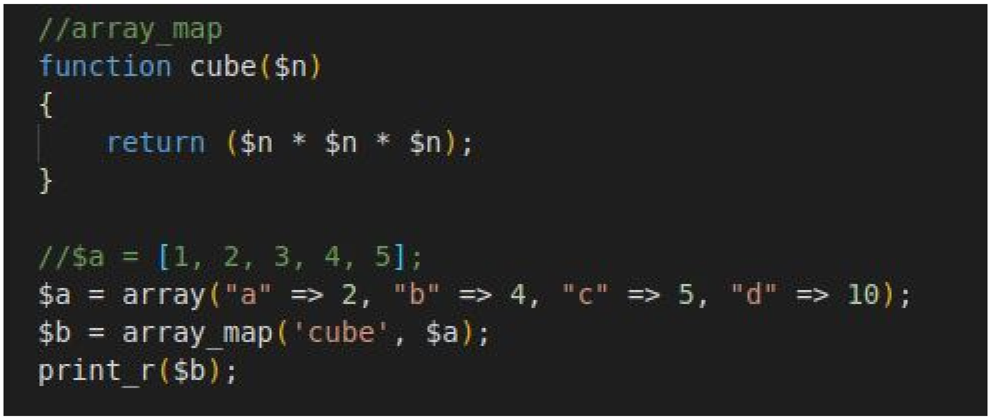
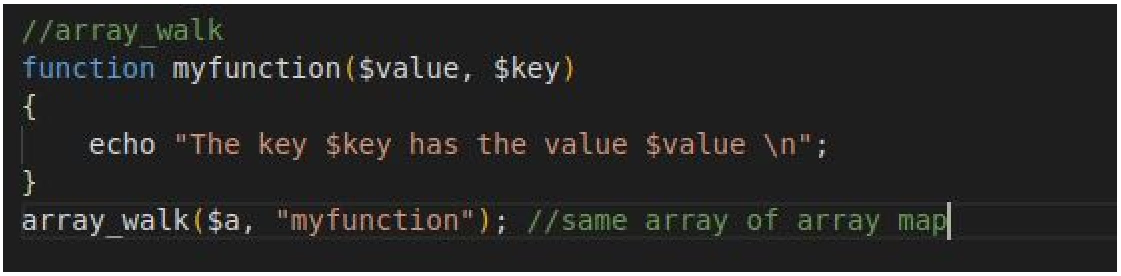
1. Difference between array_map, array_walk and array_filter. Explain with appropriate examples?

- ***The array_map()*** *is an inbuilt function in PHP and it helps to modify all elements one or more arrays according to some user-defined condition in an easy manner. It basically, sends each of the elements of an array to a user-defined function and returns an array with new values as modified by that function.*



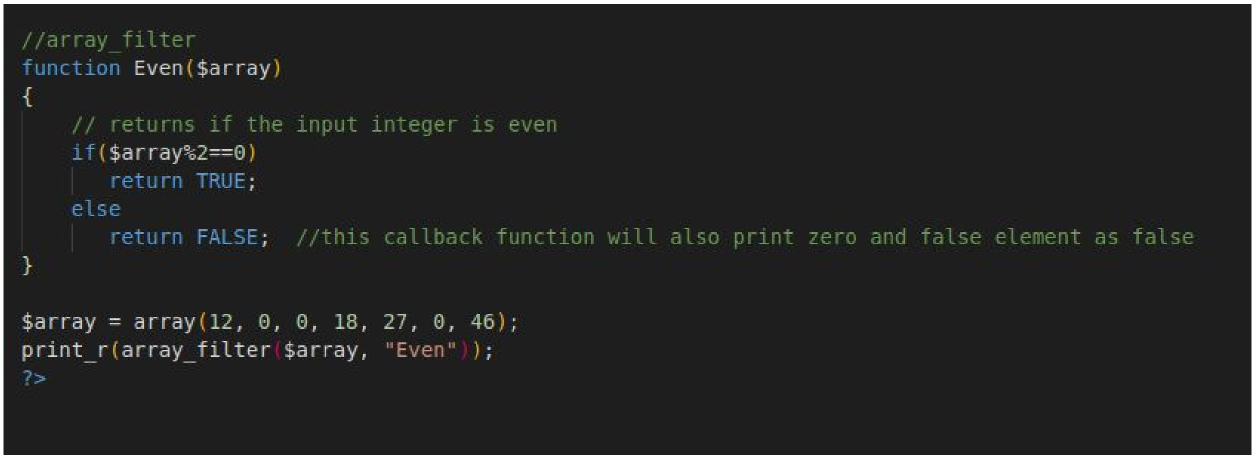
```
//array_map
function cube($n)
{
    return ($n * $n * $n);
}

//$a = [1, 2, 3, 4, 5];
$a = array("a" => 2, "b" => 4, "c" => 5, "d" => 10);
$b = array_map('cube', $a);
print_r($b);
```

- ***The array_walk()*** *function is an inbuilt function in PHP. The array_walk() function walks through the entire array regardless of pointer position and applies a callback function or user-defined function to every element of the array. The array element's keys and values are parameters in the callback function.*



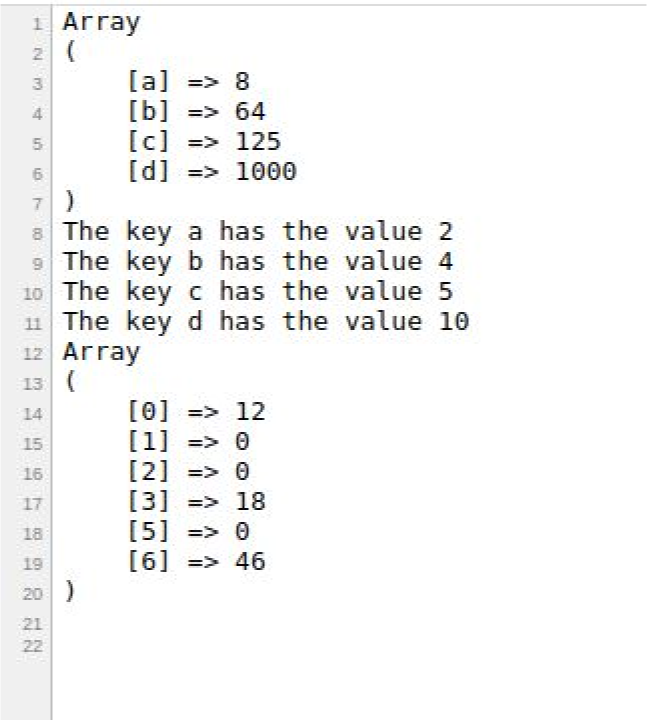
```
//array_walk
function myfunction($value, $key)
{
    echo "The key $key has the value $value \n";
}
array_walk($a, "myfunction"); //same array of array map
```

