Programmation PHP

Sessions, Mysql

David Gross-Amblard - Philippe Rigaux

6 novembre 2002

Conservatoire national des arts et métiers

Transparents disponibles la semaine prochaine sur

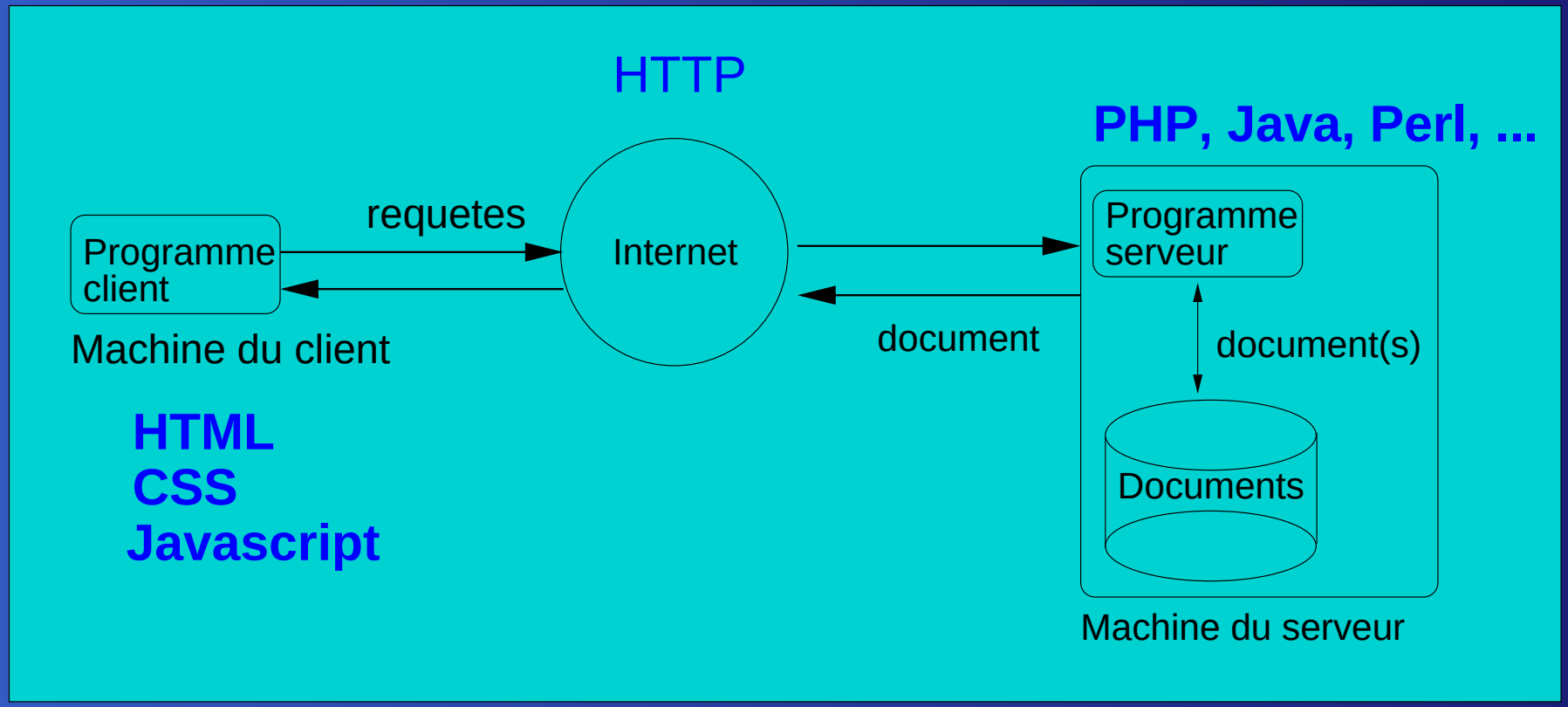
http://dept25.cnam.fr:8080/PROJET3I

Contenu

1. Rappels sur HTTP
2. Sessions PHP
   - *Cookies*
   - Sessions PHP
3. PHP et bases de données
   - mySQL
   - Interconnexion PHP/mySQL

