Basic Structure

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

If error doesn't display on php 5.5.9 ubuntu

delete ; before production value
/etc/php5/apache2

```
 ;  error_reporting
 ;  Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
 ;  Development Value: E_ALL
  Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
```

Comment

```
<?php
//this is single comment
#this is single comment
/*this is multiline comment
  comment
  comment
  */
?>
```

Printing/output (echo or print)

```
<?php
echo "printing single line in php";
echo "<p>";
echo "printing multiline <br> in php";
?>
```

Using variabel ($) and connector (.)

```
<?php
$a=3;
$b=2;
$c=$a+$b;
echo $a.'+'.$b.'='.$c;
?>
```

```
<?php
$a="php";
$b="javascript";
$c=$a.$b;
echo $a.'+'.$b.'='.$c;
?>
```

Replacement character

\n is replaced by the newline character
\r is replaced by the carriage-return character
\t is replaced by the tab character
\$ is replaced by the dollar sign itself ($)
\" is replaced by a single double-quote (")
\\ is replaced by a single backslash (\)

Variable global and local , function with parameter and return value

```
<?php
$a=8;
function number($g) // function name : number, parameter : $g
{
    GLOBAL $a; // set a as global variable, position a must be outside function
    $b=$g; // local variable $b and $c
    $c = $b+$a;
    return $c; // return value
}

echo "local variable : ".number(6);
echo "<p>";
echo "global variable : ".$a;
?>
```

```
local variable : 14

global variable : 8
```

Variable static

```
<?php
function number()
{
    STATIC $a=0;
    $a++;
    echo $a;
}

echo number();
echo number();
echo number();
echo "<br>";
?>
```

```
123
```

Static will hold the last value every time function called

Without STATIC Result $a after 3 times call will be 111

Constant

```
<?php
define("NUM",5); // number constant
define("STRING", "asri"); // string constant

function number()
{
    echo NUM;
    echo "<br>";
    echo STRING;
}

number();
?>
```

```
5
asri
```

Arithmetic Operators

Operator	Description
+, -	add, substracts
*	multiply
/	divide
%	modulus
++, --	Increment, decrement

Comparison operators

Operator	Description
==	If left equal right → if(a==b)
!=	If left not equal right
>	If left greater than right
<	If left less than right
>=	If left greater than or equal to right
<=	If left less than or equal to right

Logical operators

Operator	Description
and, &&	And operator ($a && $b)
or, ||	Or operator ($a || $b)
!	Not operator (!$a)

Assignment operators

Operator	Description
=	Place right(a+b) to left(c) → $c=$a+$b;
+= , -=	$b=$b+$a, $b=$b-$a
*= , /=	$b=$b*$a, $b=$b/$a
%=	$b=$b%$a

Conditional operator

Operator	Description
?:	if(a==true)? then …. : otherwise ….

Condition : If .. elseif .. else

```
<?php
function number($a, $b)
{
    if(($a==0) || ($b==0))
    {   $c=0;     }
    elseif($a==$b)
    {   $c=$a;   }
    else
    {   $c=$a+$b;  }
    return $c;
}

$g=number(2,5);
echo $g;
?>
```

```
7
```

Condition : Switch .. case

```
<?php
function number($g)
{
    switch($g)
    {
        case (0):
            $c="0";
            break;
        case ($g<5):
            $c=$g." less than 5";
            break;
        case ($g<10):
            $c=$g." more than or equal 5";
            break;
        default:
            $c=$g." more than or equal 10";
    }
    return $c;
}

echo number(1);
?>
```

```
1 less than 5
```

Looping : for

```
<?php
function number($g)
{
    for($a=1; $a<=$g; $a++)
    {
        $c= "cumi : ". $a . "<br>";
        echo $c;
    }
}

number(3);
?>
```

Looping : while

```
<?php
function number($g)
{
    $a=1;
    while($a<=$g)
    {
        $c= "cumi : ". $a . "<br>";
        echo $c;
        $a++;
    }
}

number(3);
?>
```

```
cumi : 1
cumi : 2
cumi : 3
```

