ARRAYS

An array is used to store a collection of values in a single variable. Arrays in PHP consist of key-value pairs. The key (or index) can either be an integer (numeric array), a string (associative array), or a combination of both (mixed array). The value can be any data type.

Numeric arrays store each element in the array with a numeric index. An array is created using the array constructor. This constructor takes a list of values, which are assigned to elements of the array.

```
$a = array(1,2,3);
```

As of PHP 5.4, a shorter syntax is available, where the array constructor is replaced with square brackets.

```
$a = [1,2,3];
```

ARRAYS

```
<body>
  <?php
      $b = array();
       $b[] = 'a';
       $b[] = 'b';
       $b[] = 'c';
  ?>
</body>
```

We can also declare an array without predefined content or size, and fill it in later.

If we do not indicate the position, the new item is added at the end.

ARRAYS

```
<?php
        $a=array("9611111","9622222",9633333);
        echo $a[0];echo $a[1];
        echo $a[2];
        $a[3]=9644444;
        echo $a[3];
?>
```

Once the array is created, its elements can be referenced by placing the index of the desired element in square brackets. Note that the index begins with zero.

Positions and values can be added at any time

ARRAYS: COUNT FUNCTION

Count all elements in an array

```
<?php
$a=array("9611111","9622222",9633333);
echo "a: (".count($a)." elements)<br>";
for($i=0;$i<count($a);$i++)
      echo "Valor ".$a[$i]."<br>";
?>
```

ASSOCIATIVE ARRAYS

```
<?php
$a=array("John"=>"96111111","Carl"=>"9622222222");
echo $a["John"];
echo $a["Carl"];

?>
```

In associative arrays, the key is a string instead of a numeric index
When creating the array the double arrow operator (=>) is used to tell which key refers to what value.
Elements in associative arrays are referenced using the element names

ARRAYS: PRINTR

```php
<?php
$a=array("9611111","9622222");
$a[]=9644444;
print_r($a);
?>
```

Result:

```
Array ( [0] => 9611111 [1] => 9622222 [2] => 9644444 )
```

When we add an element to an array without specifying position, it is added at the end

In an indexed (non-associative) array the position will be the one that corresponds in order

The print_r (array) function displays the contents of the array on the screen

ASSOCIATIVE ARRAYS

```
<?php
$a=array("John"=>"96111111","Carl"=>"9622222");
$a="9644444";
print_r($a);
```

Result:

```
Array ( [John"] => 9611111 ["Carl"] => 9622222 [0] => 9644444 )
```

When we add an element to an associative array without specifying position, it is added at the end and the key will follow the order of the positions defined as numeric, or undefined

ASSOCIATIVE ARRAYS: FOREACH LOOP

The foreach loop provides an easy way to iterate through arrays. At each iteration, the next element in the array is assigned to the specified variable (the iterator) and the loop keeps running until it has gone through the entire array.

```
foreach(ArrayName as Variable)
{
    // instructions;
}
```

There is an extension of the foreach loop to also obtain the key’s name or index by adding a key
variable followed by the double arrow operator (=>) before the iterator

```
foreach(ArrayName as position => Variable)
{
    // instructions;
}
```

STRINGS

WE CAN USE A "STRING" AS IF IT WERE AN ARRAY OF CHARACTERS

WE HAVE ALREADY SEEN THAT WE CAN PRINT STRINGS ON THE SCREEN USING THE INSTRUCTIONS echo (unformatted) OR printf (formatted)

```
$variable="Hello";
echo $variable;
echo "Hello";
$name="Marta";
echo $variable.", ".$name;
echo "Hola, $name";
```

REMEMBER THAT VARIABLES IN DOUBLE QUOTES ARE CHANGED BY THEIR VALUE

STRING FUNCTIONS

Function	Utility	Sintax
substr	Returns a portion of string	`$sub=substr($cad, init, length);`
substr_replace	Replace text within a portion of a string	`$new=substr_replace($original, $subcad, init, length);`
str_replace	Replace all occurrences of the search string with the replacement string	`$new=str_replace($cad1,$cad2,$cad3)`
strlen	Length of the string	`$len=strlen($cad);`
strpos	Find the position of the first occurrence of a substring in a string	`if(strpos($cad,$subcad)!=FALSE)`

ltrim(cad)
rtrim(cad)
trim(cad)
	Strip whitespace from left(ltrim), right(rtrim) or both (trim)	`$scad=trim($cad);`

STRING FUNCTIONS

Function	Utility	Sintax
`strtolower`	Converts a string to lowercase letters	`$min=strtolower($cad);`
`strtoupper`	Converts a string to uppercase letters	`$mai=strtoupper($cad);`
`strchr`	Finds the first occurrence of a string inside another string	`$pos=strchr($cad,$search);`
`strrchr`	Finds the last occurrence of a string inside another string	`$pos=strrchr($cad,$search);`
`strrev`	Reverses a string	`$rev=strrev($cad);`
`explode`	breaks a string into an array	`$str=explode($separator,$cad)`

YOU CAN FIND A COMPLETE GUIDE TO PHP STRING FUNCTIONS ON THE PAGES
http://www.w3schools.com/php/php_ref_string.asp
https://www.php.net/manual/en/book.strings.php

DATE/TIME FUNCTIONS

`date(format, timestamp)` formats a local date and time (timestamp), and returns the formatted date string. Current time if no timestamp is given

FORMATS:

```
d → month day  m → month  Y → year (4 digit)
l → A full textual representation of a day
```

```
echo "Today is ".date("l, d-m-Y");

Today is Saturday, 30-09-2023
```

CREATE A DATE

1. date_create_from_format

 Parses a time string according to a specified format

```
<?php
$date = date_create_from_format('j-M-Y', '15-Feb-2023')
echo date_format($date, 'd-m-Y');
?>
```

The above example will output:

```
15-02-2023
```

You have also the 'Object oriented style'

2. mktime: Get Unix timestamp for a date

```
$date=mktime(0,0,0,02,15,2023);// hours, minutes, seconds, month, day, year
```

DATE/TIME FUNCTIONS

Function getdate(data) function returns date/time information of a timestamp or the current local date/time.

```
Array
(
    [seconds] => 40
    [minutes] => 58
    [hours]   => 21
    [mday]    => 19
    [wday]    => 2
    [mon]     => 7
    [year]    => 2021
    [weekday] => Monday
    [month]   => July
    )
```

Then, once a date is created we can extract information as follows:

```
$d=mktime(21,58,40,7,19,2021);

$info=getdate($d);

echo $info["year"];

Output: 2021
```

DATE/TIME FUNCTIONS

YOU CAN FIND A COMPLETE GUIDE TO PHP DATE FUNCTIONS ON THE PAGES

http://www.w3schools.com/php/php_ref_date.asp
https://www.php.net/manual/en/ref.datetime.php