operators

Objectives:

1. To introduce and use the PHP operators.
2. The categories of PHP operators are assignment, arithmetic, logical and comparison operators.
3. Assigning variables.
4. Operations on numbers using arithmetic operators.
5. Comparing values.
6. Combining statements using Boolean logical operators.

Assignment Operators.

- Assign means to give value to a variable to be stored.
- The symbol is "**=**".
- Equal to "**= =**" is different from assign "**=**", so don't get confused!!

Assignment Operators	**Meaning**	**Example**
Simple assignments		
*Var***=***expr*	Assign the value of expression into the variable.	`$nom=3;` (value 3 is assign the variable *nom*) `$a=50;` `$b=$a;` ($b receives the value of $a, which is 30) `$hasil=($nom*3)+10;` (value produced on the right expression will be assigned to variable *hasil*)
*Var****=***expr*	Multiplication assignment.	`$nom*=3;` (similar as $nom=$nom*3;)
*Var***/=***expr*	Division assignment.	`$nom1/=$nom2;` (similar as $nom1=$nom1/$nom2;)
*Var***%=***expr*	Modulus assignment.	`$num%=2;` (similar as $num=$num%2;)
*Var***+=***expr*	Addition assignment.	`jum+=var1;` (similar as $jum=$jum+$var1;)
Var **-=***expr*	Subtraction assignment.	`$jum-=$var1;` (similar as $jum=$jum-$var1;)

Arithmetic Operators.

These operators are used in the calculation of numerical values.

Operator	Operation	Example	
		Mathematical expression	**PHP expression**
+	Addition	num1 + 3	`$num1 + 3`
-	Subtraction	pi - y	`$pi - $y`
*****	Multiplication	width x height	`$width * $height`
/	Division	a/b or a ÷ b	`$a/$b`
%	Modulus	nom modulo 3	`$nom%3`

Operators order of evaluation (precedence)

Operators	Order of evaluation
( )	First
*, /, %	Second
+, -	Last

Example of mathematical expressions transform into PHP expressions.

Mathematical expression	PHP expression
average = $\frac{a + b + c + d}{4}$	average = (a + b + c + d) / 4;
volumeSphere=4/3 x 3.142 x radius^3	volumeSphere=(4/3)*3.142 * radius * radius*radius;
Fahrenheit = (Celcius x 9/5) +32	Fahrenheit = (Celcius *(9/5)) +32;
volumeCylinder=3.142 x radius^2 x height	volumeCylinder=3.142 * radius * radius * height;
parameterCircle = 2 x 3.142 x radius	parameterCircle = 2.0 * 3.142 * radius;

EXAMPLES for Arithmetic Operations

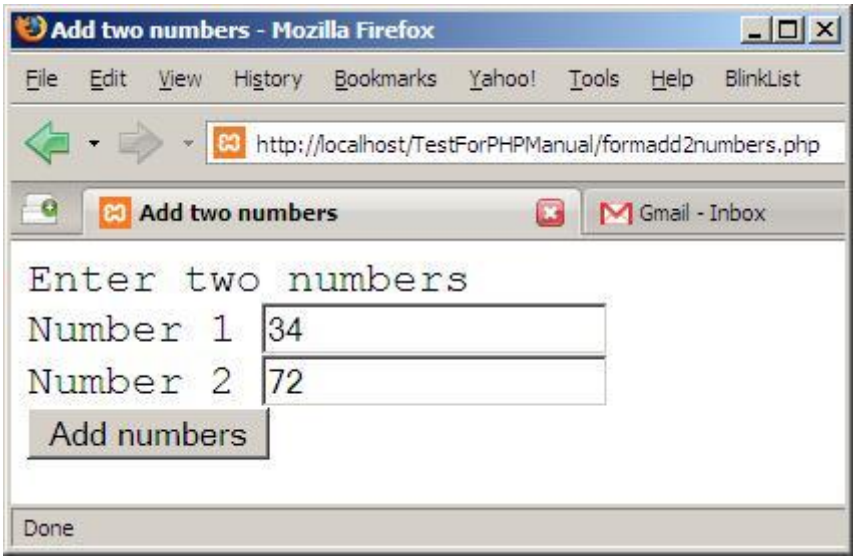
Example 1: Add 2 numbers

You're to develop a simple web application to receive two numbers from the user. The numbers is entered through a HTML form, sent to the server, and another file will receive the number, add both of them and display the result of the operation.

