

PHP

Table des matières

1 Compétences pédagogiques :	6
2 PHP c'est quoi ?	7
3 Environnement de travail :	8
4 Syntaxe du langage :	9
Emplacement des fichiers, Exemple :	9
Exemple :	9
Exemple de code d'une page :	10
Commenter des lignes de codes :	10
Création de notre premier programme en php :	11
Pourquoi Hello World ?:	11
5 Les variables :	12
Une variable ça sert à quoi ?	12
Les types de variables :	12
Déclaration d'une variable :	12
Exemple :	13
Afficher le contenu d'une variable :	13
Afficher le type d'une variable :	13
Exercice variables :	14
6 Les opérateurs :	15
Exercices Opérateurs:	16
7 Concaténation :	17
Exemple :	17
Exercices :	18
8 Les Fonctions :	19
Création d'une fonction :	19
Appel d'une fonction :	19
Exemple :	19

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

Création d'une fonction avec des paramètres :	20
Exemple :	20
Exercices :	20
9 Les conditions :	21
Opérateurs de comparaison :	21
Opérateurs logiques :	22
Exemple :	22
Test Switch case :	23
Exemple :	23
Exercices :	24
10 Les boucles :	25
Exemple boucle for:	25
Exemple boucle while :	26
Exemple boucle foreach :	27
Version tableau indexé (numéroté) :	27
Version tableau associatif :	27
Exercices :	28
11 les tableaux:	29
Déclaration d'un tableau :	29
Exemple déclaration de tableaux indexé numériquement et associatif :	30
Exemple ajouter une valeur à un tableau :	30
Exemple parcourir un tableau :	31
Exercices :	31
12 Syntaxes Alternatives :	32
Echo	32
Boucle et Condition	32
If ... Elseif ... Else :	33
Switch :	33
Foreach :	33

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

For / While:.....	33
12 Les super globales :	34
Fonctionnement GET :.....	37
Exemple transfert de données en get :	38
Fonctionnement POST :.....	40
Exemple transfert de données en post :	41
Récupération d'inputs checkbox (HTML -> formulaire) mode POST :	43
Exemple transfert de données en post :	44
Exercices :	45
13 Interaction avec une base de données :.....	46
1 Se connecter à la base de données :.....	46
2 Exécution d'une requête SQL :	47
2.1 Exemple de requête classique :	47
2.2 Exemple de requête préparée :	49
2.3 Exemple de requête préparée avec Binding de Paramètre :.....	50
2.3 Méthode alternative (utilisation de return) :.....	51
Exemple de fonction qui retourne un tableau associatif :	51
Exercices :	52
14 Modèle MVC :.....	56
Exemple :	57
Exercices :	61
15 Super Globale SESSION et connexion :.....	62
1. Super Globale SESSION :	62
2. Exemple :	62
16 Classe et objet :	67
Une classe des objets c'est quoi ?	67
Créer une classe en PHP :	67
Instancier un objet :	68
Ajouter des attributs :	68

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

Affecter une valeur à un attribut d'un objet :	69
Créer et appeler des méthodes :	70
Constructeur.....	71
Méthode toString :	72
Exercices :	73
17 Portée des objets :	75
Getter et setter :	75
1 Passer les attributs de la classe en private :	76
2 Ajouter les méthodes Getter et Setter :	76
3 modifier les méthodes existantes :	76
Exercices :	77
18 La Déclaration de Type en PHP.....	78
1 Rappel des Types.....	78
2 Comment déclarer un type	79
3 Notion de Nullable	80
4 Le Mode Strict	80
19 Héritage (Objet) :.....	81
20 Etendu des classes Héritage (Objet) :.....	84
21 Structure MVC Avancé :.....	87
1 Contenu du manager :.....	88
22 Le Routing.....	89
1 Réécriture des URL	90
2 Structure du projet en MVC	91
3 Création du routeur.....	92
23 Importer des fichiers : la superglobale \$_FILES	94
24 Le Namespacing (Espace de nom).....	98
1 Définition d'un namespace	99
2 Définition d'un sub-namespace (sous espace de nom)	99
3 Utilisation des Namespaces	100

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

4 Accès aux classes, fonctions, constantes et interfaces globales dans un Namespace	100
5 Alias de Namespaces	101

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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, IIS pour 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.