- ***Array_filter** is a built-in function in PHP is used to filter the elements of an array using a user-defined function which is also called a callback function.*



```
//array_filter
function Even($array)
{
    // returns if the input integer is even
    if($array%2==0)
        return TRUE;
    else
        return FALSE;  //this callback function will also print zero and false element as false
}

$array = array(12, 0, 0, 18, 27, 0, 46);
print_r(array_filter($array, "Even"));
?>
```

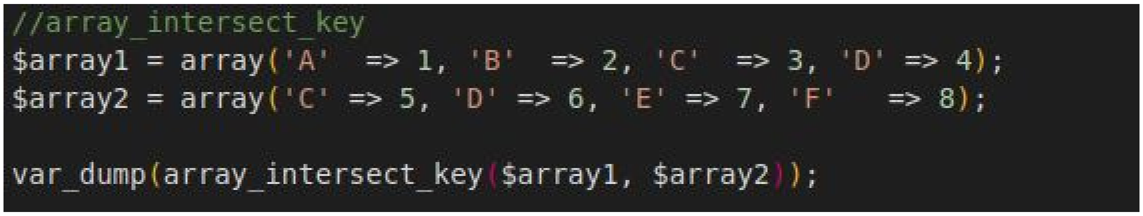
- **Outputs of all**



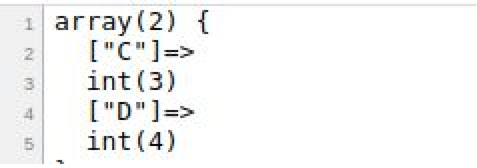
```
Array
(
    [a] => 8
    [b] => 64
    [c] => 125
    [d] => 1000
)
The key a has the value 2
The key b has the value 4
The key c has the value 5
The key d has the value 10
Array
(
    [0] => 12
    [1] => 0
    [2] => 0
    [3] => 18
    [5] => 0
    [6] => 46
)
```

2. Explain following array functions with examples : array_intersect_key, array_intersect_uassoc, array_product, compact, extract, array_pad, array_splice

- array_intersect_key — Computes the intersection of arrays using keys for comparison

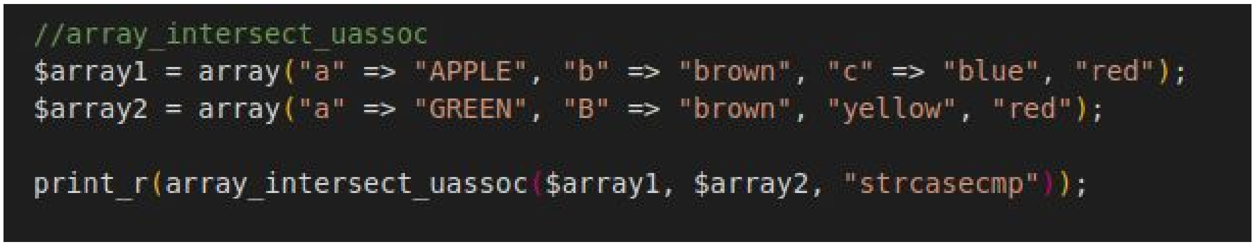


```
//array_intersect_key
$array1 = array('A'  => 1, 'B'  => 2, 'C'  => 3, 'D' => 4);
$array2 = array('C' => 5, 'D' => 6, 'E' => 7, 'F'   => 8);

var_dump(array_intersect_key($array1, $array2));
```

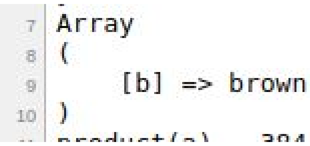


```
array(2) {
  ["C"]=>
  int(3)
  ["D"]=>
  int(4)
}
```

- array_intersect_uassoc — Computes the intersection of arrays with additional index check, compares indexes by a callback function

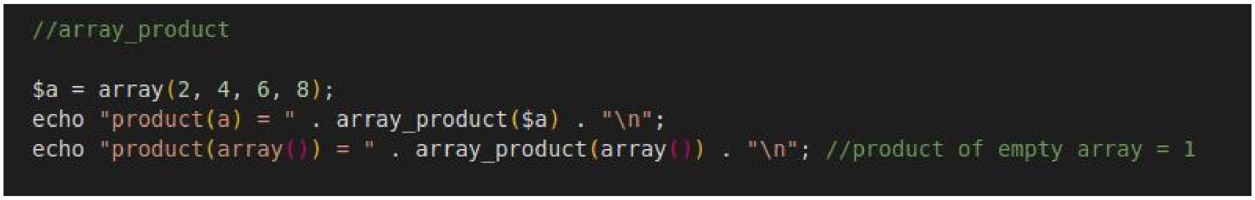


```
//array_intersect_uassoc
$array1 = array("a" => "APPLE", "b" => "brown", "c" => "blue", "red");
$array2 = array("a" => "GREEN", "B" => "brown", "yellow", "red");

print_r(array_intersect_uassoc($array1, $array2, "strcasecmp"));
```

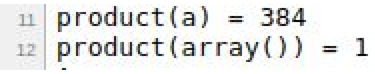


```
Array
(
    [b] => brown
)
product(a) = 384
```

- array_product — Calculate the product of values in an array

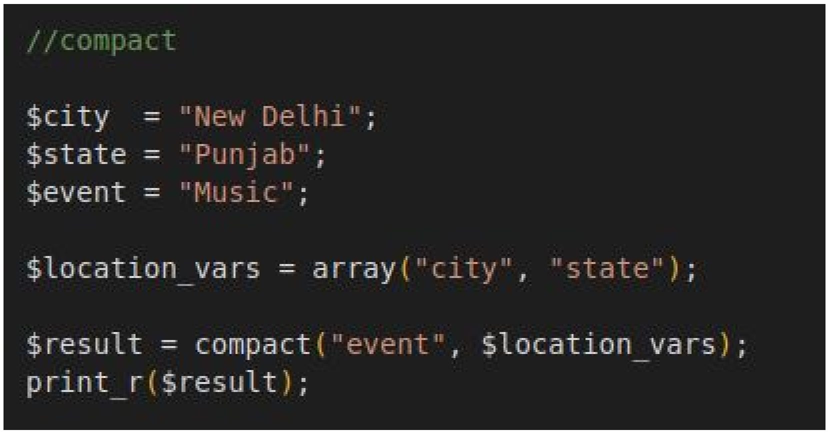


```
//array_product

$a = array(2, 4, 6, 8);
echo "product(a) = " . array_product($a) . "\n";
echo "product(array()) = " . array_product(array()) . "\n"; //product of empty array = 1
```