Step 1. Create a HTML page with a form to receive two numbers, and a submit button. Name it *formadd2numbers.php*.

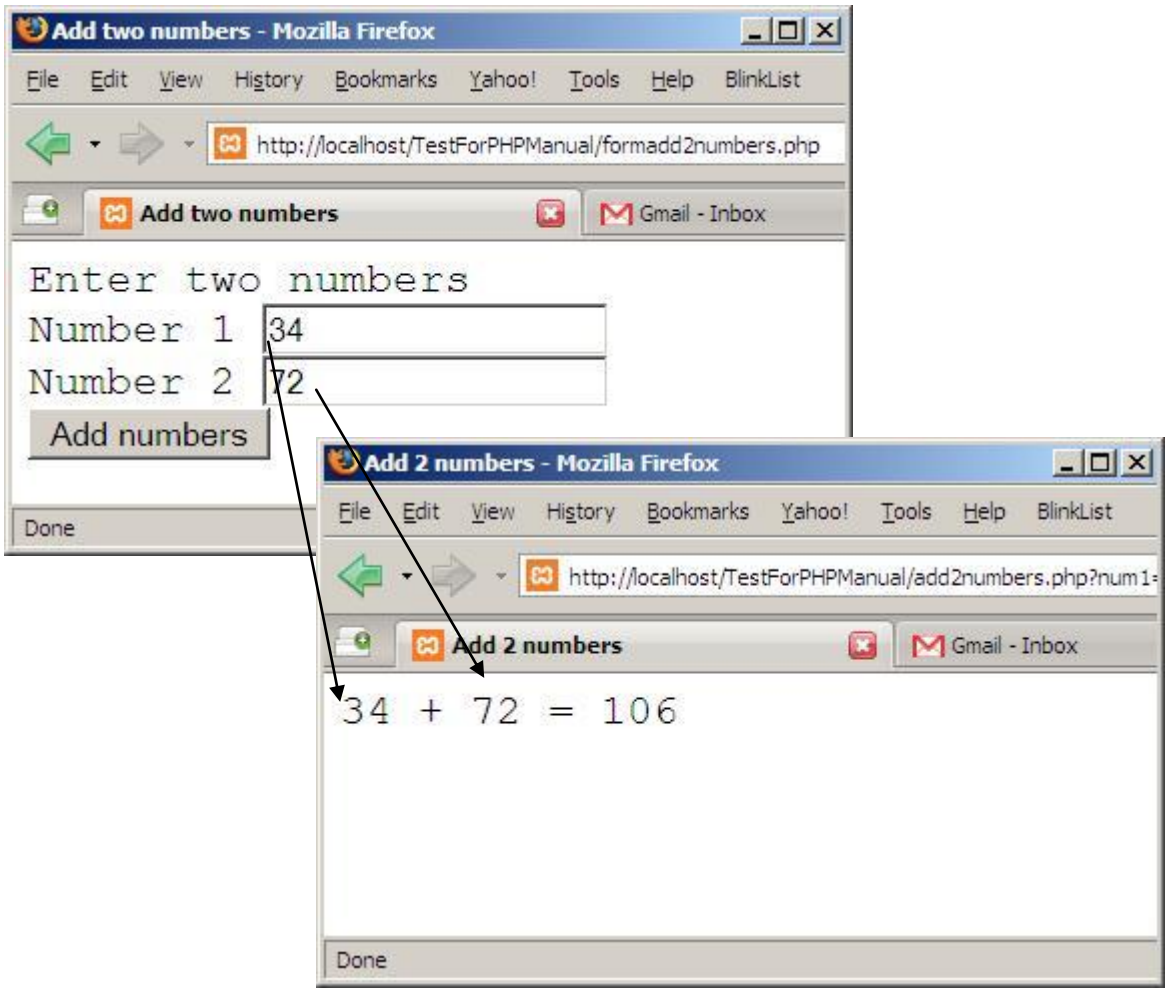
```
<html>
<head>
<title>Add two numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="formNew" action="add2numbers.php">
        Number 1      <input type="text" name="num1"><br>
        Number 2      <input type="text" name="num2"><br>
        <input type="submit" name="btnAdd" value="Add numbers">
</form>
</body>
</html>
```

Step 2. View the page in the browser. Key in any number. **Don't click the button yet.** Go to Step 3, finish the Step 3 and then you can hit the button.



