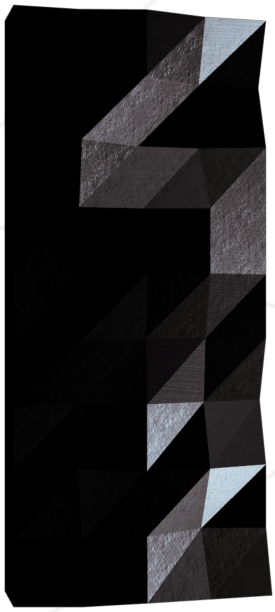

PHP Piscine

Day 09

Staff 42 piscine@42.fr

Summary:
This document is the day09's subject for the PHP Piscine.

Contents

Chapter I

Foreword

Gollum : What has he done? Stupid Hobbit! You are damaging them.

Sam : Damaging what? They‘ve got almost no meat. What we need are potatoes.

Gollum : What are potatoes my precious? What are they?

Sam : Potatoes you idiot! Broiled, mashed, fried, sliced or boiled with sauce. Nice fat sliced and fried in oil to go with fried fish. Even you won’t resist.

Gollum : Oh but we will resist. To waste perfect fish like that... We prefer fish raw and alive, keep your nasty fries.

Sam : You are impossible.

Chapter II

General Instructions

- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- Only the work submitted on the repository will be accounted for during peer-2-peer correction.
- Using a library is forbidden and will be considered cheating. Cheaters get `-42`, and this grade is non-negotiable..
- You cannot leave any additional file in your repository than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / the Internet / http://www.php.net / ....`
- Think of discussing on the Forum. The solution to your problem is probably there already. Otherwise you will start the conversation.
- By Odin, by Thor ! Use your brain !!!

Chapter III

Exercise 00 : Come blow this balloon

	Exercise 00
Come blow this balloon	
Turn-in directory : *ex*00/	
Files to turn in : `balloon.html`	
Allowed functions : `HTML, CSS, JS`	
Notes : `n/a`	

For this exercise, you will blow a balloon. To start, your HTML file will have a basic structure similar to the one seen in the introduction video. No library allowed.

Create in HTML/CSS a `div` `200px` by `200px` with a red background color. The borders will be rounded to create a perfect round shape and not a square anymore. That `div` will then be our balloon.

When a click occurs on the balloon, its size will grow by `10px` while keeping its round shape. With each click its color will go in that order from red to green, then blue and back to red again.

Usually, this type of balloon is rather resistant but if its size becomes greater than `420px` it will explode and return to its original size.

Small additional detail, when the mouse is in the balloon and leaves it, the size of the balloon shrinks by `5px` (the size of the balloon cannot go lower than `200px`) and its color changes in the reverse order than mentioned earlier.

Chapter IV

Exercise 01 : It's over 9000

	Exercise 01
It's over 9000	
Turn-in directory : $ex01/$	
Files to turn in : `calc.html`	
Allowed functions : `HTML, CSS, JS`	
Notes : `n/a`	

For this exercise, we will create a basic calculator. To start, your HTML file will have a basic structure similar to the one seen in the introduction video. No library allowed. The design doesn't matter as long as the exercise is still doable.

First, let's model our calculator. It will be composed as follows:

- A `text input` field that represents the left member of our calculation.
- A `select` field that will contain a list of the following operators as options:(`'+'`, `'-'`, `'*'`, `'/'`, `'%'`).
- A `text input` field that represents the right member of our calculation.
- A `submit input` of value `'Try me!'`.

When we click on the button `'Try me!'`, the calculation is executed and the result appears in an alert message. The result also needs to be displayed on the console of your browzer (log).

Both inputs fields can only contain positive integer values($>= 0$) for the calculation to be executed. Otherwise display an alert message must appear with the following message `'Error :('`.

Division and modulo by zero display an alert message with the following message: `"It's over 9000!"`. The result also needs to be displayed on the console of your browzer (log)

Every 30 seconds, an alert pop-up windows will appear saying `'Please, use me...'`.

Chapter V

Exercise 02 : To do or not to do

	Exercise 02 :
To do or not to do	
Turn-in directory : $ex02$: /	
Files to turn in : `index.html, todo.js`	
Allowed functions : `HTML, CSS, JS`	
Notes : `n/a`	

For this exercise, we will have to create a mini local task management. To start, your HTML file will have a basic structure similar to the one seen in the introduction video. The design does not matter as long as the structure presented below is respected. Be creative but concentrate in priority on features.

The to do list will be represented by a `div` that will have as `id` attribute the value `'ft_list'`. This bloc contains the list of "TO DO". Each TO DO is represented by a div contained in the `'ft_list'` bloc. When a TO DO is created, it is placed at the top of the list. Up to you to create the element and place it in the right spot (DOM manipulation).

There must be creation button named `'New'`. When clicked, it'll open a text window (checkout the promp function) that will allow the user to fill in a new TO DO. Once validated if not empty it must appear at the tol of the list.

To remove a TO DO from the list, all you have to do is click on it. A configuration window must open and ask whether yes or no you want to remove that TO DO. If you confirm, the TO DO must disappear permanently from the DOM, it can't just be hididen.

Small additional implementation, your TO DO list will have to be saved as a cookie. If the list contains some TO DO when you close you browser, this same list must be loaded and displayed in `'ft_list'`. If the cookie(s) do not exist, then the list will be blank.

Chapter VI

Exercise 03 : If jQuery, I'm going too

	Exercise 03 :
If jQuery, I'm going too	
Turn-in directory : $ex03$: /	
Files to turn in : `Same files than respective exercises`	
Allowed functions : `HTML, CSS, jQuery`	
Notes : `n/a`	

It's time to use that tool we love so much. You need to redo the first 3 exercises using lib jQwery. Import this lib via CDN as explained in the elearning section. No need to download the lib in your repository otherwise you will lose some points.

For the repository, create 3 disctinct folders: `ex00bis/`, `ex01bis/` et `ex02bis/`

Only the jQuery lib is allowed for these exercises.

Chapter VII

Exercise 04 : AJAX, stronger than dirt!

	Exercise 04 :
AJAX, stronger than dirt!	
Turn-in directory : $ex04$: /	
Files to turn in : `index.html, todo.js, select.php, insert.php, delete.php, list.csv`	
Allowed functions : `HTML, CSS, jQuery, PHP`	
Notes : `n/a`	

Now that you are a bit more comfortable with jQuery, it's time to throw yourself in the (bleach) bath! The school storage is great but the Cloud is better.

And as luck would have it, your Cloud won't be far. It will be on your localhost server that will execute the phpt.

Therefore you will need to replace in the previous exercise the backup system so that everything will be functional with an "online" CSV backup and no longer via cookies. The data formatting inside the CSV must be the following `'id;i am a todo'`.

It is MANDATORY that you use AJAX, your HTML page should never be refreshed.

- the `select.php` file gets the TO DO list from the CSV.
- the `insert.php` file adds a TO DO to the CSV.
- the `delete.php` file removes a TO DO from the CSV.

Any action on the page related to your list (adds, remove) will have to be transfered to the CSV using AJAX calls.

Good Luck !