

TRAVAUX PRATIQUES – Filière Développement digital

M107 - DÉVELOPPER DES SITES WEB DYNAMIQUES

43 heures

SOMMAIRE

01 - INTRODUIRE LA NOTION CLIENT/SERVEUR

Approfondir la notion client/serveur
Introduire le langage PHP
Préparer l'environnement de développement

02 - PROGRAMMER EN PHP

Maitriser le langage PHP
Traiter les données en PHP
Utiliser l'orientée objet en PHP

03 - MANIPULER LES DONNÉES

Écrire des scripts d'accès aux données
Sécuriser les données

04 - RÉALISER UN SITE WEB AVEC L'ARCHITECTURE MVC

Développer des sites dynamiques
avec MVC en mode natif
Découvrir les Web services et les API REST



MODALITÉS PÉDAGOGIQUES



1

LE GUIDE DE SOUTIEN

Il contient le résumé théorique et le manuel des travaux pratiques



2

LA VERSION PDF

Une version PDF est mise en ligne sur l'espace apprenant et formateur de la plateforme WebForce Life



3

DES CONTENUS TÉLÉCHARGEABLES

Les fiches de résumés ou des exercices sont téléchargeables sur WebForce Life



4

DU CONTENU INTERACTIF

Vous disposez de contenus interactifs sous forme d'exercices et de cours à utiliser sur WebForce Life

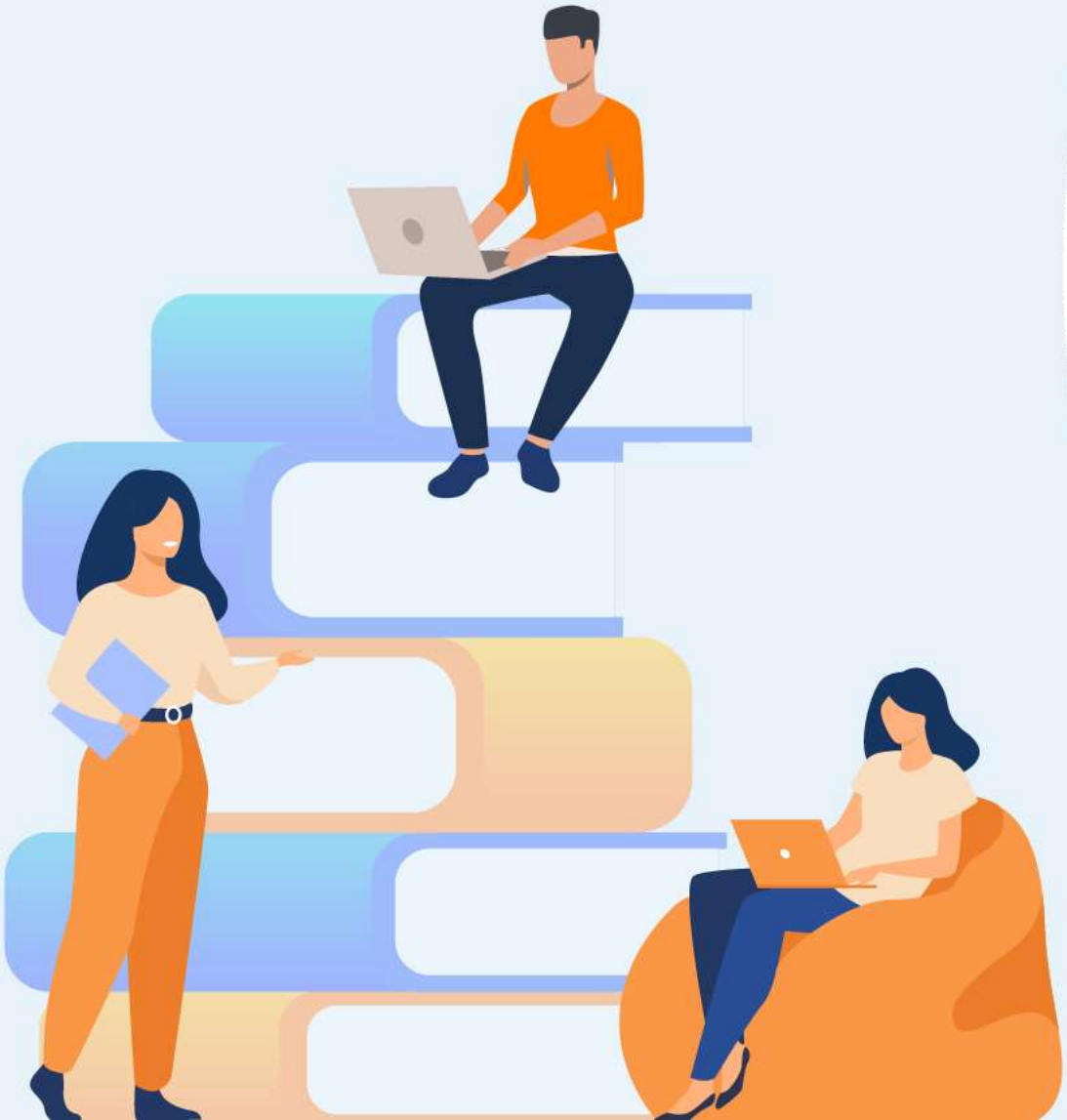


5

DES RESSOURCES EN LIGNES

Les ressources sont consultables en synchrone et en asynchrone pour s'adapter au rythme de l'apprentissage

Copyright - Tout droit réservé - OFPPT



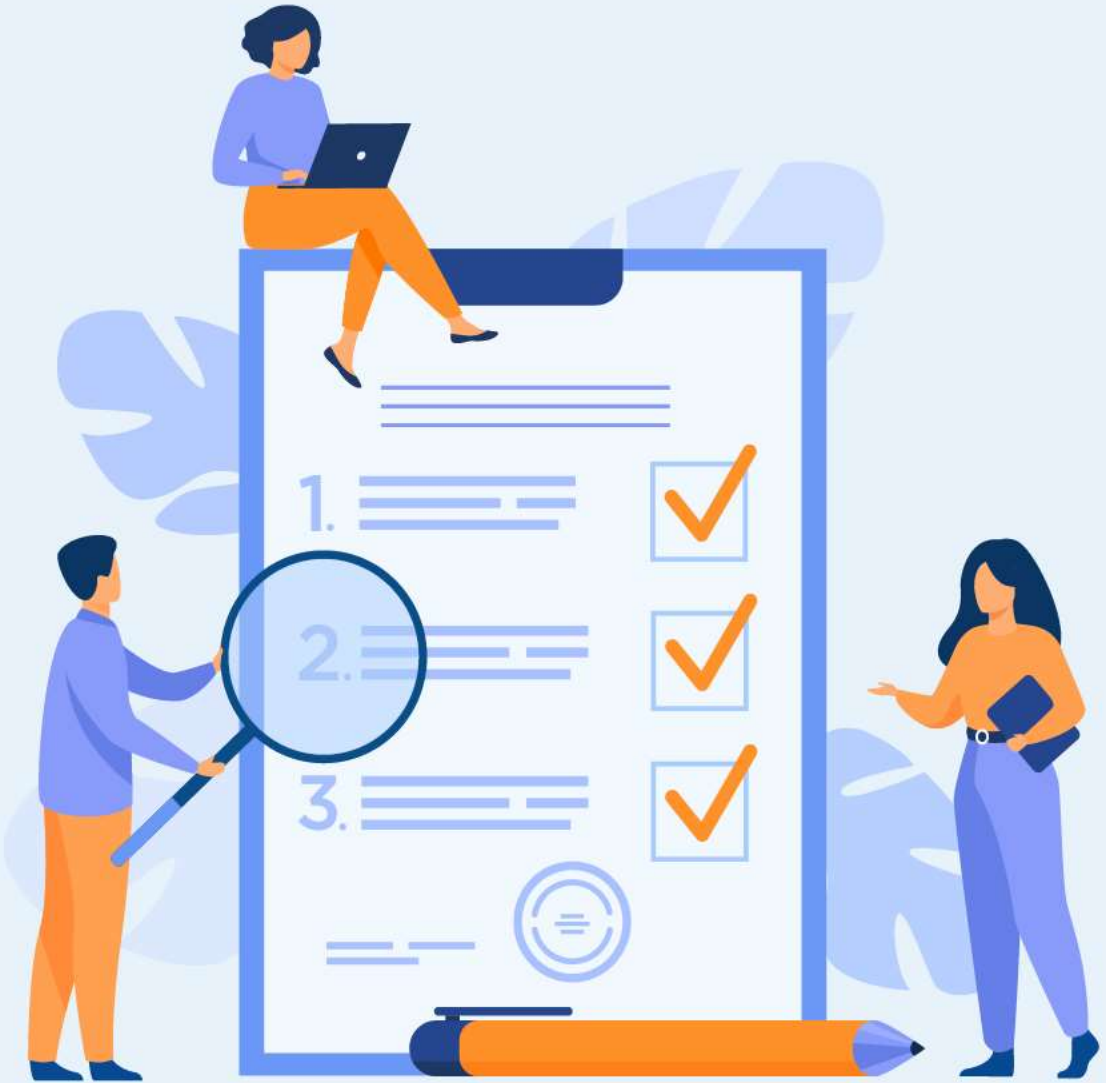
PARTIE 1

INTRODUIRE LA NOTION CLIENT/SERVEUR

Dans ce module, vous allez :

- Installer les outils de création d’une page web dynamique
- Configurer les éléments de base d'une page web dynamique
- Activer les extensions nécessaires

03 heures



Activité n°1

INTRODUIRE LA NOTION CLIENT/SERVEUR

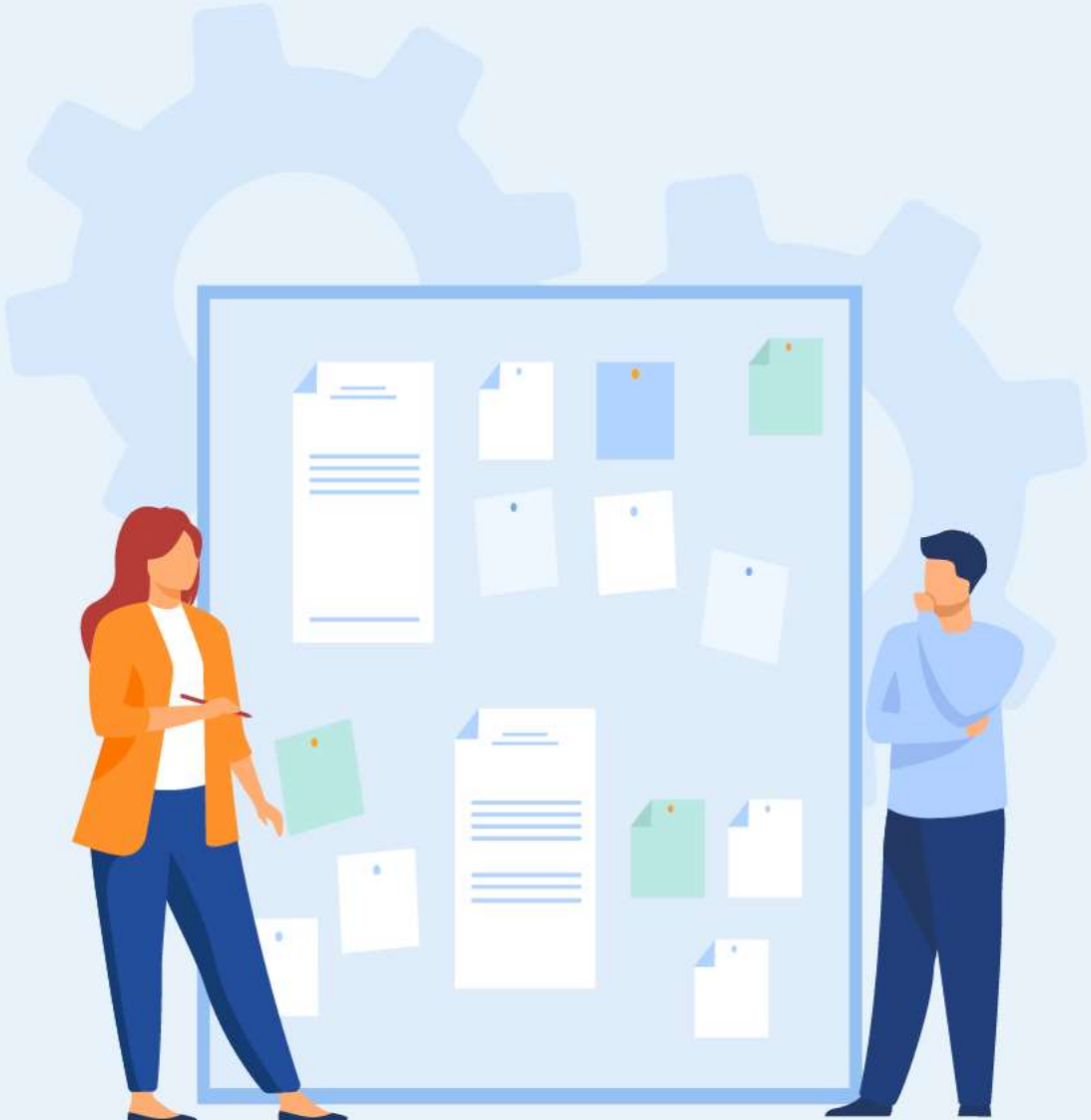
Compétences visées :

- Installer le serveur Apache
- Installer l'éditeur du code VS Code
- Installer et activer les extensions nécessaires

Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse

03 heures



CONSIGNES

1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

2. Pour l'apprenant :

- Installer Apache, PHP et VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Installer et configurer les éléments pour créer une page web dynamique
  - Manipuler les éléments PHP de base

PARTIE 1

Téléchargement et installation de Apache

- Étape 1 : Télécharger Apache depuis le site officiel https://www.apachelounge.com/download/

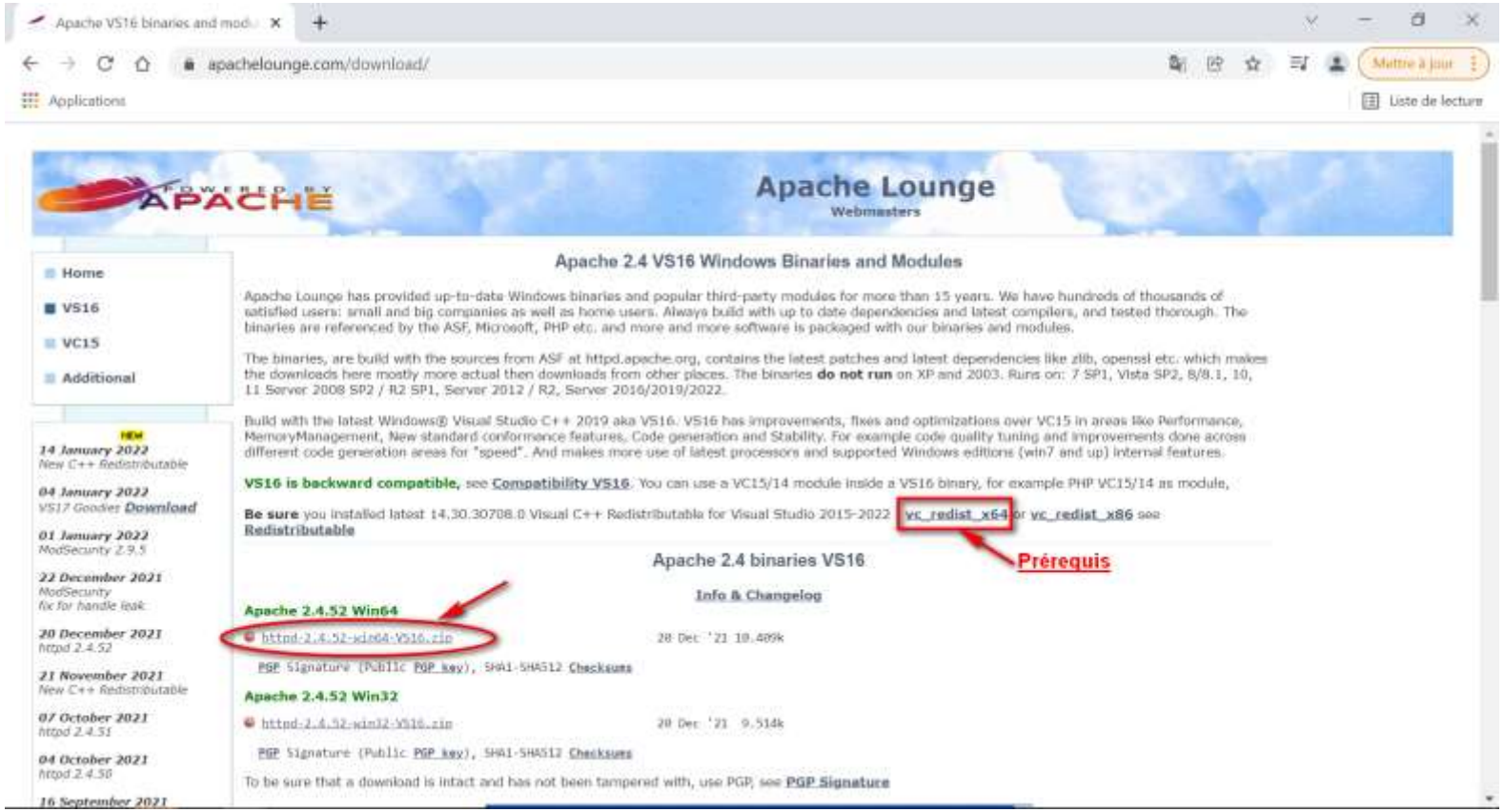


Fig. : Téléchargement du logiciel Apache

Copyright - Tout droit réservé - OFPPT

Téléchargement et installation de Apache

- Étape 2 : Double cliquer sur le fichier téléchargé ensuite dézippez.
- Étape 3 : Copier le dossier « Apache24 » et coller dans « C:\ ». Ce PC › System (C:)

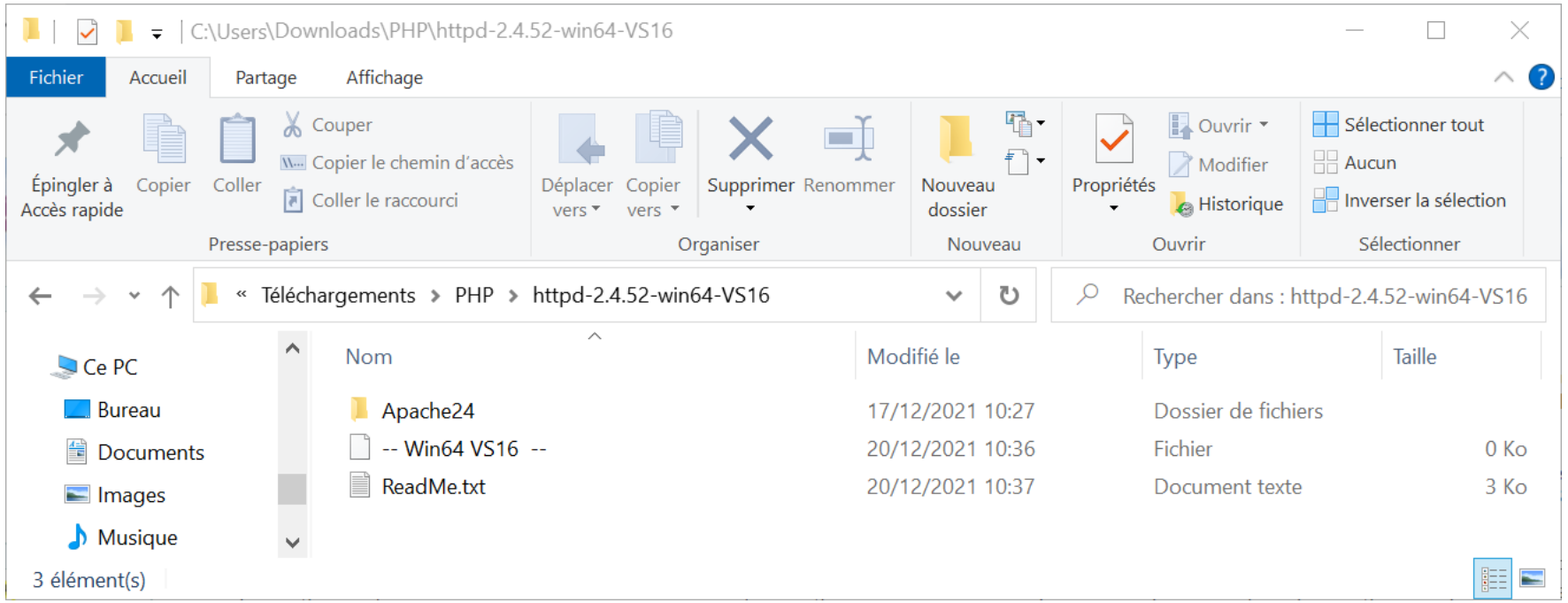


Fig. : Structure du fichier télécharger Apache

Copyright - Tout droit réservé - OFPPT

Activité 1
Apache

Configurer Apache

- Étape 1 : Accéder au « Panneau de configuration > Système et sécurité > Système > Paramètres avancés du système » pour modifier les variables d'environnement système. Cliquer sur le bouton « Variables d'environnement », dans les variables système sélectionner « Path » et ajouter « C:\Apache24\bin »
- Étape 2 : Accéder au fichier « httpd.conf » depuis « C:\Apache24\conf » et modifier les commandes « #Listen 12.34.56.78:80 » et « Listen 80 » par « #Listen 127.0.0.1:122 » et « Listen 122 » par exemple
- Étape 3 : Ouvrir l'invite de commande avec les privilèges administrateur et écrire la commande suivante « httpd –k install »
- Étape 4 : Pour lancer Apache il y a deux façons :
    - Ouvrir « Gestionnaire des tâches > Services » click droit et démarrer
    - Ouvrir l'invite de commande avec les privilèges administrateur et écrire la commande suivante « httpd –k start »
- Étape 5 : Lancer Apache dans un navigateur, mettre dans l'URL « 127.0.0.1:122 » ou « http://localhost:122/ »
- Étape 6 : Vous pouvez modifier le texte dans « index.html » présent dans « C:\Apache24\htdocs »