Looping : do .. while

```
<?php
function number($g)
{
    $a=1;
    do
    {
        $c= "cumi : ". $a . "<br>";
        echo $c;
        $a++;
    }  while($a<=$g);
}

number(3);
?>
```

Looping : foreach (looping in array)

```
<?php
function number($g)
{
    foreach($g as $k)
    {
        echo $k. "<br>";
    }
}

$myarray = array(1,4,3,2,8);
number($myarray);
?>
```

```
1
4
3
2
8
```

break and continue

```
<?php
function number($g)
{
    $a=0;
    while($a<=$g)
    {
        $a++;
        echo $a;

        if($a==4)
        {   break;    }
    }
}

number(10);
?>
```

```
1234
```

```
<?php
function number($g)
{
    $a=0;
    while($a<$g)
    {
        $a++;

        if($a==4)
        {   continue;    }

        echo $a;
    }
}

number(10);
?>
```

```
1235678910
```

arrays

```
<?php
function number() // first method to create array
{
    $number=array(1,2,3,4,5);
    foreach($number as $num)
    {   echo $num. " ";        }
}

function string() // second method to create array
{
    $string[0] = "one";
    $string[1] = "two";
    $string[2] = "three";
    $string[3] = "four";
    $string[4] = "five";

    foreach ($string as $str)
    {   echo $str. " ";  }
}

number();
echo "<br>";
string();
?>
```

```
1 2 3 4 5
one two three four five
```

associative arrays

```
<?php
function mai()
{
    $programmer = array (
                        "kuki" => "javascript",
                        "woki" => "php",
                        "aiki" => "nodejs"
                    );

    foreach ($programmer as $prog)
    {       echo $prog. " ";    }  //echo $programmer['woki'];
}

mai();
?>
```

```
$programmer['kuki'] = "javascript";
$programmer['woki'] = "php";
$programmer['aiki'] = "nodejs";
```

Passing array as parameter of function

```
<?php
function momo($a)
{
    foreach($a as $ab)
    {
        echo $ab. " ";
    }
}

$a=array(1,3,5,3,6);
momo($a);
?>
```

```
1 3 5 3 6
```

function : count() → count amount of array

```
<?php
$a=array(1,3,5,3,6,7);
echo count($a); // sizeof()
?>
```

```
6
```

```
<?php
$a=array("lion", "bear", "cow");
echo count($a);
?>
```

```
3
```

sort array (ascending), rsort array (descending)

1

```
<?php
$a=array("lion", "bear", "cow");
sort($a); // rsort
$lenghta = count($a);

echo $lenghta;
echo "<br>";
for($x=0; $x<$lenghta; $x++)
{
    if($x<$lenghta-1)
        echo $a[$x]. ", ";
    else
        echo $a[$x]. " ";
}
?>
```

```
3
bear, cow, lion
```

2

```
<?php
$a=array("lion", "bear", "cow");
sort($a); // rsort
$lenghta = count($a);

echo $lenghta;
echo "<br>";
for($x=0; $x<$lenghta; $x++)
{
    if($x==$lenghta-1)
        echo $a[$x]. " ";
    else
        echo $a[$x]. ", ";
}
?>
```

asort array (ascending), ksort array (descending)

```
<?php
$a=array("lion"=>2, "bear"=>5, "cow"=>3);
asort($a);

foreach($a as $b => $bvalue)
{
    if(!each($a))
        echo $b . "=>" . $bvalue . " ";
    else
        echo $b . "=>" . $bvalue . ", ";
}
?>
```

```
lion=>2 cow=>3 bear=>5
```