Step 3. Create a new page, write the code below, and save as *add2numbers.php* .

```
<html>
<head>
<title>Add 2 numbers </title>
</head>
<body>
<?php
        $n1=$_GET["num1"]; //retrieve the first number
        $n2=$_GET["num2"];  //retrieve the second number
        $hasil=$n1+$n2; // here is the operation
        echo " $n1 + $n2 = $hasil"; //display all the values
?>
</body>
</html>
```



Example 2: Currency converter

Convert a currency value from RM into USD. Given RM1 is equivalent to USD3.40.

Step 1. Create a HTML page with a form to receive a value in Malaysian ringgit, and a submit button. Name it *formRMUSDConverter.php*. Use the code below.

```
<html>
<head>
<title>RM to USD converter</title>
</head>

<body>
RM to USD converter<br>
<form name="formconvert" action="rmusdconverter.php" method="get">
Value of RM <input  name="txtrm"  type="text">
        <input name="btnconvert" type="submit" value="Convert to USD">

</form>

</body>
</html>
```

Step 2. Next, create a HTML page to receive the value in Malaysian ringgit, and cenvert the value into USD. Save and name it as *rmusdconverter.php*. Use the code below.

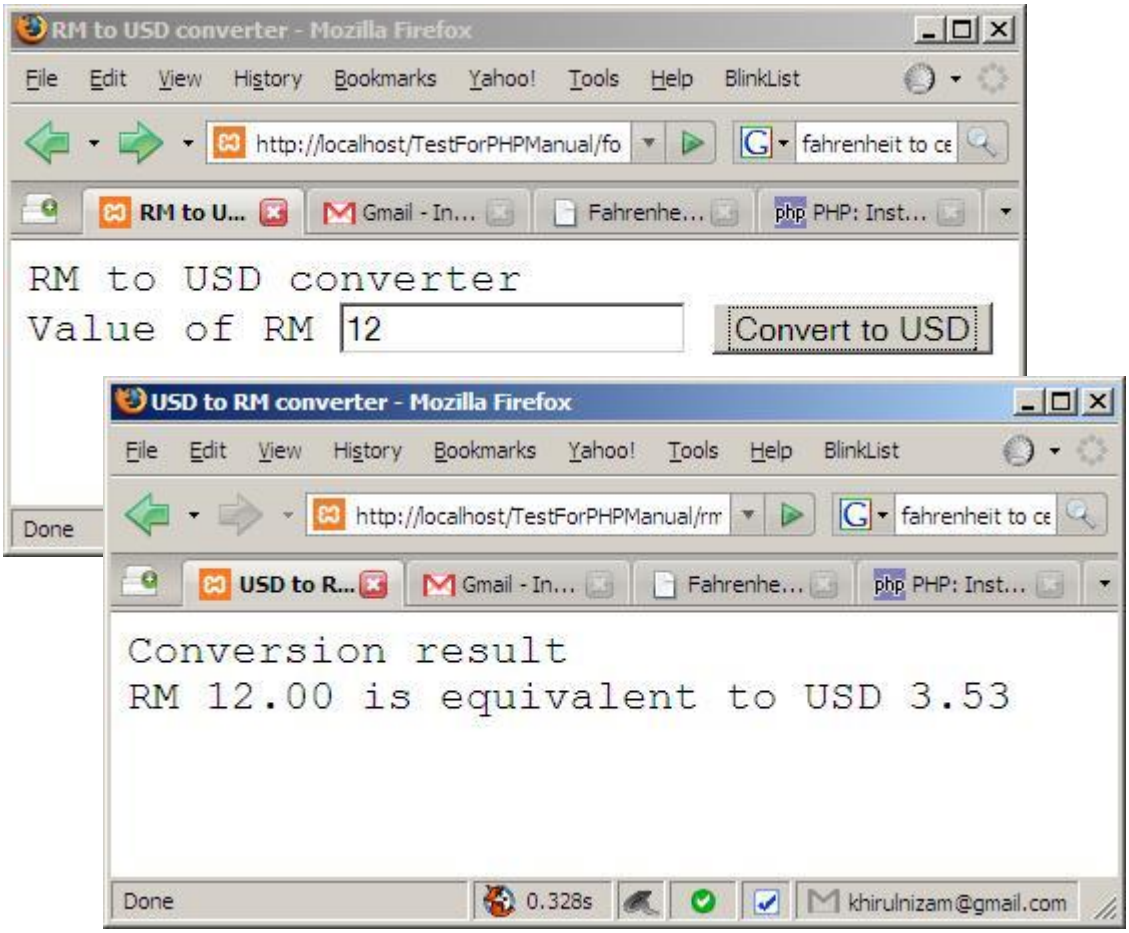
```
<html>
<head>
<title>USD to RM converter</title>
</head>

<body>
Conversion result<br>
<?php
        $rmvalue=$_GET["txtrm"];

        $usdvalue=$rmvalue/3.4;
        printf("RM %.2f",$rmvalue);
        echo " is equivalent to ";
        printf("USD %.2f",$usdvalue);

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

Step 3. Preview the page (in the browser) with the name *formRMUSDConverter.php,* key in a value in the text box and click the button “Convert to USD”. The browser should go to the next page named *rmusdconverter.php* (as defined in the form’s action) and display the value converted from RM into USD.


RM to USD converter - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/fo
fahrenheit to ce
RM to U...
Gmail - In...
Fahrenhe...
PHP: Inst...
RM to USD converter
Value of RM 12
Convert to USD
Done
USD to RM converter - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/rm
fahrenheit to ce
USD to R...
Gmail - In...
Fahrenhe...
PHP: Inst...
Conversion result
RM 12.00 is equivalent to USD 3.53
Done
0.328s
khirulnizam@gmail.com

Example 3: Simple statistical web application for calculating sum and average of five numbers.

Step 1. Create the form to receive five numbers from the user. Use the code below, and save as *form5numbers.php*.

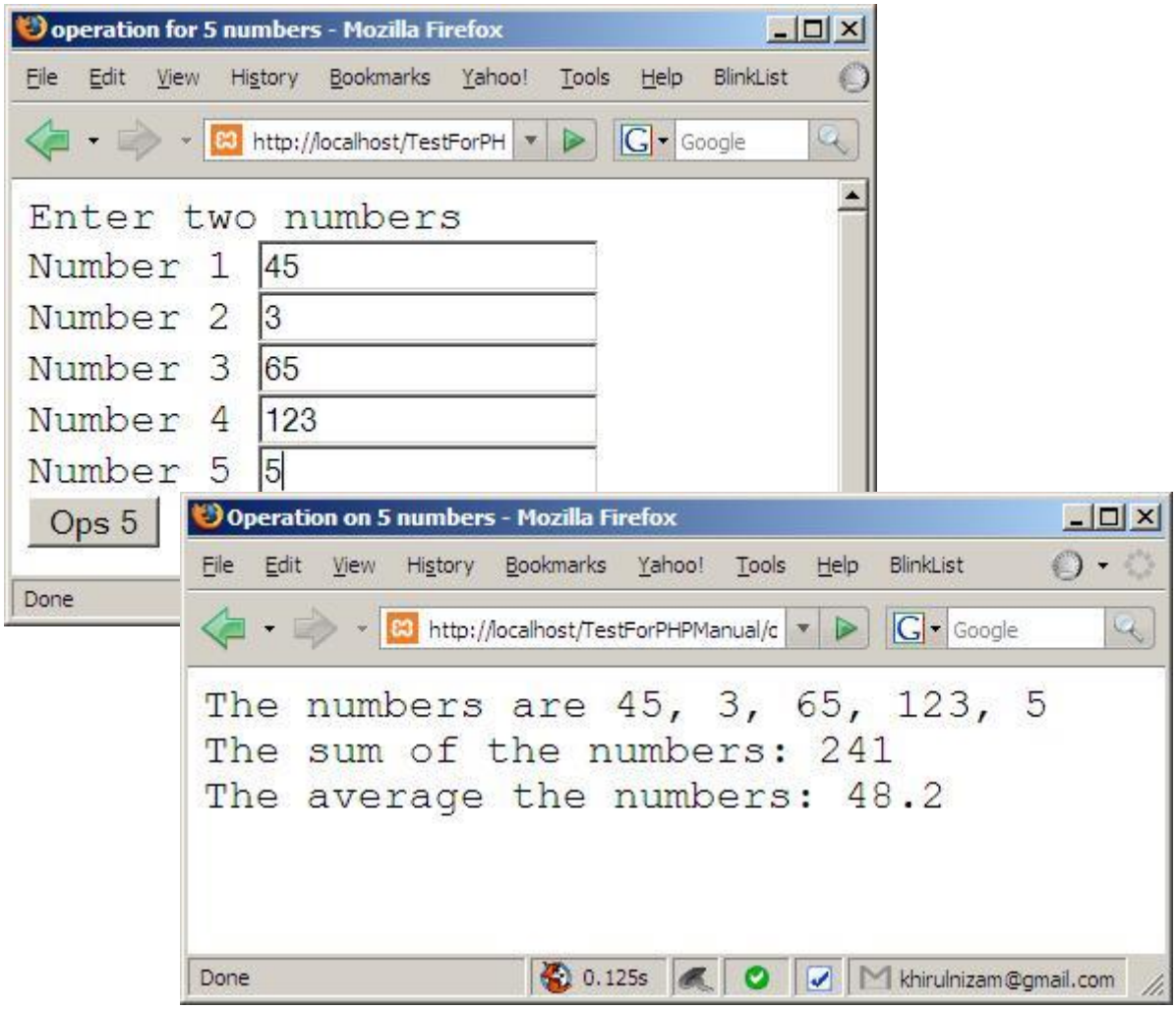
```
<html>
<head>
<title>operation for 5 numbers</title>
</head>
<body>
Enter two numbers<br>
<form method="GET" name="form5Numbers" action="ops5numbers.php">
        Number 1    <input type="text" name="num1"><br>
        Number 2    <input type="text" name="num2"><br>
        Number 3    <input type="text" name="num3"><br>
        Number 4    <input type="text" name="num4"><br>
        Number 5    <input type="text" name="num5"><br>
        <input type="submit" name="btnOps" value="Ops 5">
</form>
</body>
</html>
```

Step 2. Create another page to receive the numbers entered by the user. Save as *ops5numbers.php*.

