# Exercises: PHP Syntax, Basic Web

Problems for exercises and homework for the "Software Technologies" course @ SoftUni.

You can submit your solutions here https://judge.softuni.bg/Contests/236/PHP-First-Steps-Exercises.

# Part I: Simple operations with PHP

Taking into account the nature of PHP we will receive input **not** from the console, as in C# or Java, but from **forms**, which will be **rendered** in the **browser**.

Every task in judge will have a **skeleton**, which will be an **HTML** skeleton of the form.

In each example, you will see new column named "**Parameter name**". Every parameter, which should be passed to your program, should be received from **`input tag`** with exactly the same **`name parameter`**. All of the skeletons will be with the correct **`name parameters`**. If you write your own HTML make sure you use the **same names**.

## 1. Multiply a Number by 2

You are given a number **num**. Write a PHP script that **multiplies** the **number by 2** and prints the result. The input comes as a parameter named **num**.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 2 | 4 |

| Input | Output |
|---|---|
| 3 | 6 |

You will be given a HTML form that submits the input data:

```
<form>
    N: <input type="text" name="num" />
    <input type="submit" />
</form>
```

The name of the input parameter will be exactly as shown above – **num**.

Print the output in the HTML document, just after the HTML form.

### Hints

- In case the form was submitted and the input parameter **num** exists, take its value as integer using the function **`intval(string)`**.

```
<?php
if(isset($_GET['num'])) {
    $n = intval($_GET['num']);
}
?>
```

- Then, just print the results: **`echo $n * 2`**.

```php
if(isset($_GET['num'])) {
    $n = intval($_GET['num']);
    echo $n * 2;
}
```

- Test whether your code worked as expected:


## 2. Multiply Two Numbers

You are given a number **num1** and a number **num2**. Write a PHP script that multiplies **num1 * num2** and prints the result. The input comes as parameters named **num1** and **num2**. Print the output in the HTML page.

### Examples

| Parameters names | Input | Output |
|---|---|---|
| num1 | 2 | 6 |
| num2 | 3 | |

| Input | Output |
|---|---|
| 13 | 169 |
| 13 | |

### Hints

- This time the form which will be given to use will have 2 input elements, with names **num1** and **num2**

```php
<form>
    N: <input type="text" name="num1" />
    M: <input type="text" name="num2" />
    <input type="submit" />
</form>
```

- We must check both elements, if they have values before we perform any action

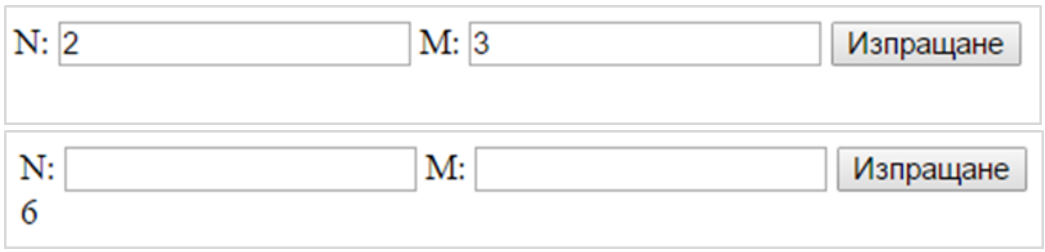
```php
<?php
if(isset($_GET['num1']) && isset($_GET['num2'])) {

}
?>
```

- When we have checked both elements we get them both and extract their values into variables and we perform the specified action:

```
if(isset($_GET['num1']) && isset($_GET['num2'])) {
    $n = intval($_GET['num1']);
    $m = intval($_GET['num2']);
    echo $n * $m;
}
```

- The result is as follows:




## 3. Multiply / Divide Numbers

You are given a number **num1** and a number **num2**. Write a PHP script that:

- Multiplies **num1 * num2** if **num2** is greater than or equal to **num1**.
- Divides **num1 / num2** if **num1** is greater than **num2**.

The input comes as parameters named **num1** and **num2**. Print the output in the HTML page.