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

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

3 Environnement de travail :

Pour développer en PHP nous allons avoir besoin :

D'un serveur, wamp (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++, 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 :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

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

Date révision :

01-12-2023



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

4 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, Xamp, Laragon, Lamp etc...) nous allons créer un fichier **index.php**.

Nos scripts php devront être rédigés entre les balises :

```
<?php
```

```
?>
```

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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;
?>
```

Commenter des lignes de codes :

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

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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 **xamp**).

-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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

5 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 :

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.

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Exemple :

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

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

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 ;
?>
```

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

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

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

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

Date création :

01-12-2023

Date révision :

01-12-2023



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

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.

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 "<br>ma chaine de caractères contient 32 caractères";
?>
```

Concaténer des variables dans des chaines de caractères :

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

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

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

Date création :

01-12-2023

Date révision :

01-12-2023



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

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 devons l'appeler par son nom.

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
    }
?>
```

Appel d'une fonction :

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

Exemple :

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

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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..).

Exemple :

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

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

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

Date création :

01-12-2023

Date révision :

01-12-2023



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

9 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).

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	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.

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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 (Ou)	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.
\$a && \$b	And (Et)	true si \$a ET \$b sont true.
\$a \$b	Or (Ou)	true si \$a OU \$b est true.

Exemple :

```
<?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";
}
?>
```

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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).

Exemple :

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

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

- ☑ Jérôme CHRETIENNE
- ☑ Sophie POULAKOS
- ☑ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

10 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:

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'.

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

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

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
    //boucle while tant que $i est plus petit que 10
    while ($i < 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).

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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

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
    $tableau = array(' valeur1', ' valeur2', ' valeur3', ' valeur4');
    foreach($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).

Version tableau associatif :

```
<?php
    $tableau = array('Nom'=>'mithridate', 'Prénom'=>'mathieu', 'Age'=> 42);
    foreach($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).

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

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 :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

11 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**.

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Exemple déclaration de tableaux indexé numériquement et associatif :

```
<?php
//déclaration d'un tableau vide (tab) :
$tab = array() ;

//déclaration d'un tableau indexé numériquement :
$tab1 = array(1,8,7,11) ;

//déclaration d'un tableau associatif :
$identite = array(
    'nom' => 'mithridate',
    'prenom' => 'mathieu',
    'age' => 41,
    'estFormateur' => true
);
?>
```

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Exemple parcourir un tableau :

```
<?php
//création d'un tableau $prenoms
$prenoms[0] = 'Mathieu';
$prenoms[1] = 'Sophie';
$prenoms[2] = 'Florence';

//ou
$prenoms = array('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);
}
?>
```

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 :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

12 Syntaxes Alternatives :

Pour certaines instructions de PHP, ce dernier intègre des syntaxes alternatives qui raccourcissent leur écriture et rend leur inclusion dans du HTML un peu plus simple.

Echo

La syntaxe alternative d'un echo est la suivante, avec un signe = qui remplace le terme php au niveau du chevron ouvrant :

```
<?= "Ce que je veux afficher" ?>
```

Exemple :

```
<?php  
    $message = "Cette Syntaxe d'Echo est un succès !";  
?>
```

```
<!--Ci Dessous le HTML-->  
<h1><?= "Voyons un Echo en Raccourcis" ?></h1>  
<p><?= $message ?></p>
```

Boucle et Condition

Dans le cas des boucles et des conditions, leur syntaxe alternative permet de prendre en compte du code qui se trouve normalement en dehors des balises PHP.

Cette syntaxe est une variation sur la forme :

```
<?php commande(): ?>  
...code à prendre en compte (comme le HTML)...  
<?php commande_complémentaire: ?>  
...code à prendre en compte (comme le HTML)...  
<?php endcommande; ?>
```

Dans ce type de syntaxe, les blocs de code sont définis par le symbole :

De plus, la syntaxe se termine toujours par une balise end (auquel est collé le nom de l'instruction).

Cette syntaxe alternative concerne les instructions if, switch, foreach, for et while.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

If ... Elseif ... Else :

```
<?php if(condition): ?>
...affichage HTML, CSS, ou JS...
<? elseif(condition): ?>
...affichage HTML, CSS, ou JS...
<? else: ?>
...affichage HTML, CSS, ou JS...
<? endif; ?>
```

Switch :

```
<?php switch ($var): ?>
<?php case 1: ?>
    ...
<?php endswitch; ?>
```

ATTENTION : il est important de ne mettre aucun espace devant les balises PHP contenant les case. En effet, ces derniers seront interprétés comme du code à afficher, ce qui résultera en une erreur du script.

Foreach :

```
<?php foreach($tab as $row): ?>
...Code HTML à afficher...
<?php endforeach; ?>
```

For / While:

```
<?php for(paramètre de for): ?>
...Code HTML à afficher...
<?php endfor; ?>
```

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

13 Le Buffering

PHP est équipé de fonctions de mise en tampon (le buffer) permettant de contrôler le moment où les données sont lues. Ces fonctions de PHP concernent les affichages par un echo, ainsi que le code qui s'écrit en dehors des balises PHP, comme du HTML par exemple.

Cela est pratique lorsque l'on souhaite réaliser des Templates. On peut ainsi écrire du HTML de manière normale, plutôt que de devoir l'écrire en tant que String.

Exemple sans Buffering :

Pour passer de l'affichage HTML à notre page, via un exemple de template, on doit écrire notre HTML dans une variable sous forme chaîne de caractère. Ceci ne rend pas le HTML très lisible.

```
<?php
$message = "<h2>Voyons le Buffering</h2><p>Ca marche du tonerre !</p>";
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemple de Buffering</title>
</head>
<body>
    <? = $message ?>
</body>
</html>
```

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Exemple avec Buffering :

En utilisant la mise en tampon, on peut écrire le HTML normalement, puis le récupérer en une fois dans une variable. C'est bien plus lisible.

```
<?php
ob_start();

<h2>Voyons le Buffering</h2>
<p>Ca marche du tonnerre !</p>

<?php
$message = ob_get_clean();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemple de Buffering</title>
</head>
<body>
    <?=$message ?>
</body>
</html>
```

Ob_start() et Ob_get_clean() :

Pour utiliser le buffering, nous avons utiliser les fonctions ob_start() et ob_get_clean().

Ob_start() va permettre de collecter et mettre en tampon tout le code (echo et html) qui le suit.

Ob_get_clean() va permettre de récolter tout le code mis en tampon, ce qui nous permet de l'assigner à une variable.

Ce ne sont pas les deux seules fonctions de bufferisation existantes. PHP en bien d'autres.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

14 Les super globales :

Le transfert de données entre des pages web est géré en PHP par le biais de variables spéciales qui s'appellent super globale. Dans cette partie nous allons voir les Super Globales suivantes : **\$_GET** et **\$_POST**.

Chacune de ces variables va récupérer dans un tableau le contenu des différents champs html d'un formulaire.

Les formulaires html possèdent 2 méthodes d'envoi possibles **get** et **post**.

Get fait passer les informations dans l'url de la page, cette méthode est dangereuse car elle affiche dans l'url de la page le nom des variables et leur contenu.

Post fait passer les informations par le body de la page cette méthode est à privilégier car elle est plus sécurisée et surtout elle permet de transférer des informations de taille plus importante.

On devra utiliser l'attribut html (**name**) pour chaque élément du formulaire.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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.

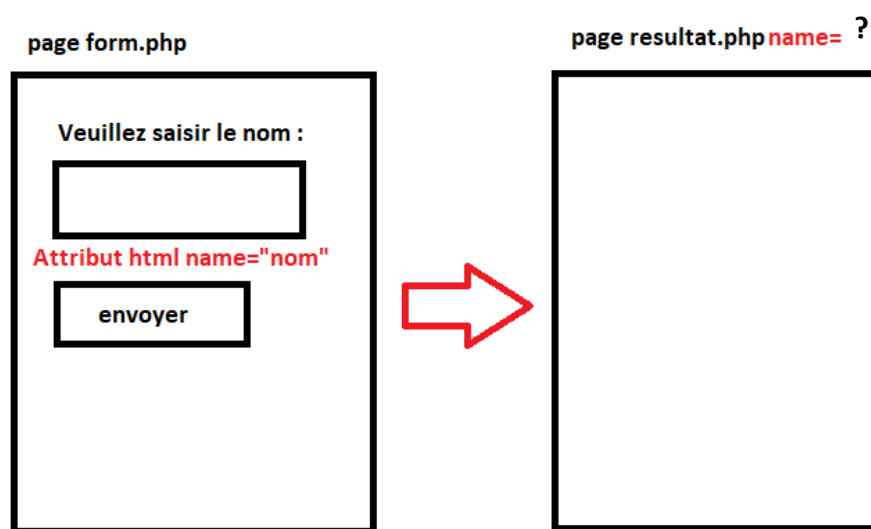
PHP

Fonctionnement GET :

Le contenu des champs de formulaire va transiter dans l'url à la condition de nommer ces champs avec l'attribut html **name**.

Schéma transfert d'informations GET :

Méthode GET



Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
☑ Sophie POULAKOS
☑ Mathieu PARIS

Date révision :

01-12-2023

PHP

Exemple transfert de données en get :

Page form.php

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>formulaire</title>
</head>
<body>
  <form action="resultat.php" method="get">
    <p>veuillez saisir votre nom :</p>
    <input type="text" name="nom">
    <br>
    <input type="submit" value="Envoyer">
  </form>
</body>
</html>
```

Cette page va envoyer à la page resultat.php le contenu du champ nom dans l'url sous la forme suivante :

<http://resultat.php?nom=valeur>.

Si l'on avait plusieurs champs dans le formulaire avec l'attribut name, ils seront séparés par le caractère & :

<http://resultat.php?nom=valeur&prenom=valeur>

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Page resultat.php

```
<?php
//test de l'existence de la super globale $_GET
if(isset($_GET['nom'])){
    $nom = $_GET['nom'];
    echo "mon nom est : ".$nom."";
}
?>
```

Dans cette page nous allons afficher le contenu de la super globale `$_GET['nom']` avec la fonction **echo**.

- 1 On vérifie l'existence de la super globale `$_GET['nom']` avec la fonction PHP **isset()** qui teste si la variable **existe** et si sa **valeur** n'est pas égal à **null**.
- 2 Ensuite on va afficher le contenu avec la méthode **echo** que l'on a vue précédemment et on concatène le résultat avec la chaîne **mon nom est Mathieu**.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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.

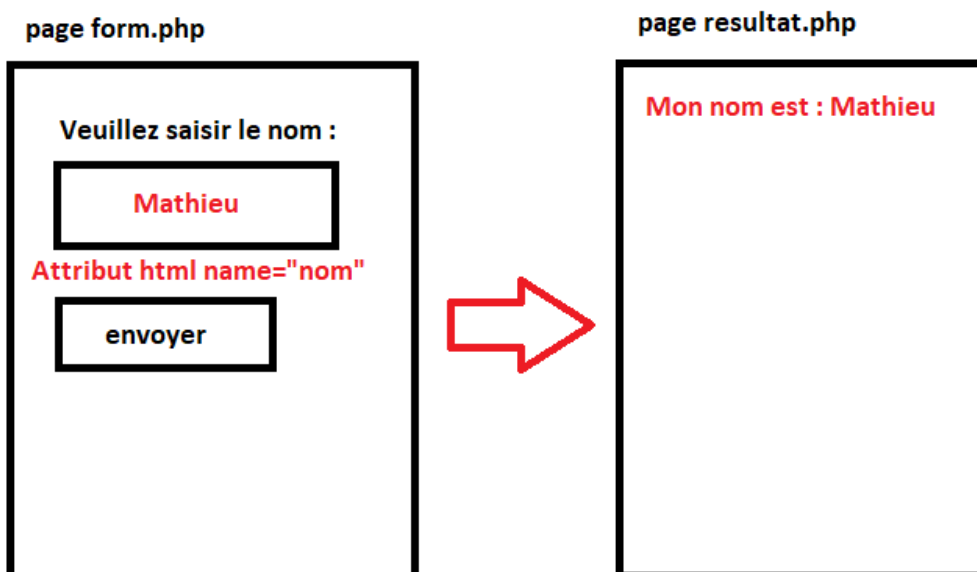
PHP

Fonctionnement POST :

Le contenu des champs de formulaire va transiter par le body de la page à la condition de nommer ces champs avec l'attribut html **name**.

Schéma transfert d'informations POST :

Méthode POST



Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023

PHP

Exemple transfert de données en post :

Page form.php

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>formulaire</title>
</head>
<body>
  <form action="resultat.php" method="post">
    <p>veuillez saisir votre nom :</p>
    <input type="text" name="nom">
    <br>
    <input type="submit" value="Envoyer">
  </form>
</body>
</html>
```

Cette page va envoyer à la page *resultat.php* le contenu du champ nom dans le **body**.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Page resultat.php

```
<?php
    //test de l'existence de la super globale $_POST
    if(isset($_POST['nom'])){
        $nom = $_POST['nom'];
        echo "mon nom est : ".$nom."";
    }
?>
```

Dans cette page nous allons afficher le contenu de la super globale `$_POST['nom']` avec la fonction **echo**.

1 On vérifie l'existence de la super globale `$_POST['nom']` avec la fonction PHP **isset()** qui teste si la variable **existe** et si sa **valeur** n'est pas égal à **null**.

2 Ensuite on va afficher le contenu avec la méthode **echo** que l'on a vue précédemment et on concatène le résultat avec la chaîne **mon nom est : Mathieu**.

NB :

Si l'on souhaite traiter les données dans la page de formulaire, dans la partie action (html) on laisse soit le champ **vide** ou on saisie **#**

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

Récupération d'inputs checkbox (HTML-> formulaire) mode POST :

Les cases à cocher (input->checkbox en HTML) se récupèrent de la façon suivante (si la case est cochée) en PHP :

1 Création d'un formulaire (méthode POST en HTML),

Dans le formulaire nous allons ajouter des inputs de type checkbox comme ci-dessous :

```
<p><input type="checkbox" name="box[]" value="1"/>1</p>
```

L'attribut **name** de chaque checkbox doit être le même, l'attribut **value** doit être unique. La super globale POST va stocker un tableau de toutes les attributs **value**.

Nous récupérerons le contenu de **value** avec une boucle **foreach** de **box[]** (car c'est un tableau, il s'appelle box pour l'exemple).

NB : Attention, les **checkbox** n'existent dans la super globale **POST** que si elles sont **cochées**.

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

Exemple transfert de données en post :

Page form.php

```
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Récupération des checkBox cochées :</title>
</head>
<body>
    <h4>Cocher une ou plusieurs checkbox :</h4>
    <!--Formulaire HTML-->
    <form action="" method="post">
        <p><input type="checkbox" name="box[]" value="1"/>1</p>
        <p><input type="checkbox" name="box[]" value="2"/>2</p>
        <p><input type="checkbox" name="box[]" value="3"/>3</p>
        <p><input type="checkbox" name="box[]" value="4"/>4</p>
        <p><input type="checkbox" name="box[]" value="5"/>5</p>
        <p><input type="submit" value="Récupérer"></p>
    </form>
    <h4>Liste des checkbox cochées :</h4>
    <!--Code PHP-->
    <?php
        //vérification de la super globale $_POST['box']
        if(isset($_POST['box'])){
            //boucle pour parcourir chaque case cochés ($value équivaut à
value en HTML)
            foreach($_POST['box'] as $value){
                echo "<p>id de la box : $value</p>";
            }
        }
        else{
            echo "<p>Veuillez cocher une ou plusieurs checkbox</p>";
        }
    ?>
</body>
</html>
```

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE

☒ Sophie POULAKOS

☒ Mathieu PARIS

Date révision :

01-12-2023



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

Exercices :

Exercice 1 :

-Créer une page de formulaire dans laquelle on aura 2 champs de formulaire de type nombre.

-Afficher dans cette même page la somme des 2 champs avec un affichage du style :

*La somme est égale à : **valeur**.*

Exercice 2 :

-Créer une page de formulaire dans laquelle on aura 3 champs de formulaire de type nombre :

1 champ de formulaire qui demande un prix HT d'un article,

1 champ de formulaire qui demande le nombre d'article,

1 champ de formulaire qui demande le taux de TVA,

-Afficher dans cette même page le prix TTC (prix HT*taux TVA*quantité) avec un affichage du style :

*Le prix TTC est égal à : **valeur €**.*

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

15 Interaction avec une base de données :

Le langage PHP permet d'interagir de façon simple et sécurisé (dans certains cas) avec des bases de données de type MYSQL (propriétaire ORACLE) ou MARIADB (équivalent open source).

Pour se faire nous devons respecter certaines étapes :

- Se connecter à la base de données,
- Exécuter la requête SQL,
- Récupérer le résultat dans une variable (pour les requêtes de type select)

1 Se connecter à la base de données :

La première des actions à effectuer pour interagir avec une base de données est de se connecter à celle-ci.

Pour se faire nous utiliserons la syntaxe suivante :

```
<?php
    //connexion à la base de données
    $bdd = new PDO('mysql:host=localhost;dbname=nom_de_la_bdd', 'root', '',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
?>
```

Cette ligne de code va stocker dans une variable \$bdd un objet **PDO** (que vous verrez dans les chapitres prochains) qui va contenir les attributs suivants :

- mysql :host = **localhost**; (base de données de type MySQL dont l'url est localhost : identique au serveur apache) et son nom dbname = **nom_de_la_bdd**
- le paramètre suivant est le nom du compte dans l'exemple ci-dessus : **'root'**,
- le paramètre suivant est le mot de passe dans l'exemple ci-dessus il est vide : **''**,
- le paramètre **array** permet de spécifier le mode de l'objet **PDO** dans l'exemple ci-dessus il active le mode d'erreur avancé (*code d'erreur SQL*).

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

- ☑ Jérôme CHRETIENNE
- ☑ Sophie POULAKOS
- ☑ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

2 Exécution d'une requête SQL :

Pour interagir avec notre base de données et exécuter des requêtes SQL il existe plusieurs méthodes nous allons en voir 2 types :

- Les requêtes classiques qui ne sont pas sécurisées (*elles sont sensibles aux attaques par injection SQL*),
- Les requêtes préparées qui elles sont plus sécurisées et bloquent l'**injection SQL**.

2.1 Exemple de requête classique :

En premier lieu nous devons nous connecter à la base de données (en utilisant le code vu dans la partie 1 du chapitre 13) :

```
<?php
    //Connexion à la base de données
    $bdd = new PDO('mysql:host=localhost;dbname=nom_de_la_bdd', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));

    //Exécution de la requête SQL avec un try catch pour la gestion des
exceptions (messages d'erreurs)
    try {
        //requête pour stocker le contenu de toute la table le contenu est
stocké dans la variable $data, $req stocke la requête SQL.
        $req = $bdd->query('SELECT * FROM utilisateur');

        //boucle pour parcourir et afficher le contenu de chaque ligne de la
table
        while($data = $req->fetch()){
            //affichage les informations d'une colonne de la bdd par son nom
d'attribut
            echo '<p>'.$data['nom_attribut'].'</p>';
        }
    }
    catch(Exception $e){
        //affichage d'une exception en cas d'erreur
        die('Erreur : '.$e->getMessage());
    }
?>
```

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

Cette requête va stocker dans une variable **\$data** le résultat de toute la requête **select**, (un tableau qui contient le résultat).

La boucle **while** va nous permettre de parcourir le contenu de la variable **\$data** et afficher pour chaque enregistrement de la base de données le contenu d'un **attribut** de la table sous la forme :

\$data['nom_attribut'],

-L'afficher avec la méthode **echo** dans un paragraphe **html** (balise **p**).

Try catch :

Le paramètre **PDO::ERRMODE_EXCEPTION** dans le fichier de connexion active la gestion des exceptions.

Le **try catch** va nous permettre d'exécuter le code dans le **try**, s'il y a une erreur (requête, connexion ou autre) le message d'erreur sera redirigé dans le **catch** et s'affichera dans la page (code erreur SQL).

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

2.2 Exemple de requête préparée :

Notre requête préparée va exécuter une requête **SQL** de type **select** similaire à la requête classique ci-dessus mais dans laquelle nous allons lui passer un **paramètre** (**\$nom_utilisateur**) qui contiendra un nom d'utilisateur.

```
<?php
    //Connexion à la base de données
    $bdd = new PDO('mysql:host=localhost;dbname=nom_de_la_bdd', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));

    //Préparation de la requête SQL nous stockons dans une variable $req la
    requête à exécuter
    $req = $bdd->prepare('SELECT * FROM utilisateur where nom_utilisateur =
:nom_utilisateur');

    //Exécution de la requête SQL création à l'aide d'un tableau qui va
    contenir le ou les paramètres à affecter à la requête SQL
    $req->execute(array(
        'nom_utilisateur' => iconv("UTF-8", "ISO-8859-1//TRANSLIT",
$nom_utilisateur),
    ));

    //boucle pour parcourir et afficher le contenu de chaque ligne de la table
    while ($donnees = $req->fetch()){
        //affichage des données d'une colonne du résultat de la requête par
        son nom d'attribut (nom champ bdd)
        echo '<p>'.$donnees['nom_attribut'].'</p>';
    }
?>
```

Cette requête effectue le même traitement que la requête classique mais de façon sécurisé.

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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

2.3 Exemple de requête préparée avec Binding de Paramètre :

Notre requête préparée est une variante de la requête précédente qui utilise du **Binding de Paramètres**. Nous utilisons un **?** comme marqueur pour injecter nos paramètres.

