String functions

```
# index  0123456789012345
$name = "Austin Weale";
$length = strlen($name);                # 16
$cmp = strcmp($name, "Linda Guo");      # > 0
$index = strpos($name, "s");            # 2
$first = substr($name, 7, 4);           # "Weal"
$name = strtoupper($name);              # "AUSTIN WEALE"
```

PHP

Name	Java Equivalent
strlen	length
strpos	indexOf
substr	substring
strtolower, strtoupper	toLowerCase, toUpperCase
trim	trim
explode, implode	split, join

Array functions

function name(s)	description
count	number of elements in the array
print_r	print array's contents
array_pop, array_push, array_shift, array_unshift	using array as a stack/queue
in_array, array_search, array_reverse, sort, rsort, shuffle	searching and reordering
array_fill, array_merge, array_intersect, array_diff, array_slice, range	creating, filling, filtering
array_sum, array_product, array_unique, array_filter, array_reduce	processing elements

Array function example

```
$tas = array("MD", "BH", "KK", "HM", "JP");
for ($i = 0; $i < count($tas); $i++) {
  $tas[$i] = strtolower($tas[$i]);
}                                    # ("md", "bh", "kk", "hm", "jp")
$morgan = array_shift($tas);         # ("bh", "kk", "hm", "jp")
array_pop($tas);                     # ("bh", "kk", "hm")
array_push($tas, "ms");              # ("bh", "kk", "hm", "ms")
array_reverse($tas);                 # ("ms", "hm", "kk", "bh")
sort($tas);                          # ("bh", "hm", "kk", "ms")
$best = array_slice($tas, 1, 2);     # ("hm", "kk")
```

- the array in PHP replaces many other collections in Java
  - list, stack, queue, set, map, ...

EXPLODE FUNCTION.

The explode function is used to "Split a string by a specified string into pieces i.e. it breaks a string into an array". The explode function in PHP allows us to break a string into smaller text with each break occurring at the same symbol. This symbol is known as the delimiter.

Using the explode command we will create an array from a string. The explode() function breaks a string into an array.

PROGRAM:

```
<?php
$arr="Twinker Tanker Tank";
$str=explode(" ",$arr);
echo "<br>" . "OUTPUT:" . "<br>";
print_r($str);
?>
```

OUTPUT: Array ([0] => Twinker [1] => Tanker [2] => Tank)

IMPLODE FUNCTION.

The implode function in PHP is used to "join elements of an array with a string".The implode() function returns a string from elements of an array. It takes an array of strings and joins them together into one string using a delimiter (string to be used between the pieces) of your choice. The implode function in PHP is easily remembered as "array to string", which simply means that it takes an array and returns a string.It rejoins any array elements and returns the resulting string, which may be put in a variable.

PROGRAM:

```
<?php
$arr=array('Twinker','Tanker','Tank');
$str=implode(" and ",$arr);
echo "<br>" . "OUTPUT:" . "<br>";
print("$str");
?>
```

OUTPUT: Twinker and Tanker and Tank.

FOREACH loop

This loop is used to iterate over arrays. For every counter of loop, an array element is assigned and the next counter is shifted to the next element.

PROGRAM:

```
<?php
echo "<br>" . "OUTPUT:" . "<br>";
$fruits = array('a'=>'apple','b'=>'banana','c'=>'pea');
foreach ($fruits as $value)
echo $value."<br>";
echo "<br>";
foreach ($fruits as $key => $value)
echo "$key: $value<br>";
?>
```

OUTPUT:

```
apple
banana
pea

a: apple
b: banana
c: pea
```

RANGE FUNCTION.

The range() function creates an array containing a range of elements.

PROGRAM:

```
<?php
$arr=range(1,10);
echo "<br>" . "OUTPUT:" . "<br>";
print_r($arr);
?>
```

OUTPUT: Array ([0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6 [6] => 7 [7] => 8 [8] => 9 [9] => 10)

MIN AND MAX FUNCTION.

min function returns smallest number in array.

max function returns largest number in array.

PROGRAM

```
<?php
$arr=array(7,3,4,5,8,9,10,2);
echo "<br>" . "OUTPUT:" . "<br>";
echo "min is" .min($arr) . "<br>" ;
echo "max is" .max($arr);
?>
```

OUTPUT: min is2
max is10

ARRAY_SLICE FUNCTION.

