 7) break;
7     $i++;
8 } while($i < 10);
9 ?>
```

এখানে লুপ শুরু করতে হয় do দিয়ে এবং শেষ করতে হয় while দিয়ে, আর কন্ডিশনটি থাকে while এর সাথেই। বাকি সব তো একই রকম।

লুপ, for দিয়ে

আমাদের আগের আর্টিকলে আমরা বলেছিলাম যে পিএইচপিতে সাধারণভাবে চার রকমের লুপ আছে এবং আমরা while ও do-while নিয়ে আলোচনা করেছিলাম। এই আর্টিকলে আমরা দেখব কিভাবে for দিয়ে লুপ তৈরী করা যায়।

```
1 <?php
```

```
2 //filename: /home/hasin/www/loop3.php
```

```
3 for($i=0;$i<10;$i++)
```

```
4 {
```

```
5     echo
6     $i."<br>";
```

```
6     if ($i>7) break;
```

```
7 }
```

```
8 ?>
```

```
1 <?php
2 //filename: /home/hasin/www/loop3.php
3 for($i=0;$i<10;$i++)
4 {
5     echo $i."<br>";
6     if ($i>7) break;
7 }
8 ?>
```

এই হল ফর লুপের উদাহরণ। চলুন একটু কোড বোঝা যাক

লাইন # ৩ এ আমরা আমাদের for লুপ শুরু করেছি। এখানে একটা জিনিস খেয়াল করুন, for লুপের কন্ডিশনে তিনটি অংশ আছে। একটা হল

$\$i=0$; এটা দিয়ে আমরা একটি ভ্যারিয়েবলের মান ঠিক করে দিয়েছি

$\$i<10$; এটা দিয়ে আমরা কন্ডিশন ঠিক করেছি, যে যতক্ষণ $\$i$ এর মান ১০ এর নিচে থাকবে ততক্ষণ পর্যন্ত আমাদের for লুপ

কাজ করবে।

`$i++` এটা দিয়ে আমরা ভ্যারিয়েবলের মান এক এক করে বাড়িয়েছি

লাইন # ৪ এ আমরা `for` লুপের স্কোপ শুরু করেছি সেকেন্ড ব্র্যাকেট দিয়ে যা শেষ হয়েছে লাইন # ৭ এ

লাইন # ৫ এ আমরা `$i` এর মান প্রিন্ট করেছি

লাইন # ৬ এ আমরা চেক করেছি যে `$i` এর মান ৭ এর বেশি কিনা। বেশী হলে আমরা `break` লিখে পিএইচপিকে বলছি `for` লুপ থেকে বের হয়ে যেতে।

চলুন একটা মজার কোড লেখা যাক। আমরা নিচের কোড টুকু দিয়ে দেখব যে একটি সংখ্যা মৌলিক সংখ্যা (prime number) কিনা। তার আগে জেনে নেই যে প্রাইম নাম্বার বের করার জন্য অনেক জটিল জটিল অ্যালগরিদম রয়েছে, কিন্তু সবচেয়ে সহজটি হল যে সংখ্যা আমরা চেক করব তার স্কয়ার রুট বের করব। এবার ২ থেকে স্কয়ার রুট পর্যন্ত একটি লুপ চালিয়ে আমরা দেখব যে আমাদের সংখ্যাটি এই সংখ্যাগুলোর কোনোটা দিয়ে ভাগ যায় কিনা (অর্থাৎ ভাগফলের অবশিষ্ট ০ কিনা) – যদি ০ হয়, তাহলে এটা প্রাইম নাম্বার নয়। চলুন দেখা যাক কোডে কিভাবে এটা লেখা যায়

```
01 <?php
```

```
02 //filename: /home/hasin/www/primenumber.php
```

```
03 $number = 29;
```

```
04 $isPrime = true;
```

```
05 $sqrt = floor(sqrt($number));
```

```
06 for ($i=2;$i<=$sqrt;$i++)
```

```
07 {
```

```
08     if($number % $i == 0)
```

```
09     {
```

```
10         $isPrime = false;
```

```
11         break;
```

```
12     }
```

```
13 }
```

```
14
```

```
15 if ($isPrime)
```

```
16     echo "{$number} is a prime number";
```

```
17 else
```

```
18     echo "{$number} is not a prime number";
```

```
19 ?>
```

```
01 <?php
02 //filename: /home/hasin/www/primenumber.php
03 $number = 29;
04 $isPrime = true;
05 $sqrt = floor(sqrt($number));
06 for ($i=2;$i<=$sqrt;$i++)
07 {
08     if($number % $i == 0)
09     {
10         $isPrime = false;
11         break;
12     }
13 }
14
15 if ($isPrime)
16 echo "{$number} is a prime number";
17 else
18 echo "{$number} is not a prime number";
19 ?>
```

কিভাবে কি হল বোঝার জন্য চলুন একটু কোড দেখি।

লাইন # ৪ এ আমরা একটি ভ্যারিয়েবল \$isPrime এর মান true রেখেছি। কারন লুপ চলার সময় কোন সংখ্যা প্রাইম নাম্বার নয় এ ব্যাপারে নিশ্চিত হলে আমরা এই ভ্যারিয়েবলের মান false করে দিব।

লাইন # ৫ এ আমরা সংখ্যাটির স্কয়ার রুট বের করেছি sqrt ফাংশন দিয়ে। এটা একটা পিএইচপির একটা বিল্ট ইন ম্যাথ ফাংশন। এরপরে আমরা একই লাইনে এই স্কয়ার রুটের দশমিক অংশ বাদ দিয়েছি floor ফাংশন দিয়ে। এই ফাংশন গুলো সম্পর্কে বিস্তারিত জানার জন্য আমাদের ম্যাথ সংক্রান্ত আর্টিকেলটি পড়ুন বাকি অংশটুকুতে আর নতুন কিছু নেই। আমাদের আগের আর্টিকেল গুলো পড়ে থাকলে আপনি অনশ্যই বুঝতে পারবেন বাকি লাইন গুলোতে আমরা কি করেছি।

লজিক এবং লুপ

পূর্বের অধ্যায় গুলোতে আমরা যে কোড লিখেছিলাম তার গতি(flow) ছিল একমুখি। এখন আমরা দেখব কিভাবে প্রোগ্রামিং এ বিভিন্ন পরিস্থিতিতে সিদ্ধান্ত নিতে হয় এবং প্রোগ্রামের গতি নিয়ন্ত্রণ করতে হয়। এছাড়াও দেখব কিভাবে লুপ ব্যবহার করে একটা কোড ব্লক নির্দিষ্ট সংখ্যক বার কল(call) করে অনেক বিশাল কোড সংক্ষিপ্ত আকারে লেখা যায়। মূলত এখান থেকেই আমরা প্রোগ্রামিং এর ডাইনামিক জগতে প্রবেশ করতে যাচ্ছি।

লজিক এবং প্রোগ্রামের গতি নিয়ন্ত্রন(Logic and Flow controle)

প্রোগ্রামিং এ প্রায়সময়ই কোন ভেরিয়েবল এর মান বা কোন এক্সপ্রেশন এর ফলাফল এর উপর নির্ভর করে অউটপুট পরিবর্তন করতে হয়।

উদাহরন হিসেবে ধরি, \$totalMarks নামে একটি ভেরিয়েবলে একজন ছাত্রের মার্কস নেওয়া হল। এখন \$totalMarks এর value অনুযায়ী তার রেজাল্ট দেখাতে হবে। নিচে এই কাজটি করতে আমরা বিভিন্ন কন্ট্রোল স্টেটমেন্ট ব্যবহার করা শিখব।

If statement

একটি কোড ব্লক এক্সিকিউট হবে কিনা তা if স্টেটমেন্ট দিয়ে নিয়ন্ত্রন করা যায়। if স্টেটমেন্ট লেখার সময় if keyword এর পর প্রথম বন্ধনীর ‘()’ মধ্যে এক্সপ্রেশন এবং দ্বিতীয় বন্ধনীর ‘{ }’ মধ্যে কোড ব্লক লিখতে হয়। এক্সপ্রেশন এর মান true হলে দ্বিতীয় বন্ধনীর মধ্যকার কোড ব্লক এক্সিকিউট হবে, নয়তো হবে না। যেমনঃ

if (expression)

