

PHP group test examples

Practice test 1.

1. **SVG generation (3 points)** Generate SVG elements based on the data structure below. An HTML element must be generated from the `type` field and HTML attributes from the `params` array. Below you can also see an example HTML (static prototype). (3 points)

```
$shapes = [
  [
    'type'    => 'rect',
    'params'  => [
      'x' => 0,
      'y' => 0,
      'width' => 50,
      'height' => 50,
      'rx' => 10,
    ]
  ],
  [
    'type'    => 'line',
    'params'  => [
      'x1' => 60,
      'y1' => 100,
      'x2' => 80,
      'y2' => 120,
      'stroke' => 'red',
    ]
  ],
  [
    'type'    => 'line',
    'params'  => [
      'x1' => 80,
      'y1' => 120,
      'x2' => 150,
      'y2' => 50,
      'stroke' => 'red',
    ]
  ],
  [
    'type'    => 'circle',
    'params'  => [
      'cx' => 150,
      'cy' => 100,
      'r' => 20,
    ]
  ],
];
```

```
<svg width="200px" height="200px" xmlns="http://www.w3.org/2000/svg">
  <rect x="0" y="0" width="100" height="100" rx="15" />
  <line x1="100" y1="100" x2="150" y2="50" stroke="black" />
  <circle cx="150" cy="150" r="50" />
</svg>
```

2. Filtering a list (3,5 pont)

- a. Generate a list based on the PHP array given below. An item in the list should look like this: `"name (age)"`. (0.5 points)
- b. It should be possible to filter this list by age given in the URL. The parameter name should be `age`. For example, if `index.php?age=10`, only 10-year-old students should be shown. (1.5 points)
If there is no `age` parameter in the URL, or it is empty or in the wrong format, i.e. it is not a number, the whole list should be displayed! (1.5 points)

```
$students = [  
    ['name' => 'Student1', 'age' => 20],  
    ['name' => 'Student2', 'age' => 10],  
    ['name' => 'Student3', 'age' => 30],  
    ['name' => 'Student4', 'age' => 20],  
    ['name' => 'Student5', 'age' => 10],  
];
```

3. Saving student records (3,5 points)

Create a web page with a form through which we can add students and save them to a file. For each student, we need the following information and validation:

- name: text input field, mandatory
- age: number, mandatory

If there are errors, display the error messages in a list above the form. (1.5 points) If there is no error, save the data in a file! (2 points) You do not have to keep the form state! *Note: when you create the file on the webprogramming server, don't forget to give it write access to everyone (F9 file properties, give "Other" W rights.)* You can use the `Storage` utility to save it to a file!

```
<h1>New student</h1>  
<form action="" method="post">  
    Name: <input type="text" name="name" /> <br>  
    Age: <input type="text" name="age" /> <br>  
    <button type="submit">Add student</button>  
</form>
```

Practice test 2.

Task1

You are given the following data structure as a PHP variable:

```
$data = [  
    [  
        "y" => 10,  
        "width" => 40,  
        "color" => "red"  
    ],  
    [  
        "y" => 30,  
        "width" => 50,  
        "color" => "lime"  
    ],  
    [  

```

```

    "y" => 50,
    "width" => 20,
    "color" => "aqua"
  ],
  [
    "y" => 70,
    "width" => 60,
    "color" => "orange"
  ],
]
```

Use this data to generate an SVG graphic of the size 200×150px with PHP. Each item in the `$data` array represents an SVG rectangle. Each rectangle has a height of 15px and an X coordinate of 10px.

The PHP generated output should look like this:

```

<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="200" height="150" >
  <rect width="40" height="15" x="10" y="10" fill="red" />
  <rect width="50" height="15" x="10" y="30" fill="lime" />
  <rect width="20" height="15" x="10" y="50" fill="aqua" />
  <rect width="60" height="15" x="10" y="70" fill="orange" />
</svg>
```

Task 2

Create a PHP program that can display the n^{th} Fibonacci number. The program should receive the input number via the `n` GET parameter in the URL.

- if the parameter is present display the n^{th} Fibonacci number on the screen.
- if no parameter is given display the following brief explanation on the usage of the program:

To calculate the n^{th} Fibonacci number add this to the url (replacing `value` with the your input): `?n=value`

A Fibonacci number is calculated by the following formula:

```

Fib(1) = 0
Fib(2) = 1
Fib(n) = Fib(n-1) + Fib(n-2) for any `n` > 1
```

Task 3

Your task is to create a page for testing the expiry date of a driving license with a form. The form consists an input field of the type "date". In the field you can enter the **date of issue** for the driving license. The form uses the POST method. You have to validate the input:

- the value exists and is not empty
- the value is formatted as date (you can use the `strtotime` function)
- the date of issue is earlier than the current date
- if there are any errors they are all shown to the user

If there are no errors:

- calculate the **expiry date** of the the driving licence: a driving licence is valid for **10 years**
- if the expiry date is in the future show it on the output (you can use the `date` function for formatting)

- if the expiry date is in the past write "Expired" to the output

The format of the input form:

```
<form>
  Date of issue:
  <input type="date">
  <button type="submit">Send</button>
</form>
```