```
<html>
<head>
<title>Operation on 5 numbers </title>
</head>
<body>
<?php
        //retrieve the numbers
        $n1=$_GET["num1"];
        $n2=$_GET["num2"];
        $n3=$_GET["num3"];
        $n4=$_GET["num4"];
        $n5=$_GET["num5"];
        $sum=$n1+$n2+$n3+$n4+$n5;
        $average=$sum/5;
        echo "The numbers are $n1, $n2, $n3, $n4, $n5<br>";
        echo "The sum of the numbers: $sum<br>";
        echo "The average the numbers: $average<br>";
?>
</body>
</html>
```

Step 3. View the form (*form5numbers.php*) in the browser and key-in a number in each of the boxes, and click the button.

Step 4. The sum and the average of the five numbers are displayed.


operation for 5 numbers - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPH
Google
Enter two numbers
Number 1 45
Number 2 3
Number 3 65
Number 4 123
Number 5 5
Ops 5
Done
Operation on 5 numbers - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help BlinkList
http://localhost/TestForPHPManual/c
Google
The numbers are 45, 3, 65, 123, 5
The sum of the numbers: 241
The average the numbers: 48.2
Done
0.125s
khirulnizam@gmail.com

Exercise 4.1: Mathematical Expression

1. What is the numerical value of each of the following expressions as evaluated by the PHP scripting language?
   i. (20/2)%2
   ii. 3%2
   iii. 10%(10/5)
   iv. 8%5*3+1
   v. 2+4*5
   vi. 10/(4+1)%3
   vii. 1+5*3/3+5%4

2. Convert the following mathematical formula into PHP expression.
   ii. $y = b^2-4ac$
   iii. $b = 2bx^2 + 3(a+b)x + 3a$
   iv. $m = \frac{y_2 - y_1}{x_2 - x_1}$
   v. $n = a^3+b^2$
   vi. $d = (x+10)(x-3)$
   vii. $C = \frac{5}{9}(F-32)$

*(*For all the problems in question 3-10, you are required to create a HTML form for the input page, and another page to display the result of the calculation).*

3. Multiplication of two numbers entered by user.

4. Receive a length in kilometer, convert the length into meter.

5. Convert a currency value from US dollar (USD) into Malaysian ringgit (RM).

6. Convert a distance in mile into kilometer.
   Given: 1 mile is equivalent to 1.609344 kilometers.

7. Convert temperature value from Fahrenheit into Celsius.
   Given: Celsius=(5/9)*(Fahrenheit-32)

8. You are given the following page with a form to receive the price of item and the quantity for the user to buy the item.

9. You are to develop a loan payment calculator. The first page is the form for the user to key in the sum of loan, the interest rate, and the number of years to settle the payment. After the user entered all the information, the user will click a button and the information will be submitted to the server. In the server there is another file waiting to calculate and display the monthly installment to be paid, and the total sum of payment to be made.

Comparison Operators.