PARTIE 1

Copyright - Tout droit réservé - OFPPT

Téléchargement et installation de PHP

- Étape 1 : Télécharger PHP depuis le site officiel https://windows.php.net/download/

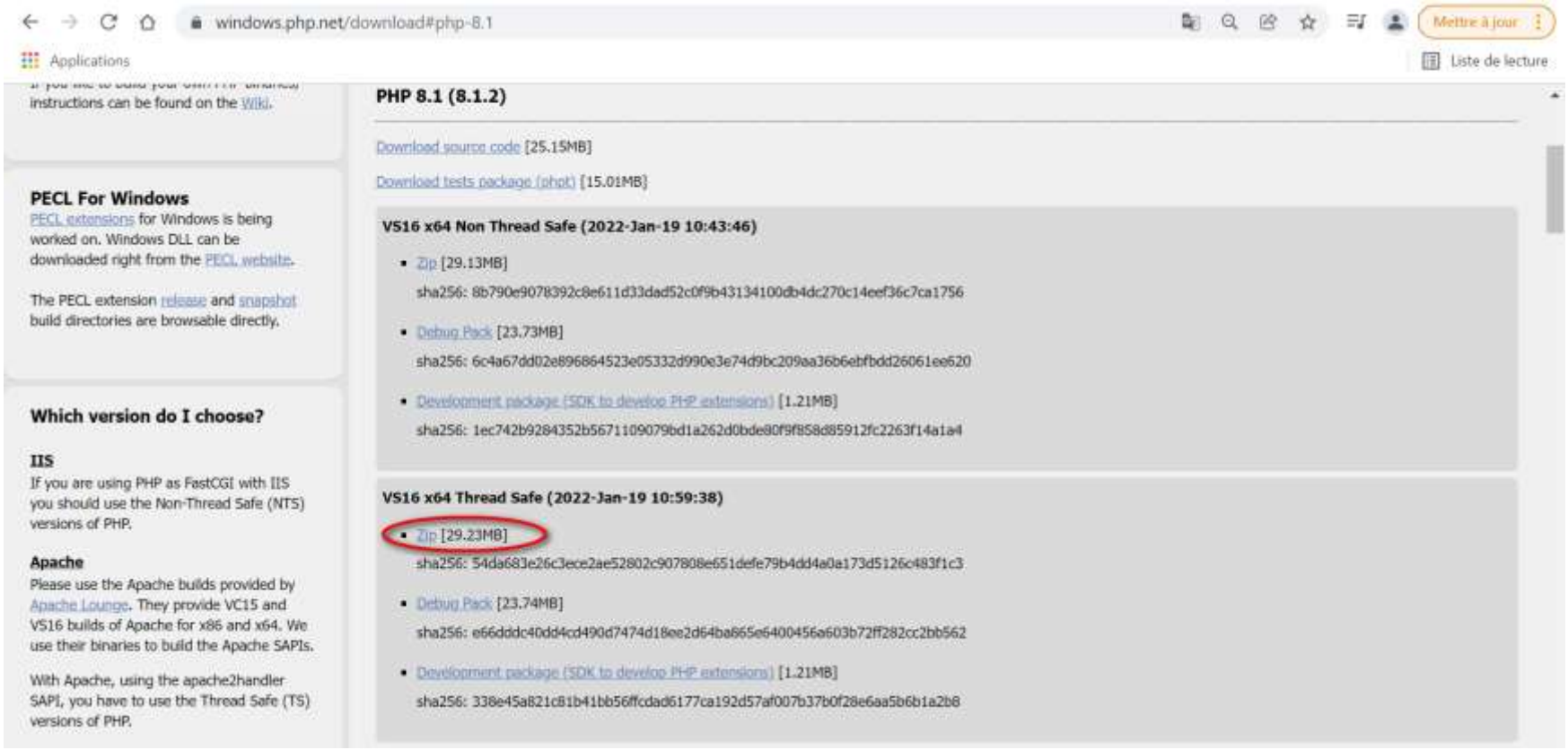


Fig. : Téléchargement de PHP

PARTIE 1

Copyright - Tout droit réservé - OFPPT

Téléchargement et installation de PHP

- Étape 2 : Double cliquer sur le fichier téléchargé ensuite dézippez.
- Étape 3 : Renommer le répertoire en « php ». Copier le dossier et coller dans « C:\ ». > Ce PC > System (C:)

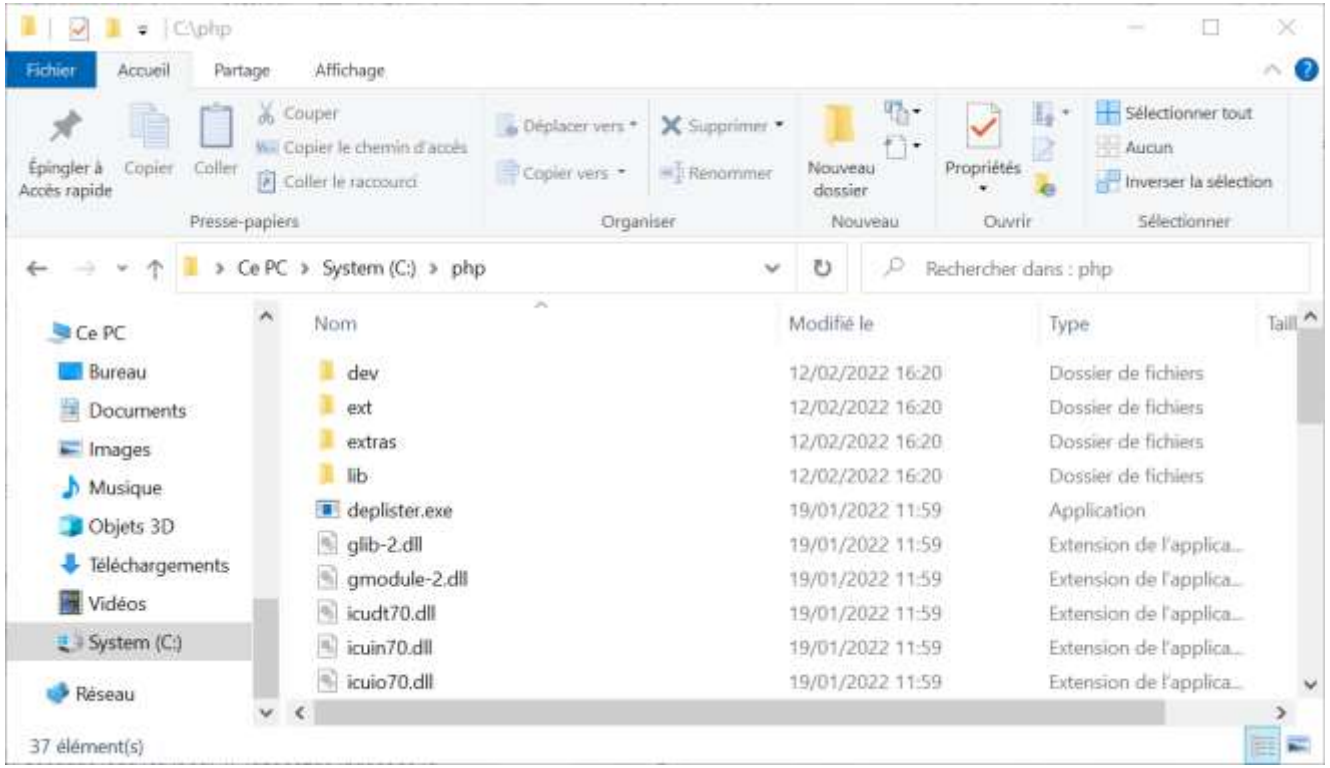


Fig. : Structure du fichier télécharger PHP

Copyright - Tout droit réservé - OFPPT

Configurer PHP

- Étape 1 : Accéder au « Panneau de configuration > Système et sécurité > Système > Paramètres avancés du système » pour modifier les variables d'environnement système. Cliquer sur le bouton « Variables d'environnement », dans les variables système sélectionner « Path » et ajouter « C:\php »
- Étape 2 : Copier le fichier « php.ini-development » et coller le dans le même emplacement en lui attribuant le nom « php.ini »
- Étape 3 : Ouvrir l'invite de commande avec les privilèges administrateur et écrire la commande suivante « php –v »

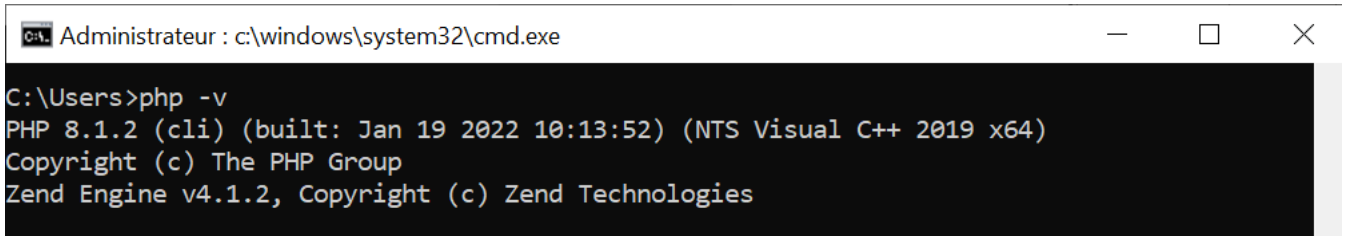


Fig. : Résultat commande php -v

Copyright - Tout droit réservé - OFPPT

Téléchargement et installation de PDO

- Étape 1 : Dans le fichier « php.ini » supprimer le symbole de commentaire de la ligne « extension=pdo_mysql » pour ajouter l'extension.

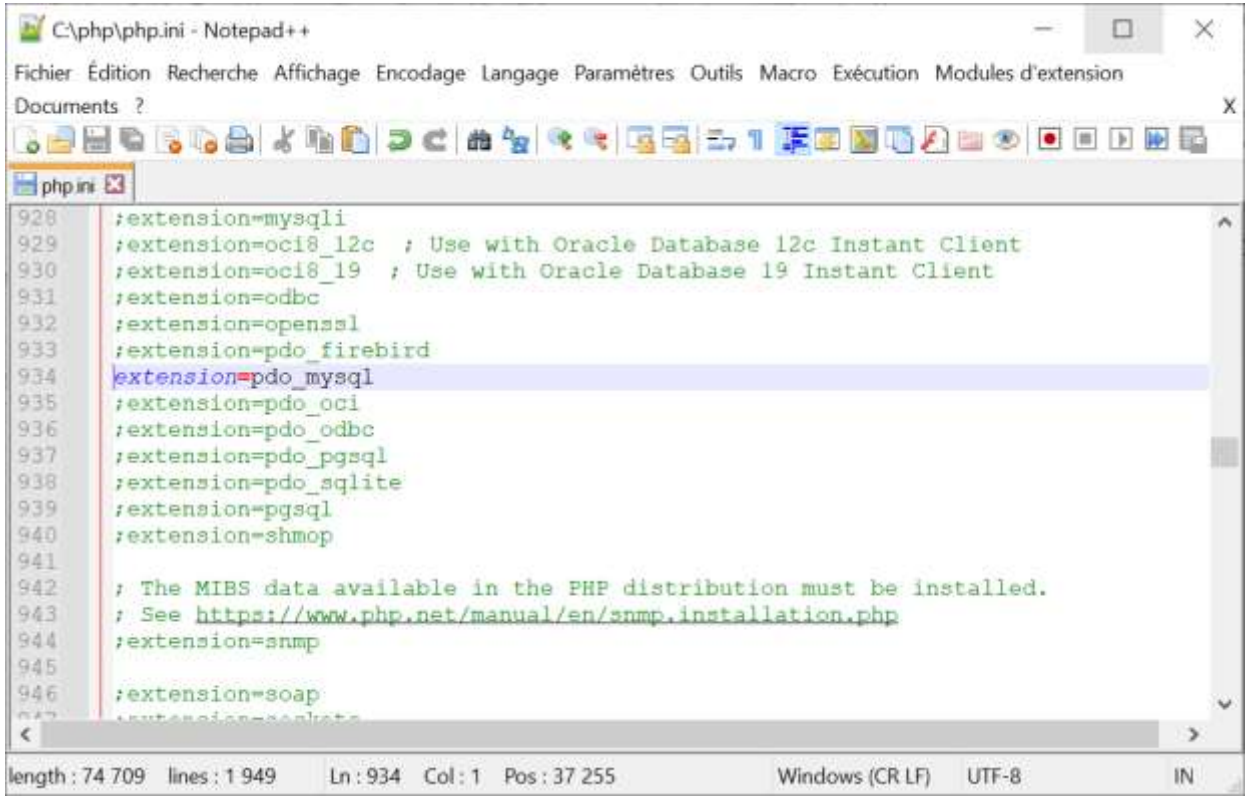


Fig. : L'extension PDO de PHP

Copyright - Tout droit réservé - OFPPT

Configurer Apache avec PHP

- Étape 1 : Accéder au fichier « httpd.conf » depuis **« C:\Apache24\conf »** et ajouter les lignes suivantes à la fin du fichier :
  - LoadModule php_module "C:\php\php8apache2_4.dll"
  - AddHandler application/x-httpd-php .php
  - PHPIniDir "C:\php"

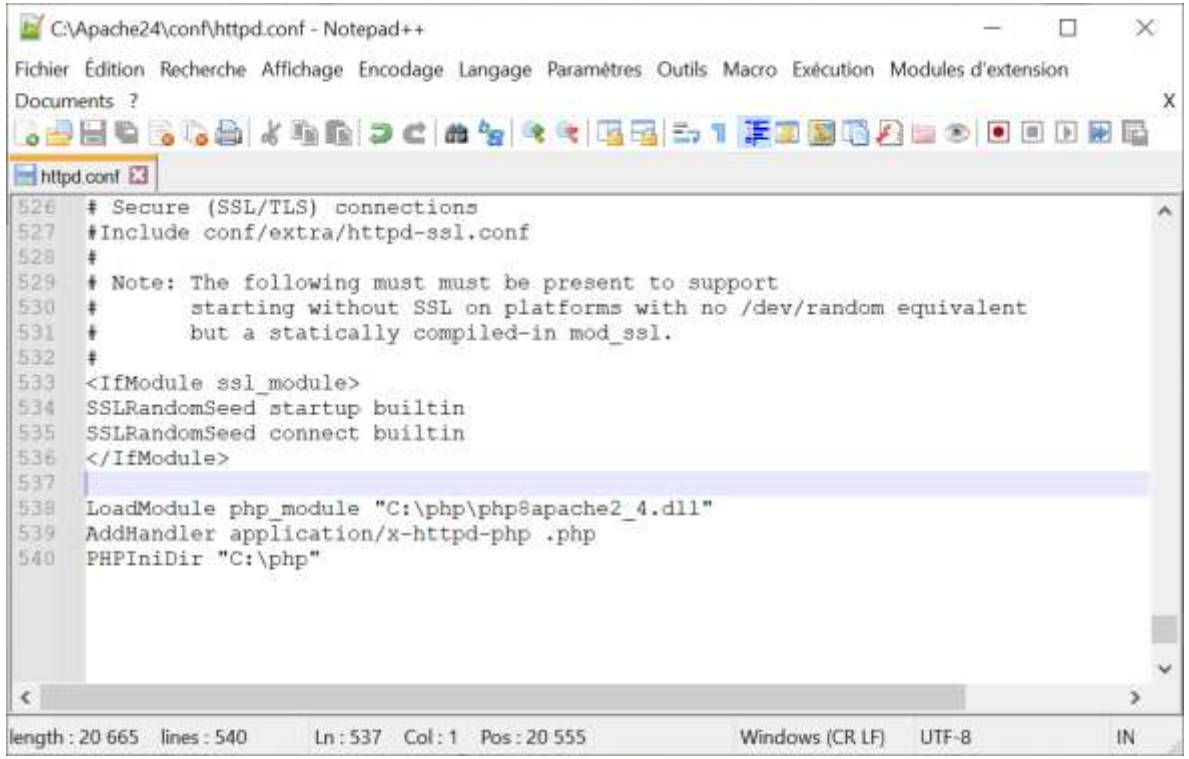


Fig. : Fichier de configuration Apache

PARTIE 1

Copyright - Tout droit réservé - OFPPT

Configurer Apache avec PHP

- Étape 2 : Renommer « index.html » présent dans « C:\Apache24\htdocs » en « index.php »
- Étape 3 : Modifier le corp du fichier « index.php » en mettant les lignes suivantes:
    - <?php

      phpinfo();

```php
<?php

phpinfo();
```

Fig. : Fichier index.php

- Étape 4 : Accéder au fichier « httpd.conf » depuis « C:\Apache24\conf » et ajouter « index.php » dans « DirectoryIndex »

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

Fig. : Fichier de configuration Apache

Copyright - Tout droit réservé - OFPPT

Configurer Apache avec PHP

- Étape 5 : Relancer Apache en utilisant la commande « httpd –k restart »



Fig. : Résultat index.php

PARTIE 1

Copyright - Tout droit réservé - OFPPT

Téléchargement et installation de VSCode

- Étape 1 : Télécharger VS Code depuis le site officiel https://code.visualstudio.com/download
- Étape 2 : Double cliquer sur le fichier téléchargé pour démarrer l'installation.

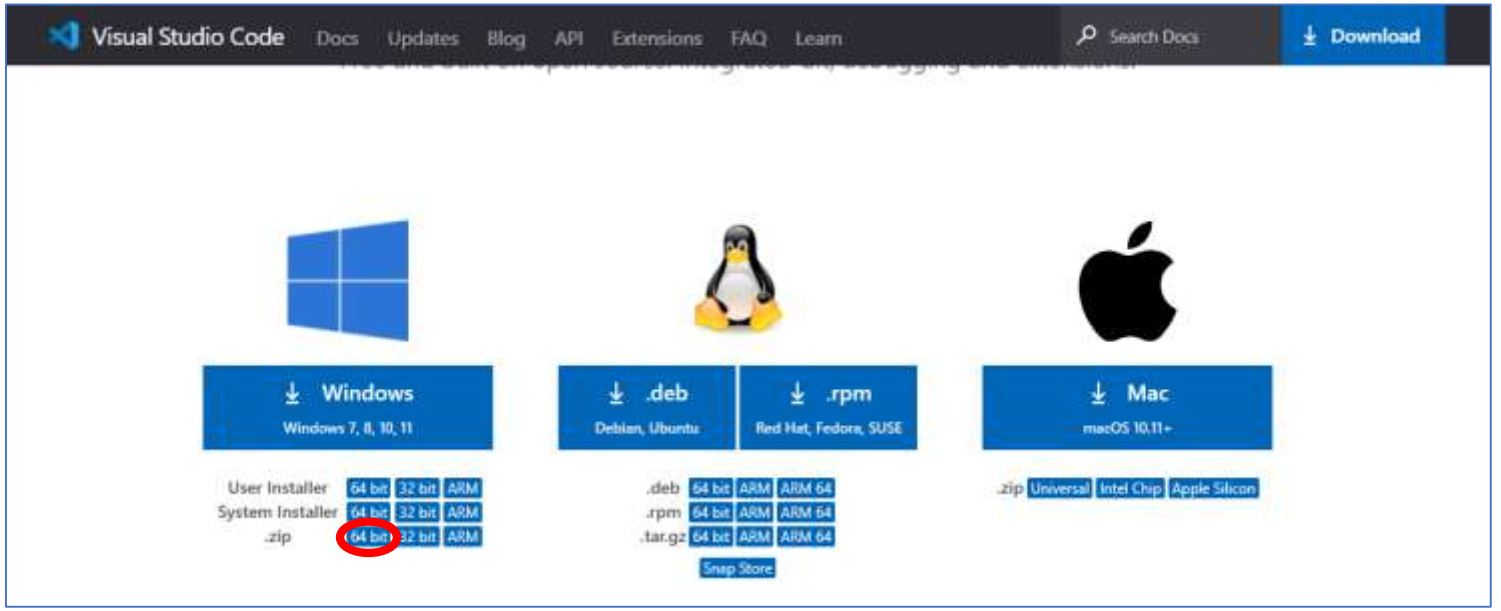


Fig. : Téléchargement de VS Code

Copyright - Tout droit réservé - OFPPT

Ajout des extensions à VS Code

- Étape 3 : Installer les extensions qui vous intéresse pour bénéficier des propriétés adaptées au code PHP