function : array_push() → add element to the end of an array

```
<?php
$a=array("lion", "bear", "cow");
array_push($a, "goat", "elephant");

foreach($a as $b)
{
    if(!each($a))
        echo $b . " ";
    else
        echo $b . ", ";
}
?>
```

```
lion, bear, cow, goat, elephant
```

```
<?php
$a=array("lion", "bear", "cow");
array_push($a, "goat", "elephant");

print_r($a);
?>
```

```
Array ( [0] => lion [1] => bear [2] => cow [3] => goat [4] => elephant )
```

function : array_splice() → replace array with new array

```
<?php
function printarr($a)
{
    foreach($a as $aa)
    {   echo $aa, " ";    }
}

$a=array(1,2,3,4,5);
$b=array(6,7);
echo "first array : ", printarr($a). "<br>";
echo "second array : ", printarr($b). "<br>";

echo "array after delete & add new array : ";
$start=1; //from array
$amount=2; //how much element in array to replace
array_splice($a,$start,$amount,$b);

printarr($a);
?>
```

```
first array : 1 2 3 4 5
second array : 6 7
array after delete & add new array : 1 6 7 4 5
```

function : array_pop() → delete last element in array

```
<?php
$a=array("php","javascript","pascal");
array_pop($a);

foreach($a as $b)
{   echo $b. " ";   }
?>
```

```
php javascript
```

function : array_count_values() → count amount of element in array

```
<?php
$a=array("php","javascript","pascal", "c++", "php", "c++", "php");
print_r(array_count_values($a)); // must use print_r to print
?>
```

```
Array ( [php] => 3 [javascript] => 1 [pascal] => 1 [c++] => 2 )
```

function : array_product() → multiply all element in array

```
<?php
$a=array(4,5,6);
echo array_product($a);
?>
```

```
120
```

function : array_replace() → replace first array with second array in right position

```
<?php
$a=array("php","javascript","pascal");
$b=array("c++","nodejs");

$c=array_replace($a, $b);

foreach ($c as $cc)
{   echo $cc. " ";   }
?>
```

```
c++ nodejs pascal
```

function : array_reverse() → reverse element in array

```
<?php
$a=array("php","javascript","pascal","c++");
$c=array_reverse($a);

foreach ($c as $cc)
{   echo $cc. " ";   }
?>
```

```
c++ pascal javascript php
```

function : array_slice() → cut array

```
<?php
$a=array("php","javascript","pascal","c++");
$c=array_slice($a,1);

foreach ($c as $cc)
{   echo $cc. " ";   }
?>
```

```
javascript pascal c++
```

function : array_sum() → sum all element in array

```
<?php
$a=array(2,5,1,3);
$c=array_sum($a);

echo $c;
?>
```

```
11
```

function : array_search() → get element position in array

```
<?php
$a=array("js","java","php","c++","pascal");
$c=array_search("c++",$a);

echo $c;
?>
```

```
3
```

function : array_unique() → remove double element in array

```
<?php
$a=array("js","java","php","c++","pascal", "php", "js");
$c=array_unique($a);

foreach($c as $cc)
{   echo $cc. " ";   }
?>
```

```
js java php c++ pascal
```

function : current() → get current value/first element in array

```
<?php
$a=array("js","java","php","c++","pascal", "nodejs", "phyton");
echo current($a) . "<br>"; // show current pointer position
echo next($a) . "<br>"; // next position
echo current($a) . "<br>"; // show current position
echo prev($a) . "<br>"; // show previous position
echo end($a) . "<br>"; // go to last element
echo reset($a) . "<br>"; // back to first element
?>
```

```
js
java
java
js
phyton
js
```

function : max(), min() → get max and min value element from array

```
<?php
$a=array(4,7,3,6,2,9,1);
echo max($a). "<br>";
echo min($a);
?>
```

```
9
1
```

function : array_merge() → merge array into one array

```
<?php
$a=array(4,7,3,6,2,9,1);
$b=array("php","js");
$c=array_merge($a,$b);

foreach($c as $cc)
{   echo $cc. " ";   }
?>
```

```
4 7 3 6 2 9 1 php js
```