### Examples

| Parameters names | Input | Output |
|---|---|---|
| num1 | 2 | 6 |
| num2 | 3 | |

| Input | Output |
|---|---|
| 13 | 169 |
| 13 | |

| Input | Output |
|---|---|
| 3 | 1.5 |
| 2 | |

## 4. Product of 3 Numbers

You are given a number **num1**, **num2** and **num3**. Write a PHP script that finds if **num1 * num2 * num3** (the product) is **negative** or **positive**. Try to do this **WITHOUT** multiplying the 3 numbers.

The input comes as parameters named **num1**, **num2** and **num3**.

### Examples

| Parameters name | Input | Output |
|---|---|---|
| num1 | 2 | Negative |
| num2 | 3 | |
| num3 | -1 | |

| Input | Output |
|---|---|
| 5 | Positive |
| 4 | |
| 3 | |

| Input | Output |
|---|---|
| -3 | Positive |
| -4 | |
| 5 | |

### Hints

- Count the **negative numbers**. If they are odd number, the result will be negative, otherwise → positive.
- Special case: one of the numbers is 0 → the product is positive.

## 5. Numbers from 1 to N

You are given a number **num**. Write a PHP script that loops through all of the numbers from **1** to **num** and prints them. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 1 2 3 4 5 |

| Input | Output |
|---|---|
| 2 | 1 2 |

## 6. Numbers from N to 1

You are given a number **num**. Write a PHP script that loops through all of the numbers from **num** to **1** and prints them. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 5 4 3 2 1 |

| Input | Output |
|---|---|
| 2 | 2 1 |

## 7. Even Numbers from 1 to N

You are given a number **num**. Write a PHP script that loops through all of the numbers from **1** to **num** and prints only the even ones. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 2 4 |

| Input | Output |
|---|---|
| 2 | 2 |

## 8. Odd Numbers from N to 1

You are given a number **num**. Write a PHP script that loops through all of the numbers from **num** to **1** and prints only the odd ones. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 5 3 1 |

| Input | Output |
|---|---|
| 2 | 1 |

## 9. N Factorial

You are given a number **num**. Write a PHP script that prints **factorial** of **num**. Factorial is calculated by multiplying all numbers to the given number. Factorial (N) = 1 * 2 * 3… * **num**. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 120 |

| Input | Output |
|---|---|
| 3 | 6 |

## 10. Not Divisor Numbers

You are given a number **num**. Write a PHP script that prints all the numbers from **num** to **1,** which are not divisors of **num.** The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 10 | 9 8 7 6 4 3 |

| Input | Output |
|---|---|
| 12 | 11 10 9 8 7 5 |

## 11. Fibonacci Sequence

You are given a number **num**. Write a PHP script that prints **num** numbers from The Fibonacci sequence, each on a new line, starting from 1. In the Fibonacci sequence, every number is formed from the sum of the previous 2. The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 5 | 1 1 2 3 5 |

| Input | Output |
|---|---|
| 10 | 1 1 2 3 5 8 13 21 34 55 |

## 12. Tribonacci Sequence

In the **"Tribonacci" sequence**, every number is formed by the **sum of the previous 3**.

You are given a number **num**. Write a PHP script that prints **num** numbers from the Tribonacci sequence, each on a new line, starting from 1. The input comes as a parameter named **num**. The value **num** will always be positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 4 | 1 1 2 4 |

| Input | Output |
|---|---|
| 8 | 1 1 2 4 7 13 24 44 |

## 13. Prime Numbers from N to 1

You are given a number **num**. Write a PHP script that prints only the **prime** numbers from **num to 1**. A prime number is a number that can be divided only by 1 and itself. **1 is not a prime number.** The input comes as a parameter named **num**. The parameter **num** will hold a positive integer.

### Examples

| Parameter name | Input | Output |
|---|---|---|
| num | 10 | 7 5 3 |

| Input | Output |
|---|---|
| 20 | 19 17 13 11 7 5 3 |

# Part 2: Manipulating HTML with PHP

## 14. HTML Buttons

You are given a number **num**. Write a PHP script that **generates HTML <button> elements.** The buttons must hold a number from **1** to **num** as their caption. The input comes as a parameter named **num**, holding a positive integer.

There is no indentation on the elements.

