

PHP

Table des matières

1 Compétences pédagogiques :	6
2 PHP c'est quoi ?	7
3 Environnement de travail :	8
4 Syntaxe du langage :	9
4.1 Emplacement des fichiers, Exemple :	9
4.2 Exemple :	9
4.3 Exemple de code d'une page :	10
4.4 Commenter des lignes de codes :	10
4.5 Création de notre premier programme en php :	11
5 Les variables :	12
5.1 Une variable ça sert à quoi ?	12
5.2 Les types de variables :	12
5.3 Déclaration d'une variable :	12
5.4 Exemple :	13
5.5 Afficher le contenu d'une variable :	13
5.6 Afficher le type d'une variable :	13
5.7 Exercice variables :	14
6 Les opérateurs :	14
6.1 Exercices Opérateurs:	15
7 Concaténation :	16
7.1 Exemple :	16
7.2 Exercices :	16
8 Les Fonctions :	17
8.1 Création d'une fonction :	17
8.2 Appel d'une fonction :	17
8.3 Exemple :	17
8.4 Création d'une fonction avec des paramètres :	18

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

8.5 Exemple :	18
8.6 Exercices :	18
9 Typage des fonctions :	19
9.1 Typage des paramètres.....	19
9.2 typage null :	20
9.3 Typage itération (iterable) :	20
10 Les conditions :	22
10.1 Opérateurs de comparaison :	22
10.2 Opérateurs logiques :	22
10.3 Exemples :	23
10.4 Test Switch case :	23
10.6 Exercices :	24
11 Les boucles :	25
11.1 Exemple boucle for :	25
11.2 Exemple boucle while :	26
11.3 Exemple boucle foreach :	26
11.4 Exercices :	27
12 les tableaux :	28
12.1 Déclaration d'un tableau :	28
12.2 Exemple déclaration de tableaux indexé numériquement et associatif :	29
12.3 Exemple ajouter une valeur à un tableau :	29
12.4 Exemple parcourir un tableau :	30
12.5 Exercices :	30
13 Les super globales :	31
13.1 Fonctionnement GET :	32
13.2 Fonctionnement POST :	34
13.3 Récupération d'inputs checkbox (HTML -> formulaire) mode POST :	36
13.4 Exercices :	38
14 Interaction avec une base de données :	39

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

14.1 Se connecter à la base de données :	39
14.2 Exécution d'une requête SQL :	40
14.3 Exemple de requête classique :	40
14.4 Exemple de requête préparée :	41
14.5 Méthode alternative (utilisation de return) :	42
14.6 Exercices :	43
15 Importer des fichiers Super Globale \$_FILES :	47
15.1 import d'une image Formulaire :	47
15.2 Script d'import PHP :	47
16 Modèle MVC :	50
16.1 Théorie Modèle MCV :	50
16.2 Exemple :	51
16.3 Exercices :	54
17 Classe et objet :	55
17.1 Une classe des objets c'est quoi ?	55
17.2 Créer une classe en PHP :	55
17.3 Instancier un objet :	56
17.4 Ajouter des attributs :	56
17.5 Affecter une valeur à un attribut d'un objet :	57
17.6 Créer et appeler des méthodes :	57
17.7 Constructeur :	59
17.8 Méthode toString :	59
17.9 Exercices :	59
18 Portée des objets (encapsulation) :	62
18.1 Getter et setter :	63
18.2 modifier les méthodes existantes :	64
18.3 Exercices :	64
19 Super Globale SESSION et connexion :	66
19.1 Super Globale SESSION :	66

