PhP programs

1) Fibonacci Series

Fibonacci series is the one in which you will get your next term by adding previous two numbers.

For example,

1. 0 1 1 2 3 5 8 13 21 34
2. Here, 0 + 1 = 1
3. 1 + 1 = 2
4. 3 + 2 = 5

and so on.

Logic:

- Initializing first and second number as 0 and 1.
- Print first and second number.
- From next number, start your loop. So third number will be the sum of the first two numbers.

Example:

We'll show an example to print the first 12 numbers of a Fibonacci series.

```
1. <?php
2. $num = 0;
3. $n1 = 0;
```

```
4. $n2 = 1;
5. echo "<h3>Fibonacci series for first 12 numbers: </h3>";
6. echo "\n";
7. echo $n1.' '.$n2.' ';
8. while ($num < 10 )
9. {
10.    $n3 = $n2 + $n1;
11.    echo $n3.' ';
12.    $n1 = $n2;
13.    $n2 = $n3;
14.    $num = $num + 1;
15. ?>
```

Output:

2) Leap Year Program

A leap year is the one which has 366 days in a year. A leap year comes after every four years. Hence a leap year is always a multiple of four.

For example, 2016, 2020, 2024, etc are leap years.

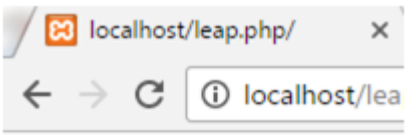
Leap Year Program

This program states whether a year is leap year or not from the specified range of years (1991 - 2016).

Example:

```php
<?php
function isLeap($year)
{
    return (date('L', mktime(0, 0, 0, 1, 1, $year))==1);
}
//For testing
for($year=1991; $year<2016; $year++)
{
    If (isLeap($year))
    {
        echo "$year : LEAP YEAR<br />\n";
    }
    else
    {
        echo "$year : Not leap year<br />\n";
    }
}
?>
```

Output:



1991 : Not leap year
1992 : LEAP YEAR
1993 : Not leap year
1994 : Not leap year
1995 : Not leap year
1996 : LEAP YEAR
1997 : Not leap year
1998 : Not leap year
1999 : Not leap year
2000 : LEAP YEAR
2001 : Not leap year
2002 : Not leap year
2003 : Not leap year
2004 : LEAP YEAR
2005 : Not leap year
2006 : Not leap year
2007 : Not leap year
2008 : LEAP YEAR
2009 : Not leap year
2010 : Not leap year
2011 : Not leap year
2012 : LEAP YEAR
2013 : Not leap year
2014 : Not leap year
2015 : Not leap year

3) Factorial Program

The factorial of a number n is defined by the product of all the digits from 1 to n (including 1 and n).

For example,

1. 4! = 4*3*2*1 = 24
2. 6! = 6*5*4*3*2*1 = 720

Note:

- It is denoted by n! and is calculated only for positive integers.
- Factorial of 0 is always 1.

The simplest way to find the factorial of a number is by using a loop.

There are two ways to find factorial in PHP:

- Using loop
- Using recursive method

Logic:

- Take a number.
- Take the descending positive integers.
- Multiply them.

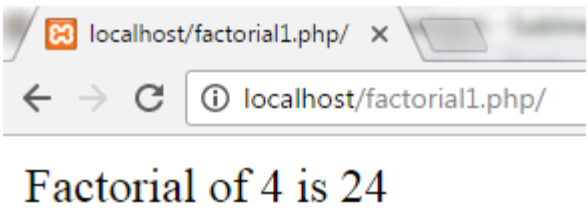
Factorial in PHP

Factorial of 4 using for loop is shown below.

Example:

```
<?php
$num = 4;
$factorial = 1;
for ($x=$num; $x>=1; $x--)
{
  $factorial = $factorial * $x;
}
echo "Factorial of $num is $factorial";
?>
```

Output:



4) Armstrong Number

An Armstrong number is the one whose value is equal to the sum of the cubes of its digits.

0, 1, 153, 371, 407, 471, etc are Armstrong numbers.

For example,

1. 407 = (4*4*4) + (0*0*0) + (7*7*7)
2. = 64 + 0 + 343
3. 407 = 407

Logic:

- Take the number.
- Store it in a variable.
- Take a variable for sum.
- Divide the number with 10 until quotient is 0.
- Cube the remainder.
- Compare sum variable and number variable.

Armstrong number in PHP

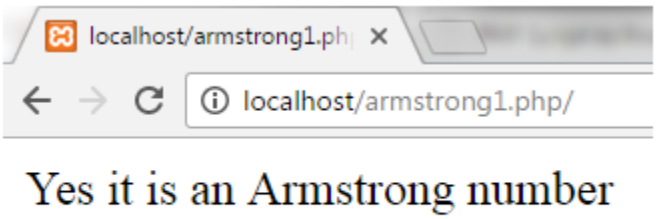
Below program checks whether 407 is Armstrong or not.

Example:

```
1. <?php
2. $num=407;
3. $total=0;
```

```
4. $x=$num;
5. while($x!=0)
6. {
7. $rem=$x%10;
8. $total=$total+$rem*$rem*$rem;
9. $x=$x/10;
10. }
11. if($num==$total)
12. {
13. echo "Yes it is an Armstrong number";
14. }
15. else
16. {
17. echo "No it is not an armstrong number";
18. }
19. ?>
```

Output:



5) Palindrome Number

A palindrome number is a number which remains same when its digits are reversed.

For example, number 24142 is a palindrome number. On reversing it we?ll get the same number.

Logic:

- Take a number.
- Reverse the input number.
- Compare the two numbers.
- If equal, it means number is palindrome

Palindrome Number in PHP

Example:

```
1. <?php
2. function palindrome($n){
3. $number = $n;
4. $sum = 0;
5. while(floor($number)) {
6. $rem = $number % 10;
7. $sum = $sum * 10 + $rem;
8. $number = $number/10;
9. }
10. return $sum;
11. }
12. $input = 1235321;
13. $num = palindrome($input);
14. if($input==$num){
15. echo "$input is a Palindrome number";
16. } else {
17. echo "$input is not a Palindrome";
18. }
19. ?>
```

Output: