BRAINSTER

Workshop 14: PHP Loops and Associative arrays

Brainster Web Development Academy

Exercise 01

Create an array containing 10 elements (each element should represent a city name).
Write code that will first print all elements on odd position, then all elements on even position and finally the element on position 5. Before printing the element on position five, to make sure the code won't throw an error in any scenario, check if there is an element on position five (hint: use the isset() function).

For example if you have the following array:
["Tokyo", "Berlin", "Lisbon", "Helsinki", "Denver", "Nairobi", "Rio" ...],
your code should print:

Elements on odd position:
Berlin, Helsinki, Nairobi ...
Elements on even position:
Tokyo Lisbon, Denver, Rio ...
Element on position five: Nairobi

If there is no element on position five (i.e. there are only 3 cities), the last sentence should be:
Element on position five: not found.

Exercise 02

Write PHP code that will print all numbers from 234 to 1987, that are divisible by 3.
Hint: these numbers, when divided by 3 give remainder 0.

The output should be:
Numbers in the range 234 - 1987 that are divisible by 3:
234, 237, 238 ... 1980, 1983, 1986

Bonus: to check if a given number is divisible by 3 we can also use a mathematical rule that says:
A number is divisible by 3 if the sum of its digits is divisible by 3.
Try to implement this code, having this rule in mind.

For example, if we want to check whether the number 375 is divisible by 3 we can sum all the number's digits (3 + 7 + 5) and check if their sum is divisible by 3. If it is, then the whole number is divisible as well.
375 % 3?
(3 + 7 + 5) % 3?
15 % 3 == 0 (this means that 375 is divisible by 3 too).

Exercise 03

Write PHP code that will print the multiplication table for given number (from 1 up until 10). Store the number for which we want the multiplication table printed in a variable called $number.

Example. If we set the value of the variable $number to be 7, our code should print:

The multiplication table for the number **7** is:
1 x 7 = 1
2 x 7 = 14
3 x 7 = 21
4 x 7 = 28
5 x 7 = 35
6 x 7 = 42
7 x 7 = 49
8 x 7 = 56
9 x 7 = 63
10 x 7 = 70

Exercise 04

Write PHP code that will print all the numbers from 1 to 50 in a separate row.

All numbers that are divisible by 2 should be printed in **blue** color.
All numbers that are divisible by 3 should be printed in **green** color.
All numbers that are divisible by both 2 and 3 should be printed in **orange** color.
All numbers that are **not** divisible by 2 and **not** divisible by 3 should remain black.

The result should look like:

1
2
3
4
5
6
7
8
9
10
....

Exercise 05

Define an array containing only integer values.
Write code that will find the smallest and the largest number in the array (min & max).

Once you have found these 2 numbers, subtract them (max – min) and store their difference in a new variable.

Check the difference between the largest and the smallest value.
If it is greater than 100, print:
The difference between the numbers in this array is huge

Else, if the difference is lower than 100, print:
The difference between the numbers in this array is small

Exercise 06

Define an associative array, where keys should be people's names and values should be their age (represented in years as a whole number, not decimals or months).

Print the names of all the people from the array who are allowed to vote on elections (people that are older than 18 years).

Example. If you have the following array:
["Jack" => 25, "John" => 15, "Alice" => 20, "Mike" =>10, "Jane" => 30]

Your code should print:
People that can vote on the upcoming elections:
Jack
Alice
Jane

Exercise 07

For this exercise, use the array from the previous exercise. Alter the names of the people manually (without any code), so that some start with an uppercase letter and some with a lowercase letter.

Next, figure out a way to print all of the names that are eligible to vote and start with uppercase letter in **green** and print all the names that are eligible to vote and start with lowercase letter in **red**.

Example. If you modify the previous array to be like this one:
[“jack” => 25, “John” => 15, “Alice” => 20, “Mike” =>10, “jane” => 30]

Your code should print:
People that can vote on the upcoming elections:
jack
Alice
jane

Break a leg!

If you haven't finished all exercises, please try to finish them at home.

Functions that can help you with the exercises in these workshop (google their usage):

trim - strip whitespace (or other characters) from the beginning and end of a string

implode - join array elements into a string

explode - breaks a string into an array

ceil - rounds a number UP to the nearest integer

floor - rounds a number DOWN to the nearest integer

strtolower - Make a string lowercase

strtoupper - Make a string uppercase