Arrays

* A composite data type
* Treated as maps, where values are associated or mapped to keys
* Keys can only be either integers or strings
* Keys must be unique
* For integer keys are numerically indexed (indexed arrays)
* For string keys are associative arrays
* Indexed arrays always begin with index 0

2 types: **Indexed** and **associated**

Indexed: search by $indexedArr[#];

$indexedArr = array(95, 98.6, ‘hello’);

Associated: search by key associated

$assocArr = array (‘grade’ => 95, ‘temp’ => 98.6, ‘greeting’ => ‘hello’ ) ;

echo $assocArr[‘greeting’]; // output hello

$dailyTemp = array (75,76,73,73);

When using function **var\_dump($dailyTemp);**

Array(4) { [0] => int(75 [1] => int(76) [2]=> int(73) [3] => int(73) }

$dailyTemp = array (

‘Sunday’ => 75,

‘Monday’ => 76,

‘Tuesday’ => 73,

‘Wednesday’ => 73,

);

echo $Array[0];

echo $dailyTemp[‘Sunday’] ;

Multi-dimension (array in an array) and print\_r() as var\_dump()

$addresses = array (

‘joe’ => array ( ‘city’ => array(‘Norfolk’ , ‘Richmond’),

‘state’ => ‘VA’

),

‘jan’ => array (‘city’ => ‘San Diego’,

‘state’ => ‘CA’),

);

echo ‘Joe lives in: ‘;

echo $addresses[‘joe’][‘city’][1];

echo ‘<br/>’;

echo ‘<br/>’;

echo ‘Jan lives in: ‘;

echo $addresses[‘jan’][‘city’];

echo ‘<br/>’;

echo ‘<br/>’;

output

Joe lives in: Richmond

Jan lives in: San Diego

**SIMPLE SCROLL**

**$matrix = array (**

**array( ‘1A’, ‘1B’, ‘1C’ ),**

**array( ‘2A’, ‘2B’, ‘2C’ ),**

**array( ‘3A’, ‘3B’, ‘3C’ ),**

**);**

**for ($i = 0; $i < count($matrix); $i++ )**

**{**

**$row = $matrix[$i];**

**$j = 0;**

**while ($j < count($row))**

**{**

**//output each value in row**

**echo $row[ $j ] . ‘ ‘;**

**$j++**

**}**

**echo ‘<br/>’;**

**}**

**FUNCTIONS**

* current() return the value of the array element at the current array cursor position
* key() returns the key of the array element at the current cursor position
* next() advance array by one
* prev() moves back
* reset() set the cursor to the first element
* end() set the cursor to the last element

|  |  |
| --- | --- |
| **NUMERIC** | **ASSOCIATED** |
| $indexArr = array (1,2,3 );  $i = 0;  while ($i < count($indexArr) )  {  // retrieve current array value  $currVal = current($indexArr);    echo “\$currVal = $currVal <br/>”;  next($indexArr); // advance 1 position  $i++;  } | **$assoc1 = array (‘One’ => 1,**  **‘Two’ => 2,**  **‘Three’ => 3);**  **$i=0;**  **reset($assoc1);**  **while ($i < count($assoc1) )**  {  $currKey = key($assoc1); // current key  $currVal = current($assoc1); // current value    echo “$currKey $currVal <br/>”;  next($assoc1); // advance 1 position  $i++;  } |

* list() assigns values to multiple variables at a time form an indexed array
* each() returns key/value AND advances the cursor; returns FALSE if the end is reached

$sample = array (‘A’, ‘B’, ‘C’ );

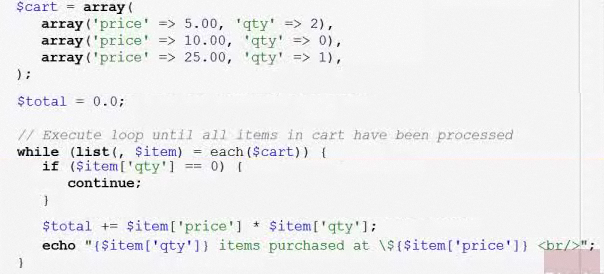
list ($zero, $one, $two) = $sample;

* $zero = ‘A’ $one = ‘B’ $two = ‘C’

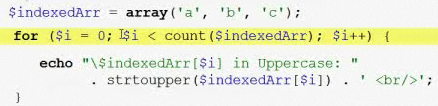
$eachRtn = each($sample);

**ANOTHER WAY TO LOOP**

|  |  |
| --- | --- |
| **NUMERIC** | **ASSOCIATED** |
| **$num1 = array ( 2, 4, 6) ;**  **reset ($num1) ; // reset array cursor**  **while (list (, $val ) = each ($num1) )**  **{**  **echo “$val <br/>”;**  **}** | **$assoc1 = array (‘One’ => 1,**  **‘Two’ => 2,**  **‘Three’ => 3);**  **reset($assoc1);**  **while (list ($key , $val ) = each ($assoc1) )**  **{**  **echo “$key = $val <br/>”;**  **}** |
|  | **One = 1**  **Two = 2**  **Three = 3** |



**MOST COMMON ONE**



**FOREACH**

* Special type of condition loop used specifically for looping over arrays
* Both continue and break works in a foreach loop
* There are 2 sintaxed, one for indexed arrays and another for associated arrays

|  |  |
| --- | --- |
| **INDEXED ARRAY** | **ASSOCIATED ARRAY** |
| $indexedArr = array( ‘a’, ‘b’, ‘c’ );  foreach ($indexedArr as $value )  {  echo stroupper ($value) . ‘ <br/>’;  }  echo ‘<br/>’;  echo $indexedArr[1]; | $assocArr = array (  ‘jan’ => ‘January’,  ‘feb’ => ‘February’,  ‘mar’ => ‘March’  );  foreach ($assocArr **as $key =>** $value )  {  echo “ ‘$key’ => ‘$value’ <br/>”;  }  echo ‘<br/>’;  echo $assocArr [‘feb’]; |
| A  B  C  B | ‘jan’ => ‘January’  ‘feb’ => ‘February’,  ‘mar’ => ‘March’  February |

**Array functions**

* count() returns numbers of items
* in\_array() returns Boolean $esta = in\_array($value , $arreglo) ;
* array\_key\_exist return Boolean $ar = array\_key\_exist( $delimiter, $srcArray);
* sort() sorts array by its values
* ksort sort array by its keys

in\_array( 9, array(10,8, 9)

in\_array( ‘smith’, array(‘c’ => ‘smith’, ‘b’ => ‘John’ ) )