The array_slice() function returns selected parts of an array.

array_slice(*ARRAY,START,LENGTH,PRESERVE*)

PROGRAM:

```
<?php
$rainbow=array('v','i','b','g','y','o','r');
$arr1=array_slice($rainbow,2,3);
$arr2=array_slice($rainbow,-5,3);
echo "<br>" . "OUTPUT:" . "<br>";
print_r ($arr1);
print_r ($arr2);
?>
```

OUTPUT:

Array ([0] => b [1] => g [2] => y) Array ([0] => b [1] => g [2] => y)

ADDING AND REMOVING ARRAY elements

array_pop() Deletes the last element of an array

array_push() Inserts one or more elements to the end of an array

array_unshift() Adds one or more elements to the beginning of an array

array_shift() Removes one or more elements to the beginning of an array

```
<?php
echo "<br>" . "OUTPUT:" . "<br>";
$mov=array('a','b','c','d');
array_shift($mov);
echo "<br>";
print_r($mov);
echo "<br>";
array_pop($mov);
print_r($mov);
echo "<br>";
array_push($mov, 'd');
print_r($mov);
echo "<br>";
array_unshift($mov, 'x');
print_r($mov);
```

REMOVAL OF DUPLICATE elements

The array_unique() function removes duplicate values from an array. If two or more array values are the same, the first appearance will be kept and the other will be removed.

```
<?php
echo "<br>" . "OUTPUT:" . "<br>";
 $arr=array('a','b','c','a','c');
 print_r($arr);
 $unique=array_unique($arr);
echo "<br>";
print_r($unique);
?>
```

OUTPUT:

```
Array ( [0] => a [1] => b [2] => c [3] => a [4] => c )
Array ( [0] => a [1] => b [2] => c )
```

RANDOMISE AND REVERSE elements

shuffle() Shuffles an array

array_reverse() Returns an array in the reverse order

PROGRAM:

```
<?php
echo "<br>" . "OUTPUT:" . "<br>";
$rainbow= array('v','i','b','g','y','o','r');
shuffle($rainbow);
print_r($rainbow);
$arr= array_Reverse($rainbow);
print_r($arr);
?>
```

OUTPUT: Array ([0] => r [1] => g [2] => y [3] => v [4] => o [5] => b [6] => i) Array ([0] => i [1] => b [2] => o [3] => v [4] => y [5] => g [6] => r)

ASSOCIATIVE SEARCHING.

The array_key_exists() function checks an array for a specified key, and returns true if the key exists and false if the key does not exist.

Tip: Remember that if you skip the key when you specify an array, an integer key is generated, starting at 0 and increases by 1 for each value. (See example 2)

array_key_exists(*KEY,ARRAY*)

```
<?php

echo "<br>" . "OUTPUT:" . "<br>";

$city=array('UP'=>'Kanpur','Delhi'=>'Delhi','Andhra Pradesh'=>'Hyderabad');

echo "array_key_exists('UP','Kanpur')";

?>
```

ASORT() AND KSORT() FUNCTION.

asort() - sort associative arrays in ascending order, according to the value.

ksort() - sort associative arrays

in ascending order, according to the key

```
<?php

$profile=array("Ename"=>'Susan',"lname"=>'Doc'
);

asort($profile);

echo "<br>" . "OUTPUT:" . "<br>";

print_r($profile);

?>
```

```
<?php

$profile=array("Ename"=>'Susan',"lname"=>'Doc'
);

ksort($profile);

print_r($profile);

?>
```

OUTPUT: Array ([lname] => Doc [Ename] => Susan) Array ([Ename] => Susan [lname] => Doc)

ARSORT() AND KRSORT() functions

arsort() - sort associative arrays in descending order, according to the value

krsort() - sort associative arrays in descending order, according to the key

```
<?php
$profile=array("Ename"=>'Susan',"lname"=>'Doc');
arsort($profile);
echo "<br>" . "OUTPUT:" . "<br>";
print_r($profile);
?>
```

OUTPUT: Array ([Ename] => Susan [lname] => Doc)

MERGING OF ARRAY.

The array_merge() is a builtin function in PHP and is used to merge two or more arrays into a single array. This function is used to merge the elements or values of two or more arrays together into a single array. The merging occurs in such a manner that the values of one array are appended at the end of the previous array. The function takes the list of arrays seperated by commas as parameter that are needed to be merged and returns a new array with merged values of arrays passed in parameter.

```
<?php
$dark=array('blue','brown','black');
$light=array('white','silver','yellow');
$color=array_merge($dark,$light);
echo "<br>" . "OUTPUT:" . "<br>";
print_r($color);
?>
```

OUTPUT:

Array ([0] => blue [1] => brown [2] => black [3] => white [4] => silver [5] => yellow)

COMPARING OF ARRAYS.

The array_intersect() function compares **the values** of two (or more) arrays, and returns the matches.

This function compares the values of two or more arrays, and return an array that contains the entries from *ARRAY1* that are present in *ARRAY2*, *ARRAY3*, etc.

Syntax

array_intersect(*ARRAY1,ARRAY2,ARRAY3...*);

```
<?php
echo "<br>" . "OUTPUT:" . "<br>";
$orange=array('R','Y');
$green=array('Y','B');
$common=array_intersect($orange,$green);
print_r($common);
echo "<br>";
$unique=array_diff($orange,$green);
print_r($unique);
echo "<br>";
?>
```

OUTPUT:

```
Array ( [1] => Y )
Array ( [0] => R )
```