```
<?php
    //Connexion à la base de données
    $bdd = new PDO('mysql:host=localhost;dbname=nom_de_la_bdd', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));

    //Préparation de la requête SQL nous stockons dans une variable $req la
    requête à exécuter
    $req = $bdd->prepare('SELECT * FROM utilisateur where nom_utilisateur =
?');

    //Binding de Paramètre, nous attribuons une valeur à chaque ? de notre
    requête, en indiquant sa position et son typage dans les paramètres
    $req->bindParam(1, $nom_utilisateur, PDO::PARAM_STR);

    //Exécution de la requête SQL création à l'aide d'un tableau qui va
    contenir le ou les paramètres à affecter à la requête SQL
    $req->execute();

    //boucle pour parcourir et afficher le contenu de chaque ligne de la table
    while ($donnees = $req->fetch()){
        //affichage des données d'une colonne du résultat de la requête par
        son nom d'attribut (nom champ bdd)
        echo '<p>'.$donnees['nom_attribut'].'</p>';
    }
?>
```

Cette requête effectue le même traitement que la requête classique mais de façon sécurisé.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

2.3 Méthode alternative (utilisation de return) :

Dans une requête (classique ou préparée) nous avons la possibilité de faire en sorte qu'elle retourne un tableau :

Associatif ou bien un tableau d'objet. Nous allons pour se faire utiliser la méthode de la classe PDO **fetchAll**.

La méthode fetchAll prend en paramètre les options suivantes :

FETCH_ASSOC -> renvoie un tableau associatif **fetchAll(PDO::FETCH_ASSOC)**,

FETCH_OBJ -> renvoie un tableau associatif **fetchAll(PDO::FETCH_OBJ)**.

Le mot clé **return** va renvoyer en sortie l'option choisie (tableau associatif ou tableau d'objet).

Exemple de fonction qui retourne un tableau associatif :

```
<?php
function showAllArticle($bdd):array{
    try{
        $req = $bdd->prepare('SELECT * FROM article');
        $req->execute();
        $data = $req->fetchAll(PDO::FETCH_ASSOC);
        return $data;
    }
    catch(Exception $e)
    {
        //affichage d'une exception en cas d'erreur
        die('Erreur : '.$e->getMessage());
    }
}
?>
```

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023

PHP

Exercices :

Exercice 1 :

a) Créer une base de données **MYSQL** avec les informations suivantes :

-Nom de la bdd : « **articles** »,

-une table nommée **article** qui va posséder les champs suivants :

id_article (clé primaire),

nom_article de type varchar(50),

contenu_article de type varchar (255),

b) Créer une page php qui va contenir un formulaire html avec comme méthode POST (balise **form**)

-A l'intérieur du formulaire rajouter les champs suivants :

Un champ input avec comme attribut html **name = «nom_article »**,

Un champ input avec comme attribut html **name = «contenu_article »**,

Un champ input de type **submit** avec comme attribut html **value = «Ajouter»**

c) Ajouter le code php suivant :

-Créer 2 variables \$name, \$content

-Importer le contenu des 2 super globales **\$_POST['nom_article']**, **\$_POST['contenu_article']** et tester les avec la méthode **isset()** dans les variables créés précédemment (**\$name** et **\$content**),

-Ajouter le code de **connexion** à la base de données en vous inspirant des exemples vus dans ce chapitre,

-Ajouter une **requête simple** qui va insérer le contenu des 2 champs dans un nouvel enregistrement (requête **SQL insert**),

d) Bonus :

-Utiliser une requête **SQL préparée avec Binding de Paramètres** à la place de la requête **simple**.

-Afficher dans un paragraphe le nom et le contenu de l'article ajouté en bdd en dessous du formulaire.

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

☒ Jérôme CHRETIENNE

☒ Sophie POULAKOS

☒ Mathieu PARIS

Date création :

01-12-2023

Date révision :

01-12-2023



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

Exercice 2 :

- a) Créer une page php,
- b) Ajouter le script php permettant de se connecter à la base de données **articles**,
- c) Ajouter le script php qui va effectuer une requête **SQL select préparée** permettant de récupérer tous les articles,
- d) Formater le résultat de la requête (dans le résultat de la boucle **while**) pour quelle l'affiche sous cette forme :

<p>numéro de l'article : id de l'article n</p>

<p>nom de l'article : nom de l'article n</p>

<p>contenu de l'article : contenu de l'article n</p>

(La liste de tous les articles devra reprendre la mise en forme ci-dessus -> a l'intérieur de la boucle **while**).

NB : On peut également utiliser **return** dans la fonction, et parcourir le résultat avec une boucle **foreach** :