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

20 Le Routing :	70
20.1 Exemple de schéma de routing :	70
20.1 Réécriture des URL :	71
20.2 Structure du projet en MVC Objet :	72
20.3 Création du routeur :	73
20.4 Script de génération de base de projet MVC Object :	74
21 Héritage (Objet) :	75
21.1 Exemple :	75
21.2 Appel des classes et méthodes :	77
21.2 Redéfinition d'une méthode dans la classe enfant :	77
22 Etendu des classes Héritage (Objet) :	78
22.1 Correction classe Utilisateur :	78
22.2 Surcharge de méthode :	78
22.3 Méthodes de la classe Admin qui utilisent des attributs de la classe Utilisateur :	78
22.4 Appel dans index.php des méthodes de la classe Admin (setActivateUser et getActivateUser) :	79
23 Opérateur de résolution de portées Héritage (Objet) :	80
23.1 Définition :	80
23.2 Utilisation de l'opérateur de portée :	81
24 Structure MVC Avancé :	82
24.1 Contenu du manager :	83
25 Template PHP (ob) :	84
25.1 découpage du template et des vues :	84
25.2 Exemple d'utilisation :	84
26 Namespace et autoloader :	86
26.1 définition :	86
26.2 Syntaxe :	87
26.3 Utiliser une classe dans un autre fichier :	87
26.4 Autolader des classes :	87

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

26.6 Exemple d'autoloader :	88
26.7 Utilisation de l'autoloader :	88
27 Méthodes et attributs static :	88
27.1 Syntaxe :	89
28 Composer :	89
28.1 installer composer :	89
28.2 Vérifier si composer est fonctionnel :	90
28.3 Installer des packages avec composer :	90
29 Mise en place de l'envoi de mails avec le package PHPMailer :	90
29.1 Génération d'un projet PHP MVC :	90
29.2 Installation du package PHPMailer :	90
29.3 Création de la base de données :	91
29.4 Création du Model Contact :	91
29.5 Ajout de la méthode addContact :	92
29.6 Création de la vue formulaire de contact :	93
29.7 Création de la classe utilitaire Utils :	94
29.8 Création du controller ContactController :	95
29.9 Ajout de la route addContact :	95
29.10 Création de la classe Messagerie :	96
29.11 Modification du controller ajout de l'envoi de mail :	98
29.12 Exercices :	99

Emplacement table des matières suite.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

1 Compétences pédagogiques :

Etre capable de comprendre le fonctionnement des variables

Etre capable de manipuler les opérateurs

Etre capable d'utiliser les instructions conditionnelles

Etre capable de manipuler un tableau

Etre capable de comprendre les boucles

Etre capable de créer et d'utiliser des fonctions

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

Etre capable de créer et utiliser les classes

Etre capable de créer et utiliser des objets

Etre capable de comprendre les notions d'héritage

Etre capable de comprendre les notions de polymorphisme

Être capable de créer des pages Web Dynamique

Etre capable de mettre en place un système d'API

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

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

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

2 PHP c'est quoi ?

PHP (Hypertext Preprocessor) est un langage de **script** conçu pour le développement d'application web.

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** pour les plus connus.

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

L'**interpréteur PHP** va alors générer une page web **HTML**.

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

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

3 Environnement de travail :

Pour développer en **PHP** nous allons avoir besoin :

D'un **serveur**, **wamp**, **xampp** (Windows) ou **Lamp** (Linux) suivant notre environnement de travail.

-**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, Notepad++, PHPStorm, 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 :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

4 Syntaxe du langage :

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

4.1 Emplacement des fichiers, Exemple :

Dans le dossier **C:\wamp64\www\exemple\index.php** (exemple du chemin avec wamp) du serveur apache (wamp, Xamp, Laragon, Lamp etc...) nous allons créer un fichier **index.php**.

Nos scripts php devront être rédigés entre les balises :

```
<?php  
?>
```

4.2 Exemple :

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

localhost/exemple/index.php

NB : le fichier doit être exécuter 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.*

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

4.3 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>
```

Chaque ligne de nos scripts devra se terminer par un ;

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

4.4 Commenter des lignes de codes :

```
<?php
  //commentaire sur une ligne
  /*
  -----
  Commentaire sur plusieurs lignes
  -----
  */
?>
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

4.5 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 :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

5 Les variables :

5.1 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).

5.2 Les types de variables :

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.

5.3 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.

Exemple d'utilisation d'une variable :

```
$variable = 10;
```

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

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

5.4 Exemple :

`$varInt` = 0 pour un entier (**int**),

`$varNom` = « nom » pour une chaîne de caractères (**String**).

5.5 Afficher le contenu d'une variable :

Pour afficher le **contenu** d'une variable nous utiliserons 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 ;
?>
```

5.6 Afficher le type d'une variable :

Pour afficher le **type** d'une variable nous utiliserons le code ci-dessous :

```
<?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);
?>
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

5.7 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**).

