

PHP Partie 1

Table des matières

1 Compétences pédagogiques :	3
2 PHP c'est quoi ?	4
3 Spécificités du langage PHP	5
PHP est un langage « interprété »	5
PHP est un langage « à typage faible » ou « non-typé »	6
4 Evolution du PHP	7
5 Environnement de travail :	8
6 Syntaxe du langage :	9
Emplacement des fichiers, Exemple :	9
Balises :	9
Exemple :	10
Exemple de code d'une page :	10
Syntaxe générale :	11
Commenter des lignes de codes :	12
Création de notre premier programme en php :	13
7 Les variables :	14
Une variable ça sert à quoi ?	14
Les types de variables :	14
Nommer une variable :	15
Déclaration d'une variable :	15
Exemple :	16
Afficher le contenu d'une variable :	16
Afficher le type d'une variable :	17
Exercice variables :	17
8 Les constantes :	18
9 Les opérateurs :	19
Les opérateurs arithmétiques :	19
Exercices Opérateurs arithmétiques :	20
Les opérateurs d'incrémentation :	21
10 Concaténation :	22

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
 ☑ Sophie POULAKOS
 ☑ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exemple :	22
Exercices :	23
Les opérateurs d'affectation :	24
11 Les Fonctions :	24
Création d'une fonction :	24
Appel d'une fonction :	25
Exemple :	25
Création d'une fonction avec des paramètres :	25
Exemple :	25
Exercices :	26
Typage des paramètres :	26
Exercices :	27

Emplacement table des matières suite.

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

1 Compétences pédagogiques :

Être capable de comprendre le fonctionnement des variables

Être capable de manipuler les opérateurs

Être capable d'utiliser les instructions conditionnelles

Être capable de manipuler un tableau

Être capable de comprendre les boucles

Être capable de créer et d'utiliser des fonctions

Être capable de comprendre le fonctionnement et l'intérêt de la programmation orienté objet

Être capable de créer et utiliser les classes

Être capable de créer et utiliser des objets

Être capable de comprendre les notions d'héritage

Être capable de comprendre les notions de polymorphisme

Être capable de créer des pages Web Dynamique

Être capable de mettre en place un système d'API

Être capable de connecter une application serveur à une base de données côté Back-end

Être capable de gérer des requêtes HTTP d'interaction côté Back-end

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

2 PHP c'est quoi ?

PHP (PHP Hypertext Preprocessor) est un langage de script libre conçu pour le développement d'applications web dynamiques.

Il s'intègre facilement dans du contenu html.

PHP est multiplateforme (Windows, Linux, Mac Os...).

Pour fonctionner PHP a besoin d'être installé sur un serveur web (Apache, NGINX, IIS sont les plus connus).

PHP est un langage qui s'exécute côté serveur et permet la génération de page web dynamique.

L'interpréteur PHP va alors générer une page web html.

De par sa polyvalence, PHP est également un langage de programmation à part entière qui peut être utilisé indépendamment d'un serveur web par exemple pour faire du scripting ou des applications graphiques (par exemple, via la librairie PHP-GTK : <http://gtk.php.net/>)

<https://www.php.net/manual/fr/intro-whatcando.php>

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

3 Spécificités du langage PHP

PHP est un langage « interprété »

On peut distinguer 2 grands types de langages de programmation : les langages interprétés et les langages compilés.

- **Langages interprétés :**
 - L'interpréteur lit le code et exécute les commandes ligne par ligne.
 - Le principal avantage de ces langages est leur côté multiplateforme : dès lors qu'un interpréteur existe sur la plateforme, un même code pourra y être exécuté.
 - Toutefois, cela a un coût en matière de ressources, puisqu'à chaque exécution du programme, l'interpréteur devra traduire les commandes avant qu'elles puissent être exécutées par la machine.
 - Exemples : PHP, Java, Python, JavaScript
- **Langages compilés :**
 - Les instructions sont compilées et traduites en langage machine avant de pouvoir exécuter le programme.
 - Le programme devra donc être compilé sur chaque machine sur lequel on souhaite l'exécuter.
 - Le principal avantage est l'optimisation des ressources nécessaires pour exécuter le programme, que ça soit au niveau de la mémoire nécessaire que de la vitesse d'exécution.
 - Exemples : C, C++, Pascal

Afin d'améliorer les performances des langages interprétés, la plupart de ces langages proposent en réalité une « pré-compilation ».

