D00 - Piscine PHP Symfony

Web Basics

Summary: *This first day will introduce you to the basics of web devlopement. On the menu :* `HTTP, HTML, CSS` *and the integration of existing scripts in the language* `Javascript`*.*

Contents

Chapter I

Foreword

This is what wikipedia told us about *Balaenoptera Musculus* :

The blue whale (Balaenoptera musculus) is a marine mammal belonging to the baleen whales (Mysticeti). At up to 30 metres (98 ft) in length and with a maximum recorded weight of 173 tonnes (190 short tons) and probably reaching over 181 tonnes (200 short tons), it is the largest animal known to have ever existed.

Long and slender, the blue whale's body can be various shades of bluish-grey dorsally and somewhat lighter underneath. There are at least three distinct subspecies: B. m. musculus of the North Atlantic and North Pacific, B. m. intermedia of the Southern Ocean and B. m. brevicauda (also known as the pygmy blue whale) found in the Indian Ocean and South Pacific Ocean. B. m. indica, found in the Indian Ocean, may be another subspecies. As with other baleen whales, its diet consists almost exclusively of small crustaceans known as krill.

Blue whales were abundant in nearly all the oceans on Earth until the beginning of the twentieth century. For over a century, they were hunted almost to extinction by whalers until protected by the international community in 1966. A 2002 report estimated there were 5,000 to 12,000 blue whales worldwide, in at least five groups. The IUCN estimates that there are probably between 10,000 and 25,000 blue whales worldwide today.[10] Before whaling, the largest population was in the Antarctic, numbering approximately 239,000 (range 202,000 to 311,000).

No whale was hurt during this subject writing.

Chapter II

General Instructions

If no other explicit information is displayed, you must assume the following versions of languages :

- `PHP - Symfony 2.7`

- `HTML 5`

- `CSS 3`

Chapter III

Exercise 00

	Exercise 00
	Exercise 00: First shell script
Turn-in directory : *ex00/*	
Files to turn in : `myawesomescript.sh`	
Allowed functions : `curl, grep, cut`	

If `Twitter` has no secrets for you, you'll probably know `bit.ly` : an useful URL shortener.

The purpose of this exercise is to write and return a shell script which displays the real address behind a `bit.ly` address (understand "The address to which the `bit.ly`'s link refers").

As explicitly written in the exercise's cartridge, you are allowed to use only the 3 following shell functions: `curl`, `grep` and `cut`. Your best starting point is to read the manual of the `curl` function. To do this, simply type `man curl` in your terminal.

Here is an example of the expected performance of your script:

```
$> ./myawesomescript.sh bit.ly/1O72s3U
    http://42.fr
$>
```

The above example shows clearly that your script must be executable. The interpreter to use is `/bin/sh`.

Turn in your script in a file `ex00` at the root of your repository.

Chapter IV

Exercise 01

	Exercise 01

Exercise 01: Your résumé in HTML.
Turn-in directory : *ex*01/
Files to turn in : `cv.html`
Allowed functions : `n/a`

You must produce a résumé in `HTML`/`css` and observe the following rules:

- You must observe the semantic of your HTML beacons and the separation between the background and the shape.

- You must produce an HTML file coherent in it structure with at least a first name, a last name, skills and experience.

- You must display at least a title with the beacon `title` and a title with the beacon `h1`.

- You must use at least a table with the beacons `table`, `th`, `tr` and `td`.

- You must use at least a list with the beacon `ul` and a list with the beacon `ol`. The components must use the beacon `li`.

- The table's borders must be visible (`solid`). The table's borders must be merged (`collapse`).

- You must use a different syntactic solution for each of the two points of the previous instruction : for the first one, use the beacon `style` in the `head` of your page. For the second one, use the symbol `style` in a beacon designed for.

- The cell at the bottom right of each table must have `#424242` as border's color.

> There is no specific instruction about the truth of informations.
> You can make a strange résumé if you want.