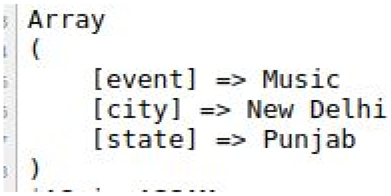


```
product(a) = 384
product(array()) = 1
```

- compact — Create array containing variables and their values

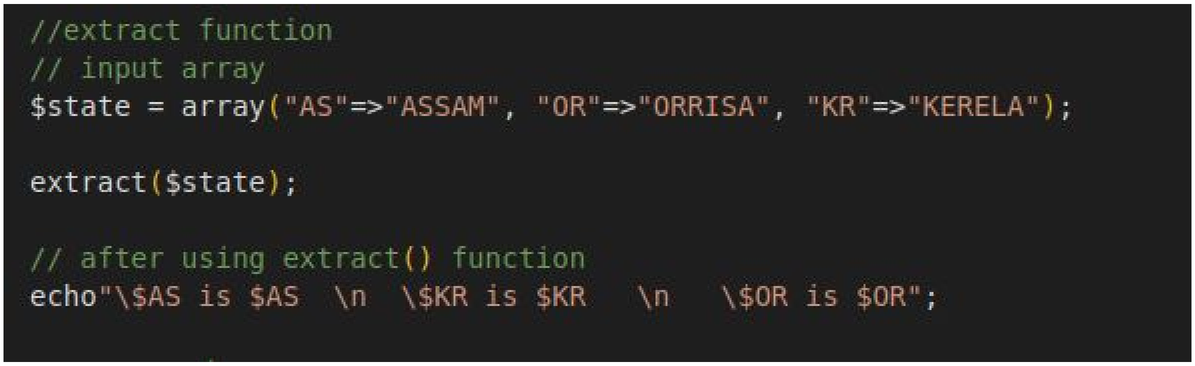


```
//compact

$city  = "New Delhi";
$state = "Punjab";
$event = "Music";

$location_vars = array("city", "state");

$result = compact("event", $location_vars);
print_r($result);
```

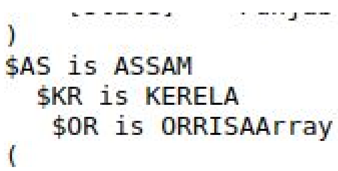


```
Array
(
    [event] => Music
    [city] => New Delhi
    [state] => Punjab
)
```

- The extract() Function is an inbuilt function in PHP. The extract() function does array to variable conversion. That is it converts array keys into variable names and array values into variable value. In other words, we can say that the extract() function imports variables from an array to the symbol table.

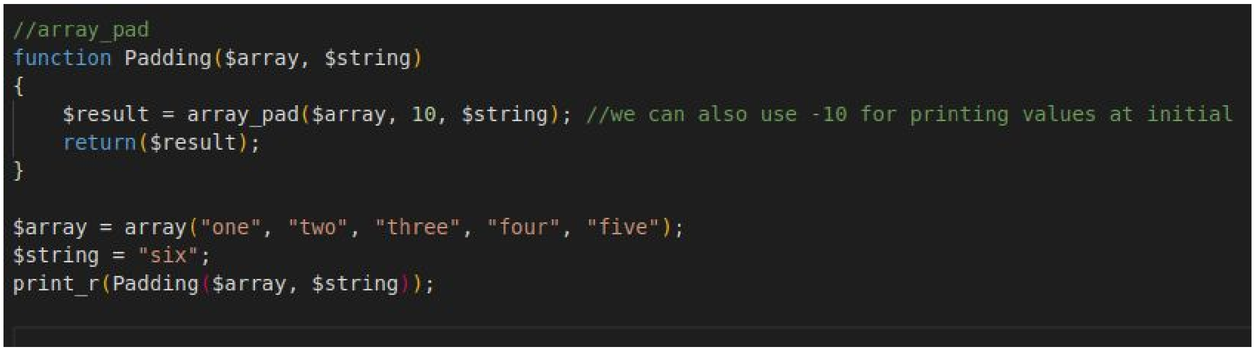


```
//extract function
// input array
$state = array("AS"=>"ASSAM", "OR"=>"ORRISA", "KR"=>"KERELA");

extract($state);

// after using extract() function
echo"\$AS is $AS  \n  \$KR is $KR   \n   \$OR is $OR";
```

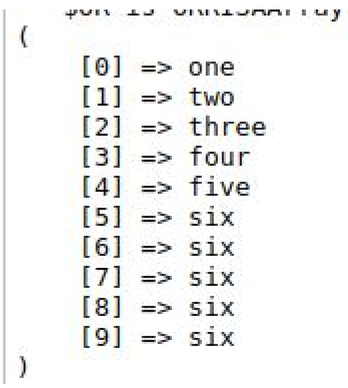


```
)
$AS is ASSAM
  $KR is KERELA
   $OR is ORRISAArray
(
```

- The array_pad() is a builtin function in PHP and is used to pad a value fixed number of time onto an array. This function inserts an element specified number of times into an array either at front or back.

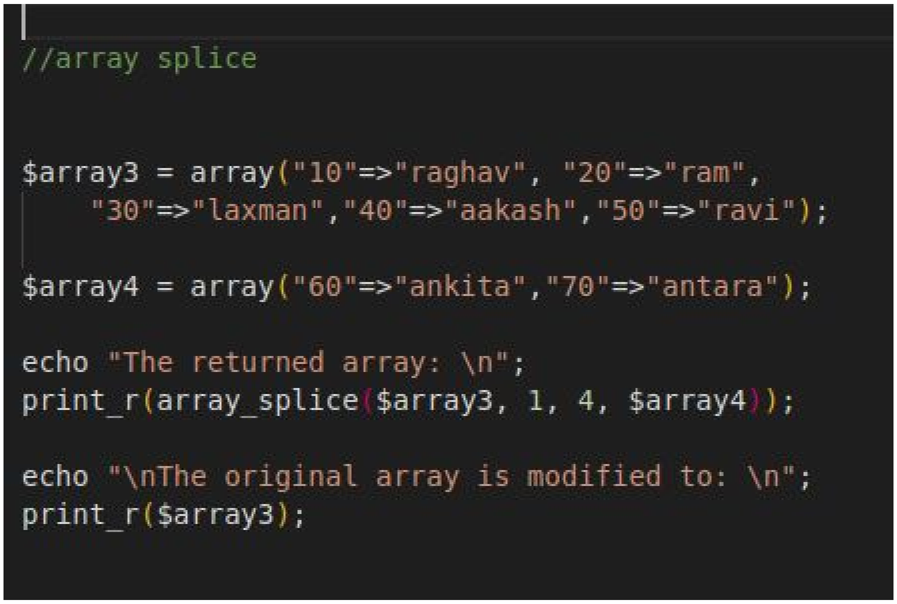


```
//array_pad
function Padding($array, $string)
{
    $result = array_pad($array, 10, $string); //we can also use -10 for printing values at initial
    return($result);
}

$array = array("one", "two", "three", "four", "five");
$string = "six";
print_r(Padding($array, $string));
```

