Créer un PDF en PHP avec FPDF

1. Objectifs

- Comprendre les concepts de base de la création de PDF
- Savoir télécharger et installer la bibliothèque FPDF sur votre serveur.
- En suivant ce tutoriel, les développeurs PHP pourront créer rapidement et facilement des documents PDF personnalisés en utilisant la bibliothèque FPDF.



2. Contexte et Définitions

- FPDF est une librairie PHP qui permet de créer des fichiers PDF à partir de données générées dynamiquement en PHP. Pour utiliser FPDF, vous devez d'abord télécharger la librairie depuis le site officiel (http://www.fpdf.org/) et l'inclure dans votre script PHP.

3. Comprendre les concepts de base de la création de PDF

- La création de documents PDF implique plusieurs concepts de base tels que les dimensions de page, les polices, les couleurs, les images, etc. Comprendre ces concepts est essentiel pour créer des documents PDF professionnels et de qualité.
- Voici quelques exemples d'explications de ces concepts de base :
  - **Dimensions de page**: Les documents PDF peuvent être créés avec des dimensions de page spécifiques, telles que A4, A5, Letter, Legal, etc. Par exemple, pour créer un document PDF avec des dimensions de page A4, vous pouvez utiliser la méthode AddPage de la bibliothèque FPDF :

```
$pdf = new FPDF();
$pdf->AddPage('A4');
```

apcpedagogie.com

- **Polices** : Les polices sont utilisées pour afficher du texte dans un document PDF. Il existe plusieurs types de polices disponibles, telles que Times New Roman, Arial, Verdana, etc. Par exemple, pour définir la police Times New Roman dans un document PDF, vous pouvez utiliser la méthode SetFont de la bibliothèque FPDF :

```
$pdf = new FPDF();
$pdf->SetFont('Times', '', 12);
$pdf->Cell(0, 10, 'Bonjour le monde');
```

- **Couleurs**: Les documents PDF peuvent utiliser différentes couleurs pour afficher du texte, des lignes, des formes, etc. Les couleurs peuvent être définies en utilisant le modèle RVB (rouge, vert, bleu) ou le modèle de couleurs HTML. Par exemple, pour définir la couleur de fond d'une page en rouge, vous pouvez utiliser la méthode SetFillColor de la bibliothèque FPDF :

```
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFillColor(255, 0, 0);
$pdf->Rect(0, 0, 100, 100, 'F');
```

- **Images**: Les images peuvent être ajoutées à un document PDF pour afficher des graphiques, des photos, des logos, etc. Les images peuvent être stockées sur le serveur ou être des images générées dynamiquement. Par exemple, pour ajouter une image à un document PDF, vous pouvez utiliser la méthode Image de la bibliothèque FPDF :

```
$pdf = new FPDF();
$pdf->AddPage();
$pdf->Image('monimage.jpg', 10, 10, 50, 50);
```

- En comprenant ces concepts de base, les développeurs PHP peuvent créer des documents PDF de qualité en utilisant la bibliothèque FPDF.

4. Télécharger et installer la bibliothèque FPDF

- Télécharger la librairie sur le site de FPDF > choisissez la version la plus récente au format ZIP.
- Extrayez les fichiers de l'archive téléchargée.
- Copiez le dossier "**fpdf**" dans le répertoire de votre projet PHP.

apcpedagogie.com

- Incluez le fichier "fpdf.php" dans votre script PHP à l'aide de la fonction "require_once". Par exemple :

```
require_once('fpdf/fpdf.php');
```

- Une fois la bibliothèque FPDF téléchargée et installée, vous pouvez commencer à l'utiliser dans votre script PHP pour créer des documents PDF.