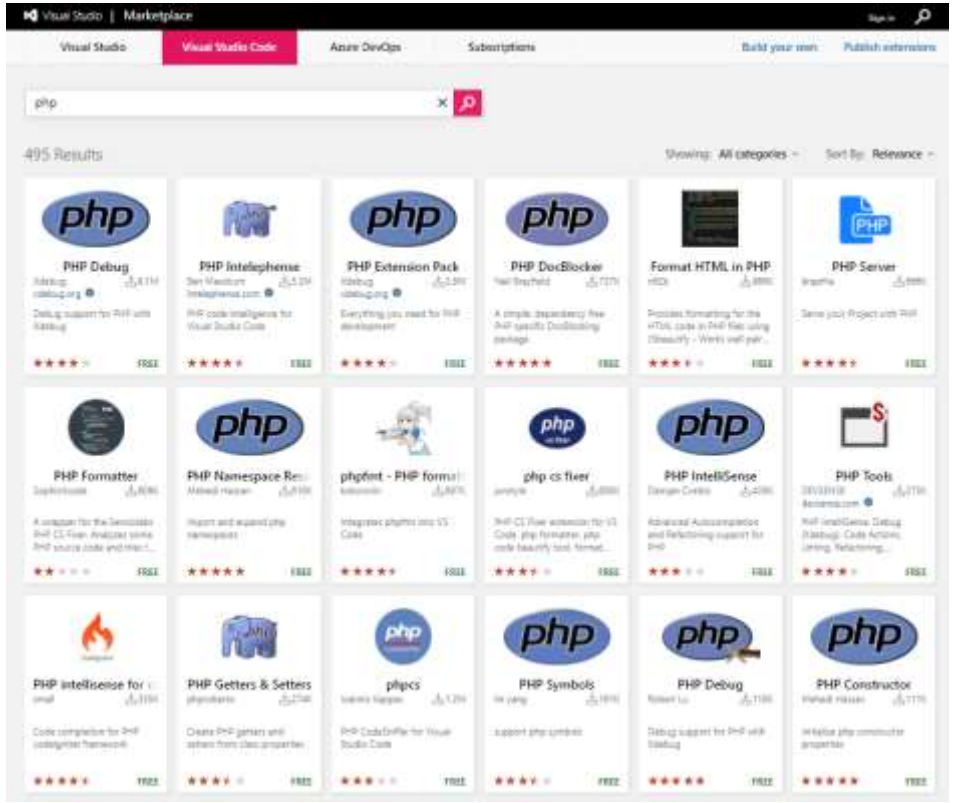


Fig. : Les extensions PHP

Copyright - Tout droit réservé - OFPPT

Ajout des extensions à VS Code

- Étape 4 : Ouvrir dans VSCode le répertoire « C:\Apache24\htdocs »
- Étape 5 : Recommandation, installer les extensions suivantes :
    - French Language Pack for Visual Studio Code :
    - PHP Debug
    - PHP Formatter
    - Format HTML in PHP

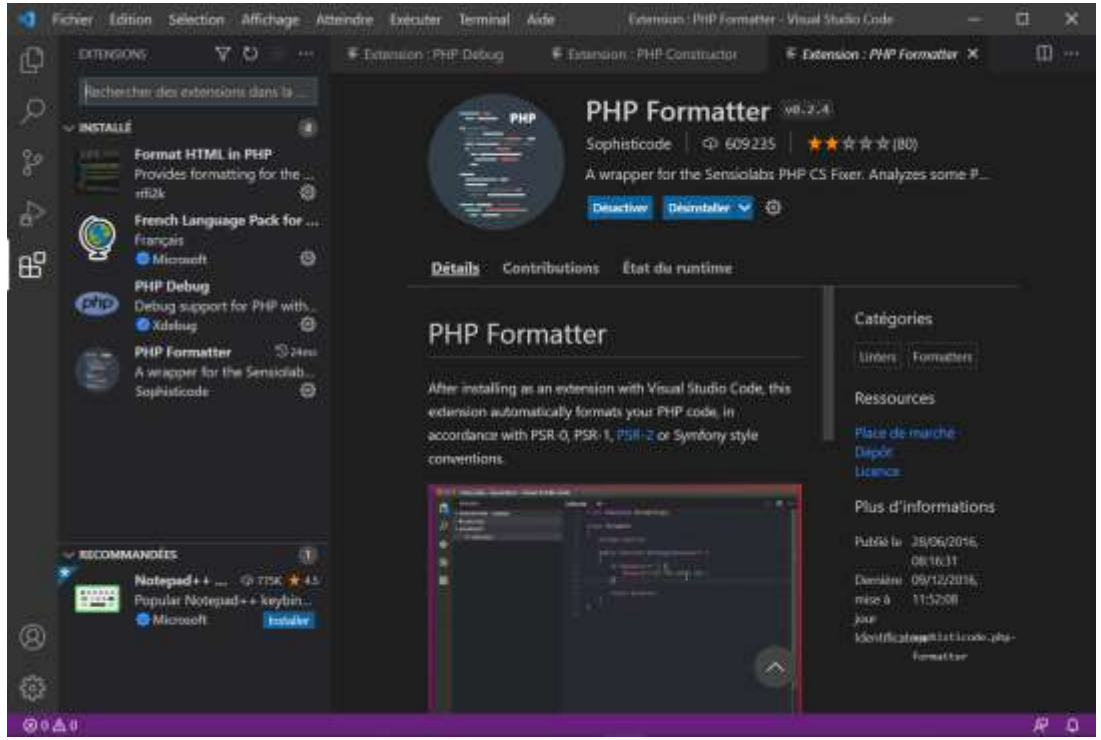
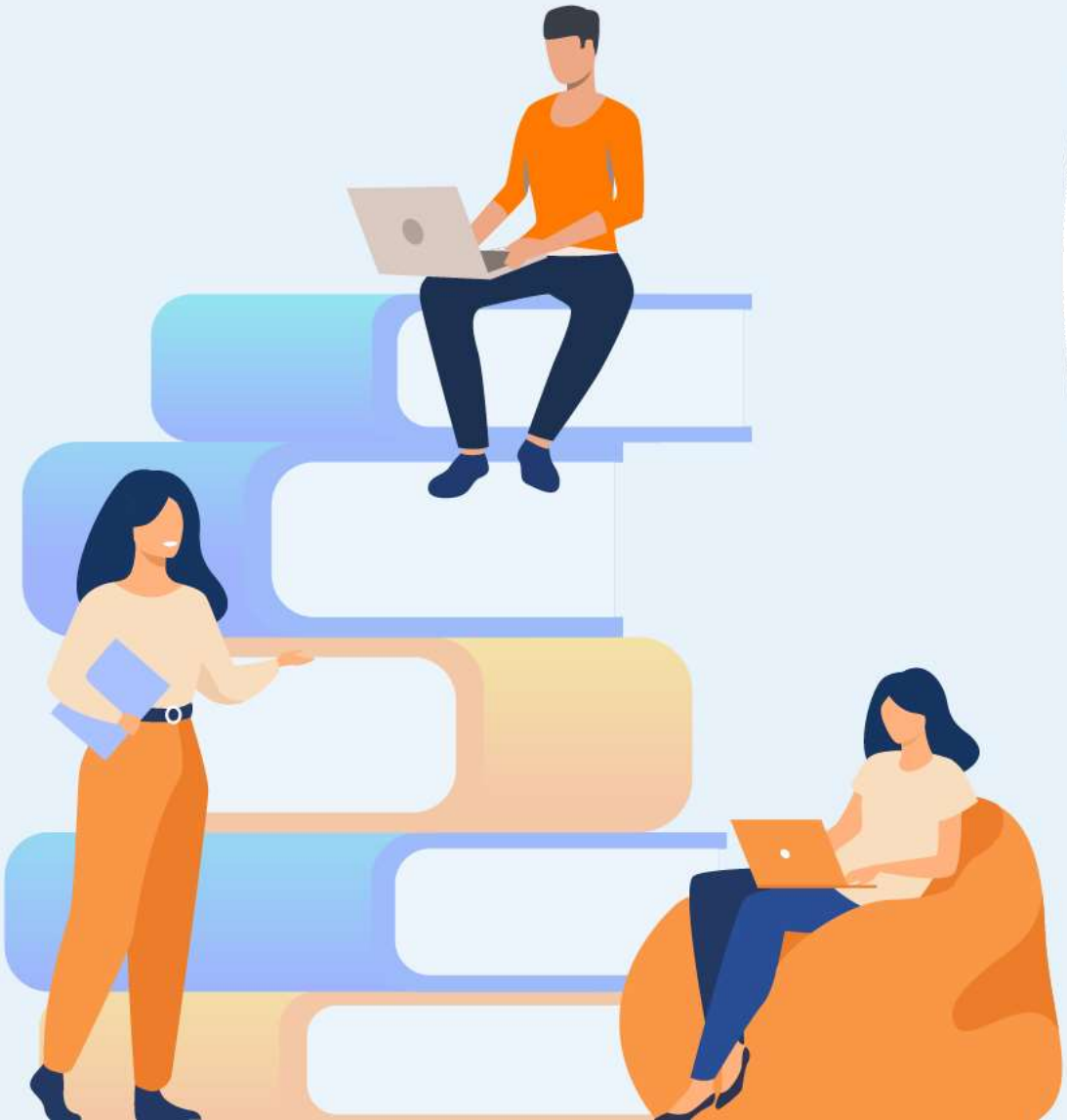


Fig. : Les extensions PHP recommandées

PARTIE 1

Copyright - Tout droit réservé - OFPPT



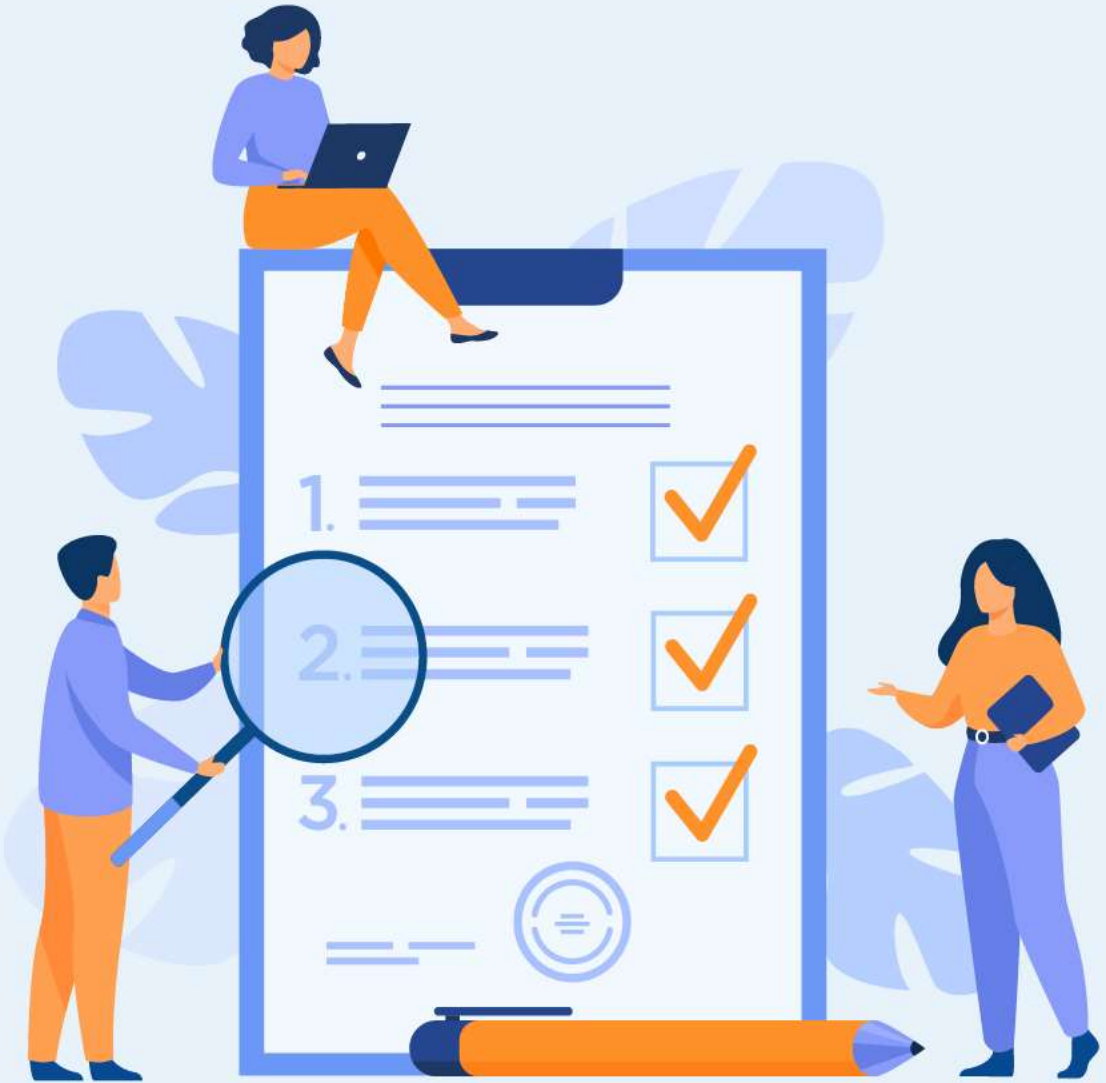
PARTIE 2

PROGRAMMER EN PHP

Dans ce module, vous allez :

- Maitriser le langage PHP
- Traiter les données en PHP
- Utiliser l'orientée objet en PHP

20 heures



Activité 1

Traiter les données

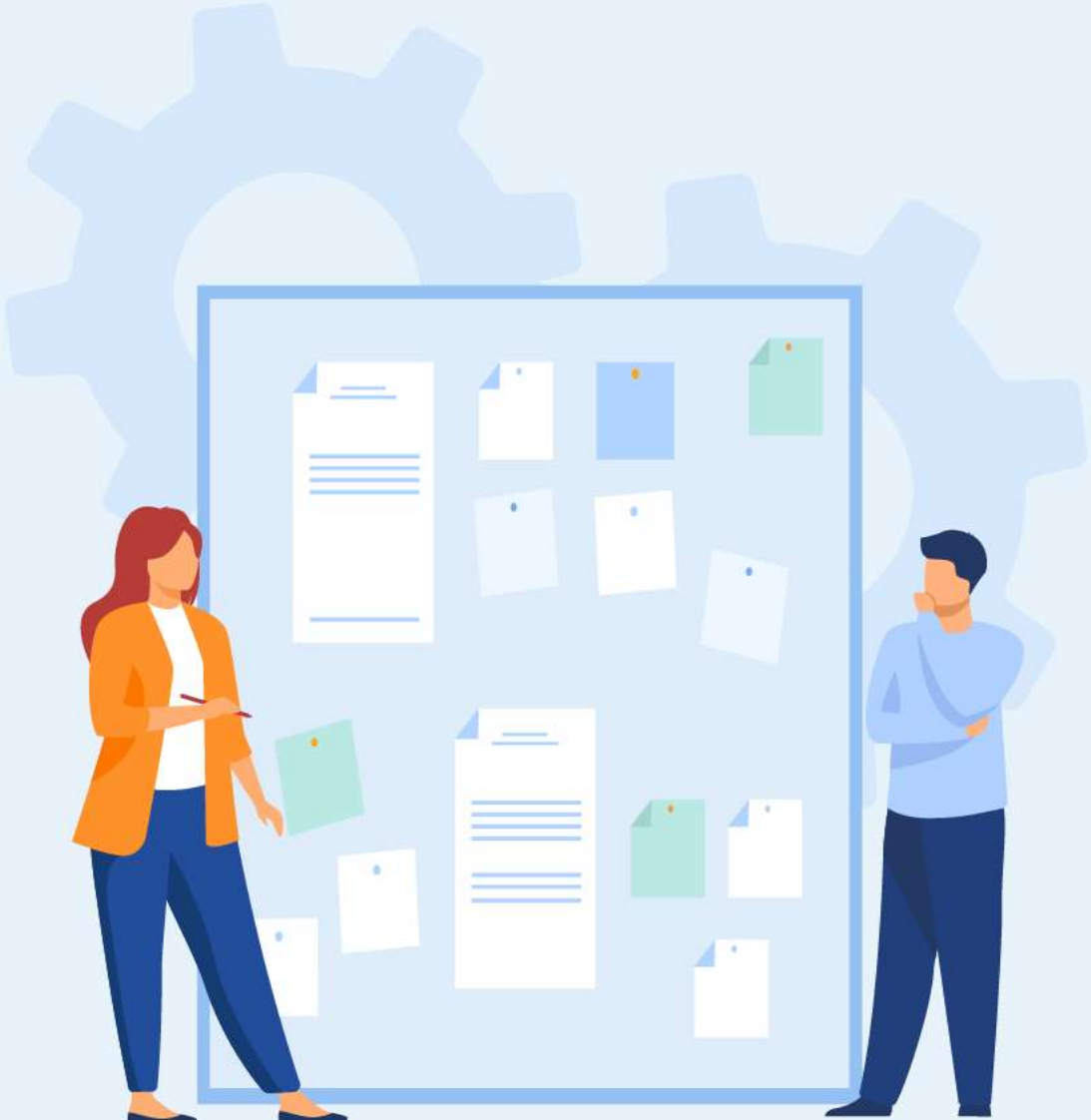
Compétences visées :

- Mettre en place des éléments pour les formulaires
- Mettre en place des éléments pour les tableaux
- Utiliser les variables d’environnement et de transmission

Recommandations clés :

- Révision générale du résumé théorique

06 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur la création des formulaires
- Rappeler les variables d’environnement

2. Pour l’apprenant :

- Mettre en place les différents composants d'un formulaire
- Définir les variables d’environnement

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Mettre en place les éléments d'un formulaire
  - Utiliser les variables prédéfinis

Manipuler les données en PHP

1. Créer une page php qui affiche simplement la chaîne de caractères « Salut PHP, nous sommes le " suivie de la date du jour sur le serveur
2. Créer une page php permettant d'afficher à la suite de la date, le message « C'est le matin » ou « C'est l'après-midi » en fonction de l'heure sur le serveur.
3. Afficher dans un tableau la signification et la valeur des variables d'environnement suivante :
    - $SERVER_ADDR
    - $HTTP_HOST
    - $REMOTE_ADDR
    - gethostbyAddr($REMOTE_ADDR)
    - $HTTP_USER_AGENT

Copyright - Tout droit réservé - OFPPT

Réponses

1. Créez une page php qui affiche simplement la chaîne de caractères « Salut PHP, nous sommes le " suivie de la date du jour sur le serveur

```
<html>
  <body>
    Salut PHP, nous sommes le <?php echo Date("j/m/Y"); ?>
  </body>
</html>
```

2. Créez une page php permettant d'afficher à la suite de la date, le message « C'est le matin" ou « C'est l'après-midi » en fonction de l'heure sur le serveur.

```
<html>
  <body>
    Salut PHP, nous sommes le <?php echo Date("j/m/Y"); ?>,
    <?php
      echo "il est ".Date("H:i:s")."<br>";
      if ( Date("H") < 12 ) {
        echo "C'est le matin";
      } else {
        echo "C'est l'après-midi";
      }
    ?>
  </body>
</html>
```


Copyright - Tout droit réservé - OFPPT

Réponses

3. Afficher dans un tableau la signification et la valeur des variables d'environnement suivante

- $SERVER_ADDR
- $HTTP_HOST
- $REMOTE_ADDR
- gethostbyAddr($REMOTE_ADDR)
- $HTTP_USER_AGENT

```
<html>
  <body>
    <table border="1">
      <caption align="top">Quelques variables d'environnement</caption>
      <tr><th align="center">Variable</th><th align="center">Valeur</th></tr>
      <tr><td>Adresse IP du serveur web</td>
      <td><?php echo $_SERVER['SERVER_ADDR'] ?></td></tr>
      <tr><td>Nom du serveur web</td>
      <td><?php echo $_SERVER['HTTP_HOST'] ?></td></tr>
      <tr><td>Adresse IP du client</td>
      <td><?php echo $_SERVER['REMOTE_ADDR'] ?></td></tr>
      <tr><td>Nom de la machine cliente</td>
      <td><?php echo gethostbyAddr($_SERVER['REMOTE_ADDR']) ?></td></tr>
      <tr><td>Navigateur du client</td>
      <td><?php echo $_SERVER['HTTP_USER_AGENT'] ?></td></tr>
    </table>
  </body>
</html>
```


Copyright - Tout droit réservé - OFPPT

Créer un formulaire

Créez un formulaire index.html comportant 2 champs de texte nom, prénom, un menu d'options sexe (M ou F), et un menu de choix multiples cours (simple, moyen, difficile). Vous ferez appel à un script index.php pour traiter les données du formulaire. Elles seront envoyées par la méthode GET.

Copyright - Tout droit réservé - OFPPT

Activité 1
Créer un formulaire

PARTIE 2

Réponses

4. Créez un formulaire index.html comportant 2 champs de texte nom, prénom, un menu d'options sexe (M ou F), et un menu de choix multiples cours (simple, moyen, difficile). Vous ferez appel à un script index.php pour traiter les données du formulaire. Elles seront envoyées par la méthode GET.

```html
<html>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
  <body>
    <form action="index.php" method="get">
      Nom: <input type="text" name="nom"><br>
      Prénom: <input type="text" name="prenom"><br>
      Sexe: <select name="sexe">
        <option>
        <option>F
        <option>M
      </select><br>
      Cours: <br>
      <select multiple name="cour[]">
        <option value="simple">Simple</option>
        <option value="moyen">Moyen</option>
        <option value="difficile">Difficile</option>
      </select><br>
      <input type="submit" value="submit Me!">
    </form>
  </body>
</html>
```

Fichier : index.html

```php
<html>
  <body>
    <h1>Affichage des données saisies</h1>
    <ul>
      <li>Nom: <?php print $_REQUEST['nom'] ?></li>
      <li>Prénom: <?php print $_REQUEST['prenom'] ?></li>
      <li>Sexe: <?php print $_REQUEST['sexe'] ?></li>
      <li>Cours:
      <ul>
        <?php
          if (isset($_REQUEST['cour']))
          foreach($_REQUEST['cour'] as $v) print "<li>$v</li>";
        ?>
      </ul>
    </ul>
    <a href="javascript:history.back()">Essayez à nouveau</a>
  </body>
</html>
```

Fichier : index.php

Copyright - Tout droit réservé - OFPPT

Gérer un formulaire

1. Créer un formulaire comprenant un groupe de champs ayant pour titre "Adresse client". Le groupe doit permettre la saisie du nom, du prénom, de l'adresse, de la ville et du code postal.
2. Les données sont ensuite traitées par un fichier PHP séparé récupérant les données et les affichant dans un tableau HTML.
3. Le résultat obtenu doit ressembler à la capture suivante :

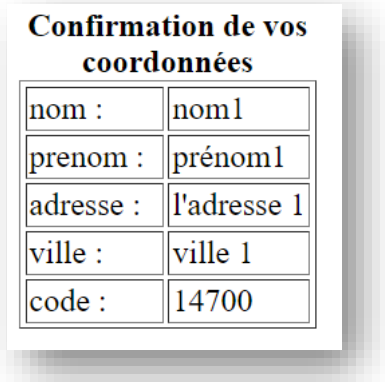


Fig. : Résultat exercice

Copyright - Tout droit réservé - OFPPT

PARTIE 2

Réponses

Fichier index.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<title>TP</title>
</head>
<body>
<form action= "index.php" method="post">
 <fieldset>
  <legend><b>Saisissez vos coordonnées </b></legend>
<table border="0" >
 <tr>
  <td>Nom : </td>
  <td><input type="text" name="nom" /></td>
 </tr>
 <tr>
  <td>Prénom : </td>
  <td> <input type="text" name="prenom" /></td>
 </tr>
```

Fichier : index.html (Partie1)

```
 <tr>
  <td>Adresse : </td>
  <td><input type="text" name="adresse" /></td>
 </tr>
 <tr>
  <td>Ville :</td>
  <td><input type="text" name="ville" /></td>
 </tr>
 <tr>
  <td>Code postal :</td>
  <td><input type="text" name="code" maxlength="5"/></td>
 </tr>
 <tr>
  <td>CONFIRMER</td>
  <td><input type="submit" value="ENVOI" /></td>
 </tr>
</table>
 </fieldset>
 </form>
</body>
</html>
```

Fichier : index.html (Partie2)

Copyright - Tout droit réservé - OFPPT

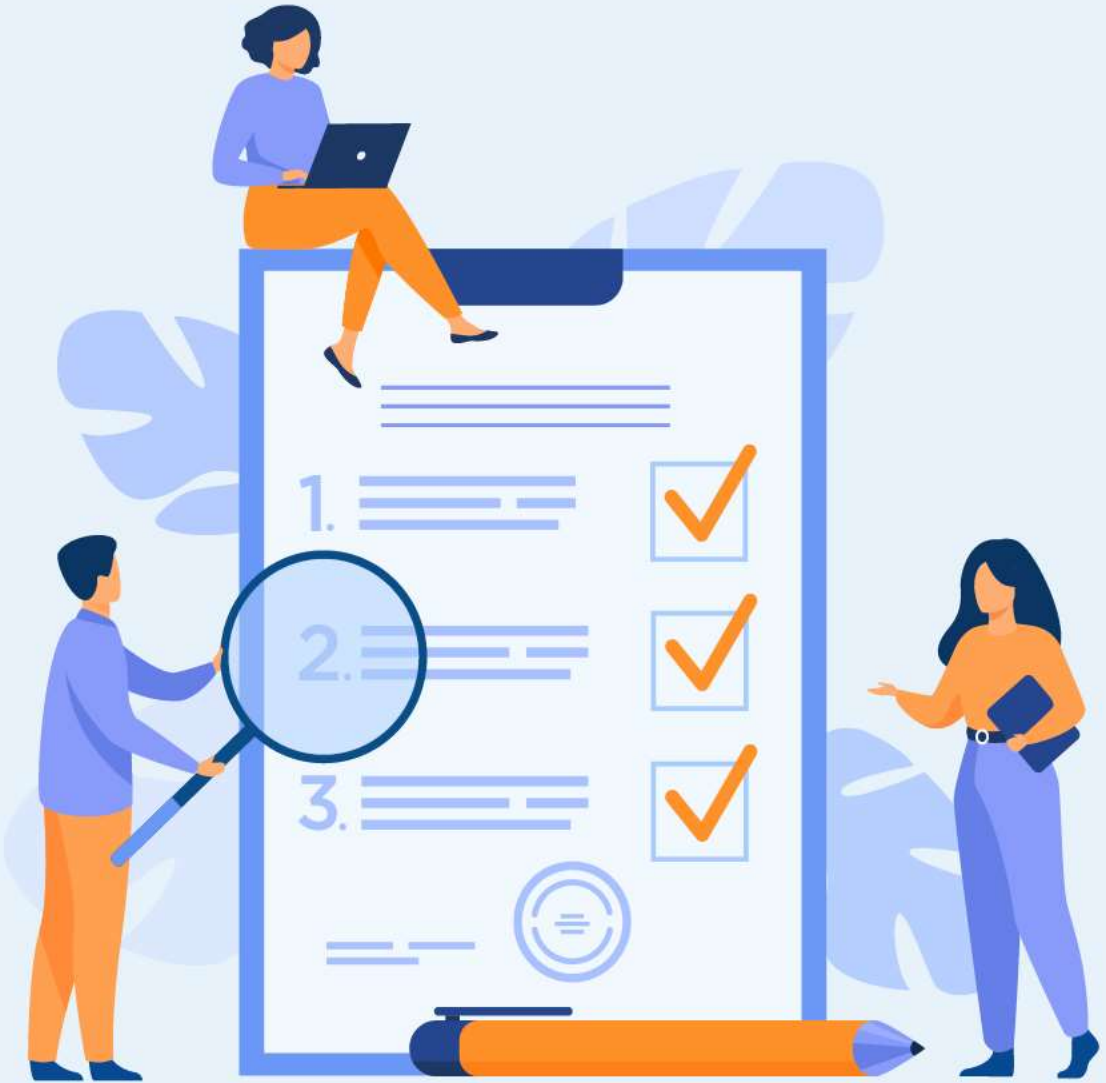
Réponses

Fichier index.php

```php
<?php
echo "<table border=\"1\" >";
echo "<caption><b>Confirmation de vos
coordonnées</b></caption>";
foreach($_POST as $cle=>$val)
{
echo "<tr> <td> $cle :  </td> <td>".stripslashes($val)
."</td></tr>";
}
echo "</table>";
?>
```

Fichier : index.php

Copyright - Tout droit réservé - OFPPT



Activité 2

Manipuler les fichiers

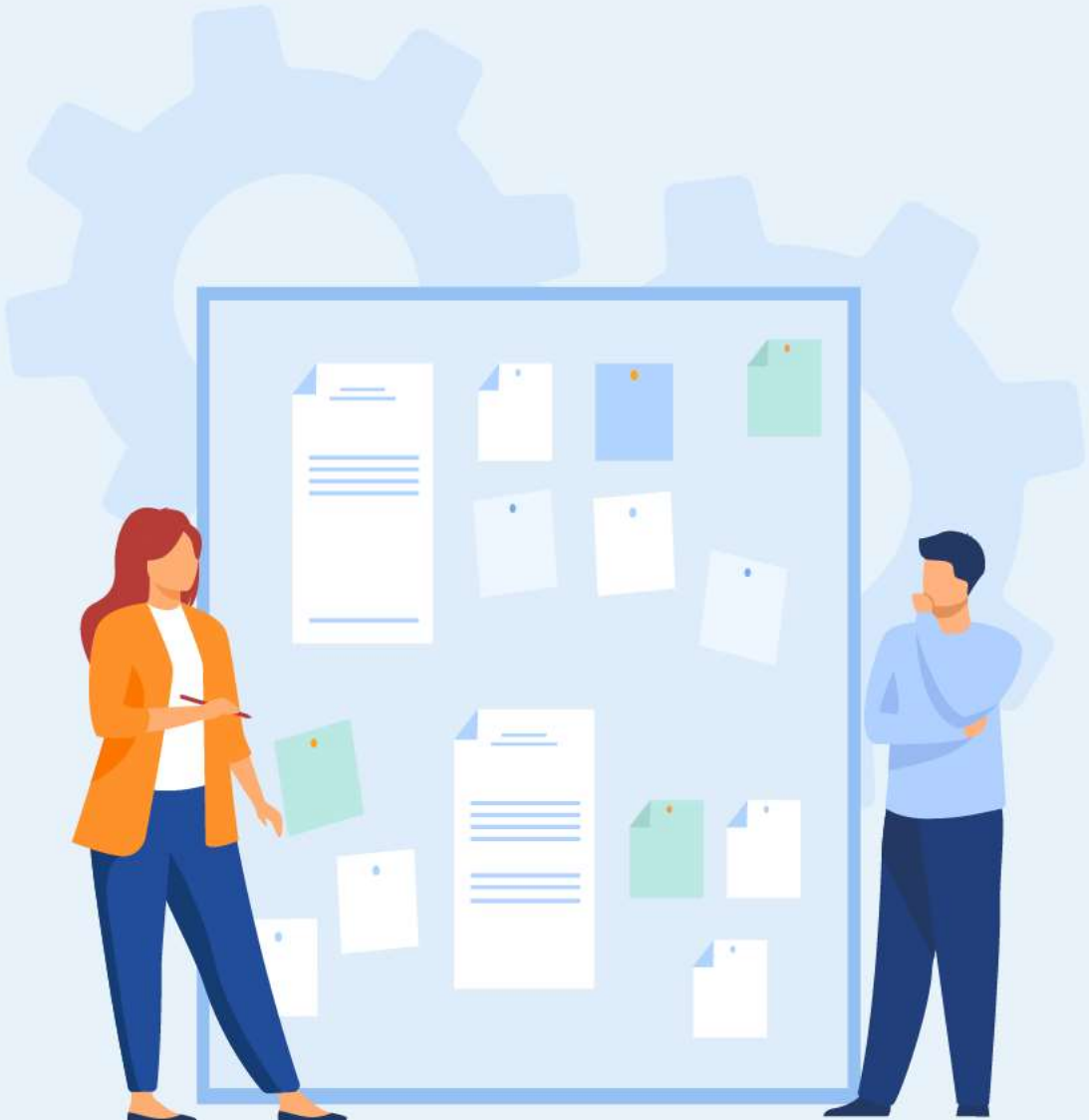
Compétences visées :

- Traiter les données d’un fichier
- Utiliser les fonctions élémentaires des fichiers

Recommandations clés :

- Révision générale du résumé théorique

06 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur la création des formulaires
- Rappeler les bases théoriques sur la gestion des fichiers

2. Pour l’apprenant :

- Mettre en place les différents composants d'un formulaire
- Manipuler les fichiers

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Définir la structure d'un formulaire
  - Mettre en place les éléments d’un fichier
  - Déterminer les valeurs des propriétés des composants

Envoi d'un fichier

Créer un formulaire n'effectuant que le transfert de fichiers ZIP et d'une taille limitée à 1 Mo. Le script affiche le nom du fichier du poste client ainsi que la taille du fichier transféré et la confirmation de réception.

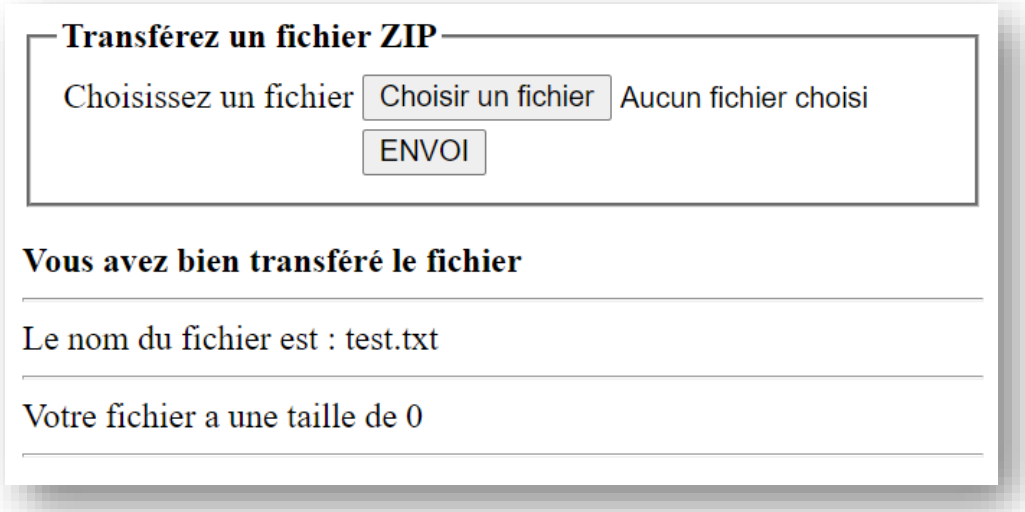


Fig. : Résultat exercice

Copyright - Tout droit réservé - OFPPT

Activité 2
Envoi d'un fichier

PARTIE 2

Réponses

```php
<?php
if(isset($_FILES['fich']))
{
 if($_POST["MAX_FILE_SIZE"] <$_FILES["fich"]["size"])
 {
  echo "<b>Taille trop grande </b><hr />";
  echo "Taille maximale autorisée :",$_POST["MAX_FILE_SIZE"],"octets<hr / >";
  echo "Taille du fichier transféré :",$_FILES["fich"]["size"],"octets<hr / >";
 }
 else
 {
  //Enregistrement et renommage du fichier
$result=move_uploaded_file($_FILES["fich"]["tmp_name"],"monfichier.zip");
if($result==TRUE)
{
 echo "<b>Vous avez bien transféré le fichier</b><hr />";
 echo "Le nom du fichier est : ",$_FILES["fich"]["name"],"<hr/>";
 echo "Votre fichier a une taille de ",$_FILES["fich"]["size"],"<hr />";
}
  else {echo "<hr /> Erreur de transfert n°",$_FILES["fich"]["error"];}
 }
}
?>
```

Fichier : index.php (Partie2)

```html
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>TP</title>
</head>
<body>
<form action="<?= $_SERVER['PHP_SELF'] ?>" method="post"
enctype="multipart/form-data">
 <fieldset>
  <legend><b>Transférez un fichier ZIP </b></legend>
<table border="0" >
 <tr>
    <td>Choisissez un fichier </td>
    <td><input type="file" name="fich" /></td>
    <td><input type="hidden" name="MAX_FILE_SIZE" accept="application/zip"
value="1000000"/></td>
  </tr>
 <tr>
  <td> </td>
  <td><input type="submit" value="ENVOI" /></td>
  </tr>
</table>
 </fieldset>
 </form>
</body>
</html>
```

Fichier : index.php (Partie1)

Copyright - Tout droit réservé - OFPPT

Activité 2
Livre d'or

Exercices

1. Créer un livre d'or qui n'affiche que les cinq derniers avis donnés par les visiteurs du site.
2. Le résultat obtenu doit ressembler à la capture suivante :



Fig. : Résultat exercice

PARTIE 2

Copyright - Tout droit réservé - OFPPT

Activité 2
Livre d'or

Réponses

Fichier index.html

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
    <title>Le livre est d'or </title>
  </head>
  <body style="background-color: #ffcc00;">
    <form action="<?= $_SERVER['PHP_SELF']?>" method="post" >
      <fieldset>
        <legend><b>Donnez votre avis sur PHP 8 ! </b></legend>
        <b>Nom :  <input type="text" name="nom" width="60" /> <br>
        Mail :  <input type="text" name="mail" width="60" /> <br>
        Vos commentaires sur le site</b><br>
        <textarea name="comment" rows="10" cols="50"></textarea> <br>
        <input type="submit" value="Envoyer " name="envoi" />
        <input type="submit" value="Afficher les avis" name="affiche" />
      </fieldset>
    </form>
```

Fichier : index.html (Partie1)

PARTIE 2

Copyright - Tout droit réservé - OFPPT

Activité 2
Livre d'or

Réponses

Fichier index.html

```php
<?php
        $date= time();
        //ENREGISTREMENT
        if(isset($_POST['envoi'])) {
           if(isset($_POST['nom']) && isset($_POST['mail']) && isset($_POST['comment'])) {
              echo "<h2>",$_POST['nom']," merci de votre avis </h2> ";
              if(file_exists("livre2.txt") ) {
                 if($id_file=fopen("livre2.txt","a")) {
                    flock($id_file,2);
                    fwrite($id_file,$_POST['nom'].":".$_POST['mail'].":".$date.":".$_POST['comment']."\n");
                    flock($id_file,3);
                    fclose($id_file);
                 }
                 else{
                    echo "fichier inaccessible";
                 }
              }
              else {
                 $id_file=fopen("livre2.txt","w");
                 fwrite($id_file,$$_POST['nom'].":".$$_POST['mail'].":".$date.":".$$_POST['comment']."\n");
                 fclose($id_file);
              }
           }
        }
```

Fichier : index.html (Partie2)

PARTIE 2

Copyright - Tout droit réservé - OFPPT

Activité 2
Livre d'or

Réponses

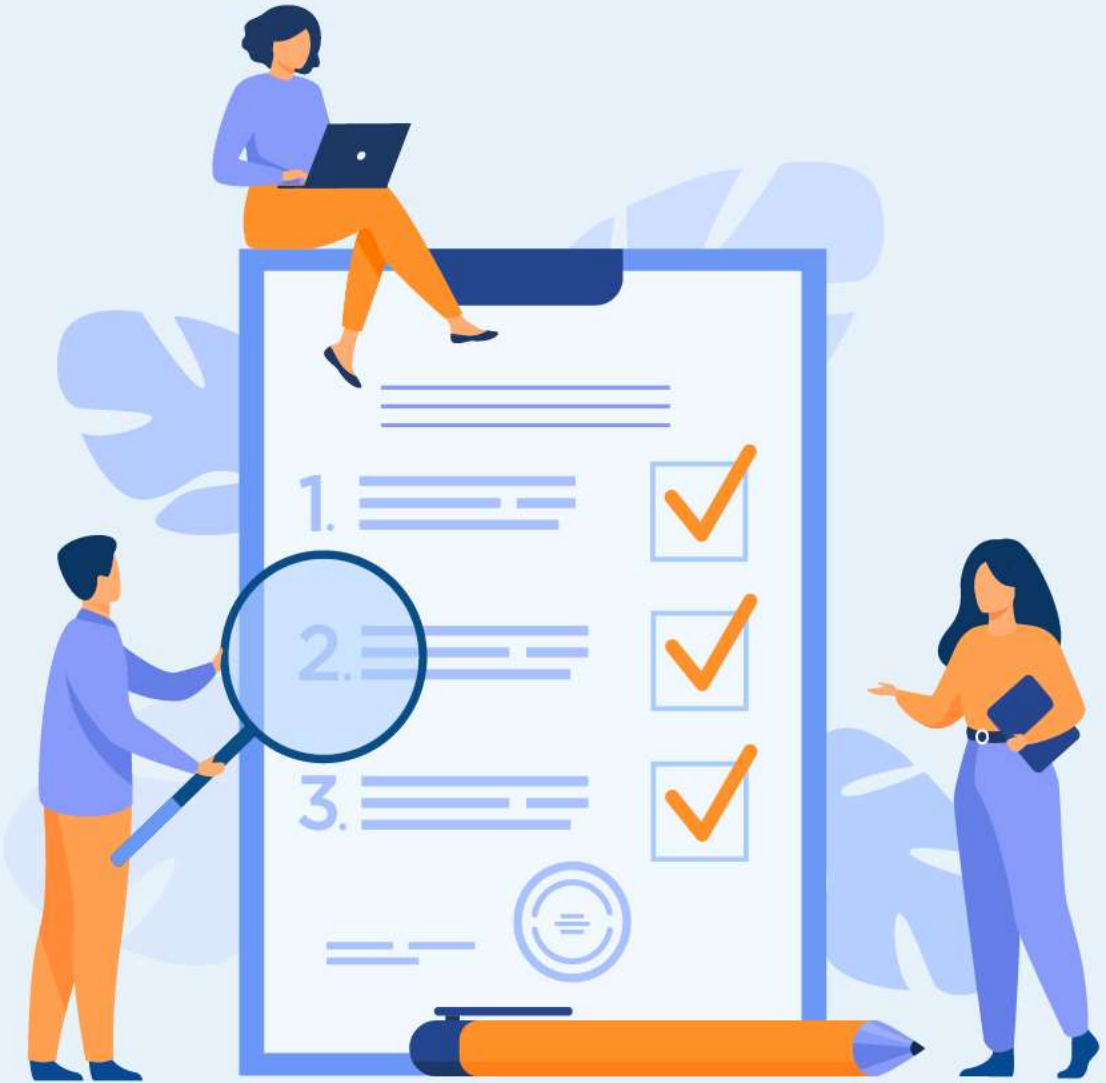
Fichier index.html

```
//LECTURE DES DONNES
        if(isset($_POST['affiche'])) {
           if($id_file=fopen("livre2.txt","r")) {
              echo "<table border=\"2\"> <tbody>";
              $i=0;
              while($tab=fgetcsv($id_file,200,":") ) {
                 $tab5[$i]=$tab;
                 $i++;
              }
              $tab5=array_reverse($tab5);
              echo "<hr />";
              for($i=0;$i<5;$i++) {
                 echo "<tr> <td>",$i+1 ,": de: ".$tab5[$i][0]." </td> <td> <ahref=\"mailto:".$tab5[$i][1]." \" >
".$tab5[$i][1]."</a></td><td>le: ",date("d/m/y H:i:s", $tab5[$i][2])," </td></tr>";
                 echo "<tr > <td colspan=\" 3 \">", stripslashes($tab5[$i][3]),"</td> </tr> ";
              }
              fclose($id_file);
           }
           echo "</tbody></table> ";
        }
        else {
           echo "<h2>Donnez votre avis puis cliquez sur 'envoyer' !</h2> ";
        }
     ?>
  </body>
</html>
```

Fichier : index.html (Partie3)

PARTIE 2

Copyright - Tout droit réservé - OFPPT



Activité 3

Utiliser l’orientée objet en PHP

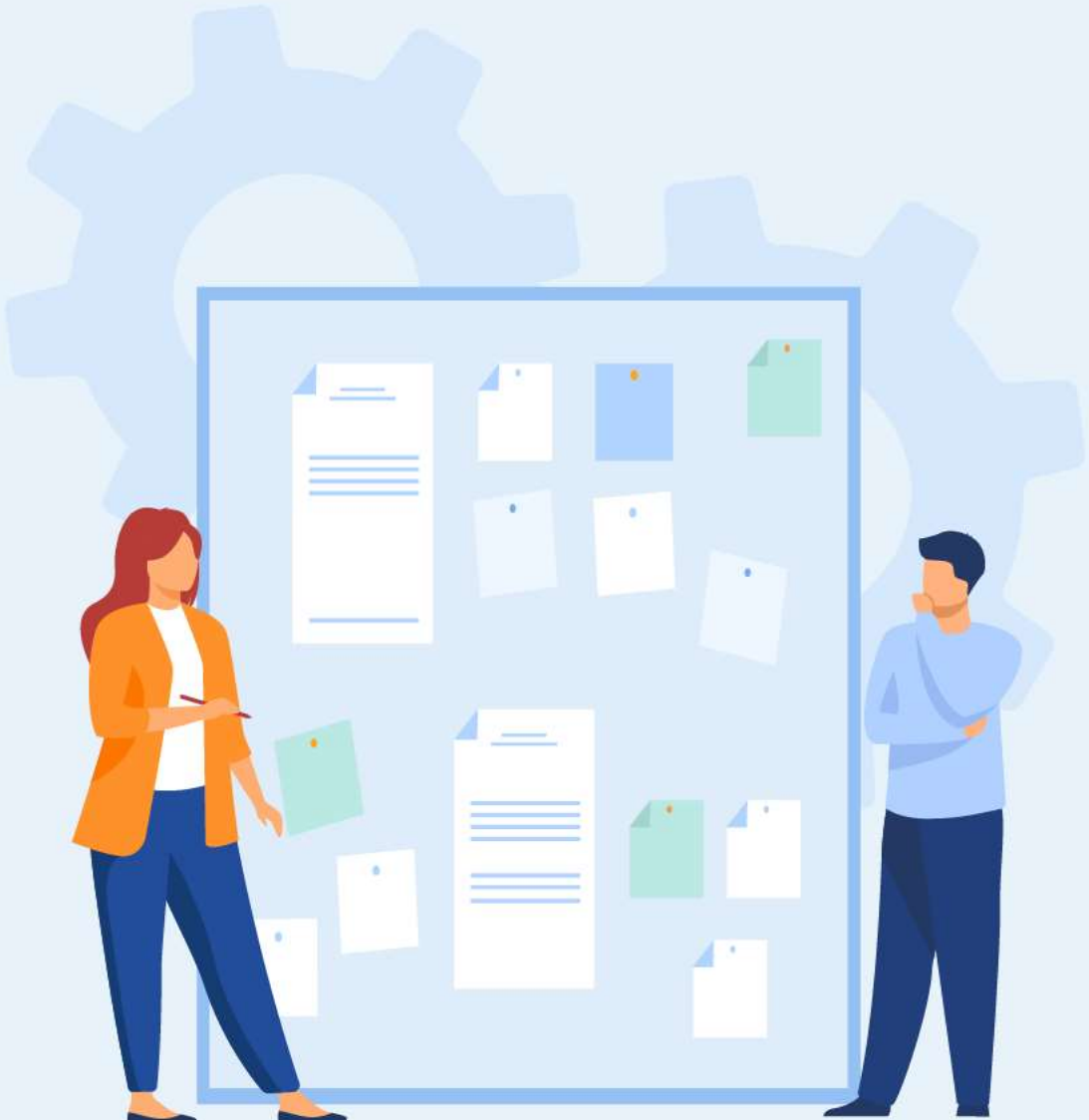
Compétences visées :

- Utiliser les classes et les fonctions

Recommandations clés :

- Révision générale du résumé théorique

08 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur la programmation orientée objet
- Rappeler les méthodes magiques en PHP

2. Pour l'apprenant :

- Appliquer les concepts de programmation orientée objet
- Instancier un objet et une classe

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Créer un objet
  - Créer une classe
  - Manipuler les objets

Fonctions et classes

1. Définir une classe Compte représentant un compte bancaire.
2. Cette classe contiendra deux attributs privés, montant et intérêt (annuel), un constructeur prenant en paramètre un montant initial et un intérêt, une méthode **get_montant** retournant le montant du compte, ainsi qu'une méthode un an modifiant le montant du compte au bout d'un an en fonction de l'intérêt.
3. Créer deux comptes, un d'un montant de 200 dirhams à 20% et l'autre d'un montant de 1000 dirhams à 2%.
4. Afficher le montant de chaque compte au bout de 10 ans.

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   class Compte{
      private $montant;
      private $interet;//En pourcentage
      public function __construct($interet, $montantInitial) {
         $this->montant = $montantInitial;
         $this->interet = $interet;
      }
      public function un_an() {
         $this->montant *= 1+ $this->interet / 100;
      }
      public function get_montant() {
         return $this->montant;
      }
   }
   $c1 = new Compte(20,200);
   $c2 = new Compte(2,1000);
   for($i = 1 ; $i <= 10 ; $i++) {
      $c1->un_an();
      $c2->un_an();
   }
   echo'<p> Valeur du compte = ' . $c1->get_montant() . ' dirhams </p>';
   echo'<p> Valeur du compte = ' . $c2->get_montant() . ' dirhams </p>';
?>
```

Fichier : index.php

Copyright - Tout droit réservé - OFPPT

Fonctions et classes

1. Définir une classe Personne possédant les propriétés nom, prénom, et une méthode presenter() qui renvoie la chaîne de caractères "je m’appelle " suivie du nom et prénom.
2. Créer un programme de test qui instancie 2 personnes, puis affiche leurs descriptions.
3. Ajoutez dans la classe Personne une propriété date de naissance et une méthode age() renvoyant l'âge
4. Tester la classe Personne

Copyright - Tout droit réservé - OFPPT

Réponses

1. Définir une classe Personne possédant les propriétés nom, prénom, et une méthode presenter() qui renvoie la chaîne de caractères « je suis » suivie du nom et prénom.

```php
<?php
   class Personne {
      // proprietes
      var $nom;
      var $prenom;
      // constructeur
      function __construct($nom, $prenom) {
         $this->nom = $nom ;
         $this->prenom = $prenom ;
      }
      // methodes
      function presenter() {
         return "je suis $this->nom $this->prenom" ;
      }
   }
?>
```

Fichier : Personne.php

Copyright - Tout droit réservé - OFPPT

Réponses

2. Créer un programme de test qui instancie 2 personnes, puis affiche leurs descriptions.

```php
<html>
  <body>
    <?php
      include "./personne.php";
      // instanciation de Personne
      $p1 = new Personne("nom1", "prénom1");
      $p2 = new Personne("nom2", "prénom2");
      // affichage
      echo $p1->presenter()."<br>" ;
      echo $p2->presenter()."<br>" ;
    ?>
  </body>
</html>
```

Fichier : index.php

Copyright - Tout droit réservé - OFPPT

PARTIE 2

Réponses

3. Ajoutez dans la classe Personne une propriété date de naissance et une méthode age() renvoyant l'âge (voir Personne.php)

4. Tester la classe Personne (voir index.php)

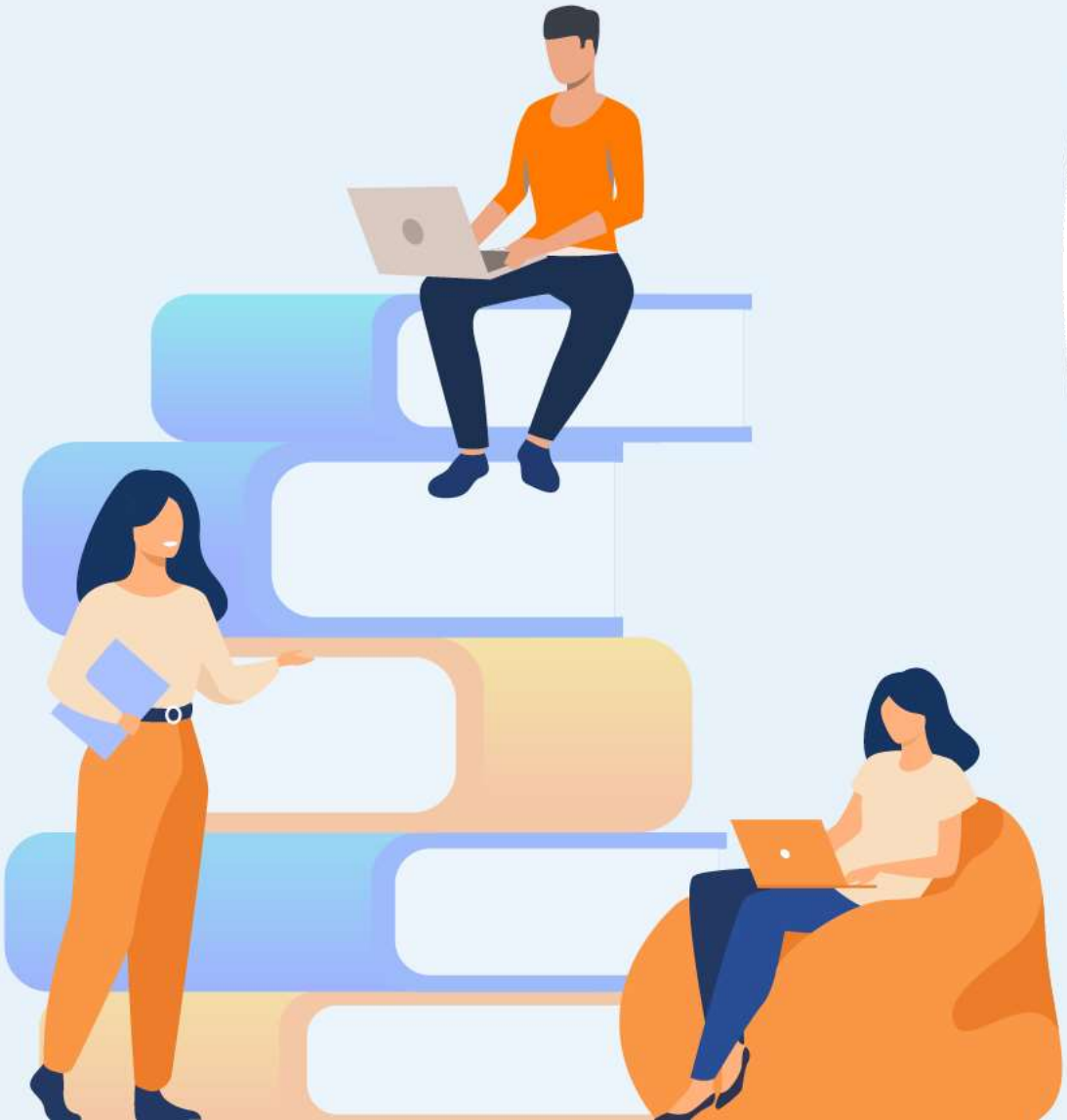
```php
<html>
  <body>
    <?php
      include "./personne.php";
      // instanciation de Personne
      $p1 = new Personne("nom1", "prénom1","31/12/1952");
      $p2 = new Personne("nom2", "prénom2","31/12/2006");
      // affichage
      echo $p1->presenter()."<br>" ;
      echo $p2->presenter()."<br>" ;
    ?>
  </body>
</html>
```

Fichier : index.php

```php
<?php
  class Personne {
    // proprietes
    var $nom ;
    var $prenom ;
    var $date_naiss ; //sous la forme jj/mm/aa
    // constructeur
    function __construct($nom, $prenom, $date_naiss) {
      $this->nom = $nom ;
      $this->prenom = $prenom ;
      $this->date_naiss = $date_naiss ;
    }
    // methodes
    function presenter() {
      $ch = "je m'appelle ".$this->nom." ".$this->prenom.", " ;
      $age = $this->age();
      $ch = $ch."j'ai ".$age." ans" ;
      return $ch ;
    }
    function age() {
      $tabdate = explode("/", $this->date_naiss);
      $t_naiss = mktime(0,0,0,$tabdate[1],$tabdate[0],$tabdate[2]);
      return (int)( ( time() - $t_naiss ) / (365*24*3600)) ;
    }
  }
?>
```

Fichier : Personne.php

Copyright - Tout droit réservé - OFPPT



PARTIE 3

Manipuler les données

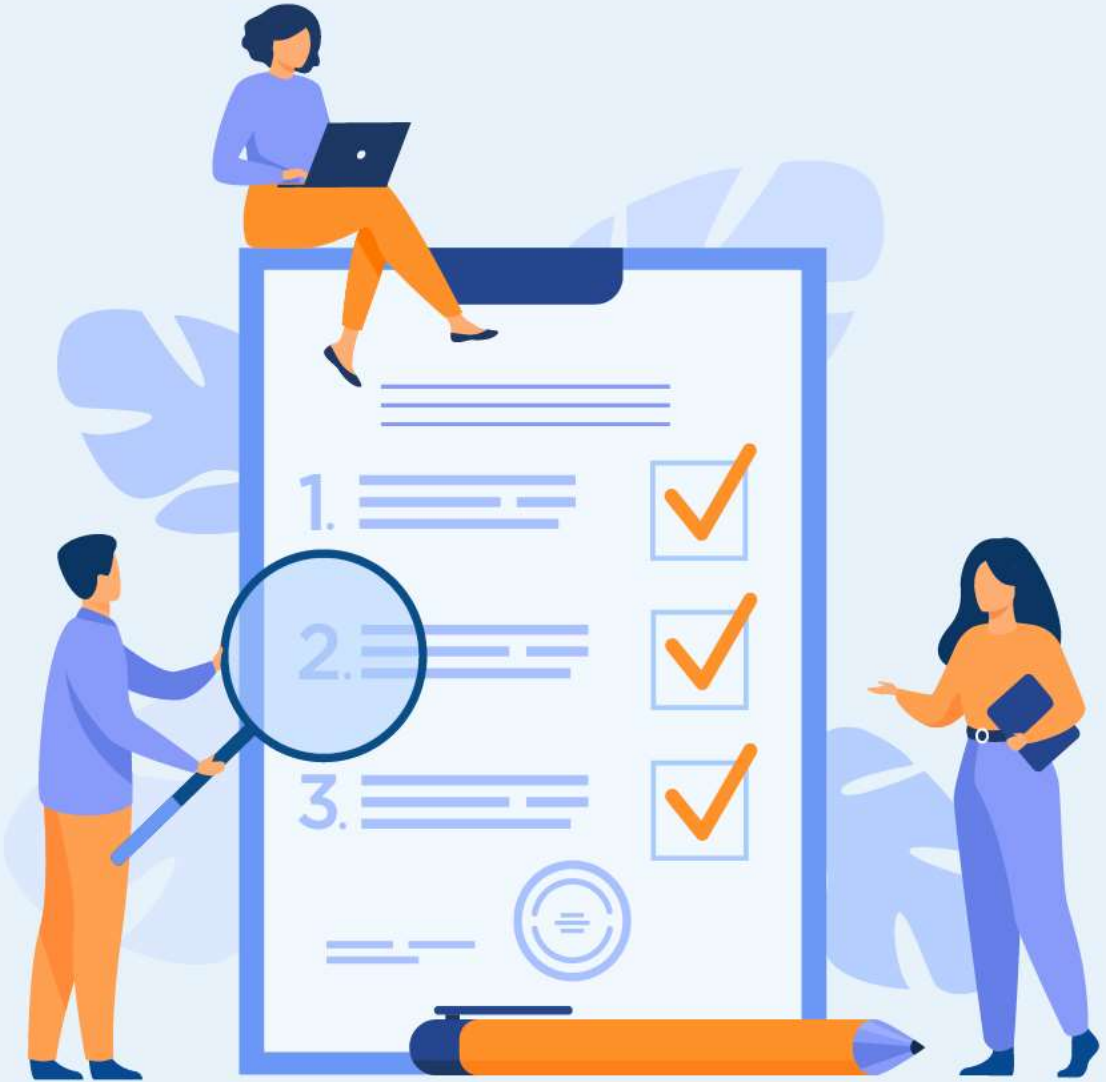
Dans ce module, vous allez :

- Maitriser les scripts d’accès aux données
- Réaliser des connexions avec une base de données

12 heures

Activité 1

Utilisation des cookies



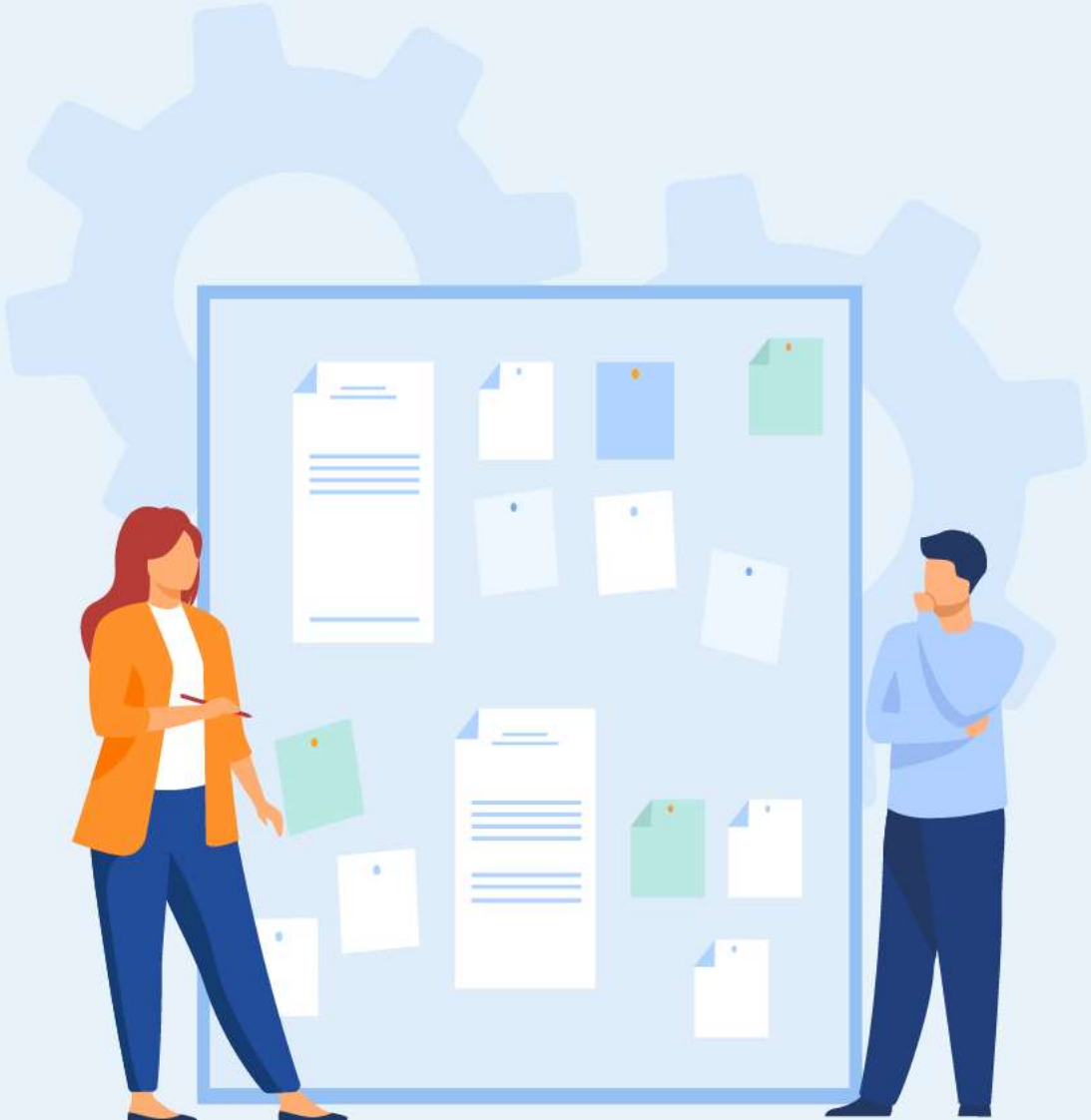
Compétences visées :

- Utiliser les cookies en PHP
- Mettre à jour les cookies

Recommandations clés :

- Bonne révision du résumé théorique

01 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur les cookies

2. Pour l’apprenant :

- Utiliser le résumé théorique
- Alimenter un cookie avec une donnée précise

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Récupérer les données d’un cookie
  - Alimenter un cookie par la donnée souhaitée
  - Mettre à jour les cookies

Activité 1
Utilisation des cookies

Exercice

Ecrire un script php qui permet de sauvegarder les heures de visites dans un cookie , et affiche les détails de visites comme suit :

C'est votre première visite :20-02-2022 21:44:30

Fig. : Première visite

Vous avez consulté cette page 4 fois , voici les détails:

- 20-02-2022 21:44:30
- 20-02-2022 21:47:54
- 20-02-2022 21:48:56
- 20-02-2022 21:49:54

Fig. : après 1ère visite la liste des heures de visite

PARTIE 3

Copyright - Tout droit réservé - OFPPT

Activité 1
Utilisation des cookies

Réponses

- Fichier index.php

```php
<?php
 // On vérifie si le cookie a été reçu dans la superglobale $_COOKIE.
 if(isset($_COOKIE["dates_visites"])){
  $liste_visites =$_COOKIE["dates_visites"];
  $dates=unserialize($_COOKIE["dates_visites"]);
  $dates[]=time();
  setcookie("dates_visites",serialize($dates));
  echo " Vous avez consulté cette page ".count($dates)." fois , voici les détails:";
  echo "<ul>";
  foreach ($dates as $key => $value) {
   echo "<li>".date("d-m-Y H:i:s",$value)."</li>";
  }
  echo "</ul>";
 }
 else{ //Le cookie n'a pas été reçu
  $dates[]=time();
  setcookie("dates_visites",serialize($dates));
  echo "C'est votre première visite :".date("d-m-Y H:i:s",time());
 }
?>
```

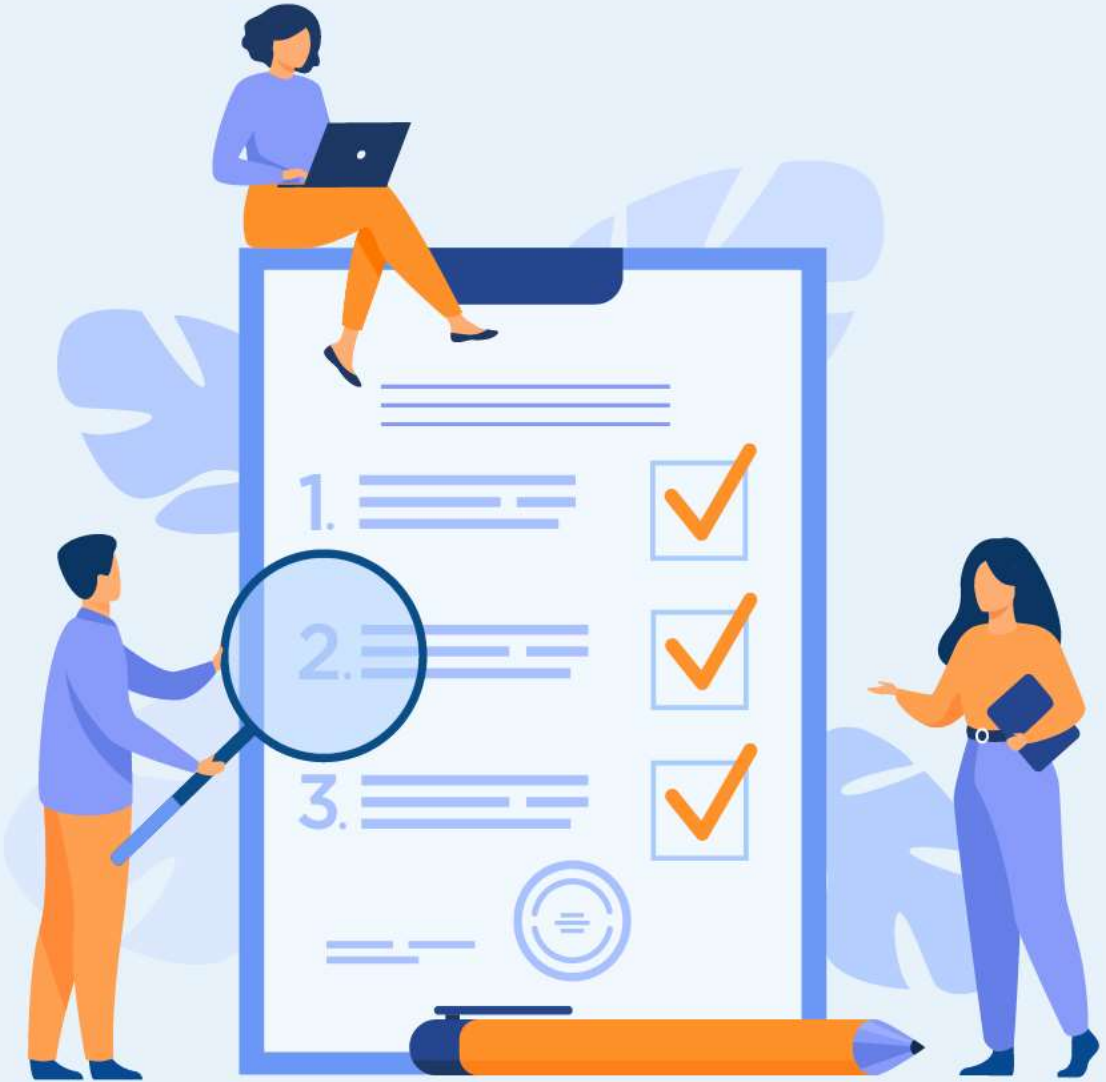
Fichier : index.php

PARTIE 3

Copyright - Tout droit réservé - OFPPT

Activité 2

Utilisation des sessions



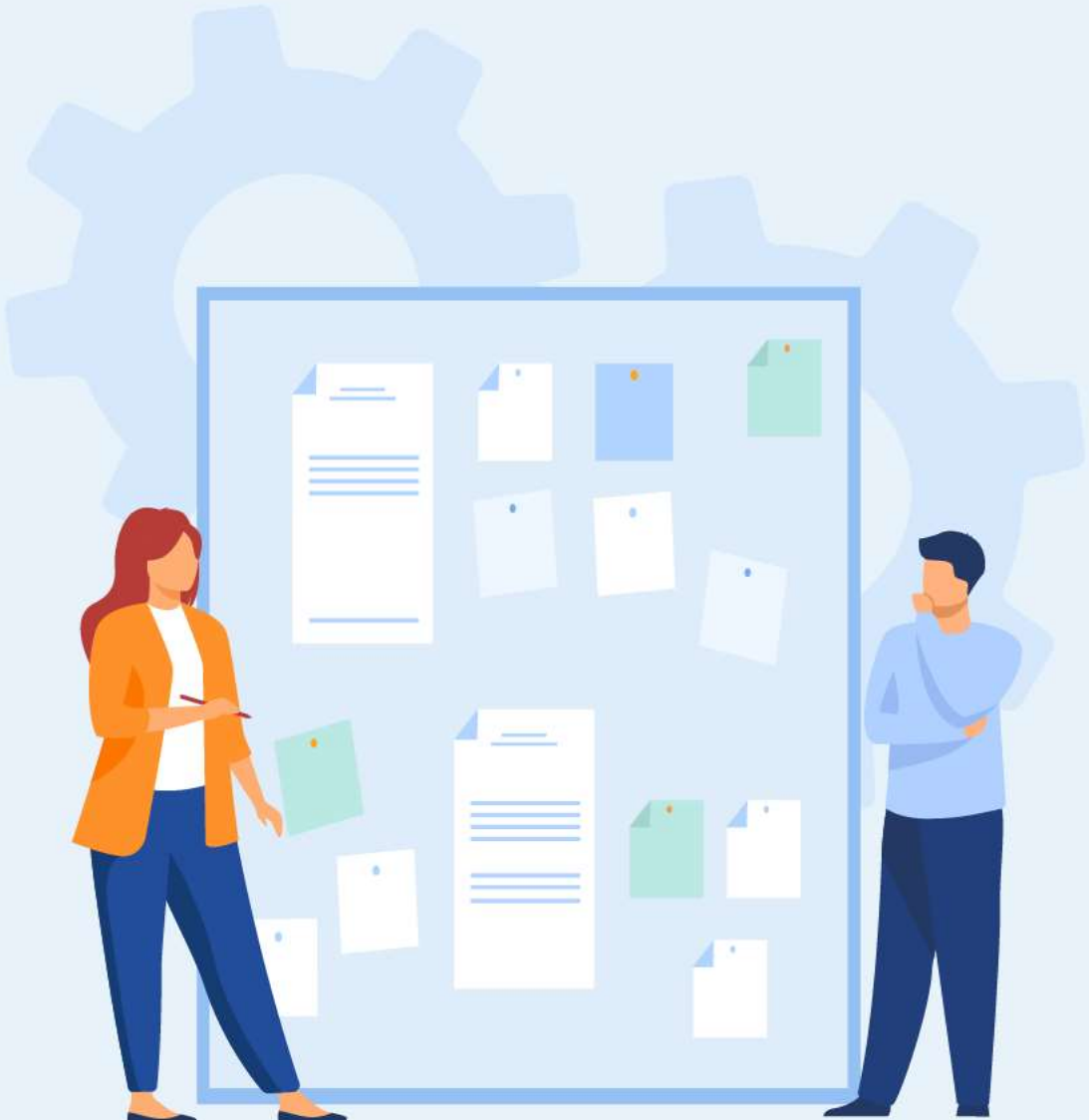
Compétences visées :

- Utiliser les sessions en PHP
- Restreindre l’utilisation de session

Recommandations clés :

- Bonne révision du résumé théorique

03 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur les sessions

2. Pour l’apprenant :

- Utiliser le résumé théorique
- Maitriser les notions de sessions

3. Conditions de réalisation :

- Support de résumé théorique accompagnant

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Définir les sessions
  - Alimenter la session par les données souhaitées
  - Authentification nécessaire pour accéder à la session

Activité 2
Utilisation des sessions

Exercice 1

Nous allons simuler un site marchant avec des pommes des poires et des bananes à acheter, on pourra les acheter une à une et supprimer toutes les pommes, toutes les poires ou toutes les bananes. Il sera également possible de vider complètement son panier.

Dans cet exercice il faut stocker les tableaux dans des variables de sessions.



Fig. : Affichage attendu

PARTIE 3

Copyright - Tout droit réservé - OFPPT

Activité 2
Utilisation des sessions

PARTIE 3

Réponses

- Fichier index.php

```php
<?php
   // Dans ce script nous utilisons les sessions
   session_start();

   // Chargement du caddie
   if (isset($_SESSION["caddie"])) {
      $caddie = $_SESSION["caddie"];
      } else {
      $caddie = array();
      $caddie["pommes"] = 0;
      $caddie["poires"] = 0;
      $caddie["bananes"] = 0;
   }
```

Fichier : index.php (Partie1)

```php
// on a demander à tout supprimer
   if (isset($_GET["action"])) {
      switch($_GET["action"]){
         case "destroy":
            session_destroy();
            header("Location: ".$_SERVER["PHP_SELF"]);
            $caddie["pommes"] = 0;
            $caddie["poires"] = 0;
            $caddie["bananes"] = 0;
            break;
      }

      // "sauvegarde" des modifications
      $_SESSION["caddie"] = $caddie;
      header("Location: ".$_SERVER["PHP_SELF"]);
      die();

   }
```

Fichier : index.php (Partie2)

Copyright - Tout droit réservé - OFPPT

Activité 2
Utilisation des sessions

PARTIE 3

Réponses

- Fichier index.php

```php
if (isset($_GET["raz"])) {
    // On a demandé à reinitialiser une valeur
    switch ($_GET["raz"]) {
        case "pomme":
            $caddie["pommes"] = 0;
            break;
        case "poire":
            $caddie["poires"]=0;
            break;
        case "banane":
            $caddie["bananes"]=0;
            break;
        }

    // "sauvegarde" des modifications
    $_SESSION["caddie"] = $caddie;
    header("Location: ".$_SERVER["PHP_SELF"]);
    die();
}
```

Fichier : index.php (Partie3)

```php
if (isset($_GET["achat"])) {
    // On a demandé à incrémenter une valeur
    switch ($_GET["achat"]) {
        case "pomme":
            $caddie["pommes"]++;
            break;
        case "poire":
            $caddie["poires"]++;
            break;
        case "banane":
            $caddie["bananes"]++;
            break;
    }
    // "sauvegarde" des modifications
    $_SESSION["caddie"] = $caddie;
    header("Location: ".$_SERVER["PHP_SELF"]);
    die();
}
```

Fichier : index.php (Partie4)

Copyright - Tout droit réservé - OFPPT

Réponses

- Fichier index.php

```
?>
Nb Pommes:<?php echo $caddie["pommes"];?><br />
<a href="<?php echo $_SERVER["PHP_SELF"];?>?achat=pomme">Acheter une pomme</a>
<a href="<?php echo $_SERVER["PHP_SELF"];?>?raz=pomme">RAZ</a><br />
<br />
Nb Poires:<?php echo $caddie["poires"];?><br />
<a href="<?php echo $_SERVER["PHP_SELF"];?>?achat=poire">Acheter une poire</a>
<a href="<?php echo $_SERVER["PHP_SELF"];?>?raz=poire">RAZ</a><br />
<br />
Nb Bananes:<?php echo $caddie["bananes"];?><br />
<a href="<?php echo $_SERVER["PHP_SELF"];?>?achat=banane">Acheter une banane</a>
<a href="<?php echo $_SERVER["PHP_SELF"];?>?raz=banane">RAZ</a><br />

<br/>
<a href="<?php echo $_SERVER["PHP_SELF"];?>?action=destroy">Détruire la session</a><br/>
```

Fichier : index.php (Partie5)

Copyright - Tout droit réservé - OFPPT

Activité 2
Utilisation des sessions

Exercice 2 :

L'objectif et de créer trois pages PHP:

- **login.php** : qui authentifie l'utilisateur. Elle contient un formulaire renfermant une zone de texte, une zone de mot de passe et un bouton d'envoi.
- **session.php** : qui représente la page à accès limité. Aucun visiteur n'a le droit de voir son contenu s'il n a pas été authentifié par la page login.php.
- **deconnexion.php** : est la page qui permet de déconnecter le client (détruire la session) et rediriger le navigateur vers la page login.php.

Si le client tente d'accéder directement à la page **session.php** alors qu'il n'est pas authentifié, il sera aussitôt redirigé vers la page **login.php**. S'il fournit un bon login et un bon mot de passe alors il sera redirigé vers la page **session.php** qu'il a désormais le droit de consulter.

Pour simplifier, nous allons définir statiquement le bon login qui est "user" et le bon mot de passe qui est "1234".

Copyright - Tout droit réservé - OFPPT

Réponses

- Fichier login.php

```php
<?php
  session_start();
  @$login=$_POST["login"];
  @$pass=$_POST["pass"];
  @$valider=$_POST["valider"];
  $bonLogin="user";
  $bonPass="1234";
  $erreur="";
  if(isset($valider)){
    if($login==$bonLogin && $pass==$bonPass){
      $_SESSION["autoriser"]="oui";
      header("location:session.php");
    }
    else
      $erreur="Mauvais login ou mot de passe!";
  }
?>
```

Fichier : login.php (Partie1)

```html
<!DOCTYPE html>
<html>
  <body onLoad="document.fo.login.focus()">
    <h1>Authentification</h1>
    <div class="erreur"><?php echo $erreur ?></div>
    <form name="fo" method="post" action="">
      <input type="text" name="login" placeholder="Login" /><br />
      <input type="password" name="pass" placeholder="Mot de passe" /><br />
      <input type="submit" name="valider" value="S'authentifier" />
    </form>
  </body>
</html>
```

Fichier : login.php (Partie2)

Copyright - Tout droit réservé - OFPPT

Activité 2
Utilisation des sessions

PARTIE 3

Réponses

- Fichier session.php

```php
<?php
  session_start();
  if($_SESSION["autoriser"]!="oui"){
    header("location:login.php");
    exit();
  }
  if(date("H")<18)
    $bienvenue="Bonjour et bienvenue dans votre espace personnel";
  else
    $bienvenue="Bonsoir et bienvenue dans votre espace personnel";
?>
```

Fichier : session.php (Partie1)

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <style>
      *{
        font-family:arial;
      }
      body{
        margin:20px;
      }
      a{
        color:#DD7700;
        text-decoration:none;
      }
      a:hover{
        text-decoration:underline;
      }
    </style>
  </head>
  <body onLoad="document.fo.login.focus()">
    <h1><?php echo $bienvenue?></h1>
    [ <a href="deconnexion.php">Se déconnecter</a> ]
  </body>
</html>
```

Fichier : session.php (Partie2)

Copyright - Tout droit réservé - OFPPT

Réponses

- Fichier deconnexion.php

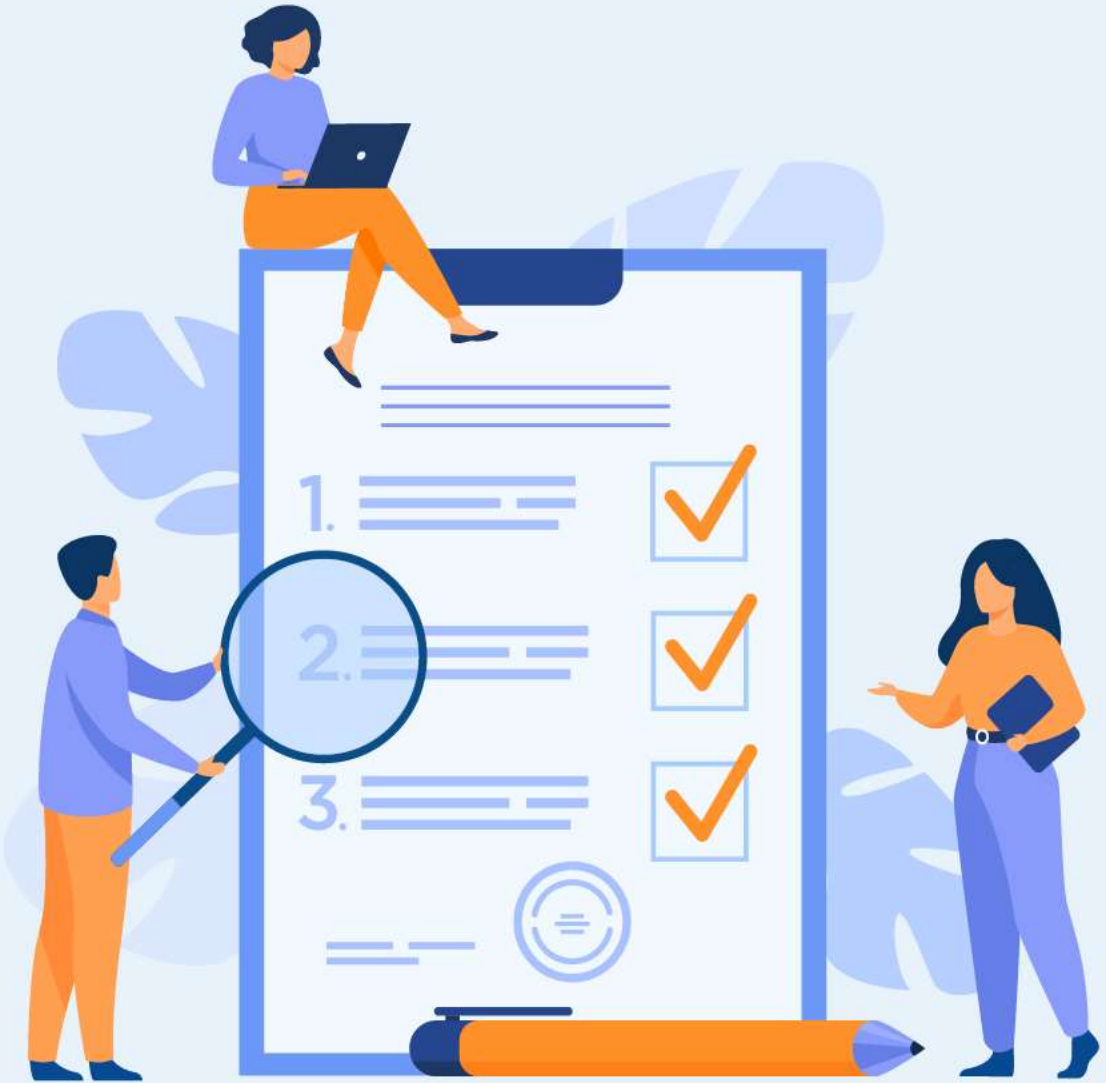
```php
<?php
   session_start();
   session_destroy();
   header("location:login.php");
?>
```

Fichier : deconnexion.php

Copyright - Tout droit réservé - OFPPT

Activité 3

Gestion des données avec MySQL



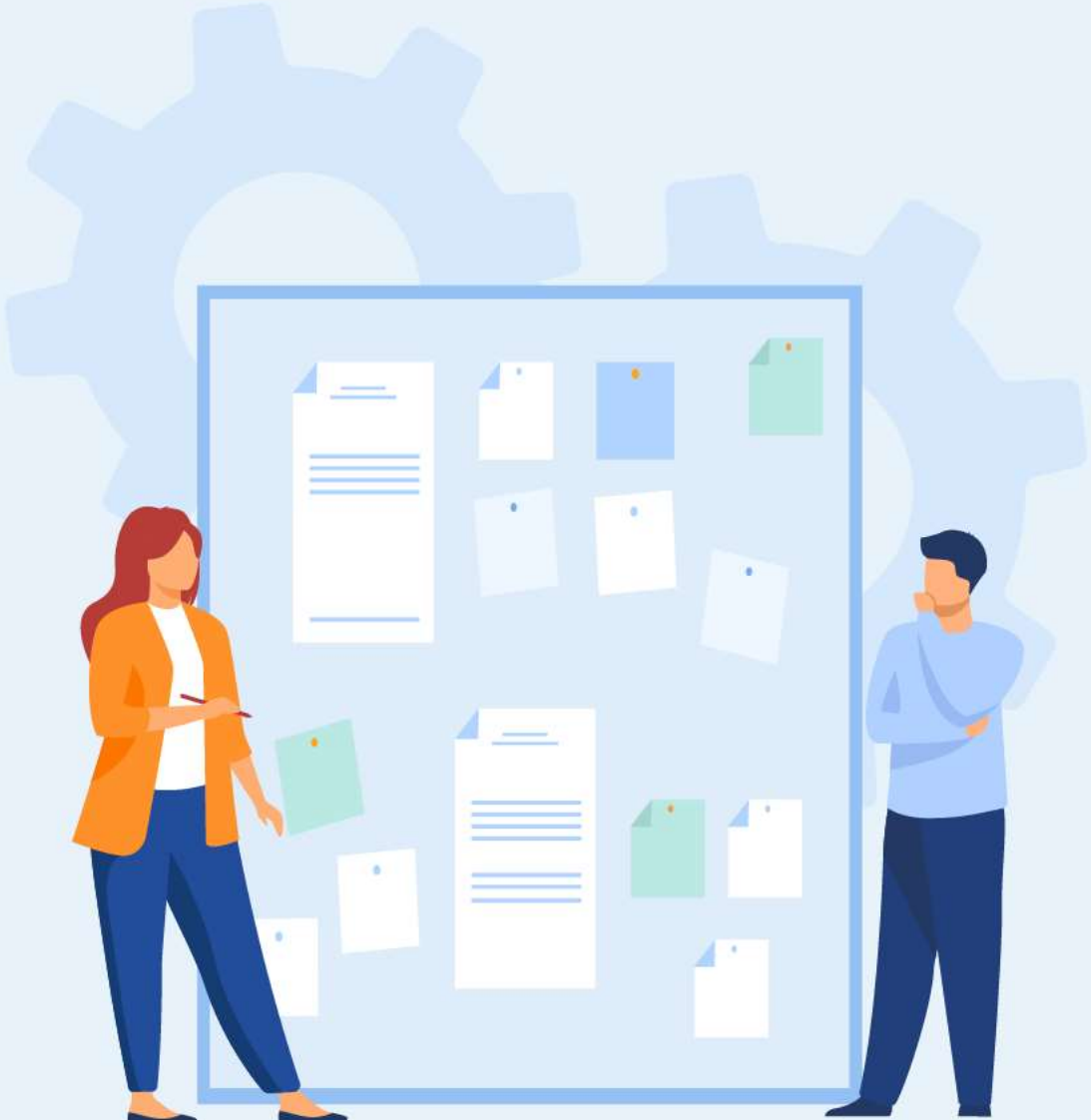
Compétences visées :

- Utilisation de gestion des données (CRUD)
- Connexion à la base de données MySQL
- Maitriser l'extension PHP Data Objects (PDO)

Recommandations clés :

- Bonne révision du résumé théorique
- Lire attentivement les consignes des exercices

08 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur la manipulation des données

2. Pour l’apprenant :

- Utiliser le résumé théorique
- Maitriser les notions de CRUD

3. Conditions de réalisation :

- Support de résumé théorique accompagnant
- Base de donnée MySQL
- L'extension PHP Data Objects (PDO)

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Appeler différent pages PHP
  - Établir une connexion avec MySQL
  - Écrire, lire et supprimer les données de la base de données

Téléchargement et Installation de MySQL

- Étape 1 : Télécharger MYSQL depuis le site officiel https://dev.mysql.com/downloads/workbench/
- Étape 2 : Suivre les étapes d'installation par défaut.