- **Java** utilise une **machine virtuelle**. Le compilateur Java va en réalité compiler le code en **byte code** et la machine virtuelle sera ensuite chargée d'interpréter ce byte code en langage machine.
- L'interpréteur **Python** propose (sans l'imposer) une option de précompilation des scripts en byte code.
- C'est un peu différent pour **PHP**. Le script est compilé à la volée lors de chaque appel serveur et est ensuite exécuté. Afin d'améliorer les performances et limiter les ressources, des **PHP cache code** ont été développés, permettant de mettre en cache sur le serveur le code compilé.

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
☑ Sophie POULAKOS
☑ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

PHP est un langage « à typage faible » ou « non-typé »

Une autre grande différence des langages de programmation est le typage des variables, celui-ci pouvant être « fort » ou « faible ».

Un langage à typage fort vérifiera que le type de variable attendu est bien celui reçu, et renverra une erreur dans le cas contraire. De plus, les opérations sur des variables de types différents seront interdites.

A contrario, un langage faiblement typé sera beaucoup plus permissif, ce qui peut parfois entraîner des comportements surprenants.

- Exemples de langages à typage fort : C++, Java, Python
- Exemples de langages à typage faible : PHP, JavaScript, C

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

4 Evolution du PHP

PHP a été créé en 1994 par Rasmus Lerdorf pour la création de son site web personnel.

Son travail a été repris en 1997 par deux étudiants : Andi Gutmans et Zeev Suraski qui ont alors publié la version 3 de PHP.

Ils ont ensuite réécrit le moteur interne de PHP, aboutissant au moteur Zend Engine qui a servi de base à la version 4 de PHP. Cette version 4 est également celle qui a commencé à intégrer la programmation orientée objet au langage PHP.

La version 5 a été la version intégrant un modèle de programmation orientée objet complet.

A la suite de ça, de nombreuses évolutions et réécritures du moteur ont permis à PHP de gagner en performance et robustesse en s'inspirant des langages comme Java : support des tests unitaires, typage des méthodes et propriétés, standards de développement (PSR).

Voici une frise illustrant son évolution au fil des années : <https://www.jetbrains.com/fr-fr/lp/php-25/>

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

5 Environnement de travail :

Pour développer en PHP nous allons avoir besoin :

D'un serveur, WAMP (Windows) ou LAMP(Linux) suivant notre environnement de travail.

- Linux / Windows : Le système d'exploitation hébergeant le serveur
- Apache (serveur web pour héberger nos différents fichiers),
- MySQL (serveur de base de données, pour héberger nos bdd),
- PHP (interpréteur PHP),

Pour concevoir nos différents fichiers :

- Un éditeur de code (Visual studio code, PhpStorm, Notepad++, Bracket, Sublime Text, etc...),

Pour tester notre code :

- Un navigateur web pour afficher nos pages tester et contrôler le rendu. (Chrome, Mozilla Firefox, Edge, Safari etc...).

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

6 Syntaxe du langage :

Pour intégrer du code PHP nous écrivons nos scripts à l'intérieur de fichier avec l'extension **php**.

Emplacement des fichiers, Exemple :

Dans le dossier **C:\wamp64\www\exemple\index.php** (exemple du chemin avec WAMP) du serveur apache (WAMP, XAMPP, Docker, LAMP, etc...) nous allons créer un fichier **index.php**.

Par défaut, le serveur web renverra le contenu du fichier **index.php** (ou **index.html**) lorsque l'on visitera le site sans préciser de fichier en particulier.

Balises :

Lorsque PHP traite un fichier, il cherche les balises d'ouverture et de fermeture lui indiquant le code qu'il devra interpréter.

