

PHP

Table des matières

1 Compétences pédagogiques :	5
2 PHP c'est quoi ?	6
3 Environnement de travail :	7
4 Syntaxe du langage :	8
Emplacement des fichiers, Exemple :	8
Exemple :	8
Exemple de code d'une page :	9
Commenter des lignes de codes :	10
Création de notre premier programme en php :	10
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:	15
7 Concaténation :	17
Exemple :	17
Exercices :	18

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

8 Les Fonctions :	19
Création d'une fonction :	19
Appel d'une fonction :	19
Exemple :	19
Création d'une fonction avec des paramètres :	20
Exemple :	20
Exercices :	20
9 Les conditions :	21
Opérateurs de comparaison :	21
Opérateurs logiques :	22
Exemple :	23
Test Switch case :	24
Exercices :	25
10 Les boucles :	26
Exemple boucle for :	26
Exemple boucle while :	27
Exemple boucle foreach :	28
Exercices :	29
11 les tableaux :	30
Déclaration d'un tableau :	30
Exemple déclaration de tableaux indexé numériquement et associatif :	31
Exemple ajouter une valeur à un tableau :	32
Exemple parcourir un tableau :	33
Exercices :	33

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

12 Les super globales :	34
Fonctionnement GET :	35
Fonctionnement POST :	38
Récupération d'inputs checkbox (HTML -> formulaire) mode POST :	41
Exercices :	43
13 Interaction avec une base de données :	44
1 Se connecter à la base de données :	45
2 Exécution d'une requête SQL :	46
Exercices :	51
14 Importer des fichiers :	55
15 Modèle MVC :	59
Exercices :	64
16 Classe et objet :	65
Une classe des objets c'est quoi ?	65
Créer une classe en PHP :	65
Instancier un objet :	66
Ajouter des attributs :	66
Affecter une valeur à un attribut d'un objet :	67
Créer et appeler des méthodes :	68
Constructeur	69
Méthode toString :	70
Exercices :	71
17 Portée des objets :	73
Getter et setter :	74

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

1 Passer les attributs de la classe en private :	74
2 Ajouter les méthodes Getter et Setter :	75
3 modifier les méthodes existantes :	76
Exercices :	76
18 Super Globale SESSION et connexion :	77
1 Super Globale SESSION :	77
19 Le Routing :	83
1 Réécriture des URL :	84
2 Structure du projet en MVC :	85
3 Création du routeur :	86
20 Héritage (Objet) :	88
21 Etendu des classes Héritage (Objet) :	91
21 Opérateur de résolution de portées Héritage (Objet) :	94
22 Structure MVC Avancé :	95
22.1 Contenu du manager :	96
23 Template PHP (ob) :	97
23.1 découpage du template et des vues :	97

Emplacement table des matières suite.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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;
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Commenter des lignes de codes :

`<?php`

`//commentaire sur une ligne`

`/*`

`Commentaire sur plusieurs lignes`

`*/`

`?>`

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 :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

```
<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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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 $\$b$)

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**)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Mathieu Mithridate	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

7 Concaténation :

En php nous pouvons concaténer des valeurs entre elles. C'est à dire ajouter des chaînes 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 chaînes de caractères et les afficher dans une page web :

```
<?php
```

```
echo "<br>ma chaîne de caractères contient 32 caractères";
```

```
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Concaténer des variables dans des chaînes 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 ' ' ;
?>
```

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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

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.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Exercice 4 :

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

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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

Exemple boucle while :

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

```
<?php
```

```

    $i = 0; //variable compteur à l'extérieur de la boucle
    while ($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).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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
```

```
);
```

```
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

12 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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