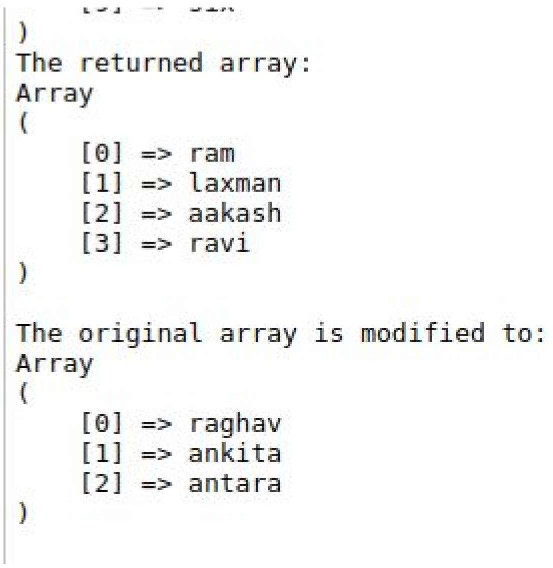


```
(
    [0] => one
    [1] => two
    [2] => three
    [3] => four
    [4] => five
    [5] => six
    [6] => six
    [7] => six
    [8] => six
    [9] => six
)
```

- This inbuilt function of PHP is an advanced and extended version of array_slice() function, where we not only can remove elements from an array but can also add other elements to the array. The function generally replaces the existing element with elements from other arrays and returns an array of removed or replaced elements.



```
//array splice


$array3 = array("10"=>"raghav", "20"=>"ram",
    "30"=>"laxman","40"=>"aakash","50"=>"ravi");

$array4 = array("60"=>"ankita","70"=>"antara");

echo "The returned array: \n";
print_r(array_splice($array3, 1, 4, $array4));

echo "\nThe original array is modified to: \n";
print_r($array3);
```

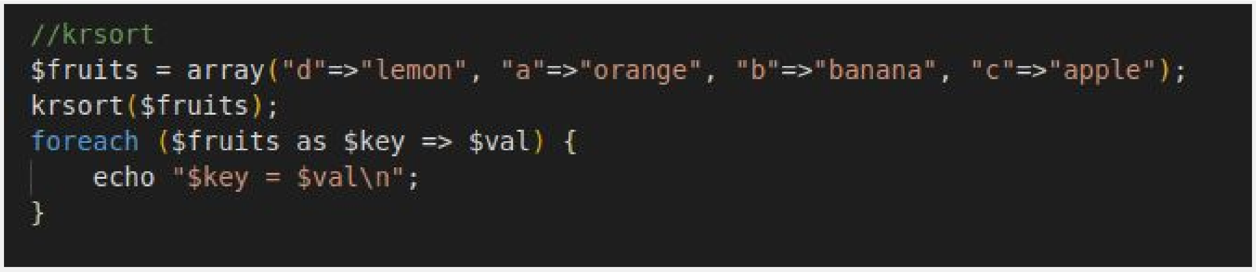


```
    [5] => six
)
The returned array:
Array
(
    [0] => ram
    [1] => laxman
    [2] => aakash
    [3] => ravi
)

The original array is modified to:
Array
(
    [0] => raghav
    [1] => ankita
    [2] => antara
)
```

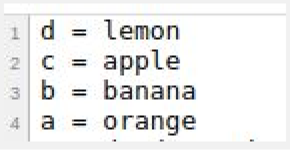
3. Explain following sort functions with examples : krsort, natsort, uasort, uksort.

- *KRSORT*

  Sort an associative array in descending order, according to the key:



```
//krsort
$fruits = array("d"=>"lemon", "a"=>"orange", "b"=>"banana", "c"=>"apple");
krsort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
```

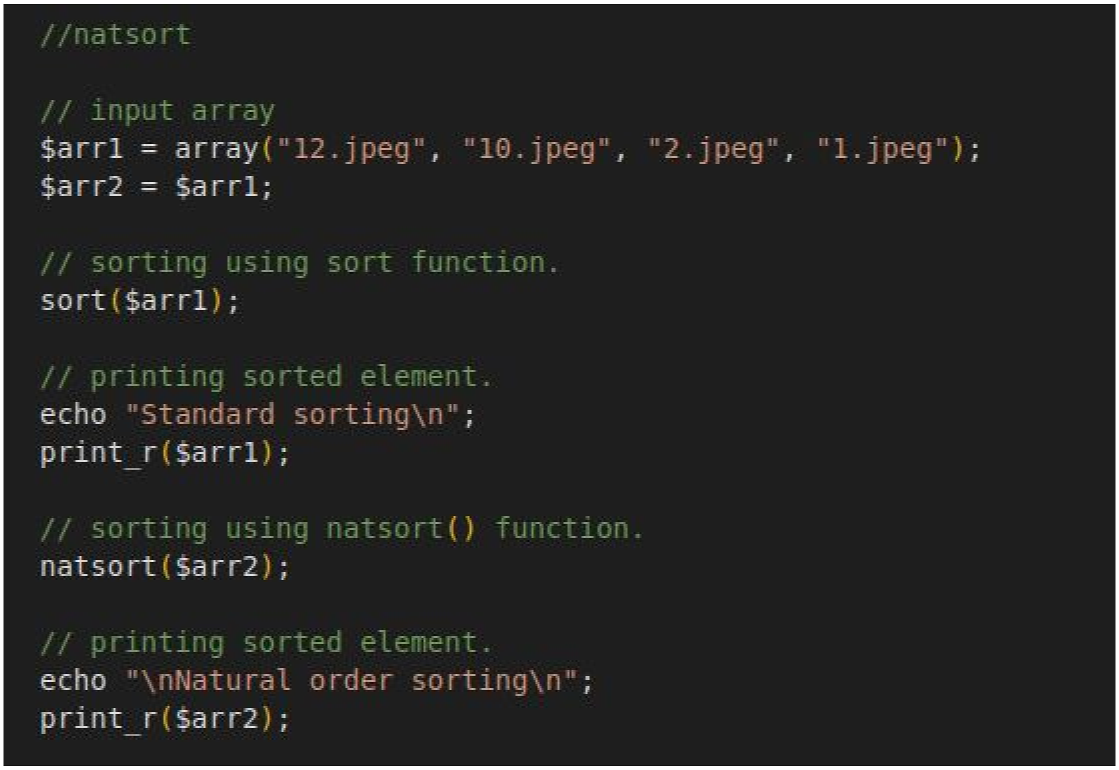


```
d = lemon
c = apple
b = banana
a = orange
```