Nos scripts PHP devront être rédigés entre les balises :

- **< ?php ...code PHP... ?>** : ce sont les balises normales.
- **< ? ...code PHP... ?>** : ce sont les balises courtes. Toutefois ces dernières pouvant être désactivées, il est conseillé de n'utiliser que les balises normales.
- **< ?= « du texte » ?>** : ces balises courtes permettent d'afficher du texte. Elles sont l'équivalent de
< ?php echo « du texte » ?>

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exemple :

La page sera accessible dans le navigateur web à l'adresse suivante :

localhost/exemple/index.php

NB : le fichier doit être exécuté et se trouver sur le serveur, si on ouvre simplement le fichier celui ne retournera rien.

Depuis l'exemple précédent nous devons avoir le fichier à l'intérieur du répertoire WWW de Wamp ou HTDOCS de Xamp et créer un sous dossier (dans le dossier à la racine de **www** ou **htdocs** en fonction du logiciel) exemple, enfin créer un fichier **index.php** dans celui-ci. On saisira dans le navigateur web l'adresse suivante (url) **localhost/exemple/index.php**, pour exécuter le fichier. L'interpréteur PHP du serveur va alors lire le fichier **.php** et exécuter le code contenu dans celui-ci.

Exemple de code d'une page :

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>ma première page php</title>
</head>
<body>
  <h1>mon premier programme</h1>
  <?php
    //le script php se trouvera entre ces balises
  ?>
</body>
</html>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023

PHP Partie 1

Syntaxe générale :

- Les instructions sont séparées par des « ; ». Bien que ça ne soit pas obligatoire, il est conseillé d'effectuer un retour à la ligne après chaque instruction.

```
<?php
    //script php;
?>
```

- Les blocs de code (classes, boucles, fonctions) sont délimités par des accolades : « { », « } ».

```
<?php
    Class MaClasse
    {
        function maFonction()
        {
            // Code de la fonction
        }
    }
?>
```

- Indentation : Bien que non obligatoire, on indente nos blocs de code pour une meilleure lisibilité (4 espaces).

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Commenter des lignes de codes :

```
<?php
    //commentaire sur une ligne

    /*
    -----
    Commentaire sur plusieurs lignes
    -----
    */

    /**
    * -----
    * PHPDoc
    * -----
    */
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Création de notre premier programme en php :

Nous allons créer un programme qui va afficher dans le navigateur internet.

Hello World

-Créer une page index.php dans votre éditeur de code et déposer là à l'intérieur de votre dossier **www/cours** du serveur apache (ou **htdocs/cours** si vous utilisez **xampp**).

-A l'intérieur de la page saisir le code ci-dessous :

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Ma première page PHP</title>
</head>
<body>
  <h1>mon premier programme</h1>
  <?php
    //programme Hello Word
    //La commande echo permet d'afficher du contenu dans une page
    html.
    echo "Hello World";
  ?>
</body>
</html>
```

Pourquoi Hello World ?:

<https://deux.io/pourquoi-hello-world/>

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

7 Les variables :

Une variable ça sert à quoi ?

Les variables permettent de stocker des valeurs (Saisies, Résultat d'un sous-programme)

Elles vont pouvoir contenir des valeurs de types différents (texte, numérique...)

Une variable est une sorte de boîte étiquetée avec un contenu.

Pour avoir accès à son contenu nous utiliserons son étiquette (son nom).

Les types de variables :

Bien que le typage en PHP soit un typage faible et dynamique (c'est-à-dire que le type est défini lorsque l'on donne une valeur à la variable), une variable en PHP possède un type de donnée.

- Le type « chaîne de caractères » ou **String** en anglais
- Le type « nombre entier » ou **Integer** en anglais,
- Le type « nombre décimal » ou **Float** en anglais,
- Le type « booléen » ou **Boolean** en anglais,
- Le type « tableau » ou **Array** en anglais,
- Le type « objet » ou **Object** en anglais,
- Le type « NULL » qui se dit également **NULL** en anglais.
- Le type « ressource » ou **Resource** en anglais faisant référence à une ressource externe (ex : stream FTP, PDF document, ...)