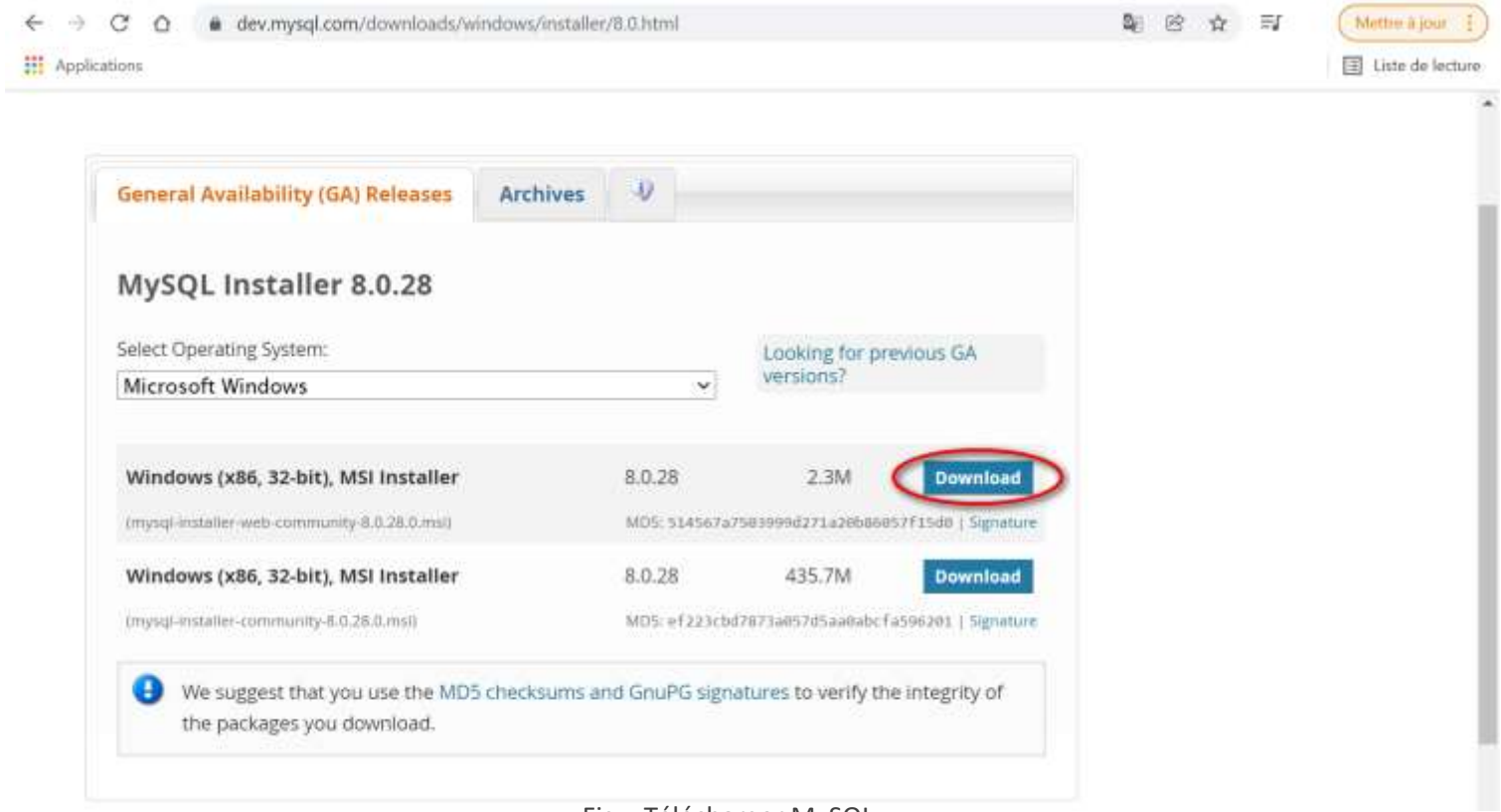


Fig. : Télécharger MySQL

Copyright - Tout droit réservé - OFPPT

Activité 3
Gestion des données avec MySQL

Exercice

L'objectif et de créer, lire, mettre à jour et supprimer (CRUD) en php les données d'une table mysql en utilisant Php Data Object (pdo) :

- La Base de donnée « mydb » contient une table « etudiant »
- La table « etudiant » contient trois colonnes « id », « nom » et « email ».
- create.php : permet la création d'un nouveau étudiant dans la base de données.
- edit.php : permet la mise à jour d'un étudiant dans la base de données.
- delete.php : permet la suppression d'un étudiant dans la base de données (Il faut avoir un message de confirmation).
- db.php : permet la connexion à la base de données.
- index.php : permet de traiter les données.
- header.php : permet d'afficher 2 lien:
    - Accueil : Affiche un tableau avec toute la liste des étudiants présent dans la base de données.
    - Créer un étudiant : permet la création d'un étudiant dans la base de données (Il faut avoir un message de confirmation).

PARTIE 3

Copyright - Tout droit réservé - OFPPT

Exercice (suite)

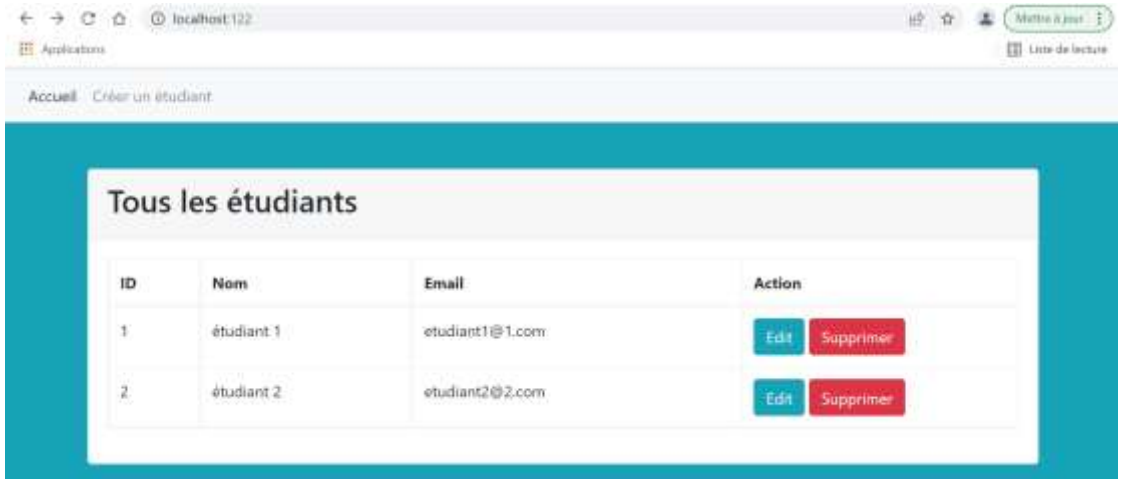


Fig. : écran Accueil

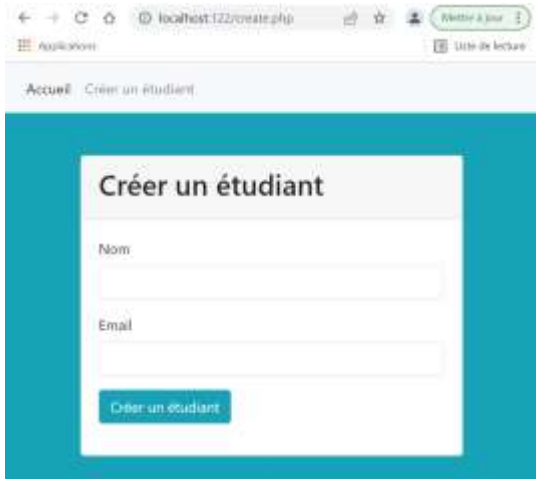


Fig. : écran créer un étudiant

PARTIE 3

Copyright - Tout droit réservé - OFPPT

Réponses

Base de données : CREATE TABLE `mydb`.`etudiant` (id int(11) auto_increment primary key,nom varchar(30) not null,email varchar(30) not null);

Fichier : db.php

```php
<?php
   $dsn = 'mysql:host=localhost;dbname=mydb';
   $username = 'root';
   $password = 'root';
   $options = [];
   try {
   $connection = new PDO($dsn, $username, $password, $options);
   }
   catch(PDOException $e) {
   }
```

Fichier : db.php

Copyright - Tout droit réservé - OFPPT

PARTIE 3

Réponses

```
<!doctype html>
<html lang="fr">
  <head>
    <title>Bonjour</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css"
integrity="sha384-PsH8R72JQ3SOdhVi3uxftmaW6Vc51MKb0q5P2rRUpPvrszuE4W1povHYgTpBfshb" crossorigin="anonymous">
  </head>
  <body class="bg-info">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="/">Accueil <span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="create.php">Créer un étudiant</a>
        </li>
      </ul>
    </div>
  </nav>
```

Fichier : header.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   require 'db.php';
   $message = '';
   if (isset ($_POST['nom'])  && isset($_POST['email']) ) {
      $nom = $_POST['nom'];
      $email = $_POST['email'];
      $sql = 'INSERT INTO etudiant(nom, email) VALUES(:nom, :email)';
      $statement = $connection->prepare($sql);
      if ($statement->execute([':nom' => $nom, ':email' => $email])) {
         $message = 'Donnée créée avec succès';
      }
   }
?>
```

Fichier : create.php (Partie 1)

Copyright - Tout droit réservé - OFPPT

Réponses

Fichier : create.php (Partie 2)

```php
<?php require 'header.php'; ?>
<div class="container">
 <div class="card mt-5">
  <div class="card-header">
   <h2>Créer un étudiant</h2>
  </div>
  <div class="card-body">
   <?php if(!empty($message)): ?>
    <div class="alert alert-success">
     <?= $message; ?>
    </div>
   <?php endif; ?>
   <form method="post">
    <div class="form-group">
     <label for="nom">Nom</label>
     <input type="text" name="nom" id="nom" class="form-control">
    </div>
    <div class="form-group">
     <label for="email">Email</label>
     <input type="email" name="email" id="email" class="form-control">
    </div>
    <div class="form-group">
     <button type="submit" class="btn btn-info">Créer un étudiant</button>
    </div>
   </form>
  </div>
 </div>
</div>
```


Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   require 'db.php';
   $id = $_GET['id'];
   $sql = 'SELECT * FROM etudiant WHERE id=:id';
   $statement = $connection->prepare($sql);
   $statement->execute([':id' => $id ]);
   $person = $statement->fetch(PDO::FETCH_OBJ);
   if (isset ($_POST['nom'])  && isset($_POST['email']) ) {
      $nom = $_POST['nom'];
      $email = $_POST['email'];
      $sql = 'UPDATE etudiant SET nom=:nom, email=:email WHERE id=:id';
      $statement = $connection->prepare($sql);
      if ($statement->execute([':nom' => $nom, ':email' => $email, ':id' => $id])) {
         header("Location: /");
      }
   }
?>
```

Fichier : edit.php (Partie 1)

Copyright - Tout droit réservé - OFPPT

PARTIE 3

Réponses

Fichier : edit.php (Partie 2)

```php
<?php require 'header.php'; ?>
<div class="container">
  <div class="card mt-5">
    <div class="card-header">
      <h2>Mise à jour d'un étudiant</h2>
    </div>
    <div class="card-body">
      <?php if(!empty($message)): ?>
        <div class="alert alert-success">
          <?= $message; ?>
        </div>
      <?php endif; ?>
      <form method="post">
        <div class="form-group">
          <label for="nom">Nom</label>
          <input value="<?= $person->nom; ?>" type="text" name="nom" id="nom" class="form-control">
        </div>
        <div class="form-group">
          <label for="email">Email</label>
          <input type="email" value="<?= $person->email; ?>" name="email" id="email" class="form-control">
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-info">Mise à jour d'un étudiant</button>
        </div>
      </form>
    </div>
  </div>
</div>
```

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
    require 'db.php';
    $id = $_GET['id'];
    $sql = 'DELETE FROM etudiant WHERE id=:id';
    $statement = $connection->prepare($sql);
    if ($statement->execute([':id' => $id])) {
        header("Location: /");
    }
```

Fichier : delete.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
        require 'db.php';
        $sql = 'SELECT * FROM etudiant';
        $statement = $connection->prepare($sql);
        $statement->execute();
        $etudiant = $statement->fetchAll(PDO::FETCH_OBJ);
 ?>
```

Fichier : index.php (Partie 1)

Copyright - Tout droit réservé - OFPPT

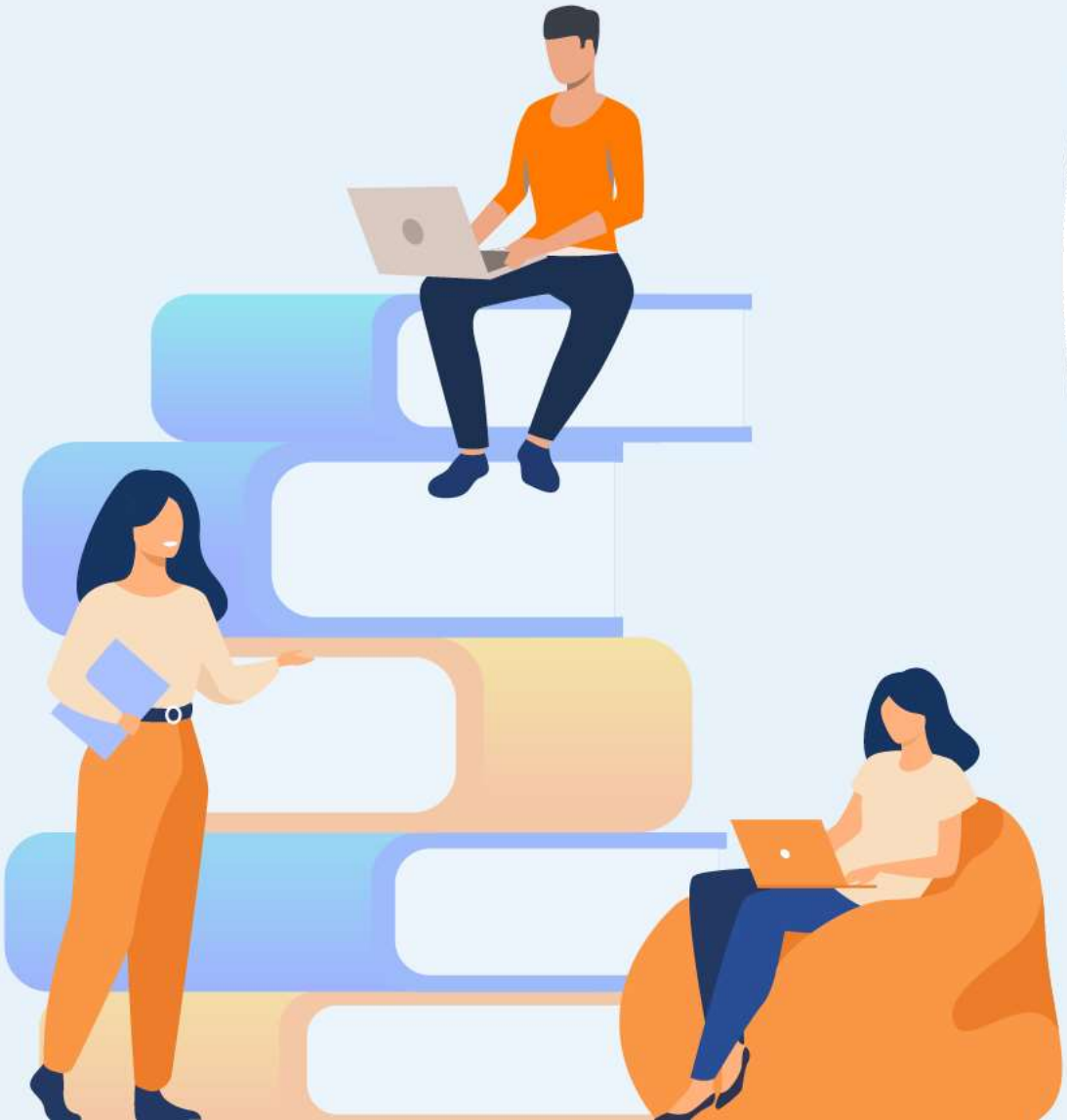
PARTIE 3

Réponses

Fichier : index.php (Partie 2)

```php
<?php require 'header.php'; ?>
<div class="container">
  <div class="card mt-5">
    <div class="card-header">
      <h2>Tous les étudiants</h2>
    </div>
    <div class="card-body">
      <table class="table table-bordered">
        <tr>
          <th>ID</th>
          <th>Nom</th>
          <th>Email</th>
          <th>Action</th>
        </tr>
        <?php foreach($etudiant as $person): ?>
          <tr>
            <td><?= $person->id; ?></td>
            <td><?= $person->nom; ?></td>
            <td><?= $person->email; ?></td>
            <td>
              <a href="edit.php?id=<?= $person->id ?>" class="btn btn-info">Edit</a>
              <a onclick="return confirm('Êtes-vous sûr de vouloir supprimer cet enregistrement?')" href="delete.php?id=<?= $person->id ?>" class='btn btn-danger'>Supprimer</a>
            </td>
          </tr>
        <?php endforeach; ?>
      </table>
    </div>
  </div>
</div>
```

Copyright - Tout droit réservé - OFPPT



PARTIE 4

Réaliser un site web avec l’architecture MVC

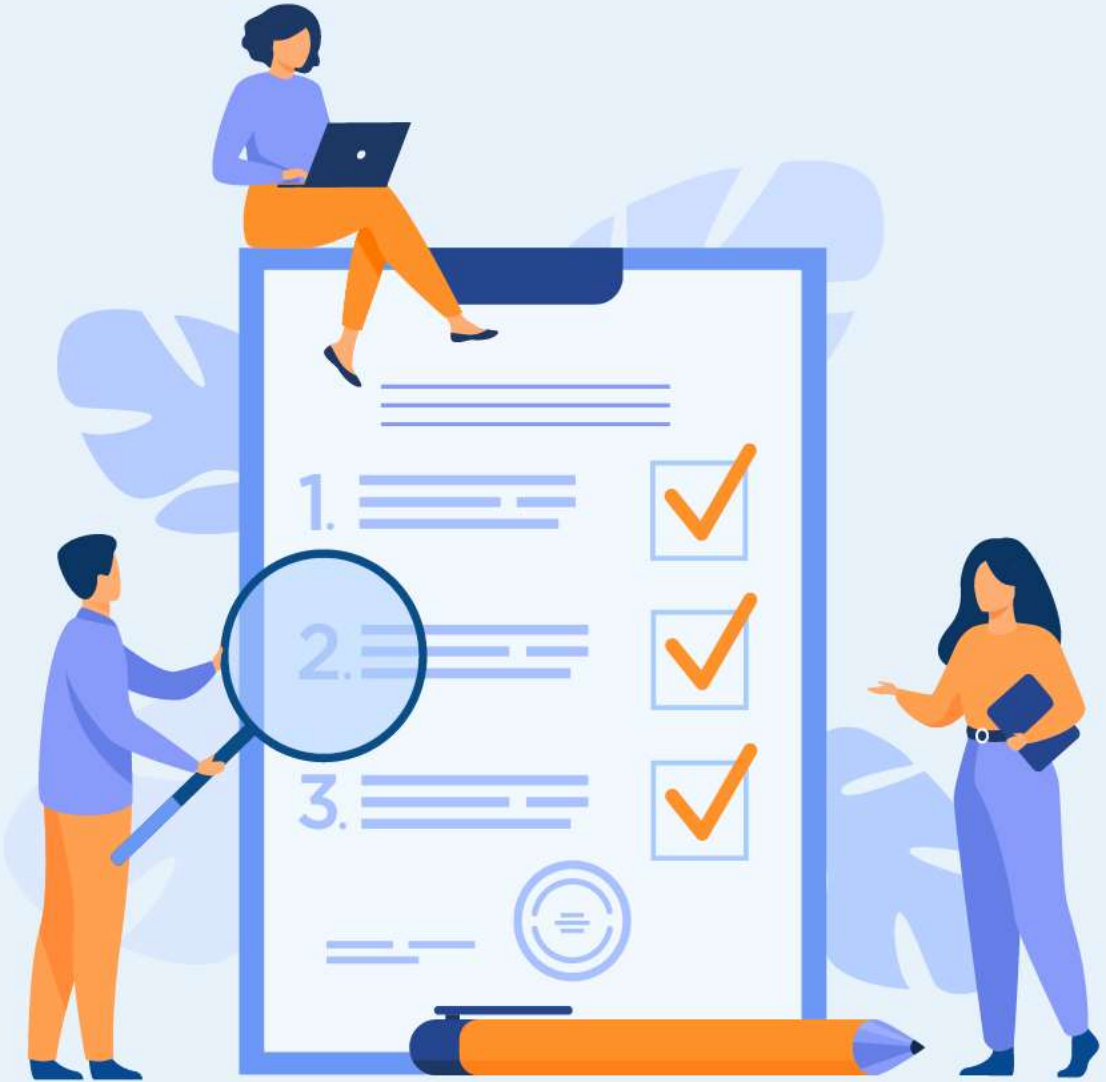
Dans ce module, vous allez :