function : array_walk() → matching array element with user function

```
<?php
function pui($a, $b)
{
    echo "No : $b, language : $a <br>";
}

$a=array(1=>"php",2=>"js");

array_walk($a, pui);
?>
```

```
No : 1, language : php
No : 2, language : js
```

function : in_array() → search element in array

```
<?php
$a=array("php","js", "c++", "nodejs");
$b='js';

if(in_array($b, $a))
{   echo $b . " found";   }
else
{   echo $b . " not found";  }
?>
```

```
js found
```

function : ceil, floor, round

```
<?php
// round numbers up to the nearest integer
echo(ceil(0.6)."<br>");
echo(ceil(0.4)."<br>");
echo(ceil(5)."<br>");
echo(ceil(5.1)."<br>");
echo(ceil(5.9)."<br>");
echo(ceil(-5.1)."<br>");
echo(ceil(-5.9)."<p>");

// round number down to the nearest integer
echo(floor(0.6)."<br>");
echo(floor(0.4)."<br>");
echo(floor(5)."<br>");
echo(floor(5.1)."<br>");
echo(floor(5.9)."<br>");
echo(floor(-5.1)."<br>");
echo(floor(-5.9)."<p>");

// round floating-point numbers
echo(round(0.60)."<br>");
echo(round(0.50)."<br>");
echo(round(0.49)."<br>");
echo(round(-4.49)."<br>");
echo(round(-4.50)."<p>");
?>
```

```
1
1
5
6
6
-5
-5

0
0
5
5
5
-6
-6

1
1
0
-4
-5
```

function : sqrt, pow

```
<?php
echo(sqrt(9)."<br>"); // square root
echo(pow(2,5)."<br>"); // 2*2*2*2*2
?>
```

```
3
32
```

function : fmod (get result of two variable dividing 0 or 1)

```
<?php
$a = 9;
$b = 3;
$c = 4;

$d = fmod($a,$b);
echo $d ."<br>";
$e = fmod($a,$c);
echo $e;
?>
```

```
0
1
```

function : bindec (convert biner to decimal) & decbin (decimal to biner)

```
<?php
$num = "0011";
$nim = bindec($num);
echo $nim;
?>
```

```
3
```

```
<?php
$num = "3";
$nim = decbin($num);
echo $nim;
?>
```

```
11
```

function : dechex (convert decimal to hexadecimal) & hexdec

```
<?php
$num = "3000";
$nim = dechex($num);
echo $nim;
?>
```

```
bb8
```

function : decoct (convert decimal to octal) & octdec

```
<?php
$num = "70";
$nim = decoct($num);
echo $nim;
?>
```

```
106
```

function : isset (is variable has been set), unset (unset a variable)

```
<?php
$word = "pui";

if(isset($word))
{   echo "Yes"; }
else
{   echo "No";  }
?>
```

```
Yes
```

```
<?php
$word = "pui";
unset($word); // if we try to put
unset function

if(isset($word))
{   echo "Yes"; }
else
{   echo "No";  }
?>
```

```
No
```

function : empty() → if a variable doesn't has a value

```
<?php
$word = "";

if(empty($word))
{   echo "variable ha no value";   }
else
{   echo "variable has a value inside";   }
?>
```

```
variable ha no value
```

function : gettype() → get what type of variable

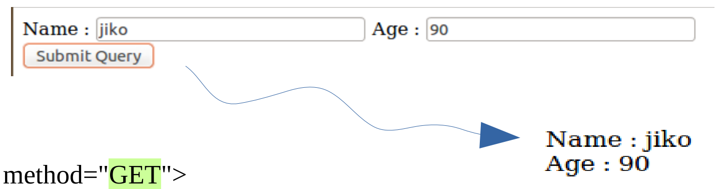
```
<?php
$a = 5;
$b = "5";
$c = "five";
$d = array(1,2,3);
$e = 5.3;
$f = true;

class pui
{
    function name()
    {   echo "my name";    }
}
$g = new pui;

echo "a " . gettype($a). "<br>";
echo "b " . gettype($b). "<br>";
echo "c " . gettype($c). "<br>";
echo "d " . gettype($d). "<br>";
echo "e " . gettype($e). "<br>";
echo "f " . gettype($f). "<br>";
echo "g " . gettype($g);
?
```

```
a integer
b string
c string
d array
e double
f boolean
g object
```