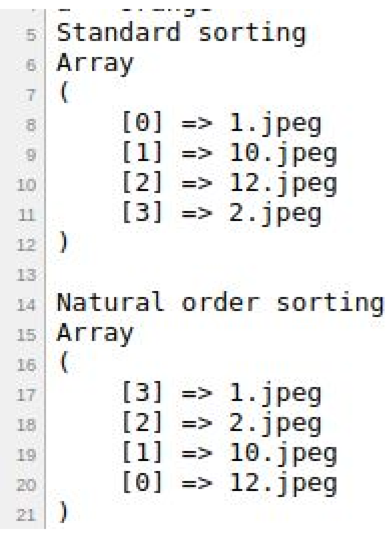
- *NATSORT*

  natsort — Sort an array using a "natural order" algorithm

  The natsort() function is an inbuilt function in PHP which is used to sort an array by using a “natural order” algorithm. The natural order tells the order to be used as a normal human being would use. That is, it does not check the type of value for comparison. For example, in string representation 30 is less than 7 according to the standard sorting algorithm as 3 comes before 7 lexicographically. But in natural order 30 is greater than 7.



```
//natsort

// input array
$arr1 = array("12.jpeg", "10.jpeg", "2.jpeg", "1.jpeg");
$arr2 = $arr1;

// sorting using sort function.
sort($arr1);

// printing sorted element.
echo "Standard sorting\n";
print_r($arr1);

// sorting using natsort() function.
natsort($arr2);

// printing sorted element.
echo "\nNatural order sorting\n";
print_r($arr2);
```



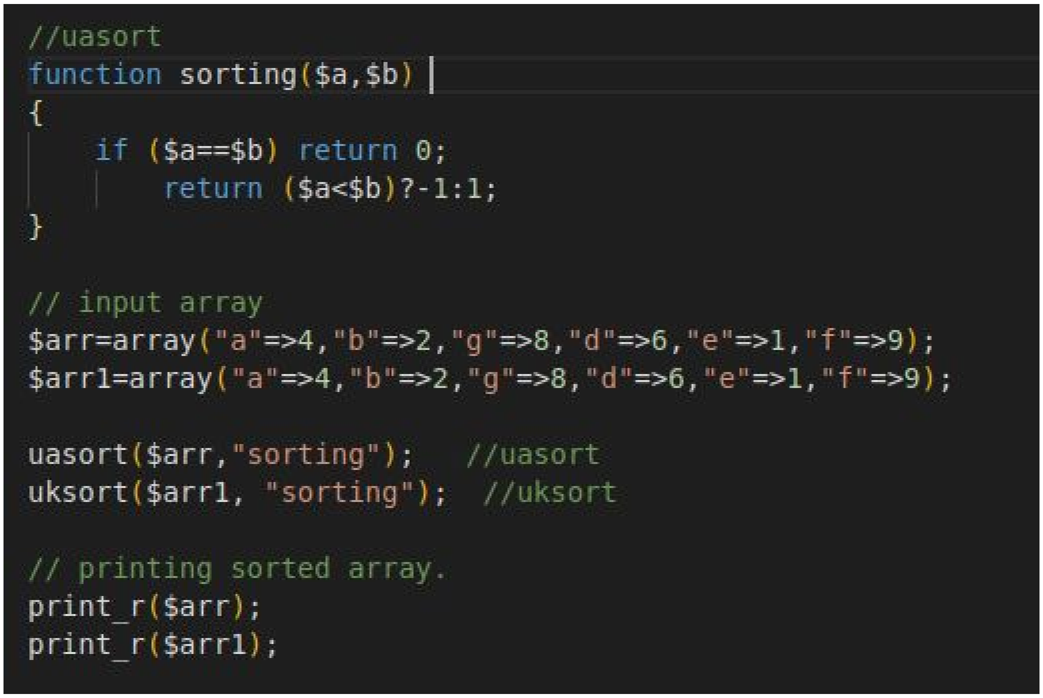
```
Standard sorting
Array
(
    [0] => 1.jpeg
    [1] => 10.jpeg
    [2] => 12.jpeg
    [3] => 2.jpeg
)

Natural order sorting
Array
(
    [3] => 1.jpeg
    [2] => 2.jpeg
    [1] => 10.jpeg
    [0] => 12.jpeg
)
```

- *UASORT*

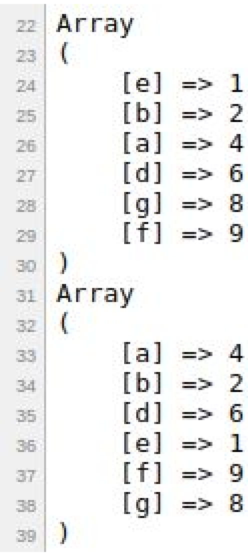
  The uasort() function is a builtin function in PHP and is used to sort an array such that array indices maintain their correlation with the array elements they are associated with, using a user-defined comparison function.

- *UKSORT*

  The *uksort()* function is a built-in function in PHP and is used to sort an array according to the keys and not values using a user-defined comparison function.



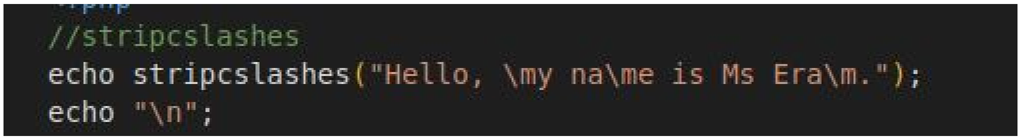
```
//uasort
function sorting($a,$b)
{
    if ($a==$b) return 0;
        return ($a<$b)?-1:1;
}

// input array
$arr=array("a"=>4,"b"=>2,"g"=>8,"d"=>6,"e"=>1,"f"=>9);
$arr1=array("a"=>4,"b"=>2,"g"=>8,"d"=>6,"e"=>1,"f"=>9);

uasort($arr,"sorting");   //uasort
uksort($arr1, "sorting");  //uksort

// printing sorted array.
print_r($arr);
print_r($arr1);
```



```
Array
(
    [e] => 1
    [b] => 2
    [a] => 4
    [d] => 6
    [g] => 8
    [f] => 9
)
Array
(
    [a] => 4
    [b] => 2
    [d] => 6
    [e] => 1
    [f] => 9
    [g] => 8
)
```