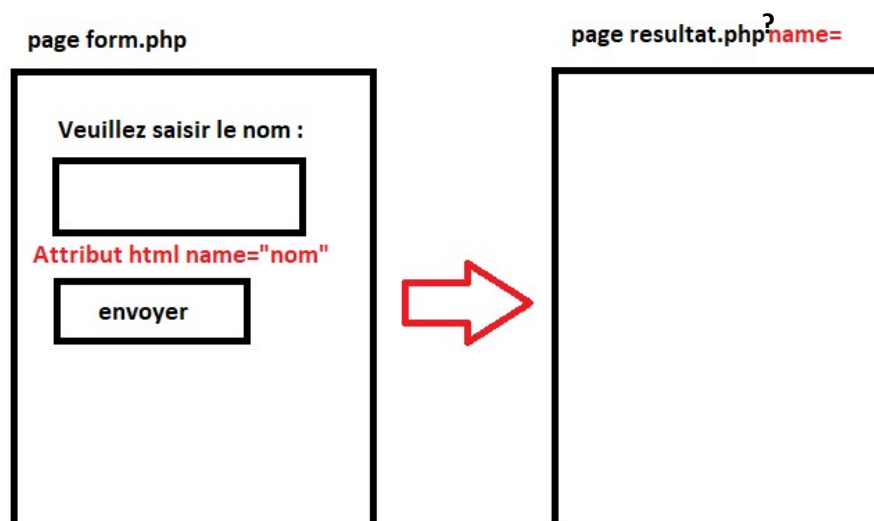
PHP

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



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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

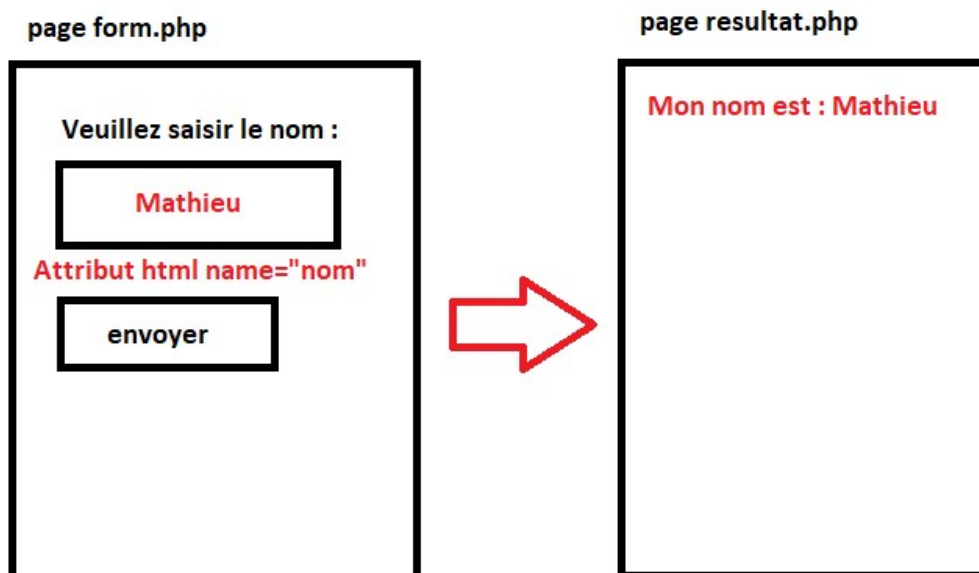
PHP

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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**.

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>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

```

<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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

13 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),

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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 :

//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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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) :

```
//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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

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.

```
//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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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 :

```
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(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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**),

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

d) Bonus :

-Utiliser une requête **SQL préparée** 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.

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** 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** :

```
$list = getAllArticle($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.*/
```

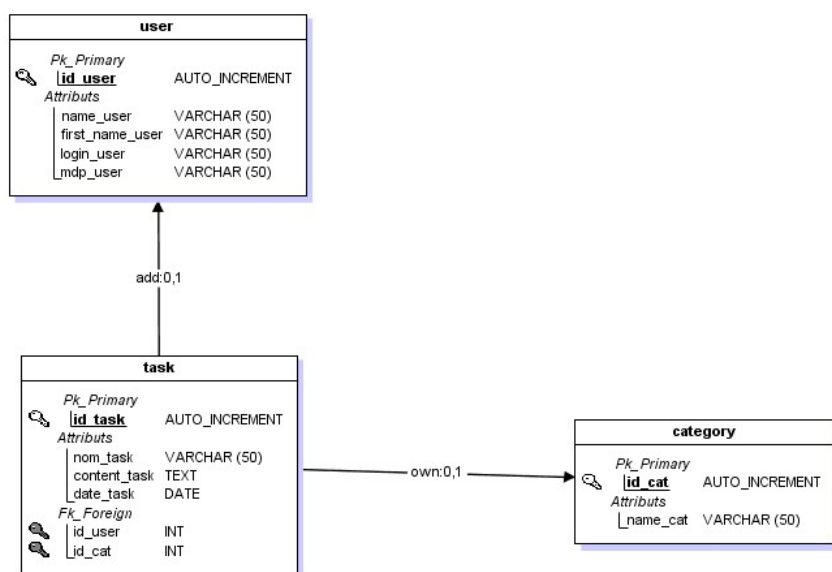
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou 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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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»**,

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Mathieu Mithridate	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

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`),

-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 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(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

PHP

14 Importer des fichiers :

Nous avons la possibilité en PHP d'importer des fichiers à l'intérieur du répertoire du projet (www\nom_du_projet\)

Pour ce faire nous allons utiliser la super globale \$_FILES.

Elle va s'utiliser comme les super globales précédentes \$_GET et \$_POST.

Quand on importe un fichier, celui-ci va se retrouver dans un dossier temporaire (à la racine du serveur apache, dans le dossier tmp) le serveur va lui donner un nom temporaire ex tmp_1569565322.jpg.

Pour importer un fichier nous allons créer un formulaire HTML comme ci-dessous (**index.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>Importer un fichier </title>
</head>
<body>
  <form action="index.php" method="POST" enctype="multipart/form-data">
    <h2>importer une image</h2>
    <input type="file" name="file">
    <p><button type="submit">importer</button></p>
  </form>
</html>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Le fichier va être importé en mode **POST** dans le dossier /tmp à la racine du serveur.

Nous allons voir ci-dessous comment le traiter et le déplacer dans le bon répertoire (*par ex le dossier image à la racine de notre projet*) :

- Nous allons créer un nouveau répertoire **import** à la racine du serveur web (*www/import ou htdocs\import*),
- Créer un dossier image à la racine du projet import,
- Créer un fichier index.php et coller à l'intérieur le code html de la page précédente,

Nous allons ajouter le code ci-dessous dans le fichier index.php pour récupérer le fichier et le déplacer dans le bon dossier (*/image à la racine du projet import*).

Code **importation** d'un fichier avec son *nom.ext* dans le dossier image à la racine du projet.

Pour ce faire nous allons :

- Vérifier si le fichier que l'on importe existe (*utilisation de la super globale \$_FILES*),
- Créer différentes variables,
- Déplacer le fichier dans le bon dossier avec la méthode (*move_uploaded_file*).

Auteur(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

PHP

```
<?php

/*-----

Test (import du fichier) :

-----*/

//test si le fichier importé existe
if(isset($_FILES['file'])){

    //stocke le chemin et le nom temporaire du fichier importé (ex /tmp/125423.pdf)
    $tmpName = $_FILES['file']['tmp_name'];

    //stocke le nom du fichier (nom du fichier et son extension importé ex : test.jpg)
    $name = $_FILES['file']['name'];

    //stocke la taille du fichier en octets
    $size = $_FILES['file']['size'];

    //stocke les erreurs (pb d'import, pb de droits etc...)
    $error = $_FILES['file']['error'];

    //déplacer le fichier importé dans le dossier image à la racine du projet
    $fichier = move_uploaded_file($tmpName, "./image/$name");

}
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

```
/*-----
```

Formulaire HTML :

```
-----*/
```

```
?>
```

```
<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>Document</title>
```

```
</head>
```

```
<body>
```

```
<form action="index.php" method="POST" enctype="multipart/form-data">
```

```
<h2>importer une image</h2>
```

```
<input type="file" name="file">
```

```
<p><button type="submit">importer</button></p>
```

```
</form>
```

```
</html>
```

Exercice TP 1 :

Réalisez le TP file import (importer des images->BDD).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

15 Modèle MVC :

Dans les chapitres précédents nous avons au sein d'une même page inclus la vue html ainsi que le code PHP.

Afin de mieux organiser notre code, pour permettre une plus grande facilité de mise à jour, nous allons lui appliquer le modèle MVC.

Dans ce modèle ou pattern chacun de nos fichiers aura un rôle bien défini :

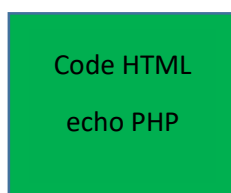
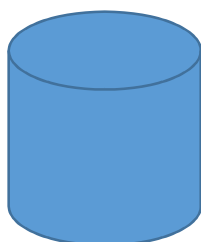
Modèle : dans cette partie nous allons déplacer toute les parties du code qui vont nous permettre l'accès aux données, afin de les préparer pour le contrôleur. C'est tout ce qui va concerner les requêtes **SQL**.

Vue : la vue se concentre sur toute la partie affichage, c'est les interfaces que l'utilisateur final va voir et avec lesquelles il va interagir dans son navigateur internet.

Dans cette partie on va retrouver toute la structure **HTML**.

Controller : C'est le controller qui va gérer toute la logique de notre page, ainsi que les calculs et traitement des données. Le controller va demander les données au modèle et adapter la vue en fonction de celle-ci. Le controller va avoir un rôle d'aiguillage.

Dans cette partie on va retrouver exclusivement du code **PHP**.

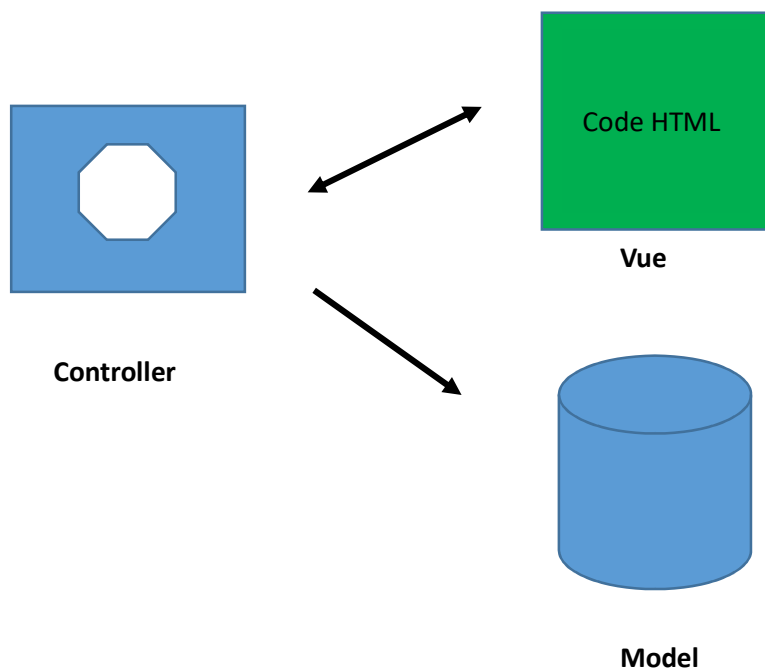


Modèle (accès à la base de données) Vue (contiens le code HTML) Controller (logique et traitement)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Echange entre les différentes couches :



Pour intégrer notre code nous allons avoir besoin d'utiliser une méthode PHP qui se nomme ***include()***.

Exemple :

Reprenons l'exercice 1 du chapitre précédent, nous allons restructurer et découper le code de cette façon :

Toute la partie html (notre formulaire) nous allons la déplacer dans un nouveau fichier que nous allons appeler **vue_article.php** comme ci-dessous :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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>ajouter un article</title>
</head>
<body>
    <form action="" method="post">
        <p>saisir le nom de l'article :</p>
        <input type="text" name="nom_article">
        <p>saisir le contenu de l'article :</p>
        <input type="text" name="contenu_article">
        <input type="submit" value="Ajouter">
    </form>
</body>
</html>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Toute la partie PHP (requête SQL) va être déplacé dans un nouveau fichier que nous allons nommer **model_article.php**. Comme ci-dessous :

```
try
{
    //Exécution de la requête SQL insert

    $reponse = $bdd->query('insert into article(nom_article,
contenu_article) values("'.$name.'", "'.$content.'")');
    echo "ajout de l'article : $name qui a comme contenu : $content";
}
catch(Exception $e)
{
    //affichage d'une exception en cas d'erreur
    die('Erreur : '.$e->getMessage());
}
```

Afin de réutiliser la connexion à la base de données dans l'ensemble de notre code nous allons déplacer la connexion dans un nouveau fichier que nous allons nommer **connect.php** comme ci-dessous :

```
<?php
//connexion à la bdd

$dbdd = new PDO('mysql:host=localhost;dbname=articles', 'root','',
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));

?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Enfin nous allons déplacer la logique (*conditions*) dans une nouvelle page (*qui sera notre **controller***) que nous allons nommer **contrôler_article.php**. Comme ci-dessous :

```
<?php
//ajout de la vue
include('vue_article.php');
//connexion à la BDD
include('connect.php');
//test existence des champs nom_article et contenu_article
if(isset($_POST['nom_article']) and isset($_POST['contenu_article']))
{
    //création des 2 variables qui vont récupérer le contenu des super globales POST
    $name = $_POST['nom_article'];
    $content = $_POST['contenu_article'];
    //ajout du model
    include('model_article.php');
}
else{
    //affichage dans la page html de ce que l'on a enregistré en bdd
    echo '<p>veuillez remplir les champs de formulaire</p>';
}
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Exercices :

Exercice 1 :

Reprendre l'exercice 1 de la partie précédente et l'adapter en MVC (se servir de l'exemple du cours) et remplacer la partie **model** par la requête **préparée**.

Intégrer la partie bonus (affichage de l'article ajouté dans un paragraphe).

Exercice 2 :

Reprendre l'exercice 2 de la partie précédente et l'adapter en MVC.

Exercice 3 :

Reprendre l'exercice 3 de la partie précédente et l'adapter en MVC.

Auteur(s)	Relu, validé et visé par :		Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT			
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM			

PHP

16 Classe et objet :

Dans ce chapitre nous voir php sous un nouvel angle, en utilisant la programmation orientée objet.

Une classe des objets c'est quoi ?

La programmation orienté objet nous permet de modéliser dans notre code des éléments de la vie réelle :

Un véhicule, un animal, un bâtiment etc...

Une **classe** fonctionne comme une recette de cuisine qui va nous permettre de créer des **objets** :

Gardons l'image d'une recette de cuisine c'est le plan (**classe**) qui va nous permettre de réaliser le plats, la recette a besoin d'ingrédients (que nous appellerons **attributs ou propriétés**), d'étapes (que nous nommerons **méthodes ou fonctions**). A la fin de la recette nous allons obtenir un plat (que nous appellerons **objet**). Chaque fois que l'on réalisera la recette nous obtiendrons un nouveau plat qui sera unique et donc un **nouvel objet**.

En programmation orienté objet cela sera la même chose. Nous créerons une **classe** qui sera notre recette, elle contiendra des **attributs** (ou **propriétés**) que l'on peut voir comme nos ingrédients et nous aurons des **méthodes ou fonctions** pour effectuer les différentes étapes de réalisation du plat qui sera notre **objet**.

Créer une classe en PHP :

Pour créer une classe en PHP nous allons créer un nouveau fichier avec la syntaxe suivante :

Ex classe véhicule :

Cette classe va nous permettre de créer des véhicules, elle contiendra des attributs et des méthodes ou fonctions. Le nom d'une classe commence toujours par une majuscule :

```
<?php
class Vehicule{
}
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Instancier un objet :

Pour créer un nouveau véhicule depuis la classe Vehicule nous utiliserons la syntaxe suivante :

```
< ?php
```

```
    //import du fichier class.php qui contient la classe Vehicule
```

```
    require './class.php'
```

```
    //création d'un nouveau véhicule depuis la classe Vehicule
```

```
    $voiture = new Vehicule();
```

```
?>
```

Ce code va nous permettre de créer un nouvel **objet** voiture depuis la **classe Vehicule**.

Ajouter des attributs :

Afin de personnaliser notre classe nous allons créer des attributs (variables), cela va nous permettre d'ajouter des propriétés dans nos objets.

```
< ?php
```

```
class Vehicule
```

```
{
```

```
    //Attributs :
```

```
    public $nomVehicule ;
```

```
    public $nbrRoue;
```

```
    public $vitesse ;
```

```
}
```

```
?>
```

Dans l'exemple ci-dessus nous avons ajouté des **attributs** pour définir un nom à notre véhicule , nombre de roues et la vitesse de notre véhicule.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Affecter une valeur à un attribut d'un objet :

Pour affecter une valeur à un attribut d'un objet (la valeur ne sera pas affectée à la classe mais à l'instance de notre objet) on utilise la syntaxe suivante :

```
<?php
```

```
//appel du fichier class.php qui contient la classe Vehicule
```

```
//require est équivalent à include
```

```
require './class.php';
```

```
//création d'un nouveau véhicule depuis la classe Vehicule
```

```
$voiture = new Vehicule();
```

```
//ajout de valeur aux attributs de la classe Vehicule
```

```
$voiture->nomVehicule = "Audi A3";
```

```
$voiture->nbrRoue = 4;
```

```
$voiture->vitesse = 250;
```

```
?>
```

Le code ci-dessus affecte des valeurs aux **attributs** de l'**objet** voiture (nomVehicule = Audi, nbrRoue = 4 et vitesse = 250)

NB : Cette syntaxe est valide uniquement quand les attributs sont en **public**. Nous Verrons plus tard qu'il est conseillé de passer les attributs en **private** ou **protected**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Créer et appeler des méthodes :

Dans notre classe nous avons la possibilité de créer des fonctions qui seront utilisable par nos objets.

Nous allons créer dans notre classe Vehicule plusieurs méthodes que nos objets pourront utiliser.

Exemple création d'une méthode démarrer :

Dans le fichier **classe Vehicule** (vehicule.php) nous allons créer une fonction qui va démarrer le véhicule elle va afficher dans une page html un paragraphe avec comme contenu :

« Démarrage de la « nom du véhicule » Vrooom !!!!»

Pour ce faire nous allons utiliser la syntaxe ci-dessous :

```
<?php
class Vehicule{
    /*-----
                        Attributs :
    -----*/
    public $nomVehicule ;
    public $nbrRoue;
    public $vitesse ;
    /*-----
                        Fonctions :
    -----*/
    //fonction démarrer le véhicule
    public function demarrer(){
        echo "<p>Démarrage de La $this->nomVehicule Vrooom !!!!</p>";
    }
}
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

La variable **\$this** correspond à l'**instance courante** de notre **objet**.

Pour utiliser cette méthode sur un objet nous utiliserons la syntaxe suivante :

```
<?php
```

```
//appel du fichier class.php qui contient la classe Vehicule
```

```
//require est équivalent à include
```

```
require './class.php';
```

```
//création d'un nouveau véhicule depuis la classe Vehicule
```

```
$voiture = new Vehicule();
```

```
//ajout de valeur aux attributs de la classe Vehicule
```

```
$voiture->nomVehicule = "Audi A3";
```

```
$voiture->nbrRoue = 4;
```

```
//utilisation de la méthode démarrer
```

```
$voiture->démarrer();
```

```
?>
```

Pour ce faire nous utilisons l'opérateur -> puis le nom de la fonction suivi de parenthèses.

NB : Pour afficher le détail d'un objet nous utilisons la fonction php **var_dump(\$objet)**.

Constructeur

Pour pouvoir affecter des valeurs à un objet directement au moment de son instanciation nous devons ajouter une méthode à la classe. Celle-ci se nomme constructeur (c'est la méthode qui est appelée au moment du new Classe()).

Par défaut un constructeur vide est disponible dans la classe (même si la méthode n'est pas écrite dans la classe).

Pour créer un constructeur personnalisé (qui va imposer de donner des valeurs à différents attributs) nous utiliserons la syntaxe suivante :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

```
<?php
class Classe{
    //attributs
    public $nom;
    public $taille;
    //constructeur
    public function __construct(string $nom, ?int $taille){
        $this->nom = $nom;
        $this->taille = $taille;
    }
}
?>
```

Quand nous instancierons des objets depuis cette classe nous devrons obligatoirement renseigner les valeurs en paramètre :

//la classe oblige à renseigner une valeur pour l'attribut nom et taille

```
$classe = new Classe('nom', 20);
```

Méthode toString :

Si l'on **echo** un **objet** on se retrouve avec une erreur de type :

Object of class NomClasse could not be converted to string. Pour la corriger nous devons recréer la **magic method** **__toString** avec la syntaxe suivante :

```
public function __toString(){
    return $this->attribut; //ici on remplace la valeur par l'attribut de la classe
    que l'on souhaite afficher (par ex le //nom)
}
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Exercices :

Exercice 1 :

Créer un fichier **test_objet.php** qui va nous servir de fichier d'exécution,

Créer une nouvelle classe Maison **maison.php** qui va contenir les attributs suivants :

-nom, longueur, largeur.

Instancier une nouvelle maison dans le fichier **test_objet.php** avec les valeurs de votre choix (**nom**, **longueur** et **largeur**),

-Créer une méthode **surface** qui calcule et affiche la superficie de la maison (**longueur * largeur**) dans la classe Maison.

-Appeler la méthode **surface** et **afficher** sous la forme suivante le résultat :

"<p>La surface de **nomMaison** est égale à : **x m2**</p>".

Bonus

Ajouter un attribut **nbrEtage** à la classe Maison,

Modifier la méthode **surface** pour qu'elle prenne en compte le paramètre **nbrEtage**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Exercice 2 :

- Créer un fichier **vehicule.php** qui va contenir la classe,
- Dans ce fichier recréer la classe Vehicule comme dans le cours (**attributs** et **méthodes**),
- Créer un fichier **test_objet.php** au même niveau que vehicule.php,
- Appeler avec **require()** ou **include()** le fichier de la classe **Vehicule**,
- Instancier 2 nouveaux véhicules dans le fichier **test_objet.php** avec les paramètres suivants :
- Objet **voiture** (nomVehicule = « Mercedes CLK », nbrRoue = 4, vitesse 250),
- Objet **moto** (nomVehicule = « Honda CBR », nbrRoue = 2, vitesse = 280),
- Créer une fonction **detect()** qui détecte si le véhicule est une moto ou une voiture (la méthode retourne une **string** **moto** ou **voiture** avec **return**) dans le fichier de classe **vehicule.php**,
- Exécuter la méthode **detect()** sur les 2 objets voiture et moto dans le fichier **test_objet.php**.
- Afficher le type de véhicule dans le fichier **test_objet.php**,
- Créer une méthode **boost** qui ajoute 50 à la vitesse d'un objet dans le fichier de classe **vehicule.php**,
- Appliquer la méthode **boost** a la voiture dans le fichier **test_objet.php**,
- Afficher la nouvelle vitesse de la voiture dans le fichier **test_objet.php**.

Bonus :

- Créer une méthode **plusRapide()** dans le fichier **vehicule.php** qui compare la vitesse des différents véhicules (**moto** et **voiture**) et retourne le véhicule le plus rapide des 2 avec un **return**.
- Exécuter la méthode **plusRapide()** dans le fichier **test_objet.php**.
- Afficher le véhicule le plus rapide dans le fichier **test_objet.php**.