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Nommer une variable :

En PHP, les variables sont représentées par un signe « \$ » suivi du nom de la variable.

De plus, le nom d'une variable en PHP doit répondre à certaines contraintes.

- Le nom d'une variable doit impérativement commencer par une lettre ou le symbole underscore « _ »
- Le nom d'une variable ne peut être composé que de lettres, chiffres et de symboles underscore.
- La casse des caractères est prise en compte. Ainsi, les variables \$variable et \$variable sont différentes.
- Par convention, une variable commencera par une lettre en minuscule, et sera écrite en camelCase. De plus, on essaiera toujours de donner un nom compréhensible à la variable pour faciliter la compréhension du code.

Déclaration d'une variable :

En PHP une variable s'écrit comme ci-dessous :

\$nomVariable = valeur;

Le symbole dollars \$ désignera une variable au moment de sa création et quand on l'utilisera.

Le type de la variable est défini lorsque l'on lui attribue une valeur.

Exemple d'utilisation d'une variable :

```
<?php
    $variable = 10;
    $total = $variable + 10; //total vaut 20 (10 de variable + 10 en
numérique).
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exemple :

```
<?php
//Définition d'une variable de type int ayant pour valeur 0
$varInt = 0;

// Définition une variable de type float (nombre à virgule flottante)
$varFloat = 3.4;

// Définition d'une variable de type bool (booléen)
$varBool1 = true;
$varBool2 = false;

/*
Définition d'une variable de type string
On peut utiliser le symbole simple quote (') ou double quote (") pour
définir une chaîne de caractères
*/
$varString1 = 'Une chaîne de caractères';
$varString2 = "Une autre chaîne de caractères";

// Définition d'une variable de type array
$varArray1 = array();
$varArray2 = [];
?>
```

Afficher le contenu d'une variable :

Pour afficher le **contenu** d'une variable nous utiliseront le code ci-dessous :

```
<?php
//initialisation d'une variable
$nbr =2 ;

//la fonction php echo permet d'afficher le contenu de la variable nbr
echo $nbr ;
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Afficher le type d'une variable :

Pour afficher le **type** d'une variable nous utiliserons la fonction « `gettype($maVariable)` » :

```
<?php
//initialisation d'une variable
$nbr =2;

//affichage dans la page web avec la fonction echo
echo $nbr ;

//utilisation de la fonction gettype pour afficher le type de la
variable
echo gettype($nbr); //Ceci affichera "integer"

// Debugger une variable.
var_dump($nbr); // Affichera "int(2)"
?>
```

Exercice variables :

Exercice 1 :

- Créer une variable de type int avec pour valeur 5,
- Afficher le contenu de la variable (utilisation de la fonction php **echo**),
- Afficher son type (utilisation de la fonction php **gettype**),
- Créer une variable de type String avec pour valeur votre prénom,
- Afficher le contenu de la variable (utilisation de la fonction php **echo**),
- Créer une variable de type booléen avec pour valeur false,
- Afficher son type (utilisation de la fonction php **gettype**).

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

8 Les constantes :

Une constante est un identifiant permettant de stocker une donnée simple.

Une fois définie, cette donnée ne peut plus être modifiée.

Par convention, le nom d'une constante est toujours écrit en majuscules.