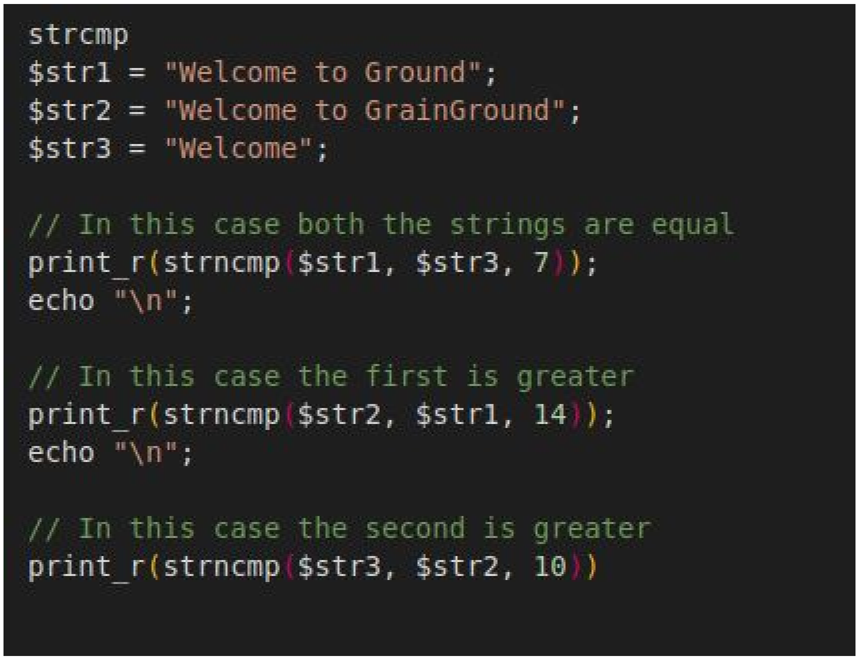
4. Explain following string functions with examples: stripcslashes, strncmp, strpbrk, strrchr, strrpos, strspn, strtr, vfprintf, vsprintf, wordwrap.

- **stripcslashes** — Un-quote string quoted with addcslashes()
  addcslashes() - Quote string with slashes in a C style



```
<?php
//stripcslashes
echo stripcslashes("Hello, \my na\me is Ms Era\m.");
echo "\n";
```

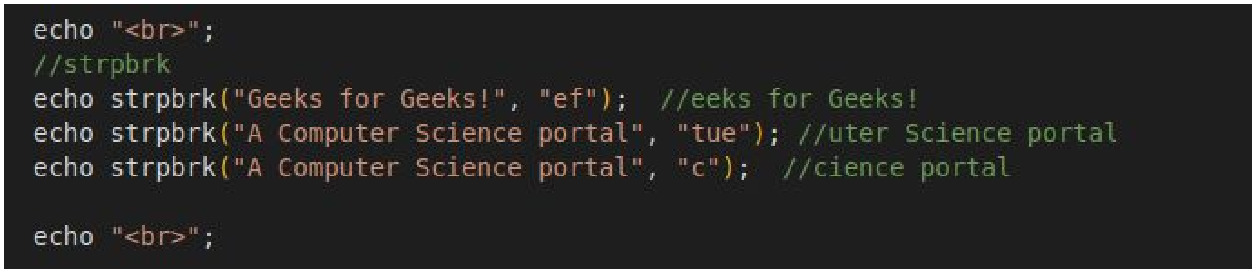
- **The strncmp()** is an inbuilt function in PHP which is used to compare first n character of two strings. This function is case-sensitive which points that capital and small cases will be treated differently, during comparison. This function compares two strings with first n character and tells whether the first string is greater or smaller or equal to the second string.



```
strcmp
$str1 = "Welcome to Ground";
$str2 = "Welcome to GrainGround";
$str3 = "Welcome";

// In this case both the strings are equal
print_r(strncmp($str1, $str3, 7));
echo "\n";

// In this case the first is greater
print_r(strncmp($str2, $str1, 14));
echo "\n";

// In this case the second is greater
print_r(strncmp($str3, $str2, 10))
```

strpbrk

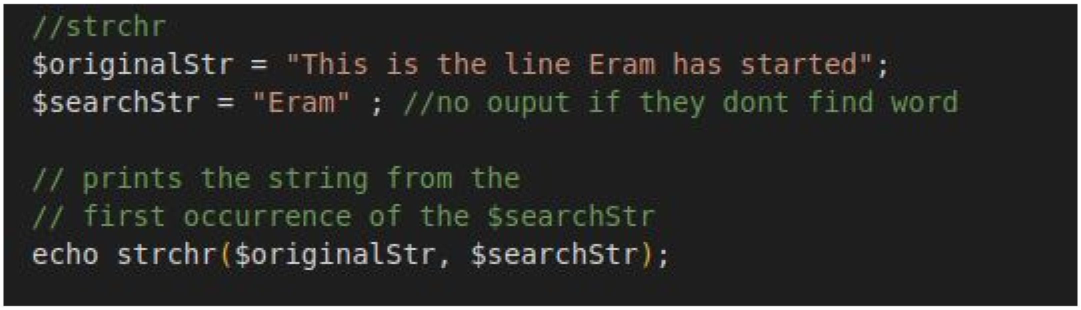
The strpbrk() function is an in-built function in PHP which searches a string for any of the specified characters. This function returns the rest of the string from where it found the first occurrence of any of the specified character. In case if none of the characters are found, it returns false. This function is **case-sensitive**.



```
echo "<br>";
//strpbrk
echo strpbrk("Geeks for Geeks!", "ef");  //eeks for Geeks!
echo strpbrk("A Computer Science portal", "tue"); //uter Science portal
echo strpbrk("A Computer Science portal", "c");  //cience portal

echo "<br>";
```