Rappels sur HTTP

Fonctionnement du Web



Quelques mots sur HTTP

Protocole (langage) d'échange d'informations propre au Web

- On ouvre une connexion
- On envoie un en-tête
- Puis une ligne blanche
- Puis éventuellement un contenu
- Et on ferme la connexion

$\Rightarrow$Pas de connexion permanente !

Les requêtes HTTP

Principaux types de requêtes :

- `GET` : on demande quelque chose au serveur
  ```
  GET /index.html HTTP/1.0
  ```
- `POST` : on envoie quelque chose au serveur
  ```
  POST /cgi-bin/prog.php HTTP/1.0
  nom=rigaux&prenom=philippe
  ```
- `HEAD` : on demande des infos sur un document

Les réponses HTTP

En-tête/ligne blanche/contenu (Démo) :

```
Date: Tue, 05 Jun 2001 20:39:56 GMT
Server: Apache/1.3.17 (Unix) mod\_jk PHP/4.0.2
Last-Modified: Tue, 05 Jun 2001 20:13:39 GMT
Content-Length: 159
Connection: close
Content-Type: text/html

<HTML><HEAD>
<TITLE>Page d'accueil</TITLE></HEAD>
<BODY><H1>Page d'accueil</H1></BODY></HTML>
```

Informations d'une réponse HTTP

Dans l'en-tête :

- Le type de contenu `Content-type`
  $\Rightarrow$indique si le contenu est du HTML, du gif,du pdf, du XML...
- la longueur du contenu (permet de savoir combien de temps ça va prendre)
- la date de dernière modification (pour gérer le cache)

Limitation de HTTP

HTTP n'établit pas de connexion permanente !
$\Rightarrow$un serveur peut difficilement suivre le parcours de l'utilisateur dans le site

Chaque connexion est vue comme une demande d'un utilisateur différent

Une solution possible : les sessions, avec ou sans *cookies*

Les *cookies*

- Le serveur demande au navigateur de stocker une variable (nom,valeur), pour une durée déterminée
  - Serveur : "stocker la variable ’maVariable’ avec la valeur ’100’ pendant 2 jours"
- Le navigateur transmet ensuite systématiquement la variable au serveur qui l’a créée
  - Navigateur : "maVariable=100, ..."

Cookies et sessions web

- Les *cookies* sont utiles pour établir une continuité dans le dialogue client/serveur
  - Quand un client se connecte la première fois : le serveur lui associe un identifiant de session
  - Le serveur transmet cet identifiant au client sous la forme d'un cookie
  - On conserve un historique des actions du client, associé à l'identifiant de session
  - Quand le client se connecte à nouveau : on sait l'identifier grâce au *cookie*

Cookie avec HTTP

L'en-tête HTTP `Set-Cookie`

- Il comprend les éléments suivants:
- `nom=valeur` : définition du cookie
- `expires` : date de fin de validité
- `domain` : le domaine pour lequel le cookie est visible. Par défaut : nom du serveur qui créé le cookie
- (`path`,`secure`,...)