```
{  
    // এক বা একাধিক কোড লাইন যা expression এর মান true হলে এক্সিকিউট হবে  
}
```

এখানে এক্সপ্রেশন হিসেবে অপারেটর এর সাহায্যে একাধিক ভেরিয়েবল এর তুলনা করা যায় অথবা একটি ভেরিয়েবল ব্যবহার করা যায়। যদি একটি ভেরিয়েবল ব্যবহার করা হয় তাহলে উক্ত ভেরিয়েবল এর মান 0, false, ""(ফাকা স্ট্রিং), "0"(স্ট্রিং এর মধ্যে শুধুমাত্র শূন্য) অথবা array() (ফাকা এর)হলে কোড ব্লক এক্সিকিউট হবে না। অন্যথায় ভেরিয়েবল এর মান true হবে এবং কোড ব্লক এক্সিকিউট হবে।

এখন মনেকরি, আমাদের উদাহরণ এর \$totalMarks এর মান 32 এর বেশি হলে "Result : Passed" প্রিন্ট করতে হবে। if স্টেটমেন্ট এর সাহায্যে কাজটি নিম্নলিখিত ভাবে করা যায়ঃ

```
if ($totalMarks > 32)  
{  
    echo "Result : Passed";  
}
```

কোড ব্লক শুধুমাত্র ১ লাইন এর হলে দ্বিতীয় বন্ধনি না লিখলেও চলে। যেমন উপরের কোডটি এভাবেও লেখা যায়ঃ

```
if ($totalMarks > 32)  
    echo "Result : Passed";
```

If এর সাথে else এর ব্যবহার

If এর সাথে else ব্যবহার করে আমরা একটি বিকল্প কোড ব্লক লিখতে পারি যা এক্সপ্রেশন এর মান true না হলে এক্সিকিউট হবে। এক্ষেত্রে if এর কোড ব্লক শেষ করে else লিখতে হবে অতঃপর দ্বিতীয় বন্ধনির '{}' মধ্যে বিকল্প কোড ব্লক লিখতে হবে। if else লেখার নিয়ম(syntax) নিম্নরূপঃ

```
if (expression)  
{  
    // এক বা একাধিক কোড লাইন যা expression এর মান true হলে এক্সিকিউট হবে  
}  
else  
{  
    // এক বা একাধিক কোড লাইন যা অন্যসব ক্ষেত্রে এক্সিকিউট হবে  
}
```

যেমন, ধরি উপরের উদাহরণ এ \$totalMarks এর মান ৩২ এর বেশি না হলে "Result : Failed" প্রিন্ট করতে হবে। if else স্টেটমেন্ট এর সাহায্যে কাজটি এভাবে করা যায়ঃ

```
$totalMarks = 87;
```

```
if ($totalMarks > 32)
{
    echo "Result : Passed";
}
else
{
    echo "Result : Failed";
}
```

উপরের কোড এ \$totalMarks এর value 87। অতএব \$totalMarks > 32 এক্সপ্রেশনটি true হবে এবং Result : Passed প্রিন্ট করবে। কিন্তু \$totalMarks এর value যদি 15 করে দিই, তাহলে \$totalMarks > 32 এক্সপ্রেশনটি false হবে। ফলে Result : Failed প্রিন্ট করবে।

একাধিক এক্সপ্রেশন এর জন্য *else if* এর ব্যবহার

যদি একাধিক এক্সপ্রেশন যাচাই করতে চাই অর্থাৎ প্রথম *if* স্টেটমেন্ট এ ব্যবহৃত এক্সপ্রেশন true না হলে *else* ব্লকে ঢোকার আগে আরও কিছু *if* স্টেটমেন্ট ব্যবহার করতে চাই তাহলে *else if* ব্যবহার করতে হবে। *else if* কে একসাথে *elseif* ও লেখা যায়। *elseif* ব্যবহারের নিয়ম নিম্নরূপঃ

```
if (expression)
{
    // কোড ব্লক যা expression এর মান true হলে এক্সিকিউট হবে
}
elseif (another expression)
{
    // কোড ব্লক যা উপরের কোন expression true না হলে
    // এবং এই ব্লকের এক্সপ্রেশন(another expression) true হলে এক্সিকিউট হবে
}
// এভাবে আরও elseif ব্লক থাকতে পারে
else
{
    // কোড ব্লক যা উপরের কোনটিই true না হলে এক্সিকিউট হবে
}
```

উপরের কোডে প্রথমে *if* এর এক্সপ্রেশন যাচাই করে true পাওয়া না গেলে পরবর্তী *elseif* এর এক্সপ্রেশন যাচাই করা হবে। তাও true না হলে তার পরের *elseif* এর এক্সপ্রেশন(যদি থাকে) যাচাই করা হবে। অর্থাৎ উপর থেকে শুরু করে প্রথম যে এক্সপ্রেশন এর মান true পাওয়া যাবে শুধুমাত্র সেই কোড ব্লক এক্সিকিউট হবে। আর কোনটিই true না হলে *else* ব্লক এক্সিকিউট হবে। অবশ্য *else* ব্লক থাকাটা বাধ্যতামূলক নয়।

আমাদের উপরের উদাহরণের ছাত্রের রেজাল্ট যদি গ্রেডিং পদ্ধতিতে দেখাতে যাই, তাহলে নিম্নলিখিত শর্তগুলো যাচাই করে দেখতে হবে-

- \$totalMarks এর value 80-100 হলে রেজাল্ট A+
- \$totalMarks এর value 70-79 হলে রেজাল্ট A
- \$totalMarks এর value 60-69 হলে রেজাল্ট A-
- \$totalMarks এর value 50-59 হলে রেজাল্ট B
- \$totalMarks এর value 40-49 হলে রেজাল্ট C
- \$totalMarks এর value 33-39 হলে রেজাল্ট D
- \$totalMarks এর value 0-32 হলে রেজাল্ট F

elseif ব্যবহার করে আমরা নিম্নলিখিতভাবে সবগুলো শর্ত যাচাই করে রেজাল্ট দেখাতে পারি-

```
$totalMarks = 56;
```

```
if ($totalMarks >= 80 AND $totalMarks <= 100)
```

```
{  
    echo "Result : A Grade";  
}
```

```
elseif($totalMarks >= 70 )
```

```
{  
    echo "Result : A Grade";  
}
```

```
elseif($totalMarks >= 60 )
```

```
{  
    echo "Result : A- Grade";  
}
```

```
elseif($totalMarks >= 50 )
```

```
{  
    echo "Result : B Grade";  
}
```

```
elseif($totalMarks >= 40 )
```

```
{  
    echo "Result : C Grade";  
}
```

```
elseif($totalMarks >= 33 )
```

```
{  
    echo "Result : D Grade";  
}
```

```
else
```



```
{  
    echo "Result : F Grade";  
}
```

এখানে \$totalMarks এর value 56. তাই পর্যায়ক্রমে ১ম(if এ ব্যবহৃত), ২য় এবং ৩য় এক্সপ্রেশন এর মান false হবে। এরপর ৪র্থ এক্সপ্রেশন (\$totalMarks >= 50) এর মান true হবে এবং Result : B Grade প্রিন্ট করবে। \$totalMarks এর মান পরিবর্তন করলে সে অনুযায়ী ভিন্ন ভিন্ন কোড ব্লক এক্সিকিউট হবে ও সঠিক গ্রেড রেজাল্ট দেখাবে।

Switch case

? অপারেটর এর ব্যবহার

লুপ

While

Do while

For

break and continue

nested loops

foreach loop

অ্যারে [Array]

ভ্যারিয়েবল নিয়ে আলোচনার একদম শুরুতেই আমরা বলেছিলাম যে পিএইচপিতে (শুধু পিএইচপি না, প্রায় সব প্রোগ্রামিং ল্যাংগুয়েজেই) অ্যারে নামে একটি কালেকশন রয়েছে, যার কাজ হচ্ছে একগুচ্ছ ডাটা স্টোর করা। একটি ভ্যারিয়েবল যেমন একক ডেটা স্টোর করে, অ্যারে কে আপনি ভাবে পারেন একটি অ্যারে আসলে অনেকগুলো ভ্যারিয়েবলের সমষ্টি (সহজ কথায়)। পিএইচপি তে অ্যারে খুবই গুরুত্বপূর্ণ এবং অ্যারে সংক্রান্ত অনেক মজার মজার ফাংশন রয়েছে পিএইচপিতে। এই পোস্টে চলুন আমরা পরিচিত হই অ্যারের সাথে। পিএইচপিতে অ্যারে ডিক্লেয়ার করা লাগে array() ফাংশন দিয়ে