```
<?php
    $list = getAllArticle($bdd); // on stocke le résultat de la fonction dans
    une variable
    foreach($list as $value){
        echo '<p>Numéro de l'article : '.$value['id_article'].'</p>' ;
    } /* on parcourt avec foreach le contenu du tableau et on echo le
    résultat.*/
?>
```

Auteur :

Yoann DEPRIESTER

Relu, validé & visé par :

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

Date création :

01-12-2023

Date révision :

01-12-2023



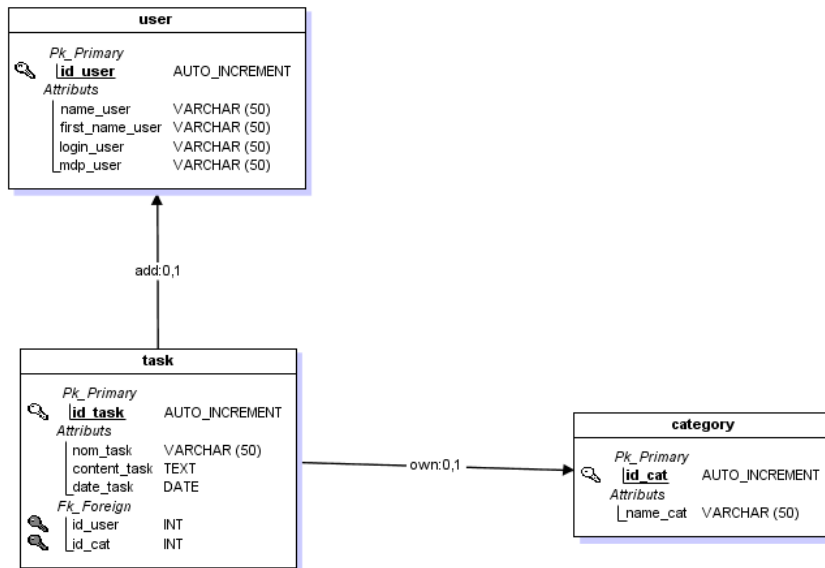
Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

PHP

Exercice 3 PROJET TASK Partie 1 :

a) Créer une base de données **MYSQL** depuis le **MLD** ci-dessous :

-Nom de la BDD : «**task**»,



b) Créer une page php qui va contenir un formulaire html avec comme méthode POST (balise **form**) cette page va nous permettre de créer nos comptes utilisateurs et les sauvegarder dans la base de données.

-A l'intérieur du formulaire ajouter les champs suivants :

Un champ input avec comme attribut html **name = «name_user»**,

Un champ input avec comme attribut html **name = «first_name_user»**,

Un champ input avec comme attribut html **name = «login_user»**,

Un champ input avec comme attribut html **name = «mdp_user»**,

Un champ input de type **submit** avec comme attribut html **value = « Ajouter »**

c) Ajouter le code php suivant :

-Créer 4 variables \$name_user, \$first_name_user, \$login_user, \$mdp_user,

-Importer le contenu des super globales \$_POST['name_user'], \$_POST['first_name_user'], \$_POST['login_user'], \$_POST['mdp_user'], et tester les avec la méthode **isset()** (dans la **condition if**) dans les variables créées précédemment (\$name_user, \$first_name_user, \$login_user, \$mdp_user),

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023

PHP

- Ajouter le code de **connexion** à la base de données en vous inspirant des exemples vus dans ce chapitre,
- Ajouter une **requête préparée avec Binding de Paramètres** qui va insérer le contenu des 4 champs dans un nouvel enregistrement (requête **SQL insert**),
- Afficher après l'insertion en base de données les informations que vous avez saisies (nom, prenom, login, mot de passe).
- d) Bonus :
 - Afficher en bas de la page la liste des comptes utilisateurs créés avec une requête **SQL select**,
 - Utiliser une requête **SQL préparée**.

Auteur :

Yoann DEPRIESTER

Date création :

01-12-2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

01-12-2023



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.