Exemple d'un en-tête avec *cookie*

```
Set-Cookie:
   MonCookie=200;
   expires=Mon,24-Dec-2000 12:00:00 GMT;
   domain=cnam.fr
```

Tout est optionnel, à l'exception de la première directive

Un script PHP pour définir des cookies

La fonction `setcookie`, en début de script, permet de modifier la réponse HTTP

```
<?php setcookie("MonCookie","chocolat");
?>
<HTML>
<BODY>
<P>Vous venez d’avaler un gâteau.</P>
</BODY>
</HTML>
```

démo

Test des *cookies*

```
<HTML>
<BODY>
<P>
<?php if (isset($MonCookie)) echo
   "Vous avez déjà eu du gâteau $MonCookie, vous !";
  else echo "Pas de gâteau";
?>
</P>
</BODY>
</HTML>
```

démo

Moins "bas niveau": sessions PHP

```
<?php
   session_start();

   session_register("maVariable");

   $maVariable="chocolat";
?>
Gloup.
```

démo

Récupération

```
<?php
session_start();
if (session_is_registered("maVariable"))
  echo "Vous avez mangé du $maVariable";
else
  echo "Puni. Pas de gâteau.";
?>
```

démo

Remarque

- Avec les sessions PHP, seul l'identifiant de session est transmis au navigateur
- Les variables sont stockées sur le serveur
- Par défaut dans un fichier, mais paramétrage possible

En résumé

- Suivre une interaction avec le même utilisateur
- Enregistrer une ou des variables sur le client
- Autre solution : champs `HIDDEN` des formulaires

PHP et les bases de données

1. MySQL

Pourquoi une BD

- Stocker de grandes quantité de données structurées
- Persistance (pannes)
- Langage de requête simple et expressif
- Efficacité (optimisation,index)

Applications

- Donnéees pour l'application (Films, Tel,...)
- Pour le fonctionnement du site
  - Stocker des variables de session
  - Stocker des statistiques d'accès (pub $$!)

Attention

- Ne pas construire une base n’importe comment...

  $\Rightarrow$cf cours du Cnam en ligne cyles A et B

Qu’est-ce que MySQL ?

- C’est un SGBD relationnel
  - Gratuit (www.mysql.com) GNU GPL
  - Disponible sous Windows et tous les Unix
  - Propose le minimum vital
- (Presque) tout le langage SQL
- Stockage, indexation, optimisation
- Qualités reconnues (outre la gratuité):
  - Simplicité
  - Efficacité pour requêtes simples

Quelques limites de MySQL

- Pas de transactions !
  - Inadapté pour les applications transactionnelles (finances, réservations)
  - Ni commit, ni rollback, une reprise sur panne rudimentaire
- Pas de requêtes imbriquées
  - Pas très grave !?
- Pas d'environnement de développement
- Alternative : PostgreSQL (www.postgresql.org)

Statut de MySQL à l'heure actuelle

- Très prisé pour les sites web
  - Très bonne intégration avec Apache et PHP
  - Tout est gratuit !
- En fort développement
  - Intégration des transactions ?
- Beaucoup de " contributions " extérieures
  - API en C++, Perl, PHP, Java (JDBC)
  - Des clients graphiques, des utilitaires

En résumé, un moteur SQL, simple et efficace

Architecture de MySQL

Le serveur mysqld

- Il est en écoute sur un port réseau (le 3306 par défaut)
- Il gère une ou plusieurs bases de données
  - La base mysql est le "dictionnaire de données"
  - Les autres bases sont des bases utilisateur
- Il dialogue avec ses clients en multi-threading (processus léger)

Client mySQL

Le client `mysql` permet de communiquer avec le serveur mySQL en ligne de commandes

```
% mysql -u rigaux -p MaBase
Password:******

mysql> [ici, n'importe quelle commande]
```

- On indique l'utilisateur et la base désirée

Utilisateurs :puissant mais compliqué

- Un utilisateur MySQL est défini par
  - Son nom
  - Le nom (Internet) de sa machine
- Problèmes :
  - `rigaux@cortes.fr`, `rigaux@cartier.fr` ne sont pas les mêmes utilisateurs !
  - Si le serveur tourne sous cortes, il faut utiliser le synonyme localhost !
- On peut utiliser des '%' : `%.cnam.fr` désigne toutes les machines du CNAM