```
1 <?php
```

```
2 //filename: /home/hasin/www/array1.php
```

```
3 $MyArray = array(11,12,13,14,15);
```

```
4 print_r($MyArray);
```

```
5 ?>
```

```
1 <?php
2 //filename: /home/hasin/www/array1.php
3 $MyArray = array(11,12,13,14,15);
4 print_r($MyArray);
5 ?>
```

উপরের কোডটুকু রান করলে আমরা দেখব স্ক্রিনে পাঁচটি এলিমেন্ট সহ এই অ্যারে টি প্রিন্ট হয়েছে। লাইন # ৩ এ array() ফাংশন দিয়ে আমরা \$MyArray নামে একটি অ্যারে তৈরী করেছি যাতে আমরা 11,12,13,14 এবং 15 এই পাঁচটি এলিমেন্ট রেখেছি। এখন অ্যারে নিয়ে আরো কথা বলার আগে দুটো জিনিস সম্বন্ধে ধারণা পরিষ্কার করা দরকার। পিএইচপিতে অন্য প্রোগ্রামিং ল্যাংগুয়েজের মত অ্যারের সাইজ (অর্থাৎ অ্যারে তে সর্বোচ্চ কয়টি এলিমেন্ট থাকবে) তা বলতে হয় না। এটা ডাইনামিক ভাবে বাড়তে থাকে। আপনি যত ইচ্ছা এলিমেন্ট রাখতে পারবেন। এলিমেন্টগুলোর পজিশনিং শুরু হয় 0 থেকে। আমাদের এই অ্যারেতে প্রথম এলিমেন্ট 11 এর পজিশন অ্যারের একদম শুরুতে, অর্থাৎ 0 তে, এবং এর পরের এলিমেন্ট 12 এর পজিশন 1 এ। এই পজিশনকে প্রোগ্রামিং এর ভাষায় বলে অফসেট (Offset), অর্থাৎ 11 এর অফসেট 0 এবং 12 এর অফসেট 1। আমরা যেকোন সময় অ্যারের যেকোন অফসেটে যেকোন এলিমেন্টের মান পরিবর্তন করতে পারি। নিচের কোডটি দেখুন কিভাবে যেকোন অফসেটে কোন এলিমেন্টের মান পরিবর্তন করা যায়

```
1 <?php
2 //filename: /home/hasin/www/array2.php
3 $MyArray = array(11,12,13,14,15);
4 for($i=0;$i<5;$i++)
5 $MyArray[$i] = $MyArray[$i]*$MyArray[$i];
6 print_r($MyArray);
7 ?>
```

```
1 <?php
2 //filename: /home/hasin/www/array2.php
3 $MyArray = array(11,12,13,14,15);
4 for($i=0;$i<5;$i++)
5 $MyArray[$i] = $MyArray[$i]*$MyArray[$i];
6 print_r($MyArray);
7 ?>
```

অ্যারের যেকোন অফসেটের এলিমেন্ট এর মান জানা বা পরিবর্তন করার জন্য অ্যারের সাথে থার্ড ব্র্যাকেটের মাঝে সেই অফসেট বলে দিতে হয়। যেমন উপরের উদাহরণে আমরা ০ থেকে ৪ পর্যন্ত অফসেটের এলিমেন্টগুলোর মান তাদের বর্তমান মান দিয়ে গুন করে পরিবর্তন করেছি (লাইন # ৪)। পিএইচপিতে অ্যারে নিয়ে কাজ করার সময় আরো একটা জিনিস মনে রাখা দরকার। অ্যারের অফসেটগুলো যদি নাম্বার হয় (যেমন 0,1,2,3,4,.....n) তাহলে আমরা সেই অ্যারেকে বলব ইনডেক্সড অ্যারে। আর যদি অফসেট হয় স্ট্রিং (তাও হয় নাকি? হ্যাঁ হয়, নিচের উদাহরণটি দেখুন) তাহলে সেই অ্যারেকে বলব অ্যাসোসিয়েটেড অ্যারে। চলুন অ্যাসোসিয়েটেড অ্যারের একটা উদাহরণ দেখা যাক

```
1 <?php
```

```
2 //filename: /home/hasin/www/array3.php
```

```
3 $MyArray = array("blue"=>"berry","red"=>"tomato","green"=>"pear","yellow"=>"banana");
```

```
4 print_r($MyArray);
```

```
5 ?>
```

```
1 <?php
2 //filename: /home/hasin/www/array3.php
3 $MyArray = array("blue"=>"berry","red"=>"tomato","green"=>"pear","yellow"=>"banana");
4 print_r($MyArray);
5 ?>
```

অ্যাসোসিয়েটেড

অ্যারের মাঝে লুপ চালানোর সময় for বা while বা do-while ব্যবহার করা যায় না। এক্ষেত্রে foreach ব্যবহার করতে হয়।

চলুন একটা উদাহরণ দেখি কিভাবে আমরা উপরের অ্যাসোসিয়েটেড অ্যারেতে লুপ চালাতে পারি।

```
1 <?php
```

```
2 //filename: /home/hasin/www/array4.php
```

```
3 $MyArray = array("blue"=>"berry","red"=>"tomato","green"=>"pear","yellow"=>"banana");
```

```
4 foreach($MyArray as $key=>$value)
```

```
5 {
```

```
6     echo "{$value} is {$key} <br>";
```

```
7 }
```

```
8 ?>
```

```
1 <?php
2 //filename: /home/hasin/www/array4.php
3 $MyArray = array("blue"=>"berry","red"=>"tomato","green"=>"pear","yellow"=>"banana");
4 foreach($MyArray as $key=>$value)
5 {
6     echo "{$value} is {$key} <br>";
7 }
8 ?>
```

এই কোডটুকু

রান করলে আমরা দেখবে নিচের মত আউটপুট এসেছে

```
1 berry is blue
```

```
2 tomato is red
```

```
3 pear is green
```

```
4 banana is yellow
```

foreach লুপে \$key তে অ্যারের এলিমেন্টের অফসেট এবং \$value তে এলিমেন্টের মান স্টোর হয়।

এর পরের পোস্টে আমরা দেখব অ্যারে নিয়ে আরো বেশ কিছু বিষয় আলোচনা করব, সেইসাথে দেখব অ্যারে সংক্রান্ত বহুল ব্যবহৃত ফাংশনগুলোর ব্যবহার।

পি.এইচ.পি. Array Library functions

ফাংশনের নাম : sizeof()
ফাংশন লিখার নিয়ম : sizeof(\$array_name);
ফাংশন এর output : size of the array

ফাংশনের কাজ

এই ফাংশনটি সাধারণত, একটি array এর মাঝে কতগুলো value আছে তা output আকারে দেখায়। এই ফাংশনটি বেশীরভাগ ক্ষেত্রে loop এর counter হিসাবে ব্যবহার করা হয়।

উদাহরণঃ

Code:

```
<?php
    $data = array("red", "green", "blue");
    echo "Array has " . sizeof($data) . " elements";
?>
```

Output:

Array has 3 elements

ফাংশনের নাম : array_values()
ফাংশন লিখার নিয়ম : array_values(\$array_name);
ফাংশন এর output : Array

ফাংশনের কাজ

এই ফাংশনটি সাধারণত, একটি array এর মাঝে কতগুলো value আছে তা দিয়ে নতুন একটি array গঠন করে output আকারে দেখায়। এই ফাংশনটি বেশীরভাগ ক্ষেত্রে associative array থেকে তার value গুলো নিয়ে আরেকটি array গঠন করার জন্য ব্যবহার করা হয়।

উদাহরণঃ

Code:

```
<?php
    $data = array("hero" => "Holmes", "villain" => "Moriarty");
    print_r(array_values($data));
?>
```

Output:

```
Array
(
    [0] => Holmes
    [1] => Moriarty
)
```

ফাংশনের নাম : array_keys()
ফাংশন লিখার নিয়ম : array_keys(\$array_name);
ফাংশন এর output : Array

ফাংশনের কাজ

এই ফাংশনটি সাধারণত, একটি array এর মাঝে কতগুলো key আছে তা দিয়ে নতুন একটি array গঠন করে output আকারে দেখায়। এই ফাংশনটি বেশীরভাগ ক্ষেত্রে associative array থেকে তার key গুলো নিয়ে আরেকটি array গঠন করার জন্য ব্যবহার করা হয়।

উদাহরণঃ

Code:

```
<?php
    $data = array("hero" => "Holmes", "villain" => "Moriarty");
    print_r(array_keys($data));
?>
```

Output:

```
Array
(
    [0] => hero
```

```
[1] => villain
)
```

ফাংশনের নাম : array_pop()
ফাংশন লিখার নিয়ম : array_pop(\$array_name);
ফাংশন এর output : একটি value মুছে দেয়ার পর যে array টি থাকে তাই array আকারে পাঠানো হয়।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array থেকে তার শেষ element টি মুছে দিতে পারি। আমরা যদি, এর আগে stack পরে থাকি, তাহলে এই ফাংশনটি আরও ভাল করে বুঝা যাবে।

উদাহরনঃ

```
Code:
<?php
    $data = array("Donald", "Jim", "Tom");
    array_pop($data);
    print_r($data);
?>

Output:
Array
(
    [0] => Donald
    [1] => Jim
)
```

ফাংশনের নাম : array_push()
ফাংশন লিখার নিয়ম : array_push(\$array_name, \$value);

ফাংশন এর output : একটি value সংযোজন করার পর যে array টি থাকে তাই array আকারে পাঠানো হয়।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর সবশেষে আমরা যেকোন value সংযোজন করতে পারি। যদি, এর আগে stack পরে থাকি, তাহলে এই ফাংশনটি আরও ভাল করে বুঝা যাবে।

উদাহরণঃ

Code:

```
<?php
    $data = array("Donald", "Jim", "Tom");
    array_push($data, "Harry");
    print_r($data);
?>
```

Output:

```
Array
(
    [0] => Donald
    [1] => Jim
    [2] => Tom
    [3] => Harry
)
```

ফাংশনের নাম : array_shift()

ফাংশন লিখার নিয়ম : array_shift(\$array_name);

ফাংশন এর output : একটি value সরানোর পর যে array টি থাকে তাই array আকারে পাঠানো হয়।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর প্রথম থেকে যেকোন value মুছতে পারব। তার মানে হল, array এর প্রথম index এ যেকোন data type এর value থাকলেও আমরা তা অতি সহজে array_shift() ফাংশনের মাধ্যমে মুছে ফেলতে পারব।

উদাহরণঃ

Code:

```
<?php
    $data = array("Donald", "Jim", "Tom");
    array_shift($data);
    print_r($data);
?>
```

Output:

```
Array
(
    [0] => Jim
    [1] => Tom
)
```

ফাংশনের নাম : array_unshift()

ফাংশন লিখার নিয়ম : array_unshift(\$array_name, \$value);

ফাংশন এর output : একটি value সংযোজন করার পর যে array টি থাকে তাই array আকারে পাঠানো হয়।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর প্রথমে যেকোন value সংযোজন করতে পারব।

উদাহরণঃ

Code:

```
<?php
    $data = array("Donald", "Jim", "Tom");
    array_unshift($data, "Sarah");
    print_r($data);
?>
```

Output:

```
Array
(
    [0] => Sarah
    [1] => Donald
)
```



```
[2] => Jim
[3] => Tom
)
```

বিঃদ্রঃ

array_shift এবং array_unshift () প্রায় উলটা array_pop() এবং array_push() ফাংশনের।

ফাংশনের নাম : each()
ফাংশন লিখার নিয়ম : each (\$array_name);
ফাংশন এর output : ব্যবহারকারীর উপর নির্ভরশীল।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর প্রতিটি key এবং value একটা নির্দিষ্ট সময় পর্যন্ত চক্রাকারে পড়তে পারি। প্রতিটি চক্রে আমরা key এবং value কে জোড়া আকারে পাই, তারপর তা পরবর্তী key এবং value জোড়াকে নির্দেশ করে।

উদাহরণঃ

```
Code:
<?php
$data = array("hero" => "Holmes", "villain" => "Moriarty");
while (list($key, $value) = each($data))
{
    echo "$key: $value \n";
}
?>

Output:
hero: Holmes
villain: Moriarty
```

ফাংশনের নাম : sort()
ফাংশন লিখার নিয়ম : sort (\$array_name);

ফাংশন এর output : sorted array(ছকাকারে সাজানো array)

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর value গুলোকে ছকাকারে সাজাতে পারি। ছকাকারে সাজানো হল, ইংরেজী অক্ষর এর ক্রমানুসারে, বাংলা/ইংরেজী সংখ্যার ক্রমানুসারে সাজানো।

উদাহরণঃ

Code:

```
<?php
    $data = array("g", "t", "a", "s");
    sort($data);
    print_r($data);
?>
```

Output:

Array

```
(
    [0] => a
    [1] => g
    [2] => s
    [3] => t
)
```

এমন আরও কিছু sorting ফাংশন আছে, যেমনঃ asort(), ksort(), rsort(), krsort(), arsort(). এগুলো নিয়ে পরে বিস্তারিত আলোচনা করা হবে।

ফাংশনের নাম : array_flip()

ফাংশন লিখার নিয়ম : array_flip (\$array_name);

ফাংশন এর output : flipped array(পুনরায় সাজানো array)

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array এর value গুলোকে প্রদত্ত array এর key এবং key গুলোকে প্রদত্ত array এর value হিসাবে পরিবর্তন করতে পারি। সাধারণত tabular ডাটা, যেখানে সারি এবং কলাম থাকে array এর key ওvalue যথাক্রমে row ও column কে নির্দেশ করে, আমরা array_flip এর মাধ্যমে row কে column এ পরিবর্তন করতে পারি।

উদাহরণঃ

Code:

```
<?php
    $data = array("a" => "apple", "b" => "ball");
    print_r(array_flip($data));
?>
```

Output:

```
Array
(
    [apple] => a
    [ball] => b
)
```

ফাংশনের নাম : array_reverse()
ফাংশন লিখার নিয়ম : array_reverse (\$array_name);
ফাংশন এর output : reverse array(পুনরায় সাজানো array)

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array কে পুনরায় সাজাতে পারি, তার উল্টা ক্রমানুসারে। এটি হতে পারে, বাংলা/ইংরেজী অক্ষরের বা সংখ্যার উল্টা ক্রমানুসারে।

উদাহরণঃ

Code:

```
<?php
    $data = array(10, 20, 25, 60);
    print_r(array_reverse($data));
?>
```

Output:

```
Array
(
    [0] => 60
    [1] => 25
    [2] => 20
    [3] => 10
)
```



ফাংশনের নাম : array_merge()
ফাংশন লিখার নিয়ম : array_merge (\$arr_1, \$arr_2, ...);
ফাংশন এর output : merged array

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা এক বা একাধিক array কে একটি array তে রূপান্তর করতে পারি। এক্ষেত্রে সাধারণত value গুলো array এর ক্রমানুসারে বসে। আর যদি কিছু array হয় associative array, আর কিছু array যদি হয় numeric সেক্ষেত্রে প্রথম array এর key এর ক্রমানুসারে array index গুলো বসে।

আমরা সাধারণত database নিয়ে কাজ করার সময় দুটি query এর output কে একটি array তে পাওয়ার জন্য প্রদত্ত ফাংশনটি ব্যবহার করি।

উদাহরণঃ

Code:

```
<?php
    $data1 = array("cat", "goat");
    $data2 = array("dog", "cow");
    print_r(array_merge($data1, $data2));
?>
```

Output:

```
Array
(
    [0] => cat
    [1] => goat
    [2] => dog
    [3] => cow
)
```

ফাংশনের নাম : array_search()
ফাংশন লিখার নিয়ম : array_search (\$search_value, \$arr);

ফাংশন এর output : প্রদত্ত search value টি array এর যে অবস্থানে আছে সে key টি।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array থেকে আমরা যেকোন value search করতে পারি। সাধারণত এই ফাংশনটি search value টির key টিকে output আকারে দেখায়। যদি প্রদত্ত value একাধিকবার থাকে, তাহলে ফাংশনটি প্রথম key টিকে output আকারে দেখাবে।

উদাহরণঃ

Code:

```
<?php
$data = array("blue" => "#0000cc",
              "black" => "#000000",
              "green" => "#00ff00"
            );
echo "Found " . array_search("#0000cc", $data);
?>
```

Output:

Found blue

ফাংশনের নাম : array_unique()
ফাংশন লিখার নিয়ম : array_unique(\$arr);
ফাংশন এর output : array, যার মাঝে কোন duplicate value নাই।

ফাংশনের কাজ

এই ফাংশনটির মাধ্যমে আমরা একটি array থেকে সকল duplicate value কে বাদ দিয়ে একটি নতুন array পেতে পারি।

উদাহরণঃ

Code:

```
<?php
$data = array(1,1,4,6,7,4);
print_r(array_unique($data));
?>
```

Output:

Array

```
(  
  [0] => 1  
  [3] => 6  
  [4] => 7  
  [5] => 4  
)
```

ফাংশনের কাজ

এই ফাংশানটি array টে একটি কী(KEY)এর ভ্যালু (value) কতবার করে আছে সেটা গুনে একটি লিস্ট আকারে প্রকাশ করে। উদাহরণে দেখা যাচ্ছে, \$var_array নামে একটি অ্যারে আছে। অ্যারেটি একটু খেয়াল করলে দেখা যাবে, "world" কী(key) ভ্যালুটি ২বার আছে। ঠিক একি রকম ভাবে "1", "4" কী(key) ভ্যালুগুলোও ২বার করে আছে। অপরদিকে "5" আছে শুধু মাত্র একবার। এই অনুযায়ী keyগুলো কে গুনে ফাংশানটি একটি অ্যারে আউটপুট দেখায়।

উদাহরণঃ

Code:

```
<?php  
  
$var_array = array(1, "world", "world", 1, 5, 4, 4);  
echo '<pre>';  
print_r(array_count_values($var_array));  
echo '</pre>';  
  
?>
```

Output:

```
Array  
(
```

```
[1] => 2  
[world] => 2  
[5] => 1  
[4] => 2  
)
```

ফাংশনের নাম : **array_filter()**

ফাংশন লিখার নিয়ম : **array_filter(\$array_name);**

ফাংশন এর output :

ফাংশনের কাজ

এই ফাংশানটি বার বার ইনপুট অ্যারের প্রত্যেকটি ভ্যালুকে কলব্যাক ফাংশান(odd and even)এ পাঠায়। যদি ওই কলব্যাক ফাংশান TRUE ভ্যালু রিটার্ন করে তবে, অ্যারের বর্তমান ভ্যালুটি রেজাল্ট অ্যারেতে চলে যাবে। এখানে একটা ব্যাপার লক্ষণীয় যে, অ্যারের কী ভালু ঠিক আগের টাই থাকবে। নিচে একটা উদাহরন আমরা দেখতে পাচ্ছি যা কলব্যাক ফাংশন দিয়ে করা হয়েছে।

উদাহরনঃ

```
Code:  
<?php  
  
function odd($var)  
{  
    return($var & 1);  
}  
  
function even($var)  
{  
    return(!$var & 1);  
}  
  
$array1 = array("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5);  
$array2 = array(6, 7, 8, 9, 10, 11, 12);  
  
echo "Odd :\n";
```

```
echo '<pre>';
print_r(array_filter($array1, "odd"));
echo '</pre>';
echo "Even:\n";
echo '<pre>';
print_r(array_filter($array2, "even"));
echo '</pre>';
?>
```

Output:

Odd :

Array

```
(
 [a] => 1
 [c] => 3
 [e] => 5
)
```

Even:

Array

```
(
 [0] => 6
 [2] => 8
 [4] => 10
 [6] => 12
)
```

কলব্যাক ফাংশন কে কল না করেও অ্যারে ফাংশান কে ব্যবহার করা যায়। সেই ক্ষেত্রে array_filter() ফাংশানটি ইনপুট এর যেসব ভ্যালু FALSE সেই সব এন্ট্রিকে মুছে ফেলবে। নিচে এর ব্যবহারের একটি উদাহরন দেওয়া হলো।

উদাহরনঃ

Code:

```
<?php
```

```
$entry = array(
```



```
0 => 'foo',
1 => false,
2 => -1,
3 => null,
4 => ""
);
```

```
print_r(array_filter($entry));
```

```
?>
```

Output:

Array

```
(
  [0] => foo
  [2] => -1
)
```

কখন ভ্যালু অ্যারের রেজাল্ট true/false হচ্ছে, সেই বিষয়ে আরো বিশদ জানার জন্য বুলিয়ান চ্যাপ্টারটি আমরা দেখতে পারি।

পিএইচপির ম্যাথমেটিকাল ফাংশন

ম্যাথ বা গণিতের কথা ভাবলেই আমার মনে পড়ে যায় রাশভারী চেহারার কোন পন্ডিতের কথা। মাঝে মাঝে ভাবতাম যে লোকটা গণিতের সূত্রপাত করেছেন তাকে যদি একবার আমার সামনে পেতাম তাহলে তাকে হাতে কলমে বুঝিয়ে দিতাম গণিত কষার কত মজা 😊

কিন্তু পিএইচপি শেখার পর থেকে মনে হয় ম্যাথ বোধহয় অতটা খারাপ কিছু না 😊 পিএইচপি তে আমি পেয়েছি অসাধারণ কিছু ম্যাথমেটিকাল ফাংশন যেগুলো দিয়ে প্রায় সব ধরনের ম্যাথমেটিকাল প্রবলেম এর সমাধান করা সম্ভব খুব সহজেই 😊

আসুন তবে দেখা যাক পিএইচপির গণিতের দৌড় কতদূর 😊 পিএইচপিতে প্রায় ৫০ টি ম্যাথ ফাংশন আছে যেগুলো আমরা কয়েক পর্বে দেখব। প্রথম পর্বে আমরা দেখব abs(), ceil(), floor(), is_finite() এবং is_infinite() – এই ৫ টি ফাংশন। তাহলে চলুন শুরু করা যাক।

abs():

যে কোন সংখ্যার ধনাত্মক মান নির্ণয় করতে আমরা এই ফাংশনটি ব্যবহার করব। যেমনঃ

```
01 <?php
```

```
02 //filename: /home/masnun/phpbook/math-1-abs.php
```

```
03 $number1 = 11;
04 echo abs($number1); // outputs: 11
05 ?>
06
07 <?php
08 $number2 = -11;
09 echo abs($number2); // outputs: 11
10 ?>
```

ceil():

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-abs.php
03 $number1 = 11;
04 echo abs($number1); // outputs: 11
05 ?>
06
07 <?php
08 $number2 = -11;
09 echo abs($number2); // outputs: 11
10 ?>
```

ilr.

ভগ্নাংশের মান বৃদ্ধি করে পরবর্তী পূর্ণ সংখ্যায় পরিণত করে। যেমনঃ

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-ceil.php
03 $number1 = 3.4;
04 echo ceil($number1); // Outputs: 4
05 ?>
06
07 <?php
08 $number2 = -3.4;
09 echo ceil($number2); // Outputs: -3
10 ?>
```

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-ceil.php
03 $number1 = 3.4;
04 echo ceil($number1); // Outputs: 4
05 ?>
06
07 <?php
08 $number2 = -3.4;
09 echo ceil($number2); // Outputs: -3
10 ?>
```

লক্ষ্য করুন, ঋনাত্মক সংখ্যার ক্ষেত্রে -৩, -৩.৪ অপেক্ষা বৃহত্তর 😊

floor():

এটি ঠিক ceil() এর উল্টোটা করে। অর্থাৎ ভগ্নাংশের মান হ্রাস করে পূর্ববর্তী পূর্ণ সংখ্যায় পরিণত করে। যেমনঃ

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-floor.php
03 $number1 = 3.4;
04 echo floor($number1); // Outputs: 3
05 ?>
06
07 <?php
08 $number2 = -3.4;
09 echo floor($number2); // Outputs: -4
10 ?>
```

is_finite():

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-floor.php
03 $number1 = 3.4;
04 echo floor($number1); // Outputs: 3
05 ?>
06
07 <?php
08 $number2 = -3.4;
09 echo floor($number2); // Outputs: -4
10 ?>
```

এই ফাংশনটির রিটার্ন ভ্যালু একটি Boolean value। অর্থাৎ যদি প্রদত্ত সংখ্যাটি সসীম হয় তবে এটি True নচেৎ False রিটার্ন করে। সহজভাবে ব্যাখ্যা করার জন্য এখানে আমি “সসীম” শব্দটি ব্যবহার করলাম। মূলত, সংখ্যাটি “float” ডাটা টাইপের সীমার ভেতরে আছে কিনা সেটা যাচাই করাই এই ফাংশনটির কাজ। যেমনঃ

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-is_finite.php
03 $number1 = 30003;
04 var_dump( is_finite($number1) ); // Outputs: bool(true)
05 ?>
06
07 <?php
08 $number2 = log(0);
09 var_dump( is_finite($number2) ); // Outputs: bool(false)
10 ?>
```

is_infinite():

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-is_finite.php
03 $number1 = 30003;
04 var_dump( is_finite($number1) ); // Outputs: bool(true)
05 ?>
06
07 <?php
08 $number2 = log(0);
09 var_dump( is_finite($number2) ); // Outputs: bool(false)
10 ?>
```

আগের ফাংশনটির ঠিক বিপরীত এই ফাংশনটি। এটি নির্দেশ করে প্রদত্ত সংখ্যাটি অসীম কিনা। যেমনঃ

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-is_infinite.php
03 $number1 = 30003;
04 var_dump( is_infinite($number1) ); // Outputs: bool(false)
05 ?>
06
07 <?php
08 $number2 = log(0);
09 var_dump( is_infinite($number2) ); // Outputs: bool(true)
10 ?>
```

