PHP Piscine

Day 09

Staff 42 piscine@42.fr

Summary:

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

Contents

Chapter 1

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 2

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 3

Exercise 00 : Come inflate the balloon

Turn-in directory : ex00/
Files to turn in: balloon.html
Allowed functions: HTML, CSS, JS

For this exercise, you will inflate 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 (not a square anymore). That **div** will become our balloon.

When you click on the balloon, its size must grow by **10px** while keeping its round shape. With each click, its color will go, in that specific 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 (without clicking it), the size of the balloon shrinks by **5px** (its cannot go lower than **200px**) and its color changes in the reverse order of the one mentioned earlier.

Chapter 4

Exercise 01 : It's over 9000

Turn-in directory : ex01/
Files to turn in: calc.html
Allowed functions: HTML, CSS, JS

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 much, as long as the exercise remains easily doable for the user.

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

- A **text input** field that represents the left member of our operation.
- 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 operation.
- A **submit input** with the 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 browser (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/or modulo by **zero** should display an alert message with the following message: "**It's over 9000!**". The result also needs to be displayed on the console of your browser (log)

Every 30 seconds, an alert pop-up window must be displayed saying '**Please, use me...**'.

Chapter 5

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

For this exercise, we will have to create a mini local task manager. 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 and prioritize the requested features.

We're going to create a to-do list, that will be represented by a **div** with an **id** attribute equal to **'ft_list'**. This bloc contains the list of all the "TO-DOs". Each TO-DO is represented by a div, contained inside the **'ft_list'** bloc. When a TO-DO is created, it is added at the top of that list. It is up to you to create the element and place it in the right spot (DOM manipulation).

In order to create a TO-DO element, you must have a button named **New**. When clicking it, it will open a text window (check out the **prompt** function) that will let the user fill in a new TO-DO element. Once validated, if it's not empty, it must be displayed at the top of the **ft_list** list.

To remove a TO-DO from the list, all you have to do is click on it. When clicking, a window must be displayed and ask the user to confirm their action (Yes or No). If the user confirms it, the TO-DO must disappear permanently from the DOM (it can't be simply hidden).

Last but not least, your TO-DO list must be saved as a cookie. If the list contains some TO-DO elements when you close you browser, this same list must be loaded and displayed in the **'ft_list'** bloc. If the cookie(s) do not exist, then the list will be empty.

Chapter 6

Exercise 03 : If jQuery, I'm going too

Turn-in directory : **ex00bis/, ex01bis/, ex02bis/**

Files to turn in: **Same files as the previous exercises**

Allowed functions: **HTML, CSS, jQuery**

It's time to use *that* tool we love so much. You need to redo the first 3 exercises, but this time using the **jQuery** library. Import this library via **CDN**, as explained in the e-learning section. You must **not** download the library in your repository.

To structure your repository, create 3 distinct folders: **ex00bis/**, **ex01bis/** and **ex02bis/**

Only the jQuery library is allowed for these exercises.

Chapter 7

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

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

Actually, your "Cloud" won't be far away. It will be on the same localhost server that executes your PHP (and it will just be a file, really).

You will need to modify what you did in the previous exercise so that the backup system works with an "online" CSV backup and no longer via cookies. The data formatting inside the CSV must be the following **'id;content of the 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 element to the CSV.
- The **delete.php** file removes a TO-DO element from the CSV.

Any action on the page related to your list (addition, removal) will have to be "transferred" to the CSV file using AJAX calls.

Good Luck !