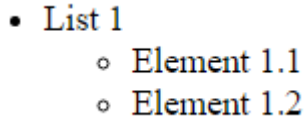
## Examples

| Parameter name | Input | Output | Picture |
|---|---|---|---|
| num | 3 | <button>1</button><br><button>2</button><br><button>3</button> |  |
| num | 5 | <button>1</button><br><button>2</button><br><button>3</button><br><button>4</button><br><button>5</button> | |
| num | 1 | <button>1</button> | |

# 15. Sub-Lists

You are given a number **num1** and a number **num2**. Write a PHP script that generates a list with **num1** elements, and each of those elements has a sub-list with **num2** elements. Each of the Lists has a caption "**List #**" where # is the current number from **1** to **num1.** And each of the elements of those lists has a caption "**Element #.#**" where the first # is the current list number and the second **#** is the current element from **1** to **num2**. The input comes as two parameters named **num1** and **num2**, holding positive integers.

## Examples

| Parameters names | Input | Output | Picture |
|---|---|---|---|
| num1 | 1 | <ul><br><li>List 1<br><ul><br><li><br>Element 1.1<br></li><br><li><br>Element 1.2<br></li><br></ul><br></li><br></ul> |  |
| num2 | 2 | | |

There is **indentation** on the elements. Each nested element is **tabbed once right** from its parent.

# 16. Draw an "S" with Buttons

Write a PHP script that draws 5 x 9 **<button>** elements with **0** and **1** in them. The **1**s should form a figure **"S"**. Make the **1**s' background color – **blue**. There is no indentation on the elements.

## Example

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

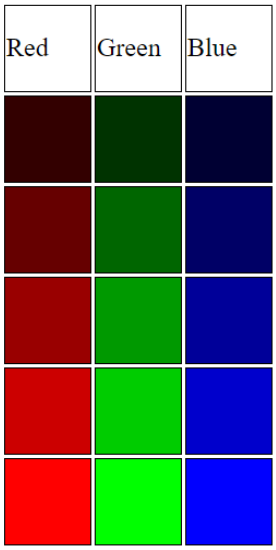
# 17. RGB Table

Write a PHP script that draws a table with 3 columns, with captions – **Red**, **Green** and **Blue**, and 5 rows, each having as **background 1** of **5** shades of those colors. Increase the color index by **51** for each row. For example, for the Red Column it will be:

- **`RGB(51, 0, 0)`**
- **`RGB(102, 0, 0)`**
- **`RGB(153, 0, 0)`**
- . . .

The caption columns should **NOT** be colored. Each cell of the table has **width** and **height** equal to **`50px`** and has **`1px solid black border`**.

## Picture:



# 18. 50 Shades of Grey

Write a PHP script that generates 5 rows of 10 **`<div>` elements**, each having as background a different shade of grey – resulting in 50 shades of grey. Every row starts with a color index – a **multiple of 51**.

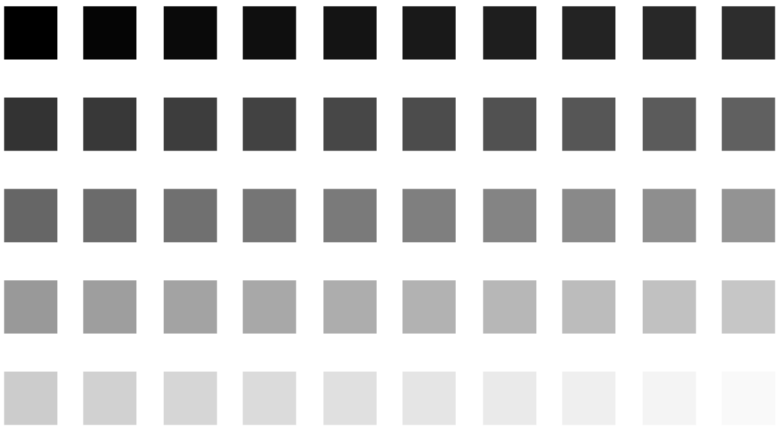
- First row – 0,
- Second row – 51,
- Third row – 102 …

Then you need to print **10 divs**, each increasing that index with **5.**

- First row – 0, 5, 10…
- Second row – 51, 56, 61…

The color index needs to be applied to **all parameters** of the **RGB function**. Example: **`rgb(51, 51, 51);`**

## Picture:



There is no indentation on the elements.