```
01 <?php
02 //filename: /home/masnun/phpbook/math-1-is_infinite.php
03 $number1 = 30003;
04 var_dump( is_infinite($number1) ); // Outputs: bool(false)
05 ?>
06
07 <?php
08 $number2 = log(0);
09 var_dump( is_infinite($number2) ); // Outputs: bool(true)
10 ?>
```

বিদ্রঃ উপরের দুটি ফাংশনের ব্যবহার

দেখাতে গিয়ে আমি $\log(0)$ ব্যবহার করেছি। $\log(0)$ এর মান অসীম বলে বিবেচিত হয়। এটিও পিএইচপির ম্যাথ ফাংশনগুলির মধ্যে গুরুত্বপূর্ণ একটি। আমরা পরবর্তী কোন পরবে এটির ব্যবহার দেখে নিব।

এই লেখাটিতে ব্যবহৃত পিএইচপি ফাইলগুলো ডাউনলোড করে নিনঃ <http://bangla-phpbook.googlecode.com/files/phpbook-math-1.zip>



phpbook-math-1.zip

সাধারণ ত্রিকোণমিতিক সমস্যার সমাধান

এসএসসি তে ত্রিকোণমিতির বেসিকস বুঝতে বুঝতেই কখন যেন সময় গড়িয়ে গেল। এইচএসসি তে এসে বুঝলাম ত্রিকোণমিতির ঝাল কেমন। সাইন, কোসাইন আর ট্যাঞ্জেন্টের ধারণা মনে রাখাটাই বড় ধরনের কৃতিত্ব মনে হত তখন। কোনটা লম্ব ভাগ অতিভুজ আর কোনটা অতিভুজ ভাগ ভূমি – এই সংজ্ঞাগুলো মনে রাখতে আয়ত্ত্বও করেছিলাম বেশকিছু ছন্দ 😊 $\sin 30$, $\cos 90$ ইত্যাদির মানতো কখনই মনে রাখতে পারতাম না। ভাগ্যিস হাতের কাছে ক্যালকুলেটর নামক যন্ত্রখানি সর্বদাই রাখা পড়ত 😊

পিএইচপির ম্যাথ ফাংশন সিরিজের দ্বিতীয় পরবে তাই আমরা ত্রিকোণমিতি সম্পর্কিত ফাংশনগুলোর দিকে নজর দিতে চাই। পিএইচপির অতি প্রচলিত ৪৮টি ম্যাথ ফাংশনের মধ্যে ১৫টিই ত্রিকোণমিতি সম্পর্কিত। এগুলো হল – $\text{acos}()$, $\text{acosh}()$, $\text{asin}()$, $\text{asinh}()$, $\text{atan2}()$, $\text{atan}()$, $\text{atanh}()$, $\text{cos}()$, $\text{cosh}()$, $\text{hypot}()$, $\text{sin}()$, $\text{sinh}()$, $\text{tan}()$, $\text{rad2deg}()$, $\text{tanh}()$ 😊 এই পর্বটিতে আমরা সাধারণ ত্রিকোণমিতিক ফাংশনগুলোর ব্যবহারিক প্রয়োগ দেখব। অর্থাৎ $\text{sin}()$, $\text{cos}()$, $\text{tan}()$ এবং $\text{hypot}()$ 😊

$\text{hypot}()$:

চমৎকার এই ফাংশনটির কথা অনেকেরই অজানা। এটির কাজ সমকোণী ত্রিভুজের অতিভুজ পরিমাপ করা। কিভাবে? নিজেই ব্যবহার করে দেখে নিন:

```
1 <?php
2 // filename: /home/masnun/phpbook/math-2/hypot.php
3
4 var_dump( hypot(3,4) ); // Outputs: float(5)
5 ?>
```

6

```
7 <?php
```

```
8 var_dump( hypot(12,5) ); // Outputs: float(13)
```

```
9 ?>
```

আমরা সমকোণী ত্রিভুজের দুই বাহুর পরিমাপ প্যারামিটার হিসেবে দিলেই এই ফাংশনটি জানিয়ে দেবে অতিভুজের হিসাব। আজ যদি পীথাগোরাস বেচে থাকতেন তবে তিনিও নির্ঘাত পিএইচপি শিখতে চাইতেন অতিভুজ সংক্রান্ত সমস্যাগুলো দ্রুত সমাধান করতে

😊কি, একটু বেশিই বলে ফেললাম নাকি ? 😊

sin():

নাম থেকেই যে ফাংশনের কাজ সম্পর্কে ধারণা করে ফেলেছেন তাতো বুঝতেই পারছি। হ্যা, প্রদত্ত কোণের সাইন অনুপাত নির্ণয় করাই এই ফাংশনের একমাত্র কাজ 😊

```
1 <?php
```

```
2 // filename: /home/masnun/phpbook/math-2/sin.php
```

```
3
```

```
4 var_dump( sin(90) ); // Outputs: float(0.893996663601)
```

```
5 ?>
```

```
6
```

```
7 <?php
```

```
8 var_dump( sin(0) ); // Outputs: float(0)
```

```
9 ?>
```

জি, sin90 এর মান পুরোপুরি ১ নয় 😊

cos():

সাইনের পরেই আমরা চলে এলাম কোসাইনের জগতে। কোন কোণের পরিমাপ থেকে তার কোসাইন অনুপাত করতে আমাদের এই ফাংশনটির শরণাপন্ন হতে হয়।

```
1 <?php
```

```
2 // filename: /home/masnun/phpbook/math-2/cos.php
```

```
3
```

```
4 var_dump( cos(90) ); // Outputs: float(-0.448073616129)
```

```
5 ?>
```

6

```
7 <?php
```

```
8 var_dump( cos(0) ); // Outputs: float(1)
```

```
9 ?>
```

tan():

সাইন অনুপাতকে কোসাইনের অনুপাত দিয়ে ভাগ করলেই যে ট্যাঞ্জেন্ট অনুপাত পাওয়া যাবে তা আমরা কম বেশি সবাই জানি। কিন্তু, কি দরকার অযথা সময় নষ্ট করার? এই ফাংশনটি থেকে আমরা একবারেই পেয়ে যাব কাঙ্ক্ষিত ফলাফল।

```
1 <?php
```

```
2 // filename: /home/masnun/phpbook/math-2/tan.php
```

```
3
```

```
4 var_dump( tan(45) ); // Outputs: float(1.61977519054)
```

```
5 ?>
```

```
6
```

```
7 <?php
```

```
8 var_dump( tan(0) ); // Outputs: float(0)
```

```
9 ?>
```

এই লেখাটিতে ব্যবহৃত উদাহরণ সমূহ ডাউনলোড করে নিন এখান থেকে – <http://bangla-phpbook.googlecode.com/files/math-2.zip>



math-2.zip

কিছু নিরাপত্তাজনিত ভুল যা আমরা সব সময় করি

১. শর্ট ট্যাগ ব্যবহার করা; যদি শর্ট ট্যাগ এনাবল না থাকে তাহলে আপনার পিএইচপি কোড সার্ভিস পার্স করবে না, ব্রাউজারে সরাসরি দেখাবে।
২. জাভাস্ক্রিপ্টে ইনপুট ভ্যালিডেশন আছে তাই সার্ভার সাইডে কোন সুরক্ষা না রাখা।
৩. register_global on রেখে কাজ করা।
৪. \$_REQUEST[] দিয়ে ভ্যারিয়েবল পাসিং।
৫. SQL স্টেটমেন্টগুলোতে কলামনেমগুলোকে সিঙ্গেল কোট এনক্লোজ না করা।
৬. এইচটিএমএল ও জাভাস্ক্রিপ্ট দিয়ে ইনপুট দেয়া প্রতিহত না করা।

৭. এটা কি ভুল সেটি বরং বিজ্ঞজনদের বলার জন্য রেখে দেই, দেখেছেন নিশ্চয়ই এসএসসি/এইচএসসি রেজাল্ট সাইট এর পিএইচপি কোড কথা নেই বার্তা নেই দেখা যাচ্ছে?

৮. input ভ্যালিডেশনে eval() ব্যবহার করা। <http://seclists.org/fulldisclosure/2006/May/0035.html>

ইনপুট ভ্যালিডেশন

আমরা ব্যবহারকারীর কাছ থেকে যেসব তথ্য নিয়ে থাকি তাকে অবশ্যই আমাদের প্রয়োজনীয় আকারে রূপান্তর করে নিতে হয়। কেননা, ব্যবহারকারী ইচ্ছাকৃত বা অনিচ্ছাকৃতভাবে এমন ভুল করতে পারে যার জন্য আপনার মাথার চুল ছেঁড়ার দশা হতে পারে কি থেকে কি হলো বুঝতে না পেরে। ধরুন, আপনার একটি খুবই সাধারণ পেজ আছে যাতে কিছু সাধারণ তথ্য নেবেন ইউজার-এর কাছ থেকে। আর এ কাজটি করার জন্য নিশ্চয়ই ফরম ব্যবহার করতে হবে তাই না?

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

```
<input name="user" type="text">
```

```
.
```

```
.
```

```
.
```

```
</form>
```

ফরমের কোড অনেকটা উপরের মতো হবে, এখন action.php ফাইলটি যখন ফরমকে প্রসেস করবে তখন যদি আমরা user নামের এই ফিল্ডটিকে ঠিকমতো যাচাই না করেই ছেড়ে দেই, তাহলে কি হবে?

- ইউজার ঠিকমতোই সব পূরণ করবে (যদি না করে?)

- ধরা যাক আমরা চাই ১০ থেকে ৪০ অক্ষরে নাম লিখবে, কিন্তু লিখলো ১০ এর কম বা অনেক অনেক বেশি।

- ইউজার এখানে এসকিউএল কোড লিখে দিতে পারে(SQL Injection পর্ব দেখুন)

- ইউজার এখানে এইচটিএমএল কোড লিখে দিতে পারে, কোনো সাইট এর url link, বড় কোনো ছবির কোড হতে পারে।

- ইউজার জাভাস্ক্রিপ্ট কোড লিখে দিতে পারে যা কোন url redirection এর কাজ করবে।

- ইউজার ইচ্ছামতো ভ্যারিয়েবল এর মান নিয়ে খেলা করতে পারে(হ্যাকিং এর উদ্দেশ্যে)। যেমন:

```
action.php?user=blah&pass=blah&... ইত্যাদি।
```

- ইউজার আপনার ফরম পেজটি কপি করে নিয়ে নিজস্ব সার্ভারে রান করতে পারে যা আপনার ডাটাবেজে এক্সেস করতে চাইবে।

এখন ধরা যাক, কেবল ক্লায়েন্ট সাইড ভ্যালিডেশন করা আছে(জাভাস্ক্রিপ্ট দিয়ে)। ইউজার এর ব্রাউজার তা সাপোর্ট নাও করতে পারে অথবা ডিজেবল করা থাকতে পারে। তাহলে তার উপর আস্থা রাখা উচিত হবে?

তাই, বুঝতে পারছি যে, ইউজার যত সাধারণই হোক তাকে কিছুতেই বিশ্বাস করা চলবে না। নাসিরুদ্দিন হোজ্জা একবার তার চাকরকে বলেছিলেন, "কাচের গ্লাসটা ভাঙিসনে যেন!" কিছু ভেবে আবার গ্লাসটি নিজ হাতে নিয়ে, চাকরকে দিলেন এক চড়। "হ্যা দেখিস সাবধানে"। তার বন্ধুরা এ অমানবিক আচরণের কারণ জানতে চাইলে, "আরে ভেঙে ফেললে তো মেরে লাভ নেই, তাই আগে ভাগেই সাবধান করে রাখলাম"।

তো, প্রোগ্রামারদের ঐ নাসিরুদ্দিন-এর চাইতেও সাবধানী হতে হয়।

কাজের কথায় আসি, user নামক ফিল্ডটি সবরমকম পরীক্ষা পাশ করিয়ে তারপর অন্যকিছু করতে হবে; এভাবে অন্য ফিল্ডও।

যেমন, **strip_tags** ফাংশনটি ব্যবহার করে এইচটিএমএল কোড ফেলে দিতে পারি। তা নাহলে, কোন এক whiz kid হয়তো আপনার সাইট এর ছোটোখাটো দফারফা করে দিতে পারে। আর সে যদি কোনো ক্র্যাকার হয় তবে তো কথাই নেই।

আবার সব কিছু পিএইচপি দিয়ে করলে তো হবে না, পুরো ফর্মটি ফিল-আপ করা হলো, তারপর.. যেই সাবমিট বাটন চাপা হলো, সেই একগাদা ভুল। আপনার সাইটের ব্যবহারকারী মনে মনে বলবে "যাহ্ আর করবোই না!"। ইউজার হারালেন। তাই পেজ ইন্টারাক্টিভ করতে ক্লায়েন্ট-সাইড ভ্যালিডেশনও চাই। শেষ রক্ষা করার জন্য অবশ্যই গুরুত্বপূর্ণ ফিল্ডসমূহ সার্ভারসাইড ভেলিডেটেড করতে হবে। আপনি জিমেইল নতুন আইডি খুলতে গেলে যেমন অ্যাভেইলেবিলিটি অপশনটি দেখায়, প্রয়োজন ও সুযোগ থাকলে সেধরণের সুবিধা রাখতে হবে। ইউজার দুবার সাবমিট বাটন চেপে ফেলতে পারে অথবা রিফ্রেশ বাটন চাপতে পারে তখন বারবার একই তথ্য এন্ট্রি হবে। সুতরাং মাল্টিপল সাবমিশন বন্ধ করতে হবে।

- Input validation
- Escaping dangerous characters. For instance using the addslashes() function in PHP to protect against SQL Injection.
- Input encoding
- Output encoding
- Other coding practices which are not prone to Code Injection vulnerabilities, such as "parameterized SQL queries" (also known as "prepared statements" and sometimes "bind variables").

PHP Tutorial

« [W3Schools Home](#)

[Next Chapter »](#)



PHP is a powerful tool for making dynamic and interactive Web pages.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

In our PHP tutorial you will learn about PHP, and how to execute scripts on your server.

PHP Array Introduction

The array functions allow you to manipulate arrays.

PHP supports both simple and multi-dimensional arrays. There are also specific functions for populating arrays from database queries.

PHP Array Functions

PHP: indicates the earliest version of PHP that supports the function.

Function	Description	PHP
array()	Creates an array	3
array_change_key_case()	Returns an array with all keys in lowercase or uppercase	4

<u>array_chunk()</u>	Splits an array into chunks of arrays	4
<u>array_combine()</u>	Creates an array by using one array for keys and another for its values	5
<u>array_count_values()</u>	Returns an array with the number of occurrences for each value	4
<u>array_diff()</u>	Compares array values, and returns the differences	4
<u>array_diff_assoc()</u>	Compares array keys and values, and returns the differences	4
<u>array_diff_key()</u>	Compares array keys, and returns the differences	5
<u>array_diff_uassoc()</u>	Compares array keys and values, with an additional user-made function check, and returns the differences	5
<u>array_diff_ukey()</u>	Compares array keys, with an additional user-made function check, and returns the differences	5
<u>array_fill()</u>	Fills an array with values	4
<u>array_filter()</u>	Filters elements of an array using a user-made function	4
<u>array_flip()</u>	Exchanges all keys with their associated values in an array	4
<u>array_intersect()</u>	Compares array values, and returns the matches	4
<u>array_intersect_assoc()</u>	Compares array keys and values, and returns the matches	4
<u>array_intersect_key()</u>	Compares array keys, and returns the matches	5
<u>array_intersect_uassoc()</u>	Compares array keys and values, with an additional user-made function check, and returns the matches	5
<u>array_intersect_ukey()</u>	Compares array keys, with an additional user-made function check, and returns the matches	5

<u>array_key_exists()</u>	Checks if the specified key exists in the array	4
<u>array_keys()</u>	Returns all the keys of an array	4
<u>array_map()</u>	Sends each value of an array to a user-made function, which returns new values	4
<u>array_merge()</u>	Merges one or more arrays into one array	4
<u>array_merge_recursive()</u>	Merges one or more arrays into one array	4
<u>array_multisort()</u>	Sorts multiple or multi-dimensional arrays	4
<u>array_pad()</u>	Inserts a specified number of items, with a specified value, to an array	4
<u>array_pop()</u>	Deletes the last element of an array	4
<u>array_product()</u>	Calculates the product of the values in an array	5
<u>array_push()</u>	Inserts one or more elements to the end of an array	4
<u>array_rand()</u>	Returns one or more random keys from an array	4
<u>array_reduce()</u>	Returns an array as a string, using a user-defined function	4
<u>array_reverse()</u>	Returns an array in the reverse order	4
<u>array_search()</u>	Searches an array for a given value and returns the key	4
<u>array_shift()</u>	Removes the first element from an array, and returns the value of the removed element	4
<u>array_slice()</u>	Returns selected parts of an array	4
<u>array_splice()</u>	Removes and replaces specified elements of an array	4
<u>array_sum()</u>	Returns the sum of the values in an array	4
<u>array_udiff()</u>	Compares array values in a user-made function and returns an array	5
<u>array_udiff_assoc()</u>	Compares array keys, and compares array values in a user-made function, and returns an array	5

<u>array_udiff_uassoc()</u>	Compares array keys and array values in user-made functions, and returns an array	5
<u>array_uintersect()</u>	Compares array values in a user-made function and returns an array	5
<u>array_uintersect_assoc()</u>	Compares array keys, and compares array values in a user-made function, and returns an array	5
<u>array_uintersect_uassoc()</u>	Compares array keys and array values in user-made functions, and returns an array	5
<u>array_unique()</u>	Removes duplicate values from an array	4
<u>array_unshift()</u>	Adds one or more elements to the beginning of an array	4
<u>array_values()</u>	Returns all the values of an array	4
<u>array_walk()</u>	Applies a user function to every member of an array	3
<u>array_walk_recursive()</u>	Applies a user function recursively to every member of an array	5
<u>arsort()</u>	Sorts an array in reverse order and maintain index association	3
<u>asort()</u>	Sorts an array and maintain index association	3
<u>compact()</u>	Create array containing variables and their values	4
<u>count()</u>	Counts elements in an array, or properties in an object	3
<u>current()</u>	Returns the current element in an array	3
<u>each()</u>	Returns the current key and value pair from an array	3
<u>end()</u>	Sets the internal pointer of an array to its last element	3
<u>extract()</u>	Imports variables into the current symbol table from	3

	an array	
<u>in_array()</u>	Checks if a specified value exists in an array	4
<u>key()</u>	Fetches a key from an array	3
<u>krsort()</u>	Sorts an array by key in reverse order	3
<u>ksort()</u>	Sorts an array by key	3
<u>list()</u>	Assigns variables as if they were an array	3
<u>natcasesort()</u>	Sorts an array using a case insensitive "natural order" algorithm	4
<u>natsort()</u>	Sorts an array using a "natural order" algorithm	4
<u>next()</u>	Advance the internal array pointer of an array	3
<u>pos()</u>	Alias of current()	3
<u>prev()</u>	Rewinds the internal array pointer	3
<u>range()</u>	Creates an array containing a range of elements	3
<u>reset()</u>	Sets the internal pointer of an array to its first element	3
<u>rsort()</u>	Sorts an array in reverse order	3
<u>shuffle()</u>	Shuffles an array	3
<u>sizeof()</u>	Alias of count()	3
<u>sort()</u>	Sorts an array	3
<u>uasort()</u>	Sorts an array with a user-defined function and maintain index association	3
<u>uksort()</u>	Sorts an array by keys using a user-defined function	3
<u>usort()</u>	Sorts an array by values using a user-defined function	3

Definition and Usage

array() creates an array, with keys and values. If you skip the keys when you specify an array, an integer key is generated, starting at 0 and increases by 1 for each value.

Syntax