The Get & Post Method

Note: Use Post for password

```
<?php
if($_GET["name"] || $_GET["age"])
{
    echo "Name : ", $_GET['name']."<br/>";
    echo "Age : ", $_GET['age']. "<br/>";
    exit();
}
?>

<html>
<body>

<form action="<?php $_PHP_SELF ?>" method="GET">
    Name : <input type="text" name="name">
    Age : <input type="text" name="age">
    <input type="submit">
</form>

</body>
</html>
```



```
Name : jiko
Age : 90
```

The Request Variable

(can receive the result from data sent with both the GET and POST methods)

```
<?php
if($_REQUEST["name"] || $_REQUEST["age"])
{
    echo "Name : ", $_REQUEST['name']."<br/>";
    echo "Age : ", $_REQUEST['age']. "<br/>";
    exit();
}
?>

<html>
<body>
    <form action="<?php $_PHP_SELF ?>" method="POST">
        Name : <input type="text" name="name">
        Age : <input type="text" name="age">
        <input type="submit">
    </form>
</body>
</html>
```

require dan include (call another php file in php)

require() function will produce error and scripts should not continue executing.
include() function will continue run scripts although there was an error.

index.php

```
<?php
require("wa.php"); //include
echo "macan";
?>
```

wa.php

```
<?php
echo "yuhu";
?>
```

open and reading file

index.php

```
<?php
$filename = "mao.txt";
$file = fopen($filename,"roooo"); // open file
if($file==false)
{
    echo("error in opening file");
    exit();
}

$filesize = filesize($filename); // get file size
$filetext = fread($file, $filesize); // read file
fclose($file); // close file

echo("File size: $filesize bytes");
echo("<pre>$filetext</pre>");
?>
```

```
File size: 22 bytes

lala
trilili
lolo
mumu
```

mao.txt

```
lala
trilili
lolo
mumu
```

write file

make file coba.txt rhen change permisson

```
<?php
$filename = "coba.txt";

if(file_exists($filename))
{
    $thefile = fopen($filename,"w");

    $data = "This is the file/n
    we want to write";
    fputs($thefile, $data); //puts = tulis
    fclose($thefile);
}
else
echo "File doesn't exist";
?>
```

"r" = mode file read only, posisi pointer pada awal file

"w" = mode file write only, membuat file baru
posisi pointer pada awal file, jika file telah ada maka isi
File langsung dihapus.

"r+" = mode read-write, pointer pada awal file,
file dapat dibaca dan ditulis.

"a" = append (hanya tulis), posisi pointer pada akhir file,
Data tidak akan dihapus namun ditambah,
jika file tidak ada maka file baru akan dibuat.

"a+" = append (baca tulis), seperti mode "a" dengan tambahan
dapat melakukan pembacaan file

class php
(function var_dump)

```
<?php
class fruit
{
    var $color="red";
    var $price=4000;
    var $taste;
    var $amount;

    function getTaste($tas)
    {
        $this->taste=$tas;
    }

    function getAmount($am)
    {   $this->amount=$am;        }
}
$mia = new fruit;
$mia->getTaste("sweet");
$mia->getAmount(20);

var_dump($mia); // var_dump = function to check all content in class
?>
```

```
object(fruit)#1 (4)
{
    ["color"]=> string(3) "red"
    ["price"]=> int(4000)
    ["taste"]=> string(5) "sweet"
    ["amount"]=> int(20)
}
```