Chapter V

Exercise 02

	Exercise 02
	Exercise 02: E-mail sending form.
Turn-in directory : *ex02/*	
Files to turn in : `form.html`	
Allowed functions : `n/a`	

You must produce a form in `HTML` which represent the usual data of any contact form. This form must dispose every of the following fields:
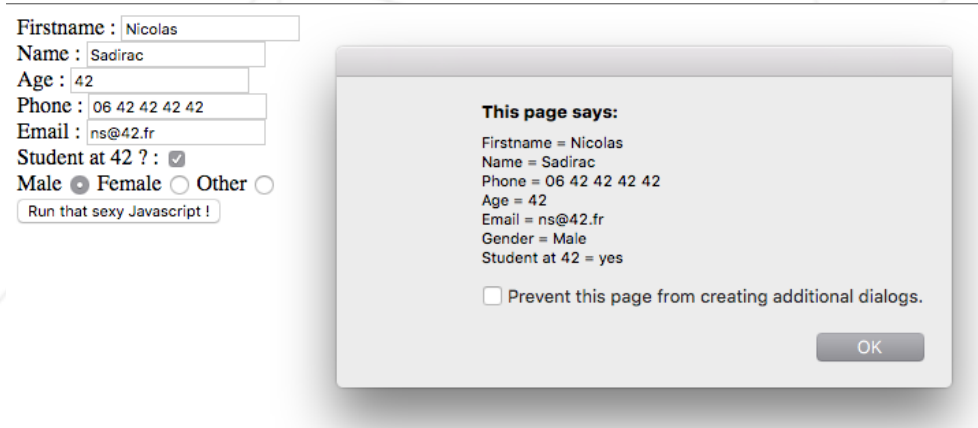
- `Firstname` : a text field.

- `Name` : also a text field.

- `Age` : you must use the `HTML5` specific numeric field.

- `Phone` : you must use the `HTML5` specific phone field.

- `Email` : you must use the `HTML5` specific email field.

- `Student at 42 ?` : you must use the checkbox field.

- `Gender` : you must use radio buttons with `Male`, `Female` and `Other` values.

- A form submission button. The `OnClick` symbol of your button must be '`displayFormContents()`

The `d00.tar.gz` tarball in annex of this subject contains a subfolder `ex02/` which itself contains a `JavaScript` file `popup.js` made by your boss's son during your internship. Because it would have been unacceptable that you made your boss's son pose as an unqualified in programming, you can't modify any of his code and you must use it as it is.

An attentive reading and a superficial understanding of the
JavaScript code provided are required to succeed this exercise.

You must correctly integrate this `JavaScript` file to your page. If your `HTML` code is correct, a click on the form button will show an ultra-modern pop-up containing every fields and values from your form. Otherwise, your `HTML` code is wrong.

Figure V.1: A non-contractual illustration of the expected result.

Chapter VI

Exercise 03

	Exercise 03
	Exercise 03: Reproduce a webpage.
Turn-in directory : *ex03/*	
Files to turn in : `copy.html`	
Allowed functions : `n/a`	

A Rival company has uploaded a more pretty webpage than your's. Through Hollywood style spying methods, your boss obtained screenshots of the page and the related `CSS`. Those two files are at your disposal in the annex of this subject contained in the `ex03/` subfolder traceable in the `d00.tar.gz` archive.

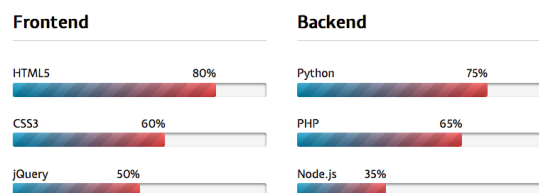You must reproduce this page the most accurately possible.

Figure VI.1: The webpage's screenshot to reproduce.

You must again observe a separation between the background and the shape, respect the beacon's semantic used and keep a logical structure in your document.

You must use the `css` file provided without editing it.

Chapter VII

Exercise 04

■	Exercise 04

Exercise 04: Integration of JS's snippets.
Turn-in directory : *ex04/*
Files to turn in : `snippets.html`
Allowed functions : `n/a`

The `d00.tar.gz` tarball in annex of this subject contains a `ex04/` subfolder which itself contains four files : `file1.js`, `file2.js`, `file3.js` and `file4.js`.

You must create and return a `snippets.html` which must import the four others scripts in such a way that the pop-up displays **correctly** (no weird character).

> ⚠ You can only import the related scripts, you can't edit them or add JavaScript in your HTML code.

Chapter VIII

Exercise 05

	Exercise 05
	Exercise 05: W3C Validation.
Turn-in directory : *ex05/*	
Files to turn in : `Your index.html and the assets's files`	
Allowed functions : `n/a`	

Having code is cool, but having beautiful code is better, to have beautiful code the best way is to follow a beautiful norm.

The `W3C` cannot be round in this sector, you must follow it when you're writing or generating `HTML` code.

You'll find a `ex05/` subfolder in the `d00.tar.gz` tarball which contains the source code of a complete webpage. Unfortunately, this page has been written by an horrible developer !

Edit the `HTML` code of the index.html file to pass the W3C validation. This process means that neither error or warnings are allowed.