Les droits

- Un utilisateur peut avoir plusieurs types de droits:
  - Sur le serveur (administration)
  - Sur les bases et tables (création, destruction)
  - Sur les données (lecture, écriture)
  - Sur d'autres utilisateurs (transmission de droits)
- Le compte root est le DBA: il a tous les droits sur tout

Création d'utilisateurs : GRANT

- Pour créer un utilisateur avec tous les droits sur la base Film:

```
GRANT ALL ON Film.*
TO admin@localhost
IDENTIFIED BY 'mdpAdmin'
```

- Pour créer un utilisateur avec des droits en lecture sur la table Artiste:

```
  GRANT select ON Film.Artiste
TO visiteur@localhost
IDENTIFIED BY 'mdpVisiteur'
```

Création de bases et de tables

- On crée des bases sous root :

```
% mysql -u root -p
mysql> CREATE DATABASE Films;
```

- Tout utilisateur ayant des droits peut créer des tables dans une base:

```
% mysql -u adminFilms -p Films
mysql> CREATE TABLE  Film (...);
```

- Les commandes SQL CREATE, ALTER, DROP sont reconnues

Types de données MySQL

- MySQL reconnaît les principaux types SQL2, et quelques autres:
  - `INTEGER` : les entiers
  - `FLOAT` : numériques flottants
  - `DECIMAL (M, D)` : numériques exacts
  - `CHAR` : chaînes de longueur fixe
  - `VARCHAR` :chaînes de longueur variable
  - `TEXT` : textes longs
  - `DATE` : dates
  - `TIME` : moments

Exemple de création de table

Insertion dans une table

- Il vaut mieux placer toutes les commandes dans un fichier de script, `script.sql`

```
% mysql -u MonNom -p MaBase < script.sql
```

Exemple d’un fichier de script

```
create table Carnet(
   nom text,
   prenom text,
   age integer);

insert into Carnet values (’Truquemuche’,’Jean’,18);
insert into Carnet values (’Chose’,’Alice’,39);
...
```

MySQL en résumé

- Tout à fait conforme à la norme SQL ANSI
  - `CREATE TABLE`, `DROP TABLE`, `ALTER TABLE`
  - `SELECT`, `INSERT`, `DELETE`, `UPDATE`
- Avec des extensions (très utiles)
  - Types de données (`TEXT`, `BLOB`)
  - Contraintes (`AUTO_INCREMENT`)
  - Beaucoup de fonctions