5. **Utiliser pour créer des documents PDF**

```
<?php
// Inclure le fichier de configuration
include_once('config.php');
$connexion=new connectionDB();
$pdo=$connexion->open();

// Inclure la bibliothèque FPDF
require('fpdf/mc_table.php');

// Créer une nouvelle instance de la classe FPDF
$pdf = new PDF_MC_Table();

// Ajouter une page
$pdf->AddPage();

// Définir la police et la taille du texte
$pdf->SetFont('Arial','B',16);

// Ajouter un titre
$pdf->Cell(0,10,'Liste des membres',0,1,'C');

// Ajouter un espace
$pdf->Ln(10);

// Définir les en-têtes de colonnes
$pdf->SetFont('Arial','B',12);
$pdf->Cell(20,10,'ID',1,0,'C');
$pdf->Cell(30,10,'Nom',1,0,'C');
$pdf->Cell(30,10,'Prénom',1,0,'C');
$pdf->Cell(80,10,'Adresse',1,1,'C');

// Récupérer les données de la base de données
```

apcpedagogie.com

```
$stmt = $pdo->query('SELECT id, first, last, adress FROM members');
$data = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Ajouter les données dans le tableau
$pdf->SetWidths(array(20, 30, 30, 80));

foreach($data as $row) {
  /*$pdf->Cell(20, 10, $row['id'], 1);
  $pdf->Cell(30, 10, $row['first'], 1);
  $pdf->Cell(30, 10, $row['last'], 1);
  $pdf->MultiCell(60, 10, $row['adress'], 1);*/
  $pdf->Row(array( $row['id'],$row['first'],$row['last'], $row['adress']));
}

// Positionnement à 1,5 cm du bas
$pdf->SetY(-20.1);
// Police Arial italique 8
$pdf->SetFont('Arial','I',8);
$pdf -> AliasNbPages();
// Numéro de page
$pdf->Cell(0,0,'Page '.$pdf->PageNo().'/{nb}',0,0,'C');


// Sortir le document PDF
$pdf->Output();


$connexion->close();
?>
```

6. Quelques méthodes

- Je vais utiliser l'exemple suivant pour analyser les méthodes utilisées dans la bibliothèque FPDF :

```
<?php
require('fpdf.php');
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Hello World !');
$pdf->Output();
```

apcpedagogie.com

```
?>
```

Dans cet exemple, la bibliothèque **FPDF** est incluse avec la fonction `require`. Ensuite, un nouvel objet PDF est créé avec `$pdf = new FPDF();`. Une nouvelle page est ajoutée avec la méthode `AddPage()`. La police de caractères est définie avec la méthode `SetFont()`, en utilisant la police "Arial" en gras et une taille de 16. Enfin, une cellule est ajoutée à la page avec la `méthode Cell()`, contenant le texte "Bonjour apcpedagogie!" et une largeur de 40 unités de mesure et une hauteur de 10 unités de mesure. Enfin, le document PDF est généré avec la méthode `Output()`.

- **La méthode `AddPage()`**
  - La méthode **AddPage()** de la bibliothèque FPDF permet d'ajouter une nouvelle page à un document PDF généré. Cette méthode ne prend pas de paramètre. Lorsqu'elle est appelée, une nouvelle page vierge est ajoutée au document en cours, en utilisant les mêmes dimensions et la même orientation que la page précédente.
  - La méthode **AddPage()** est souvent appelée après la création d'un objet PDF et avant l'ajout du contenu de la page.

- **La méthode `SetFont()`**
  - La **méthode SetFont()** de la bibliothèque FPDF permet de définir la police de caractères à utiliser pour afficher le texte dans un document PDF généré. Cette méthode prend trois paramètres :
    - **`$family`**: le nom de la police de caractères à utiliser. Il peut s'agir d'un nom de police standard (par exemple, "Arial", "Times", "Courier"), ou d'un nom de police défini par l'utilisateur (par exemple, si vous avez ajouté une police personnalisée à la bibliothèque FPDF).
    - **`$style`**: le style de la police à utiliser. Il peut s'agir d'une combinaison de plusieurs styles, tels que "**B**" pour mettre en gras, "**I**" pour mettre en italique, "**U**" pour souligner, "**D**" pour barrer, etc.
    - **`$size`**: la taille de la police en points.
  - Voici un exemple d'utilisation de la méthode **SetFont()**:

    ```
    $pdf->SetFont('Arial','B',16);
    ```

    - Dans cet exemple, la police de caractères "Arial" est définie avec un style gras et une taille de 16 points.