6 Les opérateurs :

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

On utilise les opérateurs mathématiques suivant :

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 :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

- ☒ Jérôme CHRETIENNE
- ☒ Sophie POULAKOS
- ☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

6.1 Exercices Opérateurs:

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, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC 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 :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

7 Concaténation :

En php nous pouvons concaténer des valeurs entres elles. C'est à dire ajouter des chaines de caractères, des nombres, valeur de variable au sein d'une même suite de caractères.

7.1 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
    echo "ma chaîne de caractères contient 32 caractères";
?>
```

Concaténer des variables dans des chaines de caractères :

```
<?php
    $concat1 = "ma chaîne $var";//version avec encadrement " " ;
    $concat2 = 'ma chaîne '.$var.' ';//version avec encadrement ' ' ;
    $concat3 = 'ma chaîne {$var}'; //version avec les templates String
    '' ;
?>
```

7.2 Exercices :

Exercice 1 :

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

Exercice 2 :

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

- 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 les mots suivants : **l'adrar**,
- Ajouter la variable \$a avant la phrase dans le paragraphe,
- Cela doit donner :

```
<p>bonjour l'Adrar</p>
```

8 Les Fonctions :

Les fonctions permettent de rationaliser du code qui va être exécuté plusieurs fois, plutôt que réécrire de nombreuses fois les mêmes lignes nous allons créer une fonction. La fonction va exécuter le code quelle contient (instructions entre les accolades). Pour utiliser la fonction nous devrons l'appeler par son nom.

8.1 Création d'une fonction :

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

```
<?php
function nom_de_la_fonction()
{
    echo "Ma fonction"; //affiche Ma fonction dans la page HTML
}
?>
```

8.2 Appel d'une fonction :

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

8.3 Exemple :

```
<?php
function ma_fonction() //création de la fonction
{
    echo "Ma fonction"; //affiche Ma fonction dans la page HTML
}
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

```
}  
ma_fonction();//appel de la fonction  
?>
```

8.4 Création d'une fonction avec des paramètres :

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

8.5 Exemple :

```
<? php  
ma_fonction(10,5);  
function ma_fonction($a,$b){  
    $result= $a+$b;  
    return $result;  
}  
?>
```

8.6 Exercices :

Exercice 1 :

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

Exercice 2 :

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

Exercice 3 :

- Créer une fonction qui prend en entrée **3 valeurs** et renvoie la **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).

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

9 Typage des fonctions :

Le langage PHP depuis la version **7.4** à intégrer le **typage des fonctions**. Cela consiste à déterminer le type de donnée des différents paramètres de nos fonctions ainsi que leur valeur de retour (return).

9.1 Typage des paramètres :

Nous allons pouvoir préfixer nos paramètres avec le type de données que l'on attend. Cela implique que le ou les paramètres attendus doivent respecter le typage déclaré.

Exemple de syntaxe :

```
< ?php
    function maFonction(int $nbr){
        echo 'Le nombre '.$nbr.' est bien de
type'.gettype($nbr). '<br>';
    }
```

?>

Si nous donnons une valeur qui ne correspond pas en entrée de la méthode, PHP va nous retourner une **erreur fatale** (**Attention Le script va s'arrêter ici !**).

```
< ?php
    echo '<pre>';
    var_dump(maFonction('toto'));
    echo '</pre>';
```

?>

Fatal error: Uncaught TypeError: maFonction(): Argument #1 (\$nbr) must be of type int, string given string.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

Dans le cas où nous passons en paramètre non pas un **string** mais un nombre **float** ou un **boolean**. La méthode nous renverra la **conversion** et le typage deviendra **integer**.

```
< ?php
    var_dump(maFonction(14.6));
    echo '</pre>';
    echo '<pre>';
    var_dump(maFonction(true));
    echo '</pre>';
?>
```

Le nombre 14 est bien de type integer
Le nombre 1 est bien de type integer

NB :

Il faut donc faire très attention quand on **type** des **paramètres**, car la **vérification** et la **conversion** peuvent changer le comportement de notre méthode et dans le cas **integer** ou on va passer un **string** à la place cela lève une fatale erreur.

