

ClickBlocks\Utils\PHP\Tools

General information

Inheritance	no
Child classes	no
Interfaces	no
Source	Framework/utils/php/tools.php

Contains useful methods for variety manipulations with PHP code.

Public static methods

splitClassName()

```
public static array splitClassName(string|object $class)
```

\$class	string, object	the full class name or class object.
----------------	----------------	--------------------------------------

Splits full class name into array containing two elements of the following structure: [class namespace, own class name]. Example:

```
print_r(Tools::splitClassName(new \StdClass()));  
print_r(Tools::splitClassName('MyNamespace\SpaceA\TestA'));  
print_r(Tools::splitClassName('\MyNamespace\SpaceA\TestA'));
```

The above example will output:

```
Array  
(  
    [0] => \  
    [1] => stdClass  
)  
Array  
(  
    [0] => MyNamespace\SpaceA  
    [1] => TestA  
)  
Array  
(  
    [0] => \MyNamespace\SpaceA  
    [1] => TestA  
)
```

getNamespace()

```
public static string getNamespace(string|object $class)
```

\$class	string, object	the full class name or class object.
----------------	----------------	--------------------------------------

Returns the namespace of the full class name or class object. Example:

```
echo PHP\Tools::getNamespace(new \StdClass()) . PHP_EOL;  
echo PHP\Tools::getNamespace('MyNamespace\SpaceA\TestA') . PHP_EOL;
```

```
echo PHP\Tools::getNamespace('MyNamespace\SpaceA\TestA');
```

The above example will output:

```
\
MyNamespace\SpaceA
\MyNamespace\SpaceA
```

getClassName()

```
public static string getClassName(string|object $class)
```

\$class	string, object	the full class name or class object.
----------------	----------------	--------------------------------------

Returns the class name of the full class name (class name with namespace). Example:

```
echo Tools::getClassName(new \StdClass()) . PHP_EOL;
echo Tools::getClassName('MyNamespace\SpaceA\TestA');
```

The above example will output:

```
stdClass
TestA
```

search()

```
public static array|boolean search(string $needle, string $haystack, boolean $all = false)
```

\$needle	string	the PHP code which we want to find.
\$haystack	string	the PHP code in which we want to find our PHP fragment.
\$all	boolean	determines whether all occurrences of the PHP fragment will be found.

Searches the first occurrence or all occurrences of the given PHP code in another PHP code.

Returns a numeric array of two elements or FALSE if the PHP fragment is not found. The first element is the index of the first token in the haystack.

The second element is the index of the last token in the haystack.

Example:

```
...
$res = Tools::search('$res=', file_get_contents(__FILE__), true);
print_r($res);
$res = Tools::search('$res=', file_get_contents(__FILE__), false);
print_r($res);
...
```

The execution result of the above example might be as follows:

```
Array
(
    [0] => Array
        (
            [0] => 55
            [1] => 57
        )

    [1] => Array
        (
            [0] => 86
            [1] => 88
        )
)
```

```
)
)
Array
(
    [0] => 55
    [1] => 57
)
```

in()

```
public static boolean in(string $needle, string $haystack)
```

\$needle	string	the PHP code which we want to find.
\$haystack	string	the PHP code in which we want to find our PHP fragment.

Checks whether the given PHP code is contained in other one. Example:

```
...
$res = Utils\PHP\Tools::in('var_dump ( $res )', file_get_contents(__FILE__));
var_dump($res); // will output bool(true)
...
```

replace()

```
public static string replace(string $search, string $replace, string $subject)
```

\$search	string	the PHP fragment being searched for.
\$replace	string	the replacement PHP code that replaces found \$search values.
\$subject	string	the PHP code string being searched and replaced on.

Replaces all occurrences of the PHP code fragment in the given PHP code string with another PHP fragment. Example:

```
<?php

use ClickBlocks\Utils\PHP;

require_once('connect.php');

$x = 1;
$x = $x * $x;
$x = $x++;
echo PHP\Tools::replace('$x=', '$x +=', file_get_contents(__FILE__));
```

The above example will output:

```
<?php

use ClickBlocks\Utils\PHP;

require_once('connect.php');

$x += 1;
$x += $x * $x;
$x += $x++;
echo PHP\Tools::replace('$x=', '$x += ', file_get_contents(__FILE__));
```

remove()

```
public static string remove(string $search, string $subject)
```

\$search	string	the PHP fragment being searched for.
\$subject	string	the PHP code string being searched and removed from.

Removes the given PHP code fragment from the PHP code string. Example:

```
<?php

use ClickBlocks\Utils\PHP;

require_once('connect.php');

$x = 1;
$x = $x * $x;
$x = $x++;
echo PHP\Tools::remove('$x=$x*$x', file_get_contents(__FILE__));
```

The above example will output:

```
<?php

use ClickBlocks\Utils\PHP;

require_once('connect.php');

$x = 1;
$x = $x++;
echo PHP\Tools::remove('$x=$x*$x', file_get_contents(__FILE__));
```