PHP et les bases de données

2. Interconnexion PHP/MySQL

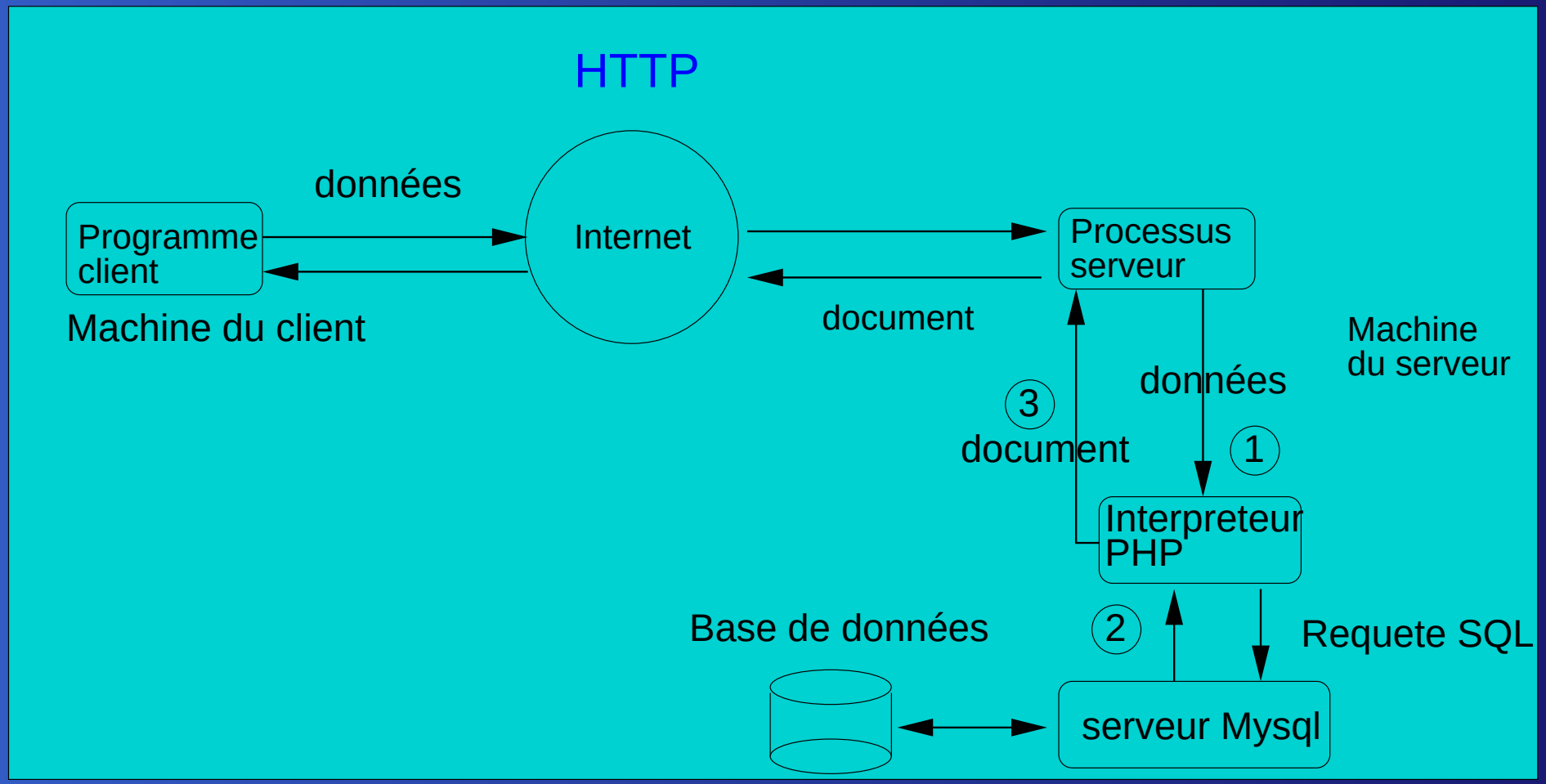
mySQL et PHP

Principe : création de documents web à partir d'une BD

- MySQL se charge du stockage, de la protection des données, de l'interface SQL
- PHP
  - extrait des données et les met en forme
  - reçoit des données et les stocke
- Le navigateur fournit l'interface graphique

$\Rightarrow$Architecture à trois pôles, à la sauce Web

Architecture avec base de données



Connexion à MySQL avec PHP

- `mysql_pconnect (serveur, nom, passe)`
  - établit une connexion
  - si OK, renvoie un identifiant non nul cnx
- `mysql_select_db(base, cnx)`
  - se place dans une base et renvoie vrai si OK
- `mysql_query (requete, cnx)`
  - exécute une requête et renvoie un identifiant
- `mysql_fetch_object (resultat)`
  - renvoie la ligne suivante sous forme d'objet

Exemple : interrogation et affichage (1)

```
<?php
$cnx=mysql_pconnect("dept25.cnam.fr","rigaux","4fRf*");
if (!$cnx){
  echo "probleme de connexion."; exit; }

mysql_select_db("Film",$cnx);

$res=mysql_query("select nom,age from Artiste",$cnx);

...
```

Exemple : interrogation et affichage (2)

```
...
echo "<TABLE>";
while($nuplet=mysql_fetch_object($res){
  echo "<TR>";
  echo "<TD>$nuplet->nom</TD>";
  echo "<TD>$nuplet->age</TD>";
  echo "</TR>";
}

echo "</TABLE>";
?>
```