- Il est important de noter que la **méthode SetFont()** doit être appelée avant d'afficher le texte dans le document PDF. Si elle est appelée après l'affichage du texte, la police de caractères ne sera pas correctement appliquée.

- **La méthode `Cell`**

  - La méthode **Cell** de la classe PDF permet d'ajouter une cellule (ou une case) à un document PDF généré avec la bibliothèque FPDF. Les propriétés de la méthode **Cell** sont les suivantes :
    - **`$w`**: largeur de la cellule en unités de mesure (par défaut, millimètres).
    - **`$h`**: hauteur de la cellule en unités de mesure (par défaut, millimètres).
    - **`$txt`**: texte à insérer dans la cellule.
    - **`$border`**: indicateur de présence d'un cadre pour la cellule (1 pour afficher un cadre, 0 pour ne pas afficher de cadre).
    - **`$ln`** : indicateur de saut de ligne (1 pour sauter à la ligne suivante, 0 pour rester sur la même ligne).
    - **`$align`**: alignement horizontal du texte dans la cellule (valeurs possibles : "L" pour aligner le texte à gauche, "C" pour centrer le texte, "R" pour aligner le texte à droite).
    - **`$fill`** : indicateur de remplissage de la cellule (1 pour remplir la cellule avec une couleur, 0 pour ne pas remplir la cellule).
    - **`$link`** : URL ou identifiant de destination pour créer un lien hypertexte dans la cellule.

    ```
    $pdf->Cell(40, 10, 'Bonjour apcpedagogie!', 1, 1, 'C', 1);
    ```

  - Dans cet exemple, une cellule est créée avec une largeur de 40 mm et une hauteur de 10 mm, contenant le texte "Bonjour apcpedagogie!", avec un cadre, un saut de ligne, un alignement centré, et un remplissage de couleur. Aucun lien hypertexte n'est créé dans cette cellule.

- **La méthode `Output()`**

  - La **méthode Output()** de la bibliothèque FPDF permet de générer et d'afficher un document PDF à partir des éléments ajoutés au document avec la bibliothèque FPDF.
  - Cette méthode peut prendre jusqu'à trois paramètres :
    - **`$name`**: le nom du fichier PDF à générer. Si ce paramètre est omis, le document PDF sera affiché dans le navigateur Web de l'utilisateur.

apcpedagogie.com

- `$dest`: la destination du fichier PDF généré. Les valeurs possibles pour ce paramètre sont "I" pour afficher le document dans le navigateur Web, "D" pour le télécharger directement sur le disque dur de l'utilisateur, "F" pour sauvegarder le fichier sur le serveur, ou "S" pour retourner le document en tant que chaîne de caractères.
- `$utf8encode`: un paramètre booléen qui indique si le texte doit être encodé en UTF-8. Par défaut, ce paramètre est défini sur false.

- Voici un exemple d'utilisation de la méthode **Output()**:

```
$pdf->Output('exemple.pdf', 'D');
```

- Dans cet exemple, un document PDF nommé "exemple.pdf" est généré et téléchargé sur le disque dur de l'utilisateur, en utilisant les éléments ajoutés avec la bibliothèque FPDF.
- Il est important de noter que la méthode Output() doit être appelée après l'ajout de tous les éléments au document PDF, sinon ceux-ci ne seront pas inclus dans le document généré.

7. Application

- **Objectifs :**
  - Créer un document PDF à partir des données stockées dans une base de données MySQL.
  - Afficher plusieurs feuilles numérotées avec des titres colorés.
- **Prérequis :**
  - Connaissance de base de PHP, MySQL et HTML.
  - Installation de la bibliothèque FPDF.
- **Étapes à suivre :**
  - **1-** Créer une base de données MySQL et une table contenant les données que vous souhaitez inclure dans le document PDF. Assurez-vous que les données contiennent les informations nécessaires pour la création de chaque feuille de votre document. Par exemple, si vous souhaitez créer un document PDF récapitulant les commandes passées par les clients, vous pouvez créer une table "commandes" contenant les champs suivants : ID, Nom du client, Date de la

apcpedagogie.com

commande, Produit, Quantité, Prix unitaire, Montant total.

- **2-**Écrire un script PHP qui se connecte à la base de données MySQL en utilisant PDO et récupère les données nécessaires pour créer le document PDF. Par exemple, vous pouvez récupérer toutes les commandes passées par les clients au cours du mois dernier
- **3-**Instancier un objet PDF de la bibliothèque FPDF en incluant le fichier fpdf.php. Par exemple, vous pouvez utiliser le code suivant pour instancier un objet PDF :

```
require('fpdf.php');
$pdf = new FPDF();
```

- **4-**Définir les propriétés du document PDF en appelant les méthodes de l'objet PDF. Par exemple, vous pouvez définir la taille et l'orientation de la page en appelant la méthode `AddPage()`:

```
$pdf->AddPage('P', 'A4');
```

- **5-**Définir la police de caractères à utiliser pour le titre de la première feuille en appelant la méthode SetFont() de l'objet PDF. Vous pouvez également définir la couleur du texte en utilisant la méthode **SetTextColor()**.

```
$pdf->SetFont('Arial', 'B', 16);
$pdf->SetTextColor(0, 0, 255);
```

- **6-**Afficher le titre de la première feuille en appelant la méthode **Cell()** de l'objet PDF.

```
$pdf->Cell(0, 10, 'Commandes passées au cours du mois dernier', 0, 1, 'C');
```

- **7-**Récupérer les données nécessaires pour créer la première feuille de votre document à partir de la base de données MySQL. Vous pouvez utiliser une requête SQL pour cela.

```
$stmt = $pdo->prepare("SELECT * FROM commandes WHERE DateCommande BETWEEN
:start_date AND :end_date");
$stmt->execute(array(':start_date' => $start_date, ':end_date' => $end_date));
$data = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

- **8-**Afficher les données de la première feuille en appelant la méthode **Cell()** de l'objet PDF. Vous pouvez également utiliser d'autres méthodes de la bibliothèque

apcpedagogie.com

FPDF pour afficher les données de manière plus sophistiquée.

```
$pdf->Cell(0, 10, 'Commandes passées au cours du mois dernier', 0, 1, 'C');
```

- **9-**Ajouter une nouvelle page au document en appelant la méthode **AddPage()** de l'objet PDF. Définir les dimensions et l'orientation de la page à votre convenance.
- **10-**Définir la police de caractères à utiliser pour le titre de la deuxième feuille en appelant la méthode **SetFont()** de l'objet PDF. Vous pouvez également définir la couleur du texte en utilisant la méthode **SetTextColor()**.
- **11-**Afficher le titre de la deuxième feuille en appelant la méthode **Cell()** de l'objet PDF.
- **12-**Récupérer les données nécessaires pour créer la deuxième feuille de votre document à partir de la base de données MySQL. Vous pouvez utiliser une requête SQL pour cela.
- **13-**Afficher les données de la deuxième feuille en appelant la méthode **Cell()** de l'objet PDF. Vous pouvez également utiliser d'autres méthodes de la bibliothèque FPDF pour afficher les données de manière plus sophistiquée.
- **14-**Répéter les étapes 9 à 13 pour chaque feuille que vous souhaitez ajouter à votre document.
- **15-**Générer le document PDF en appelant la méthode **Output()** de l'objet PDF. Vous pouvez spécifier le nom du fichier PDF et la destination souhaitée (par exemple, le télécharger directement sur le disque dur de l'utilisateur).
- **16-**Tester votre script et ajuster les paramètres à votre convenance.

- **Conclusion:**

  - Ce TP vous a permis de découvrir comment créer des documents PDF à partir de données stockées dans une base

---

Cours PHP

Source : https://apcpedagogie.com/creer-un-pdf-en-php-avec-fpdf · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com