

PHP Exercise

Objective:

Distribute a total amount randomly (within certain parameters) in a range of dates, excluding weekends.

There should be a baseline that will define the minimum amount of the value assigned to a specific date.

The returned result should be a unidimensional array with the following structure:

```
Array(
    [YYYY-mm-dd]>=>0.00, // The value should be a float value (with two decimal places) corresponding to the date used as key
    [YYYY-mm-dd]>=>0.00,
    [YYYY-mm-dd]>=>0.00,
    ...
)
```

Requirements:

1. The values should be distributed only on weekdays, therefore the weekends should be included in the array and have zero as value assigned.

2. The total sum of the values assigned in the resulting array should be within 1% above or below the total value given.

Example:

```
$total=75300;
$distributed=array_sum($result);
// $distributed value can only be a maximum of 76053 or a minimum of 74547 (1% above and below
```

75300)

3. Implementation of object programming is preferred in this exercise

4. The baseline parameter should control the amount of randomness in the distribution of values throughout the given dates and it should be given as an integer from 0 to 100. It should also guarantee a minimum value per date. The baseline indicates the amount of randomness allowed in the distribution. This method can have several approaches depending on the algorithm used, so the actual value resulting of the baseline is open to interpretation. In the following examples we use it as the percentage of the total, divided amongst the total of days that have non-zero values (weekdays).

Example 1:

```
$baseline=20;
$total=100;
$start_date='2016-12-19';
$start_date='2016-12-23';
// This set of parameters should guarantee that each day will have a minimum value of 4
// Example result
Array(
    [2016-12-19]>=>12,
    [2016-12-20]>=>4.22,
    [2016-12-21]>=>34.53,
    [2016-12-22]>=>19.47,
    [2016-12-23]>=>29.78
)
```

Example 2:

```
$baseline=100;
$total=100;
$start_date='2016-12-21';
$start_date='2016-12-27';
// This set of parameters should guarantee that each day will have a minimum value of 20
// Example result
Array(
    [2016-12-21]>=>20,
    [2016-12-22]>=>20,
    [2016-12-23]>=>20,
    [2016-12-22]>=>0, // Saturday
```

```

[2016-12-23]=>0, // Sunday
[2016-12-24]=>20,
[2016-12-25]=>20,
)

```

Example 3:

```

$baseline=50;
$total=100;
$start_date='2016-12-19';
$start_date='2016-12-24';
// This set of parameters should guarantee that each day will have a minimum value of 10
// Example result
Array(
    [2016-12-19]=>21.05,
    [2016-12-20]=>12.91,
    [2016-12-21]=>10,
    [2016-12-22]=>29.65,
    [2016-12-23]=>26.39,
    [2016-12-24]=>0 // Saturday
)

```

5. All dates within the range should be included in the array.

Evaluation criteria:

1. A user interface and/or user friendly results display using any or all of the following:
HTML, CSS, jQuery, Bootstrap
2. Asynchronous requests (using AJAX and JSON)
3. Input parsing to prevent code injection/execution
4. Responsiveness for the UI
5. Usage of Wordpress is mandatory

Input parameters:

@param	date	\$start_date
@param	date	\$end_date
@param	integer	\$total
@param	integer	\$baseline

Output:

@return	array
---------	-------