Exercice 3 Projet Task :

Créer toutes les **classes** du projet task (**user**, **task**, **category**) et ajouter les dans le dossier **model** du projet.

Créer un fichier pour chacune de nos classes (**user.php**, **task.php**, **category.php**).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Mathieu Mithridate	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

17 Portée des objets :

En programmation orienté objet chacun de nos attributs, méthode vont avoir une portée d'utilisation en fonction du paramètre devant celle-ci on parle d'encapsulation :

Public : l'attribut ou propriété, méthode sera accessible depuis n'importe où dans notre projet c'est la valeur par défaut. Ce paramètre sera utilisé au niveau des **méthodes** afin que celle-ci soit accessible depuis n'importe quel endroit de notre projet. Il est **fortement déconseillé** de l'utiliser pour les attributs de notre classe. Pour y accéder nous créerons des méthodes **getter** et **setter** pour lire et écrire les **valeurs** des **attributs**.

Private : l'attribut ou propriété, méthode sera accessible uniquement au sein de la classe c'est la valeur que nous allons attribuer par défaut à nos attributs.

Protected : l'attribut ou propriété, méthode sera accessible depuis n'importe où dans le même dossier de notre projet. Ce paramètre pourra être utilisé au niveau des **méthodes** et des **attributs** afin que ceux-ci soit accessible depuis n'importe quel fichier contenu dans le **même répertoire**. Ce paramètre assure une sécurité à l'extérieur de notre dossier, les **méthodes** et **attributs** seront inaccessibles à l'**extérieur du dossier**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

Getter et setter :

Afin de **sécuriser** les **attributs** de nos **classes** nous allons tous les passer en mode **private**.

Pour **lire** et **écrire** du contenu au sein d'un **attribut** (valeur) nous allons créer autant de **méthodes** que d'**attributs** (**propriété**).

La **méthode** **getter** va nous permettre d'accéder aux **valeurs** d'un **attribut** d'une **classe** en **private**. Cette méthode sera toujours en public.

La **méthode** **setter** va nous permettre de modifier les **valeurs** d'un **attribut** d'une **classe** en **private**. Cette méthode sera toujours en public.

1 Passer les attributs de la classe en private :

```
<?php
class Vehicule
{
    //Attributs :
    private $nomVehicule ;
    private $nbrRoue;
    private $vitesse ;
}
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

2 Ajouter les méthodes Getter et Setter :

Pour se faire nous allons utiliser la syntaxe suivante :

```
< ?php
```

```
    //Getter nomVehicule récupère Le nom du véhicule
```

```
    public function getNomVehicule()
```

```
    {
```

```
        return $this->nomVehicule;
```

```
    }
```

```
    //setter nomVehicule remplace Le nom du véhicule
```

```
    public function setNomVehicule($new_nom_vehicule)
```

```
    {
```

```
        $this->nomVehicule = $new_nom_vehicule;
```

```
    }
```

```
?>
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

PHP

3 modifier les méthodes existantes :

La fonction demarrer le véhicule va s'écrire sous la forme suivante :

```
<?php
//fonction demarrer Le véhicule
public function demarrer()
{
    $demarrage = '<p>Démarriage de La '.$this->getNomVehicule(). 'Vrooom !!!!</p>';
    return $demarrage;
}
?>
```

Exercices :

Exercice 1 :

Créer un nouveau fichier **vehicule_private.php**,

Repartir de la base de la classe **Vehicule** et passer les tous les **attributs** en **private**,

Ajouter les **getters** et **setter**,

Editer les **méthodes** afin qu'elles s'adaptent avec les nouveaux paramètres.

Exercice 2 Projet Task :

Modifier toutes les **classes** du projet task (**user**, **task**, **category**) et ajouter les dans le dossier **model** du projet. En incluant les **Getter** et les **Setter** pour chacun des attributs.

Créer toutes les méthodes ajouts (requête **insert** en mode **préparé**) que vous allez nommer :

createUser, **createCategory**, **createTask**.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Mathieu Mithridate	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	08/12/2020	
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		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.
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		