



C1 - Piscine PHP

C-COD-150

Day 09

Web at last



Day 09

repository name: pool_php_d09
repository rights: ramassage-tek



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).



+ EXERCISE 01 (2PTS)

Turn in: pool_php_d09/ex_01/index.php

Restrictions: None.

Create a file index.php that display “Hello world” when you access it from a web browser.

+ EXERCISE 02 (2PTS)

Turn in: pool_php_d09/ex_02/index.php

Restrictions: None.



Take a look at the superglobal `$_GET`.

Create a new file index.php which greets the user by his name when you access it from a web browser. If no name is specified, you will use the default name “platypus”. The user name will be given in the URL granting access to the file from a web browser.

Example:

> `http://localhost/index.php?name=Martin` -> displays “Hello Martin”



+ EXERCISE 03 (2PTS)

Turn in: pool_php_d09/ex_03/index.php

Restrictions: None.



\$_SESSION

Create a new file index.php that saves the user name given in the URL so that if you access again the file without specifying the name, the last name will be displayed.

However, if the user closes the browser, the stored data must not be kept.

Example:

- > http://localhost/index.php -> displays "Hello platypus"
- > http://localhost/index.php?name=Martin -> displays "Hello Martin"
- > http://localhost/index.php -> displays "Hello Martin"

+ EXERCISE 04 (1PT)

Turn in: pool_php_d09/ex_04/ex_04.php

Restrictions: None

Create a function remove_session that empties the session, deletes and resets it.

Prototype: void remove_session(void);



+ EXERCISE 05 (2PTS)

Turn in: pool_php_d09/ex_05/index.php

Restrictions: None

Repeat exercise 3, but this time the data stored must be kept even when the user closes the browser. For this, you will create a cookie named "name".

+ EXERCISE 06 (1PT)

Turn in: pool_php_d09/ex_06/ex_06.php

Restrictions: None

Create a function `remove_cookie` that takes as parameter the name of a cookie and deletes it.

Prototype: `void remove_cookie(string $cookie);`

+ EXERCISE 07 (1PT)

Turn in: pool_php_d09/ex_07/ex_07.php

Restrictions: None

Create a function `modify_cookie` that takes as parameter the name of a cookie as well as a value, and refreshes the value of the cookie in question.

Prototype: `void modify_cookie(string $name, mixed $value);`



+ EXERCISE 08 (2PTS)

Turn in: pool_php_d09/ex_08/index.php

Restrictions: None

We are going to ask the user to enter his name, instead of retrieving it from the URL.

Create a page index.php that will offer a form field of type “text” with attribute name “name” so that the user can enter his name.

You will also add a button to validate the form containing the text “Submit”.

Once the user has validated, the same page must greet him with the name entered.

When a greeting is displayed, the form is not displayed.



+ EXERCISE 09 (5PTS)

Turn in: pool_php_d09/ex_09/inscription.php

Restrictions: None

Create a registration form that saves a new user in a database.

You will create a page inscription.php that displays a registration form with the following input fields:

- A field of type "text" with attribute "name" with value "name"
- A field of type "text" with attribute "email" with value "email"
- A field of type "password" with attribute "password" with value "password"
- A field of type "password" with attribute "password_confirmation" with value "password_confirmation"

The name must contain between 3 and 10 characters. It must be saved in the database in the field "name". In case of error, you will need to display "Invalid name" on top of the form.

The email must be valid. It must be saved in the database in the field "email". To check that the email is valid, learn about Regular Expressions. In case of error, you will display "Invalid email" on top of the form.

The password must match its confirmation, must contain 3 to 10 characters, and must be hashed with the PHP function password_hash.

This hash must be saved in the database in the field "password". In case of error, you will display on top of the form "Invalid password or password confirmation".

The creation date of the new user must be saved in the database in the field "created_at". If the user creation succeeds, you will display the form again preceded by the message "User created".



+ EXERCISE 10 (2PTS)

Turn in: pool_php_d09/ex_10/set_color.php

pool_php_d09/ex_10/show_color.php

Restrictions: None

Create a new page set_color.php that displays a form containing a field of type "text" with attribute "name" with value "background".

You will also add a button to validate the form, containing the text "Submit".

When the form is submitted, you will check the validity of the value entered (the value can be "white", "black", "red" or "blue").

If the value is valid, you will bring the user, via a 302 redirection, to the page show_color.php that you will create.

You will also save the entered value in a cookie named "background_color".

If the value is not valid, you will display the form again preceded with the message "Invalid color".

In that case, you will need to make sure there is no more existing cookie named "background_color".

The page show_color.php has the following behavior:

- If there is not an existing cookie named "background_color" or if its value is invalid, you will redirect the user to the page set_color.php with a 302 redirection.
- If a cookie named "background_color" exists and its value is valid, you will apply a background color on the element <body> of the page, corresponding to the value saved in the cookie.