```
array(key => value)
```

Parameter	Description
key	Optional. Specifies the key, of type numeric or string. If not set, an integer key is generated, starting at 0
value	Required. Specifies the value

Example 1

```
<?php
$a=array("a"=>"Dog","b"=>"Cat","c"=>"Horse");
print_r($a);
?>
```

The output of the code above will be:

```
Array ( [a] => Dog [b] => Cat [c] => Horse )
```

Example 2

```
<?php
$a=array("Dog","Cat","Horse");
print_r($a);
?>
```

The output of the code above will be:

```
Array ( [0] => Dog [1] => Cat [2] => Horse )
```

PHP [array_change_key_case\(\)](#) Function

Definition and Usage

The `array_change_key_case()` function returns an array with all array KEYS in lower case or upper case.

Syntax

```
array_change_key_case(array,case)
```

Parameter	Description
array	Required. Specifies the array to use
case	Optional. Possible values: <ul style="list-style-type: none">• CASE_LOWER - Default value. Returns the array key values in lower case.• CASE_UPPER - Returns the array key values in upper case.

Tips and Notes

Note: If two or more array keys will be the same after running this function, the last array will override the others. (See example 2)

Example 1

```
<?php
$a=array("a"=>"Cat","b"=>"Dog","c"=>"Horse");
print_r(array_change_key_case($a,CASE_UPPER));
?>
```

The output of the code above will be:

```
Array ( [A] => Cat [B] => Dog [C] => Horse )
```

Example 2

```
<?php
$a=array("a"=>"Cat","b"=>"Dog","c"=>"Horse","B"=>"Bird");
print_r(array_change_key_case($a,CASE_UPPER));
?>
```

The output of the code above will be:

```
Array ( [A] => Cat [B] => Bird [C] => Horse )
```

PHP [array_chunk\(\)](#) Function

[Complete PHP Array Reference](#)

Definition and Usage

The `array_chunk()` function splits an array into chunks of new arrays.

Syntax

```
array_chunk(array,size,preserve_key)
```

Parameter	Description
array	Required. Specifies the array to use
size	Required. Specifies how many elements each new array will contain
preserve_key	Optional. Possible values: <ul style="list-style-type: none">• true - Preserves the keys from the original array.• false - Default. Does not preserve the keys from the original array.

Example 1

```
<?php
$a=array("a"=>"Cat","b"=>"Dog","c"=>"Horse","d"=>"Cow");
print_r(array_chunk($a,2));
```



```
?>
```

The output of the code above will be:

```
Array (  
[0] => Array ( [0] => Cat [1] => Dog )  
[1] => Array ( [0] => Horse [1] => Cow )  
)
```

Example 2

```
<?php  
$a=array("a"=>"Cat","b"=>"Dog","c"=>"Horse","d"=>"Cow");  
print_r(array_chunk($a,2,true));  
?>
```

The output of the code above will be:

```
Array (  
[0] => Array ( [a] => Cat [b] => Dog )  
[1] => Array ( [c] => Horse [d] => Cow )  
)
```

PHP [array_combine\(\)](#) Function

 [Complete PHP Array Reference](#)

Definition and Usage

The `array_combine()` function creates an array by combining two other arrays, where the first array is the keys, and the other array is the values.

Syntax

```
array_combine(array1,array2)
```

Parameter	Description
-----------	-------------

array1	Required. An array, specifying the keys
array2	Required. An array, specifying the values

Tips and Notes

Note: Both parameters must have equal number of elements.

Example

```
<?php
$a1=array("a","b","c","d");
$a2=array("Cat","Dog","Horse","Cow");
print_r(array_combine($a1,$a2));
?>
```

The output of the code above will be:

```
Array ( [a] => Cat [b] => Dog [c] => Horse [d] => Cow )
```

PHP [array_count_values\(\)](#) Function

[Complete PHP Array Reference](#)

Definition and Usage

The `array_count_values()` function returns an array, where the keys are the original array's values, and the values is the number of occurrences.

Syntax

```
array_count_values(array)
```

Parameter	Description
array	Required. Specifying an array.

Example

```
<?php
$a=array("Cat","Dog","Horse","Dog");
print_r(array_count_values($a));
?>
```

The output of the code above will be:

```
Array ( [Cat] => 1 [Dog] => 2 [Horse] => 1 )
```

PHP [array_diff\(\)](#) Function

[Complete PHP Array Reference](#)

Definition and Usage

The `array_diff()` function compares two or more arrays, and returns an array with the keys and values from the first array, only if the value is not present in any of the other arrays.

Syntax

```
array_diff(array1,array2,array3...)
```

Parameter	Description
array1	Required. The first array is the array that the others will be compared with
array2	Required. An array to be compared with the first array
array3	Optional. An array to be compared with the first array

Tips and Notes

Tip: You can compare the first array with one array, or as many as you like.

Note: Only the value is used in the comparison.

Example

```
<?php
$a1=array(0=>"Cat",1=>"Dog",2=>"Horse");
$a2=array(3=>"Horse",4=>"Dog",5=>"Fish");
print_r(array_diff($a1,$a2));
?>
```

The output of the code above will be:

```
Array ( [0] => Cat )
```

 [Complete PHP Array Reference](#)

PHP `array_diff()` Function

 [Complete PHP Array Reference](#)

Definition and Usage

The `array_diff()` function compares two or more arrays, and returns an array with the keys and values from the first array, only if the value is not present in any of the other arrays.

Syntax

```
array_diff(array1,array2,array3...)
```

Parameter	Description
array1	Required. The first array is the array that the others will be compared with
array2	Required. An array to be compared with the first array
array3	Optional. An array to be compared with the first array

Tips and Notes

Tip: You can compare the first array with one array, or as many as you like.

Note: Only the value is used in the comparison.

Example

```
<?php
$a1=array(0=>"Cat",1=>"Dog",2=>"Horse");
$a2=array(3=>"Horse",4=>"Dog",5=>"Fish");
print_r(array_diff($a1,$a2));
?>
```

The output of the code above will be:

```
Array ( [0] => Cat )
```

 [Complete PHP Array Reference](#)

[ইন্টারনেট হতে সংগ্রহীত](#)

প্রয়োজনীয় বাংলা বই ফ্রী ডাউনলোড করতে চাইলে নিচের লিংক গুলো দেখতে পারেনঃ

☆ http://techtunes.com.bd/tuner/tanbir_cox

☆ http://tunerpage.com/archives/author/tanbir_cox

☆ <http://somerwhereinblog.net/tanbircox>

☆ http://pchelplinebd.com/archives/author/tanbir_cox

☆ http://prothom-aloblog.com/blog/tanbir_cox

Tanbir Ahmad Razib

📞 Mobile No:→ **01738 -359 555**

✉ E-Mail: → tanbir.cox@gmail.com

👤 Facebook: → <http://facebook.com/tanbir.cox>

📖 e-books Page: → <http://facebook.com/tanbir.ebooks>

🌐 Web Site : → <http://tanbircox.blogspot.com>



I share new interesting & Useful Bangla e-books(pdf) everyday on my facebook page & website .

Keep on eye always on my facebook page & website & update ur knowledge .

If You think my e-books are useful , then please share & Distribute my e-book on Your facebook & personal blog .