The comparison operators are used to compare values.

Operator	Meaning	Example
==	Equal to	($nombor == 100)
!=	Not equal to	($huruf != 'Z')
<	Less than	($nom1 < $nom2)
>	Greater than	($markah > 80)
<=	Less than or equal to	($bil <= 10)
>=	Greater than or equal to	($bil >= $input)

Logical Operators.

The logical operators are used for Boolean expression or logical comparison. The Boolean expression produces **`true`** if the statement is true and **`false`** if the statement is false.

Operator	Meaning	Example
&&	Logical AND	($x==1) && ($y==2)
||	Logical OR	($huruf=='A') \|\| ($nom>10)
!	Logical NOT	(!EOF) !($nombor > 100)

Examples of expression:

Statement	convert to PHP expression
x is between 10 to 50. (*Pembolehubah x bernilai di antara 10 hingga 50*)	`($x>=10)&&($x<=50)`
y is not equal to f or q. (y tak sama dengan f atau q)	`($y!=$f)||( $y!=$q)`

z is multiplication of 5.
(*z adalah gandaan 5*)
 	
z is not multiplication of 5.
(*z bukan gandaan 5*)
	`($z%5 == 0)`
	
	`($z%5 != 0)`

Var1 is negative number. (*Var1 ialah nombor negatif*)	`($Var1 < 0)`
Var2 is between 0 to 100, but 20 to 50 are excluded. (*Var2 adalah antara 0 hingga 100, tapi antara 20 hingga 50 tak termasuk*)	`(($Var2>=0)&&($Var2<=100)&&` `(!(($Var2>=20)&&($Var2<=50)))`

Other Operators

Operator	Meaning	Example

++	Unary pre-increment (add 1 to the value in a variable)
	Unary post-increment (add 1 to the value in a variable *later*)
		`$nom++;`
		
		`++$nom;`

--	Unary pre-decrement (subtract 1 from the value in a variable)
	Unary post-decrement (subtract 1 from the value in a variable *later*)
		`$jum--;`
		
		`--$jum;`

{ }	Parenthesis (block of statements)	`if($val==0){`
		`.....`
		}

[ ]	Array subscript (index number for array variable)	`$j[1]=100;`
.	Concatenate (merge several separated strings)	`$name="Kerul"." Rahman";` `$name+=" Abd Rahman";`
;	Semicolon (to end a statement)	`$a=5;` `$a+=10;`
( )	Bracket As the arithmetic expression separator.	 `$average=($a+$b+$c)/3;`
//	Line comment	
/* …*/	Block comment	

Exercise 4.2: Boolean Expressions

1. What is the value of the following Boolean expressions (true/false)?
   a. `4 > 1`
   b. `!((4+1)>5)`
   c. `'c'=='C'`
   d. `!(false)&&(true)`
   e. `(4>4)||(100%10==5)`

2. Create the Boolean expression (in PHP format) for each of the following statements.
   a. $X is positive integer.
   b. $Z is negative integer.
   c. $W is between –100 to 100.
   d. $Y is multiplication of 10.
   e. $A is multiplication of 3.
   f. $num1 is not equal to 10.
   g. $num2 is even number.
   h. $num3 is odd number.
   i. $num4 is a prime number.
   j. $var2 is devisable by 5.
   k. $gaji is greater than 5000.
   l. $huruf1 is vocal.
   m. $huruf2 is consonant.
   n. $huruf3 is upper case.
   o. $huruf4 is lower case.
   p. $huruf5 is equal to 'm'.
   q. $huruf6 is not equal to 'Z'.