9.2 typage null :

De la même manière que nous avons vu le typage (format de données) PHP nous permet de donner un type null à nos paramètres, retour de fonction. Nous utiliserons le ? pour spécifier ce typage.

Exemple :

```
< ?php
    function exemple (?string $variable) :?string{
        return strtoupper($variable) ;
    }
?>
```

NB : Veuillez noter que à partir de **PHP 9** nous ne pourrons plus utiliser le type **null** avec les **chaines de caractères**. Que ce soit avec les fonctions **natives** de PHP ou nos **propres fonctions**. PHP lèvera une **erreur fatale**.

9.3 Typage itération (iterable) :

PHP nous permet d'utiliser un type d'itération, pour passer un ensemble de paramètres et les stocker dans un tableau. Nous n'avons pas besoin donc de les lister. Imaginons que nous avons une fonction ou l'on souhaite ajouter la valeur de tous les paramètres (faire une addition) et que nous ne savons pas à l'avance le nombre de valeurs à ajouter. Le type iterable va nous permettre de résoudre le problème.

```
< ?php
    function argumentVariables2(...$tab){
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

```
$sum = 0;
$cpt = 0;
foreach ($tab as $value) {
    $sum += $value;
    $cpt++;
}
return 'Il y a :'.$cpt.' arguments, La somme vaut :'.$sum;
}
//La méthode fonctionnera peu importe le nombre de valeur que nous
lui assignerons à son exécution.
echo argumentVariables2(12,22,33,55,46) ;
echo argumentVariables2(5,2,3) ;
?>
```

NB : Attention le type **iterable** ne peut être utilisé qu'en **dernier paramètre** d'une fonction.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

10 Les conditions :

Les conditions vont nous permettre de tester, vérifier des valeurs et exécuter dans le cas où la condition se trouve vérifiée le code correspondant. Pour cela nous allons rédiger la syntaxe suivante :

if (si vérifié),

else if (sinon si vérifié, nous pouvons en utiliser plusieurs),

else (sinon tous les autres cas).

10.1 Opérateurs de comparaison :

Exemple	Nom	Résultat
\$a == \$b	Egal	true si \$a est égal à \$b après le transtypage.
\$a === \$b	Identique	true si \$a est égal à \$b et qu'ils sont de même type.
\$a != \$b	Différent	true si \$a est différent de \$b après le transtypage.
\$a <> \$b	Différent	true si \$a est différent de \$b après le transtypage.
\$a !== \$b du même type.	Différent	true si \$a est différent de \$b ou bien s'ils ne sont pas du même type.
\$a < \$b	Plus petit que	true si \$a est strictement plus petit que \$b.
\$a > \$b	Plus grand que	true si \$a est strictement plus grand que \$b.
\$a <= \$b	Inférieur ou égal	true si \$a est plus petit ou égal à \$b.
\$a >= \$b	Supérieur ou égal	true si \$a est plus grand ou égal à \$b.
\$a <=> \$b	Combiné	Un entier inférieur, égal ou supérieur à zéro lorsque \$a est Inférieur, égal, ou supérieur à \$b respectivement.

10.2 Opérateurs logiques :

Nous allons également avoir besoin des opérateurs logiques :

Exemple	Nom	Résultat
\$a and \$b	And (Et)	true si \$a ET \$b valent true.
\$a or \$b	Or (Où)	true si \$a OU \$b valent true.
\$a xor \$b	XOR	true si \$a OU \$b est true, mais pas les deux en même temps.
! \$a	Not (Non)	true si \$a n'est pas true.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

<code>\$a && \$b</code>	And (Et)	true si \$a ET \$b sont true.
<code>\$a \$b</code>	Or (Où)	true si \$a OU \$b est true.

10.3 Exemples :

```
<?php
    $a = 6;
    if($a<=3 and $a >0)
    { //test si $a est plus petit que 3 et est supérieur à 0
        echo "la valeur de la variable \$a est plus petite que 3";
    }
    else if($a>=3 && $a <5)
    { //test si $a est plus grand ou égal et 3 et inférieur à 5
        echo "la valeur de la variable \$a est comprise entre 3 et 5";
    }
    else
    { //test autre cas
        echo "la valeur de la variable \$a est supérieur à 5";
    }
?>
```

10.4 Test Switch case :

Le switch case va nous permettre d'exécuter du code en fonction de la valeur d'une variable. Nous allons gérer des **cas**. Le switch case permet de vérifier différents **cas** (valeurs), le code associé sera alors exécuté si le **cas** est **vérifié** (la valeur correspond).

10.5 Exemple Switch Case :