Strrchr

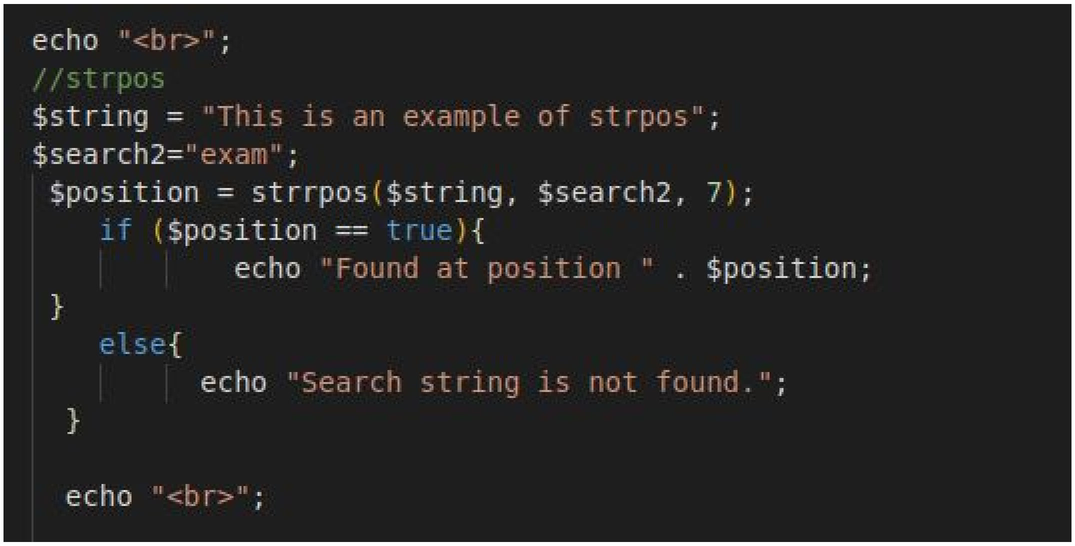
The **strchr()** function is a built-in function in PHP and is used to search for the first occurrence of a given string(say *searchStr*) in another string(say *originalStr*) and returns the rest of the string from *originalStr* starting from the first occurrence of *searchStr* in *orignalStr*.



```
//strchr
$originalStr = "This is the line Eram has started";
$searchStr = "Eram" ; //no ouput if they dont find word

// prints the string from the
// first occurrence of the $searchStr
echo strchr($originalStr, $searchStr);
```

- ## Strrpos

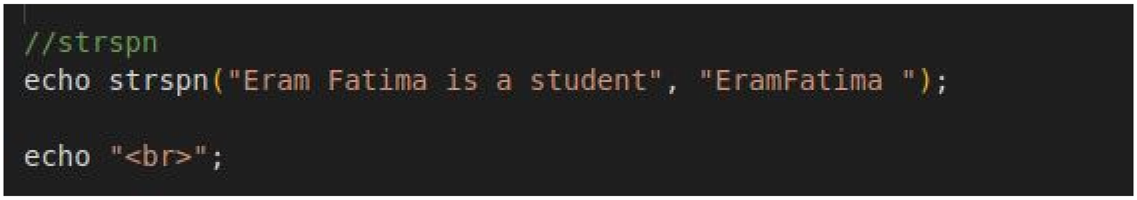
  strrpos() function helps us to find the position of the last occurrence of a string in another string. This function returns an integer value corresponding to the position of the last occurrence of the string. This function is **case-sensitive**, which means it treats uppercase and lowercase characters differently.



```
echo "<br>";
//strpos
$string = "This is an example of strpos";
$search2="exam";
 $position = strrpos($string, $search2, 7);
    if ($position == true){
            echo "Found at position " . $position;
 }
    else{
          echo "Search string is not found.";
  }

  echo "<br>";
```

- ## Strspn

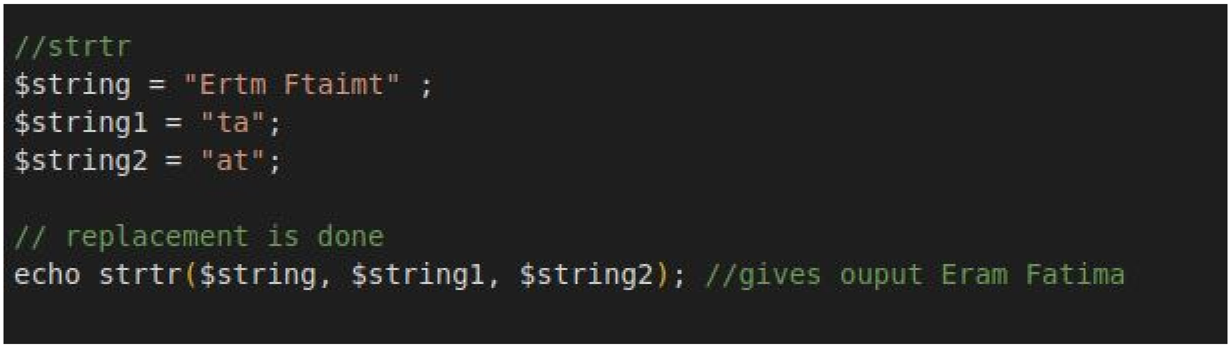
  The strspn() function is an in-built function in PHP which finds the length of the initial segment of a string consisting entirely of characters contained within a given list of characters passed as a parameter to the function.



```
//strspn
echo strspn("Eram Fatima is a student", "EramFatima ");

echo "<br>";
```

- ## Strtr

  The **strtr()** is an inbuilt function in PHP which is used to replace a substring in a string to a given string of characters. It also has the option to change a particular word to a different word in a string. The function is case sensitive.



```
//strtr
$string = "Ertm Ftaimt" ;
$string1 = "ta";
$string2 = "at";

// replacement is done
echo strtr($string, $string1, $string2); //gives ouput Eram Fatima
```

- ## Vfprintf

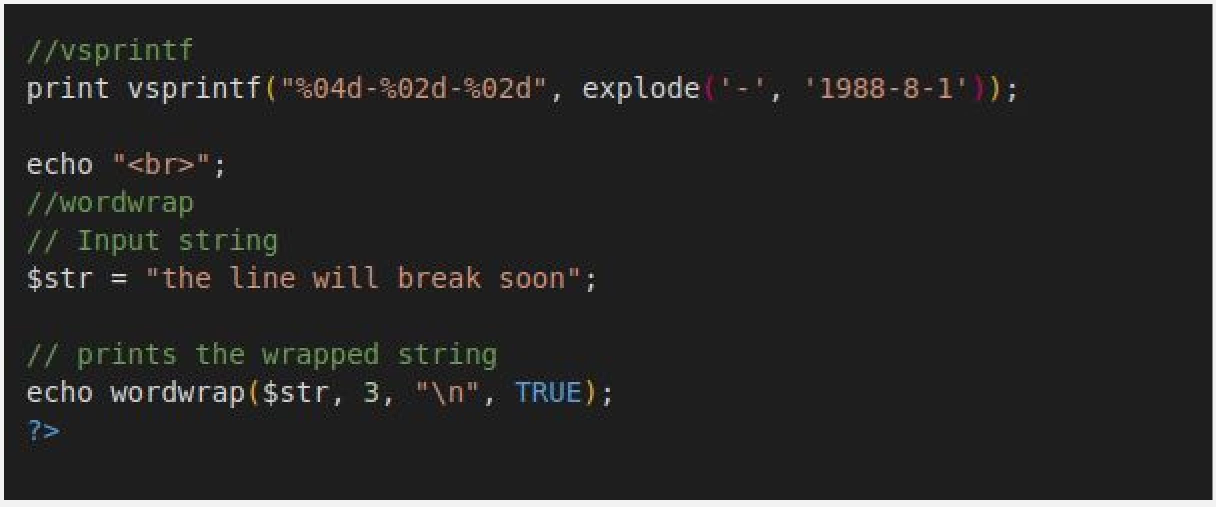
  vfprintf — Write a formatted string to a stream

