

String Functions

```
int strpos($haystack , mixed $needle [, $offset])
Find the position of the first occurrence of a
substring in a string
string substr ($string , $start [, $length ] )
Return part of a string
string strrchr($haystack , mixed $needle )
This function returns the portion of haystack
which starts at the last occurrence of needle
and goes until the end of haystack.
string strstr($haystack , mixed $needle [, bool
$before_needle = false ] )
return first occurrence of a string
string implode ($glue , array $pieces)
array explode ($delimiter, $string[, $limit])
string strtok ($str, $token)
Tokenize string
string trim ($str [, $charlist])
```

Date

```
date($format[, $timestamp = time()])
d -> 01-31 m-> 01-12 y->13 Y->2013
h-> 1pm H-> 13 i->00-59 s->00-59
c-> ISO 8601 U ->unix time
```

Regex

```
int [preg_match|preg_match_all]($pattern,
$subject[, &$matches [, int $flags = 0 [, int
$offset = 0 ]]]) )
flags:
PREG_OFFSET_CAPTURE
PREG_PATTERN_ORDER (all)
PREG_SET_ORDER (all)
mixed preg_replace(mixed $pattern, mixed
$replacement, mixed $subject[, int $limit = -1 [,
int &$count]])
mixed preg_replace_callback($pattern, callable
$callback, mixed $subject[, $limit[, $count]])
array preg_split($pattern, $subject[, $limit[,
$flags]])
array preg_grep($pattern, $input[, $flags])
int preg_last_error()
return:
PREG_NO_ERROR
PREG_INTERNAL_ERROR
```

Regex (cont)

```
PREG_BACKTRACK_LIMIT_ERROR
PREG_RECURSION_LIMIT_ERROR
PREG_BAD_UTF8_ERROR
PREG_BAD_UTF8_OFFSET_ERROR
```

Constants

```
__LINE__
__FILE__
__DIR__
__FUNCTION__
__CLASS__
__TRAIT__
__METHOD__
__NAMESPACE__
```

Class example

```
class ClassName extends FatherClass
implements InterfaceClass{
const CONSTANT = 'text';
[public|protected|private] $attribute = 'default';
[public|protected|private] function [final]
functionName(array $p1 = 'default', $p2){}
public function __construct(){}
public function __destruct(){}
public function __get(){}
public function __set(){}
public function __call(){}
public function __callStatic(){}
public function __isset(){}
public function __unset(){}
public function __sleep(){}
public function __wakeup(){}
public function __toString(){}
public function __invoke(){}
public function __set_state(){}
public function __clone(){}
}
```

Namespace

```
namespace MyProject\Sub\Level;
use My\Full\Classname as Another;
class teste{
}
```