function : strlen() = length of string

```
<?php
function mai($acem)
{
    $mongki= strlen($acem);
    return $mongki;
}

$g = "mamamia lezatoz";
echo "Amount of Character : ". mai($g);
?>
```

```
Amount of Character : 15
```

function : strpos() = search for a string or character within a string

```
<?php
function mai($acem)
{
    $koki = "mamamia lezatoz";
    $mongki = strpos($koki, $acem);
    return $mongki . " at " . $koki;
}

$g = "lezatoz";
echo "Start position of " . $g ." is : ". mai($g);
?>
```

```
Start position of lezatoz is : 8 at mamamia lezatoz
```

function : str_repeat() = repeat variable

```
<?php
    $a = "eat";
    $b = str_repeat($a. " ",5); // repeat $a 5x
    echo $b;
?>
```

```
eat eat eat eat eat
```

function : strtoupper & strtolower = convert to uppercase & lowercase

```
<?php
$a = "eat";
$b = strtoupper($a);
echo $a . " " . $b;
?>
```

```
eat EAT
```

```
<?php
$a = "EAT";
$b = strtolower($a);
echo $a . " " . $b;
?>
```

```
EAT eat
```

function : substr = take string from ... to …

```
<?php
    $a = "javascript";
    $b = substr($a, 0,4);
    echo $a . "<br>" . $b;
?>
```

```
javascript
java
```

function : substr_count = count substr from variable

```
<?php
    $a = "javascript javascript";
    $b = substr_count($a, "pt");
    echo $a . "<br>" . $b;
?>
```

```
javascript javascript
2
```

integer and string

```
<?php
$integer=123;
$string="123";

echo "Integer number : ";
echo $integer;
echo "<br>";
echo "String number : ";
echo $string;
echo "<p>";
echo "Adding integer with integer 2 : ";
echo $integer + 2;
echo "<br>";
echo "Adding string with string 2 : ";
echo $string . "2";
echo "<p>";
echo "Adding integer with string 2 : ";
echo $integer . "2";
echo "<br>";
echo "Adding string wth integer 2 : ";
echo $string + 2;
echo "<p>";
echo "Length of integer : ";
echo strlen($integer);
echo "<br>";
echo "Length of string : ";
echo strlen($string);
echo "<p>";
echo "Adding integer and string ";
$integerstring = $integer+$string;
echo $integerstring;
?>
```

```
Integer number : 123
String number : 123
Adding integer with integer 2 : 125
Adding string with string 2 : 1232

Adding integer with string 2 : 1232
Adding string wth integer 2 : 125

Length of integer : 3
Length of string : 3

Adding integer and string 246
```

convert integer to string

```
<?php
$integer=123;
$integertostring = strval($integer); // way 1 (convert integer to string using strval)
echo $integertostring+4; // after convert we still can add string to integer
echo "<p>";
$integertostring2 = (string) $integer; // way 2 (convert integer to string using string)
echo $integertostring2+4;
echo "<p>";
?>
```

```
127

127
```

convert string to integer

```
<?php
$string="123";
$stringtointeger = intval($string); // way 1 (convert string to integer using intval)
echo $stringtointeger+"4"; // after convert we still can add integer to string
echo "<p>";
$stringtointeger2 = (integer) $string; // way 2 (convert string to integer using integer)
echo $stringtointeger2+"4";
echo "<p>";
?>
```

function : ucwords (change uppercase every first word) & ucfirst (only first word)

```
<?php
    $a = "javascript javascript";
    $b = ucwords($a);
    $c = ucfirst($a);
    echo $a . "<br>" . $b . "<br>" . $c;
?>
```

```
javascript javascript
Javascript Javascript
Javascript javascript
```

