

Cours

Programmation avancée pour le web

Credit : 4

Coefficient : 2

Mode d'évaluation :

- Examen (60%) - contrôle continu (40%)

Author:

Aimad Eddine Debbi

Chap VII PHP

PHP:

PHP stands for « *hypertext preprocessor* » est un langage de script côté serveur Il permet d'apporter de la logique, du dynamisme via l'interfaçage avec des bases de données.

-

Chap VII PHP

PHP :

syntax

- Un script PHP commence par `<?php` et se termine par `?>`.

```
<?php
```

```
// script php ...;
```

```
?>
```

Chap VII PHP

PHP :

syntax

- Un fichier .php contient normalement du code html et le des scripts php.

```
<!DOCTYPE html>
<html>
<body>
<h1> PHP page </h1>

<?php
echo" php  prompt!";
?>

</body>
</html>
```

Chap VII PHP

PHP:

syntax

- php n'est pas *case sensitive* (*sensible aux majuscules/minuscules*) au niveau des mots réservés tels (**if else while echo**) et les noms de fonctions. Mais sensible aux Maj/min au niveau des variables.

```
<body>
<h1> PHP page </h1>
<?php
$i=2; $I=5;
While($i){
    echo" php  prompt! in the loop pass" .$i. "<br>";
    EchO" PHP  PROMPT! in the loop pass" .$I. "<br>";
    $i--; }
?>
</body>
```

Rendu

Chap VII PHP

PHP:

Les sorties avec **echo** et **print**

- Les appels echo et print permettent l'émission d'affichages. L'appel echo et print s'établie avec ou sans parenthèses.

```
<body>
<h1> PHP page </h1>
<?php
$title1= " Main section ";
$title2= " Sub section ";
        echo("<h1>".$title1."<h1><br>");
        print("<h2>".$title2."<h2><br>");
?>
</body>
```

Rendu

Chap VII PHP

PHP:

Les variables

- Les variables php doivent commencer par: \$
- Mais les variables globales (déclarée en dehors des fonctions) ne sont pas reconnues en dedans des fonctions.

```
<body>
<h1> PHP page </h1>
<?php
$var1= " var1 as global";
function show__var(){ $var2= " local var2";
    echo("<p> global var is :".$var1."</p>");
    print ("<p> local is :".$var2."</p>");
}
show__var();
?>
</body>
```

Rendu

Chap VII PHP

PHP:

Les variables (les types)

- Les types reconnus en php sont : String, Integer, Float, Boolean, Array, Object and Null.
- L'appel `var_dump()` donne le type et la valeur.

```
<body>
<h1> PHP page </h1>
<?php
$intvar= 2538; $tablevar= array("wrd1","wrd2");
    var_dump($intvar); // int(2538)
    var_dump($tablevar);
                        //array(2) {[0]-> string(4) "wrd4" .....
?>
</body>
```

Rendu

Chap VII PHP

PHP :

Les SuperGlobals

- Un ensemble de variables prédéfinies sont dits **superglobal** et sont accessibles depuis partout (en dedans et en dehors des fonctions et des classes).

```
$GLOBALS  
$_SERVER  
$_REQUEST  
$_POST  
$_GET  
$_FILES  
$_ENV  
$_COOKIE  
$_SESSION
```

Rendu

Chap VII PHP

PHP:

Les SuperGlobals

- Le **superglobal \$GLOBALS** est utilisé pour accéder les variables globales utilisateur

```
<?php
$var1= " var1 as global";
function show__var(){ $var2= " local var2";
    echo("<p> global var is :". $GLOBALS['var1'].
        "</p>");
    print ("<p> local is :". $var2."</p>");
}
show__var();
?>
```

Rendu

Chap VII PHP

PHP :

Les SuperGlobals

- Le **superglobal** `$_SERVER` contient de l'information concernant les headers, les chemins (paths), le script ...
- Les éléments renseignés par `$_SERVER` :

Element	Designation
<code>\$_SERVER ['PHP_SELF']</code>	Nom de fichier script
<code>\$_SERVER ['SERVER_ADDR']</code>	Adresse IP du host (server)
<code>\$_SERVER ['SERVER_NAME']</code>	Nom du serveur
<code>\$_SERVER ['SERVER_PROTOCOL']</code>	Protocol et version (HTTP/1.1)
<code>\$_SERVER ['REQUEST_METHOD']</code>	(get / post /)
<code>\$_SERVER ['QUERY_STRING']</code>	La requête query si elle existe

Chap VII PHP

PHP :

Les SuperGlobals

- Le **superglobal** `$_SERVER` contient de l'information concernant les headers, les chemins (paths), le script ...
- Les éléments renseignés par `$_SERVER` :

Element	Designation
<code>\$_SERVER ['SCRIPT_FILENAME']</code>	Chemin absolu du fichier script
<code>\$_SERVER ['REMOTE_ADDR']</code>	Adresse IP du client
<code>\$_SERVER ['REMOTE_HOST']</code>	Nom du client
<code>\$_SERVER ['SERVER_PORT']</code>	Port utilisé sur le hôte client

