

Instantiation

changekey

pick

sort

transpose

calendar

Matrix

The matrix class offers a couple of generic array utilities and a month calendar for a specified date.

Namespace: \

File location: lib/matrix.php

Instantiation

Return class instance

```
$matrix = \Matrix::instance();
```

The Matrix class uses the Prefab (prefab-registry) factory wrapper, so you can grab the same instance of that class at any point of your code.

changekey

Change the key of a two-dimensional array element

```
null changekey ( array &$var, string $old, string $new )
```

example:

```
$array=array(
    'id'=>array(456,123,345,234),
    'name'=>array('ringo','paul','george','john'),
    'sales'=>array(0.13,0.35,0.57,0.79)
);
$matrix->changekey($array,'sales','percent');
/* $array is now:
array(
    'id'=>array(456,123,345,234),
    'name'=>array('ringo','paul','george','john'),
    'percent'=>array(0.13,0.35,0.57,0.79)
)*/
```

pick

Retrieve values from a specified column of a multi-dimensional array variable

```
array pick ( array $var, mixed $col )
```

example:

```
$array=array(
    array('id'=>123,'name'=>'paul','sales'=>0.35),
    array('id'=>456,'name'=>'ringo','sales'=>0.13),
    array('id'=>345,'name'=>'george','sales'=>0.57),
    array('id'=>234,'name'=>'john','sales'=>0.79)
);

$result = $matrix->pick($array,'name');
/* returns:
array('paul','ringo','george','john')
*/
```

sort

Sort a multi-dimensional array variable on a specified column

```
bool sort ( array &$amp;var, mixed $col [, int $order=SORT_ASC ] )
```

example:

```

$array=array(
    array('id'=>123,'name'=>'paul','sales'=>0.35),
    array('id'=>456,'name'=>'ringo','sales'=>0.13),
    array('id'=>345,'name'=>'george','sales'=>0.57),
    array('id'=>234,'name'=>'john','sales'=>0.79)
);

$matrix->sort($array,'sales');

/* $array is now:
array(
    array('id'=>456,'name'=>'ringo','sales'=>0.13),
    array('id'=>123,'name'=>'paul','sales'=>0.35),
    array('id'=>345,'name'=>'george','sales'=>0.57),
    array('id'=>234,'name'=>'john','sales'=>0.79)
)
*/

```

transpose

Rotate a two-dimensional array variable

```

null transpose ( array &$var )

```

example:

```

$array=array(
    array('id'=>123,'name'=>'paul','sales'=>0.35),
    array('id'=>456,'name'=>'ringo','sales'=>0.13),
    array('id'=>345,'name'=>'george','sales'=>0.57),
    array('id'=>234,'name'=>'john','sales'=>0.79)
);

$matrix->transpose($array);
/* $array is now:
array(
    'id'=>array(123,456,345,234),
    'name'=>array('paul','ringo','george','john'),
    'sales'=>array(0.35,0.13,0.57,0.79)
)*/

```

calendar

Return month calendar of specified date, with optional setting for first day of week (0 for Sunday)

```

array calendar ( string $date [, int $first = 0 ] )

```

Returns an array containing one array per week of the given month. Possible indexes are from 0 to 6, 0 for the first day of the week as given by the `$first` parameter, while 6 being the 6th day after *that first day* of the week. Let's take an example: if you want a calendar with weeks starting on Monday, the days

in the arrays of weeks in the returned calendar will start from the index 0, and Sundays, being 6 days later, will get the index 6.

Examples:

```
$cal = $matrix->calendar('2014-06', 1); // with weeks starting on Monday
/*
array(
    array(6=>1), // 1st day of the month is a Sunday,
index 6
    array(0=>2,1=>3,2=>4,3=>5,4=>6,5=>7,6=>8),
    array(0=>9,1=>10,2=>11,3=>12,4=>13,5=>14,6=>15),
    array(0=>16,1=>17,2=>18,3=>19,4=>20,5=>21,6=>22),
    array(0=>23,1=>24,2=>25,3=>26,4=>27,5=>28,6=>29),
    array(0=>30)
) */
```

and the same calendar, with weeks starting on Sunday:

```
$cal = $matrix->calendar('2014-06', 0); // with weeks starting on Sunday
/*
array(
    array(0=>1,1=>2,2=>3,3=>4,4=>5,5=>6,6=>7), // 1st day of the month is a Sunday,
index 0
    array(0=>8,1=>9,2=>10,3=>11,4=>12,5=>13,6=>14),
    array(0=>15,1=>16,2=>17,3=>18,4=>19,5=>20,6=>21),
    array(0=>22,1=>23,2=>24,3=>25,4=>26,5=>27,6=>28),
    array(0=>29,1=>30)
) */
```