function : is_numeric (this variable is a number/integer or not)

```
<?php
$integer=123;
$string="123";
$string2="satu";
$string3=satu;

if(is_numeric($integer))
{    echo "{$integer} is integer<br>";    }
else
{    echo "{$integer} is string<br>";}

if(is_numeric($string))
{    echo "{$string} is integer<br>";     }
else
{    echo "{$string} is string<br>";  }

if(is_numeric($string2))
{    echo "{$string2} is integer<br>";    }
else
{    echo "{$string2} is string<br>";     }

if(is_numeric($string3))
{    echo "{$string3} is integer<br>";    }
else
{    echo "{$string3} is string<br>";     }
?>
```

```
123 is integer
123 is integer
satu is string
satu is string
```

function : is_string (this variable is string or not) is_int (integer or not)

```
<?php
$integer=123;
$string="123";
$string2="satu";
$string3=satu;
```

```
123 is integer
123 is string
satu is string
satu is string
```

```
if(is_string($integer))
{    echo "{$integer} is string<br>";     }
else
{    echo "{$integer} is integer<br>";    }

if(is_string($string))
{    echo "{$string} is string<br>";  }
else
{    echo "{$string} is integer<br>";     }

if(is_string($string2))
{    echo "{$string2} is string<br>";     }
else
{    echo "{$string2} is integer<br>";    }

if(is_string($string3))
{    echo "{$string3} is string<br>";     }
else
{    echo "{$string3} is integer<br>";    }
?>
```

```
<?php
$integer=123;
$string="123";
$string2="satu";
$string3=satu;
```

```
123 is integer
123 is string
satu is string
satu is string
```

```
if(is_int($integer))
{    echo "{$integer} is integer<br>";    }
else
{    echo "{$integer} is string<br>";     }

if(is_int($string))
{    echo "{$string} is integer<br>";     }
else
{    echo "{$string} is string<br>";  }

if(is_int($string2))
{    echo "{$string2} is integer<br>";    }
else
{    echo "{$string2} is string<br>";     }

if(is_int($string3))
{    echo "{$string3} is integer<br>";    }
else
{    echo "{$string3} is string<br>";     }
?>
```

function : explode & implode

```
<?php
    $days = "sunday monday wednesday thursday friday saturday";
    print_r (explode(" ",$days));
?>
```

```
Array ( [0] => sunday [1] => monday [2] => wednesday [3] => thursday [4] => friday [5] => saturday )
```

```
<?php
    $days = array('sunday', 'monday', 'wednesday', 'thursday', 'friday', 'saturday');
    echo (implode(" ",$days));
?>
```

```
sunday monday wednesday thursday friday saturday
```

function : rand() = generate a random number with-in a given range
function : srand() = specifies the seed number as its argument

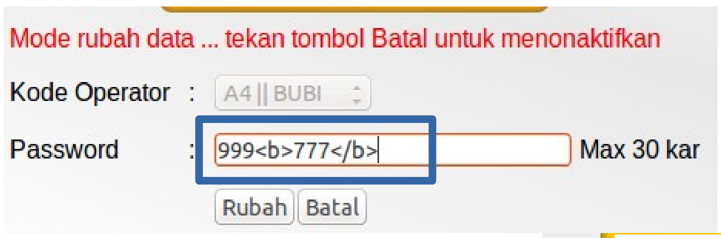
```
<?php
function mixer()
{
    srand(microtime()*1000000);
    $num=rand(1,4);
    switch($num)
    {
        case 1: $im = "js.svg";       break;
        case 2: $im = "css.svg";      break;
        case 3: $im = "html.svg";     break;
        case 4: $im = "php.svg";          break;
    }
    echo "Random image <br> <img src=$im>";
}

mixer();
?>
```

Random image

function : htmlspecialchars

Get variable input without htmlspecialchars, if we try to fill data with html tag <b>777</b>, the result will return only number without tag

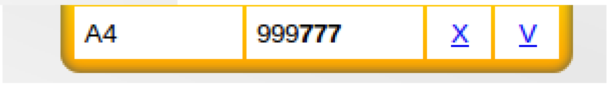
$pass = $_GET['pass'];



Mode rubah data ... tekan tombol Batal untuk menonaktifkan

Kode Operator : A4 || BUBI

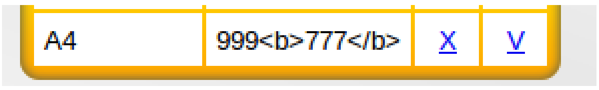
Password : 999<b>777</b> Max 30 kar



A4	999777	X	V

if we using htmlspecialchars, the result will show number and tag all at once.