Chap VII PHP

PHP:

Les SuperGlobals

- Le **superglobal** `$_REQUEST` rassemble les données issues d'une émission de requête formulaire.

```
<body>
<form method="post"
      action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="frst_name">
  <input type="submit">
</form>
<?php if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $name = $_REQUEST['frst_name'];
      if (empty($name)) {
          echo "Name is empty";
      } else {echo $name;    }
    }
?>
</body>
```

Chap VII PHP

PHP:

Les SuperGlobals

- Le **superglobal** `$_POST` rassemble les données issues d'une requête POST .

```
<body>
<form method="post"
      action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="frst_name">
  <input type="submit">
</form>
<?php if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['frst_name'];
    if (empty($name)) {
        echo "Name is empty";
    } else {echo $name;    }
}
?>
</body>
```

Chap VII PHP

PHP:

Les SuperGlobals

- Le **superglobal** `$_GET` rassemble les données issues d'une requête GET .

```
<body>
<a href="handle_get.php?topic=PHP&location=school">
do GET</a>
</body>
```

Le fichier `handle_get.php` :

```
<?php
echo "Study " . $_GET['topic'] . " at " . $_GET['location'];

?>
```

Chap VII PHP

PHP:

Travailler avec bases de données MySQL

Pour accéder à la base de données MySQL une connexion doit être au préalable établie.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```


Chap VII PHP

PHP :

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
$creation__reque = "CREATE DATABASE My__DBase";
$table__creation = "CREATE TABLE medecine( id INT(6) UNSIGNED
AUTO_INCREMENT PRIMARY KEY, medcn_name VARCHAR(30) NOT NULL,
labo VARCHAR(30) NOT NULL, provider VARCHAR(30) NOT NULL,
quantity INT(6) UNSIGNED, price FLOAT) ";
$conn = new mysqli($servername, $username, $password);
if($conn->query($creation__reque )===true){
    echo "successfull creation";
    if($conn->query($table__creation)===true)
        echo "table creation";
    } else echo " creation echec";
?>
```

Chap VII PHP

PHP:

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//insertion of one row.
$insert__request = " INSERT INTO medecine (medcn_name, labo,
provider, quantity, price)
VALUES ('voltarene' , 'Novartis' , 'Betta_pharm' , 12 , 15.5) ";

if ($conn->query($insert__request) === TRUE) {
    echo "successfull insertion";
} else {
    echo "Error: " . $insert__request . "<br>" . $conn->error;
}
?>
```

Chap VII PHP

PHP:

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//insertion of second row.
$insert__request = " INSERT INTO medecine (medcn_name, labo,
provider, quantity, price)
VALUES ('sulpiride', 'Sanofi', 'Sigma_pharm', 08, 40.7) ;

if ($conn->query($insert__request) === TRUE) {
    echo "successfull insertion";
} else {
    echo "Error: " . $insert__request . "<br>" . $conn->error;
}
?>
```

Chap VII PHP

PHP:

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//insertion of second row.
$insert__request = " INSERT INTO medecine (medcn_name, labo,
provider, quantity, price)
VALUES ('paracetamol' , 'Sanofi' , 'Alpha_pharm' , 05, 22.2) ;

if ($conn->query($insert__request) === TRUE) {
    echo "successfull insertion";
} else {
    echo "Error: " . $insert__request . "<br>" . $conn->error;
}
?>
```

Chap VII PHP

PHP :

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//l'interrogation et la clause SELECT.
$sel__request = " SELECT * FROM medecine "
$result = $conn->query($ sel__request);
if($result->num_rows > 0){
    echo " rows :" . $result->num_rows . "<br>" ;
    while($row = $result->fetch_assoc())
        echo " med__name :" . $row[1]. "<br>" ;
}

$conn->close;
?>
```

Chap VII PHP

PHP :

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//l'interrogation et la clause ORDER.
$order__sel__request = " SELECT medcn_name FROM
                        medicine ORDER BY quantity "
$result = $conn->query($order__sel__request);
if($result->num_rows > 0){
    echo " rows :" . $result->num_rows . "<br>" ;
    while($row = $result->fetch_assoc())
        echo " med__name :" . $row[1]. "<br>" ;
}

$conn->close;
?>
```

Chap VII PHP

PHP:

Travailler avec bases de données MySQL

La création, l'insertion, l'interrogation et la suite de traitements se font principalement avec une requête `query()`.

```
<?php
//la mise à jour "update" .
$update__request = " UPDATE medcn_name SET
medcn_name='agripal' WHERE id=5 "

if ($conn->query($update__request) === TRUE) {
    echo "successfull update";
} else {
    echo "Error: " . $update__request . "<br>" . $conn->error;
}

$conn->close;
?>
```