Association avec formulaire

```
<FORM ACTION="mysql.php" METHOD=POST>
Nom de l'artiste : <INPUT TYPE=TEXT NAME=nom> <BR>
Prenom : <INPUT TYPE=TEXT NAME=prenom> <BR>
Age : <INPUT TYPE=TEXT NAME=age> <BR>
<INPUT TYPE=SUBMIT NAME=ajout VALUE="ajouter">
<INPUT TYPE=SUBMIT NAME=suppression VALUE="supprimer">
</FORM>
```

Traitement

- On récupère donc les variables `$nom`, `$prenom`, `$age`, `$ajout` et `$suppression`
- On construit la bonne requête SQL
  - Un ordre INSERT pour des insertions
  - Un ordre DELETE pour une destruction
- Dans tous les cas la fonction `mysql_query` permet d'exécuter l'ordre

Traitement : mysql.php

```
<?php
   ..............(partie connexion)
if ($ajout)
$requete="insert into Artiste values
                  ('$nom','$prenom','$age')";
else
 if ($supression)
 $requete="delete from Artiste where
                  nom=$nom and prenom=$age";

   mysql_query($requete,$cnx);
?>
```

Informations sur le schéma

- Etant donné le résultat d'une requête :
  - `mysql_num_fields($res)` donne le nombre d'attributs
  - `mysql_field_name($res, $i)` donne le nom de l'attribut
  - `mysql_fetch_row ($res)` renvoie une ligne du résultat sous la forme d'un tableau indicé.
- Application : phpMyAdmin

Conclusion

- mySQL : sgbd libre, puissant, répandu
- Interconnexion avec PHP :
- production de documents HTML à partir de requêtes SQL
- gestion des interactions avec l ’utilisateur (requêtes/données de formulaires)

Promenades PHP:

- Internationalisation
- PDF
- Images

Internationalisation

- Gerer facilement les sites multi-lingue
- Utilise la librairie Gettext
- ...ou se simule facilement

Principe

- Scripts en anglais
- Table de traduction anglais $\Rightarrow$...
- Centralisé dans un repertoire

Repertoire

- locale/
  - fr/
  - en/
  - es/
  - de/

Dans chacun, un fichier myPHPApp.

Exemple

Dans locale/fr/myPHPApp:

```
Welcome to my PHP Application:Bienvenue
    dans mon application PHP
Hello:Bonjour
email:mél
...
```

Exemple

```
<?php

putenv ("LANG=fr");

bindtextdomain ("myPHPApp", "./locale");

textdomain ("myPHPApp");

print (gettext ("Welcome to My PHP Application"));
?>
```

Production d’images

Exemple : un bouton dont le contenu varie

Contenu en paramètre, appelé par :

```
<img src="button.php?text">
```

Production d’images

```
<?php
    Header("Content-type: image/png");
    $string=implode($argv," ");
    $im=imageCreateFromPng("images/button1.png");
    $orange=ImageColorAllocate($im,220,210,60);
    $px=(imagesx($im)-7.5*strlen($string))/2;
    ImageString($im,3,$px,9,$string,$orange);
    ImagePng($im);
    ImageDestroy($im);
?>
```

Production de PDF

```
$pdf = pdf_new();
pdf_open_file($pdf, "test.pdf");
pdf_set_info($pdf,"Title","Test for PHP");
pdf_begin_page($pdf,595,842);
pdf_add_outline($pdf,"Page 1");
pdf_set_font($pdf, "Times-Roman",30,"host");
pdf_show_xy($pdf, "Times Roman outlined",50,750);
pdf_moveto($pdf,50,740);
pdf_lineto($pdf,330,740);
pdf_stroke($pdf);
pdf_end_page($pdf);
pdf_close($pdf);
pdf_delete($pdf);
```