- Maitriser les scripts d’accès aux données
- Réaliser l’architecture MVC
- Créer une API RESTful

08 heures

Activité 1

Vers une architecture MVC



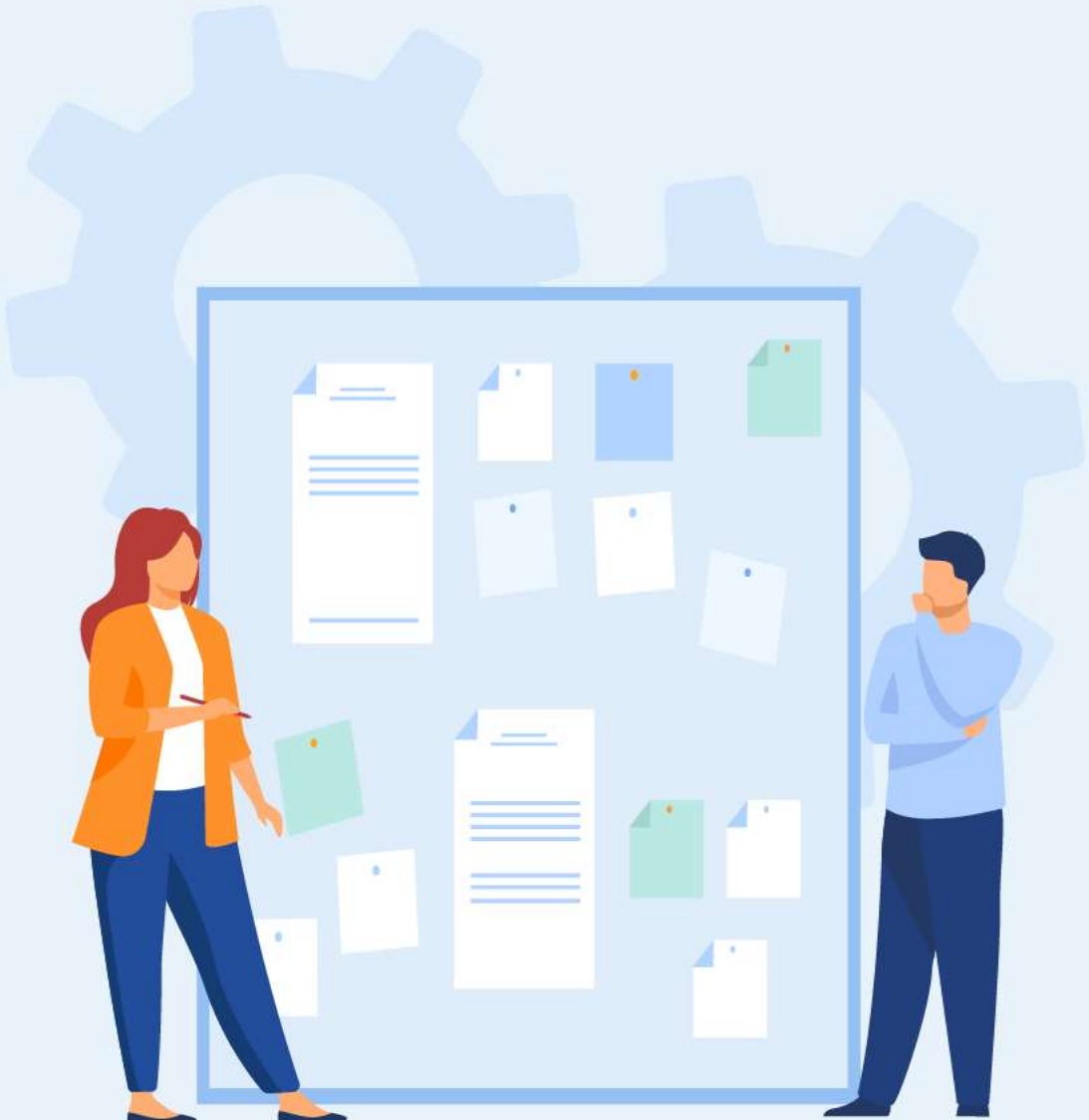
Compétences visées :

- Mettre en place l'architecture MVC
- Réaliser une API RESTful
- Mettre en place les connaissances PHP de base

Recommandations clés :

- Bonne révision du résumé théorique
- Lire attentivement les consignes des exercices

08 heures



CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur l’architecture MVC

2. Pour l’apprenant :

- Utiliser le résumé théorique
- Maitriser les notions de MVC
- Utiliser un paramètre comme jeton d’authentification

3. Conditions de réalisation :

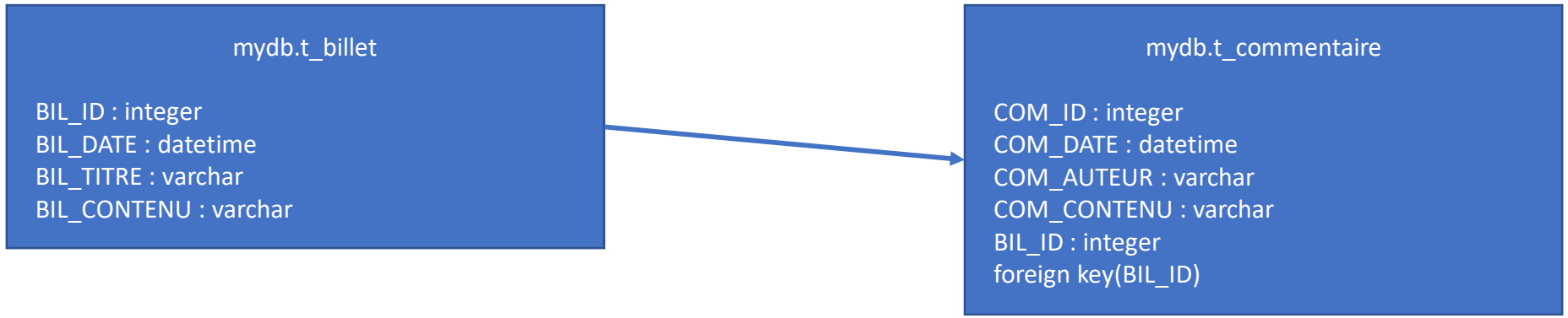
- Base de donnée MySQL
- L'extension PHP Data Objects (PDO)
- Structure de base de MVC

4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Établir une connexion avec MySQL
  - Écrire, lire et supprimer les données de la base de données
  - Utiliser les notions MVC
  - Créer une API RESTful

Exercice

1. Créer une base de données qui se compose de deux tables, l'une stockant les billets (articles) du blog et l'autre les commentaires associés aux articles.



Copyright - Tout droit réservé - OFPPT

Réponses

```
drop table if exists T_COMMENTAIRE;
drop table if exists T_BILLET;
create table T_BILLET (
  BIL_ID integer primary key auto_increment,
  BIL_DATE datetime not null,
  BIL_TITRE varchar(100) not null,
  BIL_CONTENU varchar(400) not null
) ENGINE=INNODB CHARACTER SET utf8 COLLATE utf8_general_ci;
create table T_COMMENTAIRE (
  COM_ID integer primary key auto_increment,
  COM_DATE datetime not null,
  COM_AUTEUR varchar(100) not null,
  COM_CONTENU varchar(200) not null,
  BIL_ID integer not null,
  constraint fk_com_bil foreign key(BIL_ID) references T_BILLET(BIL_ID)
) ENGINE=INNODB CHARACTER SET utf8 COLLATE utf8_general_ci;
```


Copyright - Tout droit réservé - OFPPT

Exercice

2. Insérer des données pour avoir le résultat suivant:

	BIL_ID	BIL_DATE	BIL_TITRE	BIL_CONTENU
▶	1	2022-02-21 00:25:42	Premier billet	Bonjour monde ! Ceci est le premier billet sur mon blog.
	2	2022-02-21 00:25:42	Au travail	Il faut enrichir ce blog dès maintenant.

Fig. : Données de la table t_billet

	COM_ID	COM_DATE	COM_AUTEUR	COM_CONTENU	BIL_ID
▶	1	2022-02-21 00:25:42	A. Nonyme	Bravo pour ce début	1
	2	2022-02-21 00:25:42	Moi	Merci ! Je vais continuer sur ma lancée	1

Fig. : Données de la table t_commentaire

Copyright - Tout droit réservé - OFPPT

Réponses

```
insert into T_BILLET(BIL_DATE, BIL_TITRE, BIL_CONTENU) values
(NOW(), 'Premier billet', 'Bonjour monde ! Ceci est le premier billet sur mon blog.');
insert into T_BILLET(BIL_DATE, BIL_TITRE, BIL_CONTENU) values
(NOW(), 'Au travail', 'Il faut enrichir ce blog dès maintenant.');

insert into T_COMMENTAIRE(COM_DATE, COM_AUTEUR, COM_CONTENU, BIL_ID) values
(NOW(), 'A. Nonyme', 'Bravo pour ce début', 1);
insert into T_COMMENTAIRE(COM_DATE, COM_AUTEUR, COM_CONTENU, BIL_ID) values
(NOW(), 'Moi', 'Merci ! Je vais continuer sur ma lancée', 1);
```

Fig. : Requêtes d'insertion

Copyright - Tout droit réservé - OFPPT

Exercice

3. Créer un répertoire Modele qui contient le code d'accès aux données :
    - Modele.php : Classe abstraite Modèle. Centralise les services d'accès à une base de données et utilise l'API PDO.
    - Billet.php : Permet d'afficher les informations demandées de la table t_billet. Contient deux fonctions:
        - getBillets() : Renvoie la liste des billets du blog.
        - getBillet($idBillet) : Renvoie les informations sur un billet.
    - Commentaire.php : Une classe sur le même modèle que la classe Billet. Contient deux fonctions:
        - getCommentaires($idBillet) : Renvoie la liste des commentaires associés à un billet.
        - ajouterCommentaire($auteur, $contenu, $idBillet) : Ajoute un commentaire dans la base de données.

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   abstract class Modele {
      private $bdd;
      protected function executerRequete($sql, $params = null) {
         if ($params == null) {
            $resultat = $this->getBdd()->query($sql);
         }
         else {
            $resultat = $this->getBdd()->prepare($sql);
            $resultat->execute($params);
         }
         return $resultat;
      }
      private function getBdd() {
         if ($this->bdd == null) {
            $this->bdd = new PDO('mysql:host=localhost;dbname=mydb;charset=utf8',
                  'root', 'root',
                  array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
         }
         return $this->bdd;
      }
   }
```

Fig. : Modele.php

Copyright - Tout droit réservé - OFPPT

PARTIE 4

Réponses

```php
<?php
  require_once 'Modele/Modele.php';
  class Billet extends Modele {
    public function getBillets() {
      $sql = 'select BIL_ID as id, BIL_DATE as date,'
           . ' BIL_TITRE as titre, BIL_CONTENU as contenu from T_BILLET'
           . ' order by BIL_ID desc';
      $billets = $this->executerRequete($sql);
      return $billets;
    }
    public function getBillet($idBillet) {
      $sql = 'select BIL_ID as id, BIL_DATE as date,'
           . ' BIL_TITRE as titre, BIL_CONTENU as contenu from T_BILLET'
           . ' where BIL_ID=?';
      $billet = $this->executerRequete($sql, array($idBillet));
      if ($billet->rowCount() > 0)
        return $billet->fetch();  // Accès à la première ligne de résultat
      else
        throw new Exception("Aucun billet ne correspond à l'identifiant '$idBillet'");
    }
  }
```

Fig. : Billet.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   require_once 'Modele/Modele.php';
   class Commentaire extends Modele {
      public function getCommentaires($idBillet) {
         $sql = 'select COM_ID as id, COM_DATE as date,'
               . ' COM_AUTEUR as auteur, COM_CONTENU as contenu from T_COMMENTAIRE'
               . ' where BIL_ID=?';
         $commentaires = $this->executerRequete($sql, array($idBillet));
         return $commentaires;
      }

      public function ajouterCommentaire($auteur, $contenu, $idBillet) {
         $sql = 'insert into T_COMMENTAIRE(COM_DATE, COM_AUTEUR, COM_CONTENU,
BIL_ID)'
              . ' values(?, ?, ?, ?)';
         $date = date(DATE_W3C);  // Récupère la date courante
         $this->executerRequete($sql, array($date, $auteur, $contenu, $idBillet));
      }
   }
```

Fig. : Commentaire.php

Copyright - Tout droit réservé - OFPPT

Exercice

4. Créer un répertoire Vue regroupant le code d'affichage :

- gabarit.php : (template en anglais) Ce modèle contiendra tous les éléments communs et permettra d'ajouter les éléments spécifiques à chaque vue.
- vueAccueil.php : Permet d'afficher la l'ensemble des titres billets (lien qui renvoie à la vueBillet.php) avec leurs date.
- vueBillet.php : Permet d'afficher l'ensemble des commentaires du Billet sélectionné. Permet d'ajouter un nouveau commentaire pour ce billet.
- vueErreur.php : Permet d'afficher un message d'erreur. Exemples :
  - http://localhost:122/index.php?action=billet&id=UIF => Affiche : Identifiant de billet non valide (toujours avec l'entête et le pied de la page)
  - http://localhost:122/index.php?action=billet => Affiche : Paramètre 'id' absent (toujours avec l'entête et le pied de la page)
- Vue.php : La classe Vue dont le rôle sera de gérer la génération des vues.

Copyright - Tout droit réservé - OFPPT

Réponses

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="Contenu/style.css" />
    <title><?= $titre ?></title>
  </head>
  <body>
    <div id="global">
      <header>
        <a href="index.php"><h1 id="titreBlog">TP Blog</h1></a>
        <p>Je vous souhaite la bienvenue sur ce blog.</p>
      </header>
      <div id="contenu">
        <?= $contenu ?>
      </div>
      <footer id="piedBlog">
        Blog réalisé dans le cadre d'un TP.
      </footer>
    </div>
  </body>
</html>
```

Fig. : gabarit.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php $this->titre = "Mon Blog"; ?>

<?php foreach ($billets as $billet): ?>
<article>
    <header>
        <a href="<?= "index.php?action=billet&id=" . $billet['id'] ?>">
            <h1 class="titreBillet"><?= $billet['titre'] ?></h1>
        </a>
        <time><?= $billet['date'] ?></time>
    </header>
    <p><?= $billet['contenu'] ?></p>
</article>
<hr />
<?php endforeach; ?>
```

Fig. : vueAccueil.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php $this->titre = "Mon Blog - " . $billet['titre']; ?>
<article>
  <header>
    <h1 class="titreBillet"><?= $billet['titre'] ?></h1>
    <time><?= $billet['date'] ?></time>
  </header>
  <p><?= $billet['contenu'] ?></p>
</article>
<hr />
<header>
  <h1 id="titreReponses">Réponses à <?= $billet['titre'] ?></h1>
</header>
<?php foreach ($commentaires as $commentaire): ?>
<p><?= $commentaire['auteur'] ?> dit :</p>
<p><?= $commentaire['contenu'] ?></p>
<?php endforeach; ?>
<hr />
<form method="post" action="index.php?action=commenter">
  <input id="auteur" name="auteur" type="text" placeholder="Votre pseudo" required /><br />
  <textarea id="txtCommentaire" name="contenu" rows="4" placeholder="Votre commentaire" required></textarea><br />
  <input type="hidden" name="id" value="<?= $billet['id'] ?>" />
  <input type="submit" value="Commenter" />
</form>
```

Fig. : vueBillet.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php $this->titre = "Mon Blog - Erreur !"; ?>

<p><?= $msgErreur ?></p>
```

Fig. : vueErreur.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
   class Vue {
      private $fichier;
      private $titre;
      public function __construct($action) {
         $this->fichier = "Vue/vue" . $action . ".php";
      }
      public function generer($donnees) {
         $contenu = $this->genererFichier($this->fichier, $donnees);
         $vue = $this->genererFichier('Vue/gabarit.php', array('titre' => $this->titre, 'contenu' => $contenu));
         echo $vue;
      }
      private function genererFichier($fichier, $donnees) {
         if (file_exists($fichier)) {
            extract($donnees);
            ob_start();
            require $fichier;
            return ob_get_clean();
         }
         else {
            throw new Exception("Fichier '$fichier' introuvable");
         }
      }
   }
```

Fig. : Vue.php

Copyright - Tout droit réservé - OFPPT

Exercice

5. Créer un répertoire Controleur regroupant le code d'affichage :

- ControleurAccueil.php : Est une classe qui permet de gérer l'accueil. Contient deux fonctions:
  - __construct() : Crée un nouveau objet Billet
  - accueil() : Affiche la liste de tous les billets du blog
- ControleurBillet.php : Permet de gérer l'affichage d'un billet. Contient deux fonctions:
  - __construct() : Crée deux nouveau objets Billet et Commentaire.
  - billet($idBillet) : Affiche les détails sur un billet
  - commenter($auteur, $contenu, $idBillet) : Ajoute un commentaire à un billet
- Routeur.php : le routeur dont la méthode principale est d'analyser la requête entrante pour déterminer l'action à entreprendre. Contient 4 fonctions:
  - __construct() : Crée deux nouveau objets ControleurAccueil et ControleurBillet.
  - routerRequete() : Route une requête entrante : exécution l'action associée
  - erreur($msgErreur) : Affiche une erreur
  - getParametre($tableau, $nom) : Recherche un paramètre dans un tableau

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
require_once 'Modele/Billet.php';
require_once 'Vue/Vue.php';
class ControleurAccueil {
   private $billet;
   public function __construct() {
      $this->billet = new Billet();
   }
 public function accueil() {
      $billets = $this->billet->getBillets();
      $vue = new Vue("Accueil");
      $vue->generer(array('billets' => $billets));
   }
}
```

Fig. : ControleurAccueil.php

Copyright - Tout droit réservé - OFPPT

Réponses

```php
<?php
require_once 'Modele/Billet.php';
require_once 'Modele/Commentaire.php';
require_once 'Vue/Vue.php';
class ControleurBillet {
    private $billet;
    private $commentaire;
    public function __construct() {
        $this->billet = new Billet();
        $this->commentaire = new Commentaire();
    }
    public function billet($idBillet) {
        $billet = $this->billet->getBillet($idBillet);
        $commentaires = $this->commentaire->getCommentaires($idBillet);
        $vue = new Vue("Billet");
        $vue->generer(array('billet' => $billet, 'commentaires' => $commentaires));
    }
    public function commenter($auteur, $contenu, $idBillet) {
        $this->commentaire->ajouterCommentaire($auteur, $contenu, $idBillet);
        $this->billet($idBillet);
    }
}
```

Fig. : ControleurBillet.php

Copyright - Tout droit réservé - OFPPT

PARTIE 4

Réponses

```php
public function routerRequete() {
    try {
        if (isset($_GET['action'])) {
            if ($_GET['action'] == 'billet') {
                $idBillet = intval($this->getParametre($_GET, 'id'));
                if ($idBillet != 0) { $this->ctrlBillet->billet($idBillet); }
                else throw new Exception("Identifiant de billet non valide");
            }
            else if ($_GET['action'] == 'commenter') {
                $auteur = $this->getParametre($_POST, 'auteur');
                $contenu = $this->getParametre($_POST, 'contenu');
                $idBillet = $this->getParametre($_POST, 'id');
                $this->ctrlBillet->commenter($auteur, $contenu, $idBillet);
            }
            else throw new Exception("Action non valide");
        }
        else { $this->ctrlAccueil->accueil();}
    }
    catch (Exception $e) { $this->erreur($e->getMessage()); }
}
```

Fig. : fonction routerRequete()

```php
<?php
    require_once 'Controleur/ControleurAccueil.php';
    require_once 'Controleur/ControleurBillet.php';
    require_once 'Vue/Vue.php';
    class Routeur {
        private $ctrlAccueil;
        private $ctrlBillet;
        public function __construct() {
            $this->ctrlAccueil = new ControleurAccueil();
            $this->ctrlBillet = new ControleurBillet();
        }
        //public function routerRequete() { ... }
        private function erreur($msgErreur) {
            $vue = new Vue("Erreur");
            $vue->generer(array('msgErreur' => $msgErreur));
        }
        private function getParametre($tableau, $nom) {
            if (isset($tableau[$nom])) { return $tableau[$nom]; }
            else throw new Exception("Paramètre '$nom' absent");
        }
    }
```

Fig. : Routeur.php

Copyright - Tout droit réservé - OFPPT

Exercice

6. Créer un répertoire Contenu regroupant le style:

- Style.css : Une feuille de style CSS est utilisée afin d'améliorer le rendu HTML

Copyright - Tout droit réservé - OFPPT

PARTIE 4

Réponses

```css
html, body {
   height: 100%;
}
body {
   color: #bfbfbf;
   background: black;
   font-family: 'Futura-Medium', 'Futura', 'Trebuchet MS', sans-serif;
}
h1 {
   color: white;
}
.titreBillet {
   margin-bottom : 0px;
}
```

Fig. : style.css (Partie 1)

```css
#global {
   min-height: 100%;
   background: #333534;
   width: 70%;
   margin: auto;
   text-align: justify;
   padding: 5px 20px;
}
#contenu {
   margin-bottom : 30px;
}
#titreBlog, #piedBlog {
   text-align: center;
}
#titreReponses {
   font-size : 100%;
}
#txtCommentaire {
   width: 50%;
}
```

Fig. : style.css (Partie 2)

Copyright - Tout droit réservé - OFPPT