- ## Vsprintf

  vsprintf — Return a formatted string

- ## wordwrap

  wordwrap — Wraps a string to a given number of characters

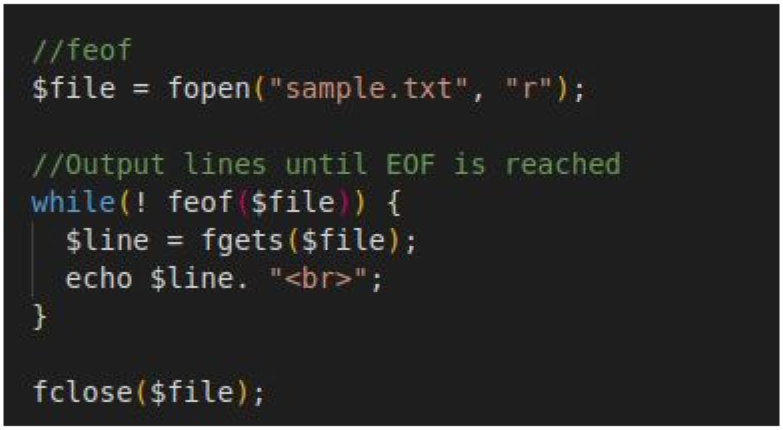


```
//vsprintf
print vsprintf("%04d-%02d-%02d", explode('-', '1988-8-1'));

echo "<br>";
//wordwrap
// Input string
$str = "the line will break soon";

// prints the wrapped string
echo wordwrap($str, 3, "\n", TRUE);
?>
```

5. Explain following file system functions with examples: feof, fgetcsv, fscanf, fseek, glob, umask, unlink, fileperms, filemtime

- ### FEOF

  The feof() function in PHP is an inbuilt function which is used to test for the end-of-file on a file pointer. It checks if the “end-of-file” has been reached or not. The feof() function is used for looping through the content of a file if the size of content is not known beforehand.



```
//feof
$file = fopen("sample.txt", "r");

//Output lines until EOF is reached
while(! feof($file)) {
  $line = fgets($file);
  echo $line. "<br>";
}

fclose($file);
```

- ### FGETCSV

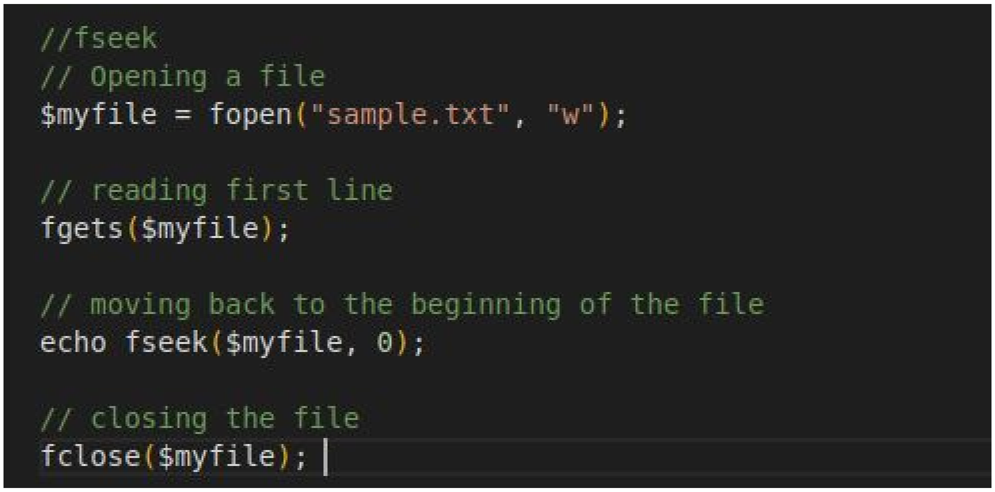
  Similar to fgets() except that fgetcsv() parses the line it reads for fields in CSV format and returns an array containing the fields read.

- ### FSCANF

  The fscanf() function parses the input from an open file according to the specified format.

- ## FSEEK

  The *fseek()* function in PHP is an inbuilt function which is used to seek in an open file. It moves the file pointer from its current position to a new position, forward or backward specified by the number of bytes. The file and the offset are sent as parameters to the *fseek()* function and it returns 0 on success, or -1 on failure.



```
//fseek
// Opening a file
$myfile = fopen("sample.txt", "w");

// reading first line
fgets($myfile);

// moving back to the beginning of the file
echo fseek($myfile, 0);

// closing the file
fclose($myfile);
```

- ## GLOB

The glob() function returns an array of filenames or directories matching a specified pattern. The glob() function returns.

1. An array containing the matched files/directories,
2. Returns an empty array if no file is matched,
3. FALSE on error.

```
<?php

foreach (glob("*.txt") as $filename) {

    echo "$filename size " . filesize($filename) . "\n";

}

      ?>
```

- ## UMASK

  Usually when you see `umask(0)` it should be followed directly by a call to `chmod()` to explicitly set the permissions needed on the newly created file or directory to something other than world-writable.

```
       <?php
$old = umask(0);
chmod("/path/some_dir/some_file.txt", 0755);
umask($old);

// Checking
if ($old != umask()) {
    die('An error occurred while changing back the umask');
}
      ?>
```

- ## UNLINK

  The unlink() function is an inbuilt function in PHP which is used to delete files. It is similar to UNIX unlink() function. The $filename is sent as a parameter which needs to be deleted and the function returns True on success and false on failure.

```
unlink( $filename, $context )
```

- **FILEPERMS**

  The fileperms() function in PHP is an inbuilt function which is used to return the permissions given to a file or a directory. The filename of the file whose permissions have to be checked is sent as a parameter to the function and it returns the permissions given to the file in the form of numbers on success and False on failure.

  ```
  echo fileperms("gfg.txt");
  ```

- **FILEMTIMES**The filemtime() function in PHP is an inbuilt function which is used to return the last time of a specified file when its content was modified. The filemtime() function returns the last time the file was changed as a Unix Timestamp on success and False on failure.

  ```
  echo filemtime("gfg.txt");
  ```