Il existe deux façons de déclarer une constante :

```
<? php
define('MA_CONSTANTE_1', 'Hello');
const MA_CONSTANTE_2 = 'World!';

echo MA_CONSTANTE_1 . ' ' . MA_CONSTANTE_2;
//affichera "Hello World!"
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

9 Les opérateurs :

Les opérateurs arithmétiques :

Pour effectuer des opérations mathématiques sur des types numériques (int, long, float, etc...)

On utilise les opérateurs mathématiques suivants :

Addition :

$\$a + \b

Soustraction :

$\$a - \b

Multiplication :

$\$a * \b

Division :

$\$a / \b

Modulo :

$\$a \% \b (reste de la division de **$\$a$** divisé par **$\b**)

Exponentielle :

$\$a ** \b (Résultat de l'élévation de **$\$a$** à la puissance **$\b**)

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exercices Opérateurs arithmétiques :

Exercice 1 :

- Créer 2 variables \$a et \$b qui ont pour valeur 12 et 10,
- Stocker le résultat de l'addition de \$a et \$b dans une variable \$total,
- Afficher le résultat (utilisez la fonction **echo**)

Exercice 2 :

- Créer 3 variables \$a, \$b et \$c qui ont pour valeur \$a =5, \$b =3 et \$c = \$a+\$b,
- Afficher la valeur de chaque variable (utilisez la fonction **echo**),
- passer la valeur de \$a à 2,
- Afficher la valeur de \$a,
- passer la valeur de \$c à \$b - \$a,
- Afficher la valeur de chaque variable (utilisez la fonction **echo**).

Exercice 3 :

- Créer 2 variables \$a et \$b qui ont pour valeur 15 et 23,
- Afficher la valeur de chaque variable (utilisez la fonction **echo**),
- Intervertissez les valeurs de \$a et \$b,
- Afficher la valeur de \$a et \$b (utilisez la fonction **echo**).

Exercice 4 :

- Ecrire un programme qui prend le prix HT d'un article (de type float), le nombre d'articles (de type integer) et le taux de TVA (de type float), et qui fournit le prix total TTC (de type float) correspondant.
- Afficher le prix HT, le nbr d'articles et le taux de TVA (utilisez la fonction **echo**),
- Afficher le résultat (utilisez la fonction **echo**).

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Les opérateurs d'incrémentation :

Pré-incrémentation :

++\$a (incrmente \$a de 1 ($a = a + 1$) puis retourne la valeur de \$a)

Post-incrémentation :

\$a++ (retourne la valeur de \$a puis incrmente de 1 ($a = a+1$))

Pré-décrémentation :

--\$a (décrémte \$a de 1 ($a = a - 1$) puis retourne la valeur de \$a)

Post-décrémentation :

\$a-- (retourne la valeur de \$a puis décrémte de 1 ($a = a - 1$))

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

10 Concaténation :

En PHP nous pouvons concaténer des valeurs entres elles. C'est à dire ajouter des chaines de caractères, des nombres, valeurs de variables au sein d'une même chaine de caractères.

Exemple :

Ecrire le nom d'une variable dans une page web :

```
<?php
$nom = 'test';
//on va utiliser le symbole \devant le nom de la variable, ce
caractère permet //d'annuler l'interprétation du caractère qui va
suivre, dans ce cas il va afficher le nom de la variable et non son
contenu.
echo 'affichage de la variable s'appelant \$test' ;
?>
```

Ecrire la valeur d'une variable dans une page web :

```
<?php
$nom = 'test';
echo "affichage du contenu de la variable \$nom : $nom ";
?>
```

Concaténer des chiffres, des chaines de caractères et les afficher dans une page web :

```
<?php
$str = 'une chaine de caractères';
$strLength = strlen($str);
echo "<br>La chaine de caractères '$str' contient $strLength
caractères.";
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Concaténer des variables dans des chaînes de caractères :

```
<?php
$var = 'du texte';

//version avec encadrement ""
$concat1 = "ma chaîne $var";

//version avec encadrement ''
$concat2 = 'ma chaîne ' . $var;
?>
```

Exercices :

Exercice 1 :

- Créer une variable \$a qui a pour valeur « **bonjour** »,
- Afficher le **nom de la variable et sa valeur**.

Exercice 2 :

- Créer 1 variable \$a qui a pour valeur « **bon** »,
- Créer 1 variable \$b qui a pour valeur « **jour** »,
- Créer 1 variable \$c qui a pour valeur **10**,
- Concaténer **\$a, \$b et \$c + 1**,
- Afficher le **résultat** de la concaténation.