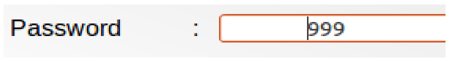
$pass = htmlspecialchars($_GET['pass']);



A4	999<b>777</b>	X	V

function : trim = cut all space

$pass = trim($_GET['pass']);

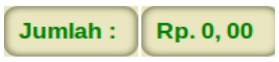


Password : 999

cut all space when we input a number or string.

function : number_format

echo "Rp. ". number_format(doubleval("$rowtotaljual[jual]"),2, ", ", ".");



Jumlah : Rp. 0, 00

function : preg_match = must number

```
if (preg_match("/\D/",$jlhber))
   {   echo "Jumlah barang harus angka"; }
```

Get variable and send variable to other file using form

```
$koper= $_POST['koper'];
echo "<form method='POST' action='lihat_semua_nota.php'>";
        echo "<input type='hidden' name='koper' value='$koper'>";
        echo "<input type='submit' value='' class='button btback' title='tampil semua tanggal berisi nota'>";
echo "</form>";
```

function : getdate

```
<?php
    $tglsekarang=getdate();
    $tgl=$tglsekarang[0];
    $tglsek=date('Y-m-d', $tgl);

    echo $tglsek;
?>
```

```
2016-07-29
```

```
<?php
    $tglsekarang=getdate();
    $tgl=$tglsekarang[0];
    $tglsek=date('l-d-F-Y', $tgl);

    echo $tglsek;
?>
```

```
Friday-29-July-2016
```

```
<?php
    $tglsekarang=getdate();
    $tgl=$tglsekarang[0];
    $tglsek=date('w S', $tgl);

    echo "day : ". $tglsek; . " in month ";
?>
```

```
day : 5 th in month
```

```
<?php
    $tglsekarang=getdate();
    $tgl=$tglsekarang[0];
    $tglsek=date('z S', $tgl);

    echo "day : ". $tglsek . " in year ";
?>
```

```
day : 210 th in year
```

funtion : mail (send email)

```
<?php
    $from = "From: bob@gmail.com";
    $to = "sus@yahoo.com";
    $subject = "Hello";
    $fill = "tihs is only for you";

    mail($to,$subject,$fill,$from);
?>
```

version of php

php.ini-production contains settings which hold security, performance and best practices at its core. But please be aware, these settings may break compatibility with older or less security conscience applications. We recommending using the production ini in production and testing environments.

php.ini-development is very similar to its production variant, except it's much more verbose when it comes to errors. We recommending using the development version only in development environments as errors shown to application users can inadvertently leak otherwise secure information.

This is php.ini-production INI file.

We recommending using the development version only in development environments as errors shown to application users can **inadvertently** **leak** otherwise secure information.

Kami merekomendasikan menggunakan versi pengembangan hanya dalam lingkungan pengembangan sebagai kesalahan ditampilkan kepada pengguna aplikasi **secara tidak sengaja** dapat **membocorkan** informasi dinyatakan aman .

Hasil translate

Kami merekomendasikan menggunakan versi pengembangan hanya dalam lingkungan pengembangan dimana kesalahan ditampilkan kepada pengguna aplikasi secara tidak sengaja dapat membocorkan informasi dinyatakan aman

documenting properties

You should document each of your properties with a PHPDoc block. The comment includes a short
description, any pertinent or confusing information, the @var tag to signal that this is a variable,
and the type of variable expected, such as string, integer, float, array, Boolean, or object.

```
/**
* The phone number of this cell phone
* @var string
*/
```