```
<?php
    $value = 5; //variable value qui vaut 5
    switch($value){//vérification de la valeur contenue dans $value.
        case 1 : //cas si $value vaut 1
            echo '\$value est égale à 1'; //affiche $value est égale à 1
            break; //cette instruction arrête le code ici, arrêt on sort du
switch.
        case 2 : //cas si $value vaut 2 //affiche $value est égale à 2
            echo '\$value est égale à 2'; //affiche $value est égale à 2
            break; //cette instruction arrête le code ici, arrêt on sort du
switch.
        case 5 : //cas si $value vaut 5
            echo '\$value est égale à 5'; //affiche $value est égale à 5
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

```
break; //cette instruction arrête le code ici, arrêt on du  
switch.  
}  
?>
```

10.6 Exercices :

Exercice 1 :

-Créer une fonction qui teste si un nombre est **positif** ou **négatif** (**echo** dans la page web).

Exercice 2 :

-Créer une fonction qui prend en entrée **3 valeurs** et retourne le nombre le plus **grand** (**echo** dans la page web).

Exercice 3 :

-Créer une fonction qui prend en entrée **3 valeurs** et retourne le nombre le plus **petit** (**echo** dans la page web).

Exercice 4 :

-Créer une fonction qui prend en entrée **1 valeur** (l'âge d'un enfant). Ensuite, elle informe de sa **catégorie** (**echo** dans la page web) :

- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" après 12 ans

Bonus : Refaire l'exercice en utilisant le **switch case**.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

- ☒ Jérôme CHRETIENNE
- ☒ Sophie POULAKOS
- ☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

11 Les boucles :

Comme dans tous les langages de programmation, PHP gère les structures de boucle.

La boucle est un élément de base d'un langage de programmation. Les boucles permettent de répéter plusieurs fois une ou plusieurs instructions tant qu'une condition est vérifiée ou bien jusqu'à ce qu'elle soit vérifiée. Les boucles permettent également de parcourir des chaînes de caractères, tableaux et des objets.

Pour écrire une boucle (**for** « **pour** »), nous allons utiliser la syntaxe ci-dessous :

11.1 Exemple boucle for :

Tant que \$i est inférieur à 10 on répète l'opération :

```
<?php
// for (valeur initiale ; condition ; opération)
for ($i=0; $i<10; $i++) //boucle for
{
    echo 'Ceci est une boucle for en PHP';
    echo '<br>';
}
?>
```

La boucle va afficher 10 fois 'Ceci est une boucle for en PHP'.

Pour écrire une boucle (**while** « **tant que** ») :

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

11.2 Exemple boucle while :

Tant que \$i est inférieur à 10 on répète l'opération :

```
<?php
    $i = 0; //variable compteur à l'extérieur de la boucle
    while ($i < 10) //boucle while tant que $i est plus petit que 10
    {
        //j'affiche la valeur de $i
        echo $i;
        //à chaque tour j'incrémente $i (+1)
        $i++;
        //je saute une ligne
        echo '<br>';
    }
?>
```

La boucle va afficher 0 1 2 3 4 5 6 7 8 9 (en sautant une ligne à chaque tour).

Pour écrire une boucle (**foreach** « pour chaque ») :

11.3 Exemple boucle foreach :

Ce type de boucle est en général utilisée pour parcourir un tableau, des chaînes de caractères ou des objets.

Version tableau indexé (numéroté) :

```
<?php
    $tableau = ['valeur1', 'valeur2', 'valeur3', 'valeur4'];
    foreach($tableau as $valeur)
    {
        echo "$valeur<br/> ";
    }
?>
```

Cette boucle va afficher le contenu de chaque colonne du tableau avec un saut à la ligne (valeur1, valeur2, valeur3, valeur4).