Exercice 3 :

- Créer une variable \$a qui a pour valeur **bonjour**,
- Afficher un paragraphe (**balise html**) et à l'intérieur la phrase suivante :

l'adrar

- Ajouter la variable \$a avant la phrase dans le paragraphe,
- Cela doit donner :

<p>bonjour l'adrar</p>

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Les opérateurs d'affectation :

`$a = 5` (affecte la valeur 5 à la variable \$a)

`$a += 3` (affecte la valeur \$a+3 à la variable \$a : équivalent à `$a = $a + 3`)

`$b = 'Bonjour '`; (affecte la valeur 'Bonjour ' à la variable \$b)

`$b .= 'le monde.'`; (Concatène 'le monde.' Et l'affecte à la variable \$b : équivalent à `$b = $b . 'le monde.'`)

11 Les Fonctions :

Les fonctions permettent de générer du code qui va être exécuté chaque fois que l'on va appeler la fonction par son nom. Les fonctions sont créées pour exécuter une tâche précise et permettre de mutualiser le code afin d'améliorer la qualité et maintenabilité du code.

Nous avons déjà utilisé la fonction `gettype()` qui est une fonction interne à PHP.

Nous pouvons trouver la liste des fonctions internes sur la documentation officielle de PHP (<https://www.php.net/manual/fr/funcref.php>) ou sur le site w3school (<https://www.w3schools.com/php/default.asp>).

Une fonction peut prendre des paramètres en entrée et retourner un résultat.

Création d'une fonction :

Pour créer une fonction en php nous allons utiliser la syntaxe suivante :

```
<?php
function nomDeLaFonction()
{
}
?>
```

Le nom d'une fonction devra répondre au mêmes critères de nommage que les variables (commencer par une lettre ou le symbole underscore et ne comporter que lettres, chiffres ou symboles underscore. Par convention le nom d'une fonction sera écrit en camelCase.

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Appel d'une fonction :

Pour appeler une fonction on va saisir le nom de la fonction suivi de `()`

Exemple :

```
<?php
function maFonction()
{
}
maFonction();
?>
```

Création d'une fonction avec des paramètres :

Une fonction avec des paramètres va nous permettre d'exécuter le code de celle-ci et d'adapter son traitement en fonction de ce que l'on va passer en paramètre. Le mot clé **return** permet de retourner (int, string, boolean etc..).

Exemple :

```
<? php
maFonction(10,5);

function maFonction($a,$b)
{
    $result= $a+$b ;

    return $result ;
}
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exercices :

Exercice 1 :

- Créer une fonction qui soustrait à **\$a** la variable **\$b** (2 paramètres en entrée),
- la fonction doit retourner le résultat (**return**).

Exercice 2 :

- Créer une fonction qui prend en entrée un nombre à virgule (**float**),
- la fonction doit retourner l'arrondi (**return**) du nombre en entrée (utiliser une fonction interne au langage).

Exercice 3 :

- Créer une fonction qui prend en entrée **3 valeurs** et retourne **somme** des 3 valeurs.

Exercice 4 :

- Créer une fonction qui prend en entrée **3 valeurs** et retourne la **valeur moyenne** des 3 valeurs (saisies en paramètre).

Typage des paramètres :

Le version 7 de PHP introduit le typage des paramètres d'entrée et de sortie des fonctions.

Cela permet un meilleur contrôle des entrées, ce qui est impératif pour du code de qualité et devra donc être utilisé autant que possible. En effet, si le paramètre reçu par la fonction n'est pas du bon type, le serveur lancera une erreur.

Le type de sortie d'une fonction en retournant aucune valeur est « void ».

Si un paramètre peut être de plusieurs types, on peut utiliser le symbole pipe « | » pour indiquer plusieurs paramètres (exemple : int|float).

Dans l'exemple ci-dessous, on a défini que les paramètres \$a et \$b étaient de type « float », et que le paramètre de sortie était également de type « float ».

```
<? php
maFonction(10,5);

function maFonction(float $a,float $b):float
{
    return $a+$b ;
}
?>
```

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP Partie 1

Exercices :

Exercice :

Modifier les fonctions précédemment écrites en typant les paramètres d'entrée et de sortie.

Auteur :

M. Mithridate / G. Rodrigues

Date création :

25/08/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx-xx-2023



Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.