ADVANCE ARRAY FUNCTIONS

- **Array_keys() :**

It returns the keys of input array in form of new array where the are stored as values.

The keys of new array are automatically incremented ingers, started from 0.

Array_values() :

It works exactly same as Array_keys() functions excepts that stored values are values from original array.

ADVANCE ARRAY FUNCTIONS

- **Array_conut_values() :**

This function take as array as arguments and returns new array.

In new array all the values from original array (passed array) will become the keys of returned array(new array) and the new values are the number of times the old value occurs in original array.

ADVANCE ARRAY FUNCTIONS

Array_flip() :

This functions changes the keys of array into values and values into keys.

Array_reverse():

Returns new array with key/values pair in reverse order.

ADVANCE ARRAY FUNCTIONS

Merging array :

array_merge(array1,array2)

This functions takes two or more arguments and returns renumbered new array.

Array_pad() :

It create some leading or following increasing size of array.

It takes input array as first argument, number of element to incrase as second argument and value to assigned as third element.

if second argument is positive then element will pad to end of array, if negative number then element will pad at starting of array.

if second argument is smaller then size of array then no padding will occurs.

STACK AND QUEUES

Stack and queue are data structure, frequently used in computer science.

Stack :

- ✓ **A stack is a container that stores values and supports LIFO(Last In First Out) behavior.**
- ✓ **This means that stack maintains an order of value stored.**

Act of adding into stack is Pushing in to stack.

Act of taking off the top is called poping the stack.

STACK AND QUEUES

Queue :

Queue is similar to the stack, but its behavior is FIFO (First In First Out).

Array_push() : This function takes an initial array argument and then any number of elements to push into the stack

The elements will in inserted at end of array from left to right.

Array_pop() : it takes an array and removes the element at end, returning it.

EXPLODE()

explode() is used to split a string into pieces.

Cutting of the string depending upon a substring or a single character, and returns the chunked form of the string as an array.

For example you have a string

```
$text="Ali,Jeff,Joel,Cortex,Charly";

$arr = explode(",", $text);
```

IMPLODE()

function implode() is oppisite to the explode function.

It rejoins any array elements and returns the resulting string, which may be put in a variable.

Your have an array
$arr = Array("Ali","Jeff","Joel","Cortex","Charly");

and you wish to combine it in a string, by putting a seperator '/' between each element of the array.

$str = implode("/",$arr);

Now $str will be Ali/Jeff/Joel/Cortex/Charly

EXTRACT() FUNCTION

The extract() function imports variables into the local symbol table from an array.

This function uses array keys as variable names and values as variable values.

For each element it will create a variable in the current symbol table.

This function returns the number of variables extracted on success.

EXTRACT() FUNCTION CONT.

Syntax

extract(array, extract_rules, prefix)

array : Required. Specifies the array to use

extract_rules : Optional.

Desc : The extract() function checks for invalid variable names and collisions with existing variable names. This parameter specifies how invalid and colliding names are treated.

EXTRACT() FUNCTION CONT.

Possible values:

- **EXTR_OVERWRITE - Default. On collision, the existing variable is overwritten**
- **EXTR_SKIP - On collision, the existing variable is not overwritten**
- **EXTR_PREFIX_SAME - On collision, the variable name will be given a prefix**
- **EXTR_PREFIX_ALL - All variable names will be given a prefix**

Continue....

EXTRACT() FUNCTION CONT.

- EXTR_PREFIX_INVALID - Only invalid or numeric variable names will be given a prefix
- EXTR_IF_EXISTS - Only overwrite existing variables in the current symbol table, otherwise do nothing
- EXTR_PREFIX_IF_EXISTS - Only add prefix to variables if the same variable exists in the current symbol table
- EXTR_REFS - Extracts variables as references. The imported variables are still referencing the values of the array parameter

EXTRACT() FUNCTION CONT.

- **Optional.**

- **Description :**

If EXTR_PREFIX_SAME, EXTR_PREFIX_ALL, EXTR_PREFIX_INVALID or EXTR_PREFIX_IF_EXISTS are used in the extract_rules parameter, a specified prefix is required. This parameter specifies the prefix. The prefix is automatically separated from the array key by an underscore character.