



W1 – Piscine PHP

Jour 04

Responsables Pédagogiques

pedagowac@epitech.eu

Bienvenue dans cette quatrième journée de piscine Web@cadémie ! 😊

Au cours de cette journée nous allons aborder les opérateurs et les structures conditionnelles.

Sommaire

Exercice 01 (1 pts)	3
Exercice 02 (1 pts)	4
Exercice 03 (1 pts)	5
Exercice 04 (1 pts)	6
Exercice 05 (1 pts)	7
Exercice 06 (1 pts)	8
Exercice 07 (1 pts)	9
Exercice 08 (1 pts)	10
Exercice 09 (1 pts)	11
Exercice 10 (1 pts)	12
Exercice 11 (1 pts)	13
Exercice 12 (1 pts)	14
Exercice 13 (1 pts)	15
Exercice 14 (1 pts)	16
Exercice 15 (1 pts)	17
Exercice 16 (1 pts)	18
Exercice 17 (1 pts)	19
Exercice 18 (1 pts)	20
Exercice 19 (1 pts)	21
Exercice 20 (1 pts)	22

Exercice 01 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_01.php

Restrictions : Aucune

Écrire une fonction qui affiche la variable passée en paramètre suivi d'un retour à la ligne si celle-ci est supérieure à 0. Sinon elle n'affiche rien.

Prototype : void print_positive(int \$value);

Exemple :

```
print_positive(42);  
// affiche "42"
```

Exercice 02 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_02.php

Restrictions : N'utiliser qu'une seule fois le mot-clé "if".

Écrire une fonction affiche "Je suis encore mineur." si la variable passée en paramètre est inférieure à 18, ou "Youpi, je suis majeur !" si elle supérieure ou égale à 18. Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_status(int \$age);

Exemple :

```
print_status(42);  
// affiche "Youpi, je suis majeur !"
```

Exercice 03 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_03.php

Restrictions : N'utiliser qu'une seule fois le mot-clé "if".

Écrire une fonction qui affichera "Enfantillage..." pour une variable passée en paramètre inférieure ou égale à 0, "I'm the best." si la valeur est 42, sinon elle affichera "Mon age est cool.". Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_age(int \$age);

Exemple :

```
print_age(42);  
// affiche "I'm the best."
```

Exercice 04 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_04.php

Restrictions : Utiliser le mot-clé "while"

Écrire une fonction qui affiche tous les entiers de 0 à la valeur de la variable passée en paramètre et retourner « true ». Si la valeur est négative, la fonction doit afficher "Nunuche !" et retourner « false ». Un retour à la ligne sera effectué après chaque affichage.

Prototype : bool print_until(int \$max);

Exercice 05 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_05.php

Restrictions :

- Interdiction d'utiliser le mot-clé "if"
- Obligation d'utiliser le mot-clé "do while"

Écrire une fonction qui affiche tous les entiers de la variable passée en paramètre jusqu'à la valeur 42. Si la valeur est supérieure à 42, elle devra quand même être affichée une fois. Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_range(int \$min);

Exercice 06 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_06.php

Restrictions : Obligation d'utiliser le mot-clé "for"

Écrire une fonction qui retourne une chaîne de caractères composée d'autant de "woof" que la valeur de la variable passée en paramètre, suivi d'un retour à la ligne.

Prototype : string get_angry_dog(int \$nbr);

Exemple :

```
get_angry_dog(3);
```

```
// retourne "woofwoofwoof"
```


Exercice 07 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_07.php

Restrictions : Obligation d'utiliser le mot-clé "foreach"

Écrire une fonction qui affiche toutes les valeurs du tableau passé en paramètre, chacune séparée par un retour à la ligne.

Prototype : void print_array(array \$my_array);

Exercice 08 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_08.php

Restrictions : Aucune

Écrire une fonction qui affiche toutes les clés et valeurs du tableau passé en paramètre sous la forme « clé : valeur ». Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_array_with_key(array \$my_array);

Exercice 09 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_09.php

Restrictions : Aucune

Créer une fonction qui affiche (par ordre croissant) tous les multiples compris entre 0 et 150 000 du nombre entier donné en paramètre. Un retour à la ligne sera effectué après chaque affichage.