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

Version tableau associatif :

```
<?php
    $tableau = [ ' Nom'=>'Mithridate', ' Prénom'=>'Mathieu', ' Age'=> 42];
    foreach($tableau as $cle=> $valeur)
    {
        echo "$cle : $valeur<br/> ";
    }
?>
```

Cette boucle va afficher le nom de la colonne et la valeur associée pour chaque entrée du tableau avec un saut à la ligne (Nom : Mithridate, Prénom : Mathieu, Age : 42).

11.4 Exercices :

Exercice 1 :

Créer un script qui affiche les nombres de 1 -> 5 (méthode echo).

Exercice 2 :

Ecrire une fonction qui prend un nombre en paramètre (variable **\$nbr**), et qui ensuite affiche les dix nombres suivants. Par exemple, si la valeur de nbr équivaut à : 17, la fonction affichera les nombres de 18 à 27 (méthode echo).

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

12 les tableaux :

Un tableau PHP a pour fonction de stocker et manipuler des informations.

Les tableaux, aussi appelés **arrays** en anglais, sont des types de données structurés permettant de grouper des informations ensemble. Les tableaux peuvent stocker une ou plusieurs valeurs à la fois (de types différents).

Lors de la déclaration d'un tableau, il est inutile de préciser sa dimension et le type de données qu'il va contenir. PHP s'en charge tout seul. Chaque fois que l'on va ajouter une nouvelle entrée enregistrée dans le tableau, PHP agrandit sa taille de 1 élément.

Le langage PHP propose également deux types distincts de tableaux : les tableaux à **index numériques** et les tableaux **associatifs**.

12.1 Déclaration d'un tableau :

La déclaration d'un tableau vide se fait de la même manière qu'une variable, c'est à dire avec un signe dollars (\$) et un nom.

Pour déclarer un nouveau tableau, il suffit d'utiliser la structure de langage **array()**. Cette fonction prend en paramètres facultatifs (séparés par une virgule), les valeurs que l'on souhaite insérer dans le tableau pour l'initialiser. Si rien n'est précisé en paramètre, le tableau créé sera vide. Un tableau **commence** toujours à l'**index 0**.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

12.2 Exemple déclaration de tableaux indexé numériquement et associatif :

```
<?php
//déclaration d'un tableau vide (tab) :
$tab = array() ;
//ou version []
$tab = [] ;
//déclaration d'un tableau indexé numériquement :
$tab1 = array(1,8,7,11) ;
//ou version []
$tab1 = [1,8,7,11) ;
//déclaration d'un tableau associatif :
$identite = array(
    'nom' => 'Mithridate',
    'prenom' => 'Mathieu',
    'age' => 41,
    'estFormateur' => true
);
//ou version []
$identite = ['nom' => 'Mithridate', 'prenom' => 'Mathieu', 'age' =>
41, 'estFormateur' => true] ;
?>
```

12.3 Exemple ajouter une valeur à un tableau :

```
<?php
// Ajout d'un élément a un tableau indexé numériquement il sera
ajouté à la dernière position.
$legumes[] = 'salade';
// Ajout d'un élément a un tableau indexé numériquement à une
position (2° position).
$legumes[1] = 'salade';
// Ajout de la taille de la personne dans le tableau associatif
$identite['taille'] = 180;
?>
```

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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

12.4 Exemple parcourir un tableau :

```
<php?
//création d'un tableau $prenoms
$prenoms[0] = 'Mathieu';
$prenoms[1] = 'Sophie';
$prenoms[2] = 'Florence';
//ou
$prenoms = ['Mathieu', 'Sophie', 'Florence'];
//parcours de tout le tableau
foreach ($prenoms as $key => $value) {
    echo '<br>';
    //Affiche le contenu de la case à chaque tour.
    print_r($value);
}
?>
```

12.5 Exercices :

Exercice 1 :

-Créer une fonction qui affiche la valeur la plus **grande** du tableau.

Exercice 2 :

-Créer une fonction qui affiche la **moyenne** du tableau.

Exercice 3 :

-Créer une fonction qui affiche la valeur la plus **petite** du tableau.

Auteur :

Mathieu MITHRIDATE

Date création :

08 / 12 / 2022

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

xx / xx / 20xx



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.