Prototype : void multiples(int \$nb)

Exercice 10 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_10.php

Restrictions : Obligation d'utiliser le mot-clé "switch"

Écrire une fonction qui prend un nombre entier en paramètre. Si la valeur est 3, elle affiche "Les trois freres". Si la valeur est 6, elle affiche "Sixieme sens". Pour 23, elle affiche "Le nombre 23". Et pour 28, elle affiche "28 jours plus tard". Sinon, elle affiche "Je ne connais pas.". Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_film_from_nbr(int \$nbr);

Exemple :

```
print_film_from_nbr(23);
```

```
// affiche "Le nombre 23"
```

Exercice 11 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_11.php

Restrictions : Aucune

Écrire une fonction qui inclut le fichier dont le nom est passé en paramètre.

Prototype : `void include_file(string $filename);`

Exercice 12 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_12.php

Restrictions : Aucune

Écrire une fonction qui teste les 2 variables passés en paramètre. Si les variables ont la même valeur, elle affiche "Same value.". Si les variables ont le même type, elle affiche "Same type.". Si les variables ont le même type et la même valeur, elle affiche "Same type and value.". Un retour à la ligne sera effectué après chaque affichage.

Prototype : void is_same(mixed \$a, mixed \$b);

Exercice 13 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_13.php

Restrictions :

- Interdiction d'utiliser les mots-clés "if" et "else"
- Obligation d'utiliser les ternaires

Écrire une fonction qui retourne "true" si la valeur de la variable passée en paramètre est supérieure ou égale à 18. Sinon elle retourne "false".

Prototype : bool is_major(int \$age);

Exercice 14 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_14.php

Restrictions :

- Obligation d'utiliser le mot-clé "goto"
- Interdiction d'utiliser "do while", "while", "for"
- Interdiction d'utiliser plus d'une fois "echo"

Écrire une fonction qui affiche 42 fois le mot "woop", suivi d'un retour à la ligne à chaque fois.

Prototype : void brouhaha(void);

Exercice 15 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_15.php

Restrictions : Obligation d'utiliser le mot-clé "foreach"

Écrire une fonction qui remplace toutes les valeurs du tableau passé en paramètre par la chaîne de caractères "poney".

Prototype : void make_poneys(array &\$my_array);

Exercice 16 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_16.php

Restrictions : Obligation d'utiliser le mot-clé "foreach"

Écrire une fonction qui affiche toutes les valeurs du tableau à deux dimensions passé en paramètre. Un retour à la ligne sera effectué après chaque affichage.

Prototype : void print_double_array(array \$my_array);

Exercice 17 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_17.php

Restrictions : Aucune

Créez une fonction "array_key" qui devra retourner la clé de l'élément du tableau dont la valeur est passée en paramètre.

Prototype : mixed array_key(array \$arr, mixed \$value);

Exercice 18 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_18.php

Restrictions : Aucune

Créez une fonction "return_calls" qui ne prend aucun paramètre et qui retourne le double du nombre de fois qu'elle est appelée.

Prototype : int return_calls(void);

Exemple :

```
return_calls();  
return_calls();  
echo return_calls();  
// Le retour du 3ème appel à la fonction devra retourner : 6
```

Exercice 19 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_19.php

Restrictions :

- Interdiction d'utiliser le mot-clé "continue"
- Obligation d'utiliser le mot-clé "break"

Écrire une fonction qui affiche ce qui est lu sur l'entrée standard. La fonction s'arrête après avoir rencontré et affiché la ligne « FIN ».

Prototype : void break_cat(void);

Exercice 20 (1 pts)

Nom du rendu : Piscine_PHP_Jour_04/ex_20.php

Restrictions :

- Obligation d'utiliser le mot-clé "continue"
- Interdiction d'utiliser le mot-clé "break"

Écrire une fonction qui affiche ce qui est lu sur l'entrée standard. La fonction s'arrête après avoir rencontré et affiché la ligne « FIN ».

Prototype : void continue_cat(void);