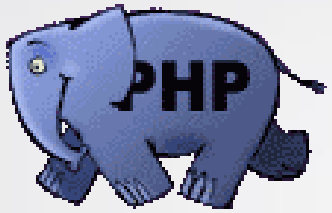
esprit
Se former autrement
HONORIS UNITED UNIVERSITIES



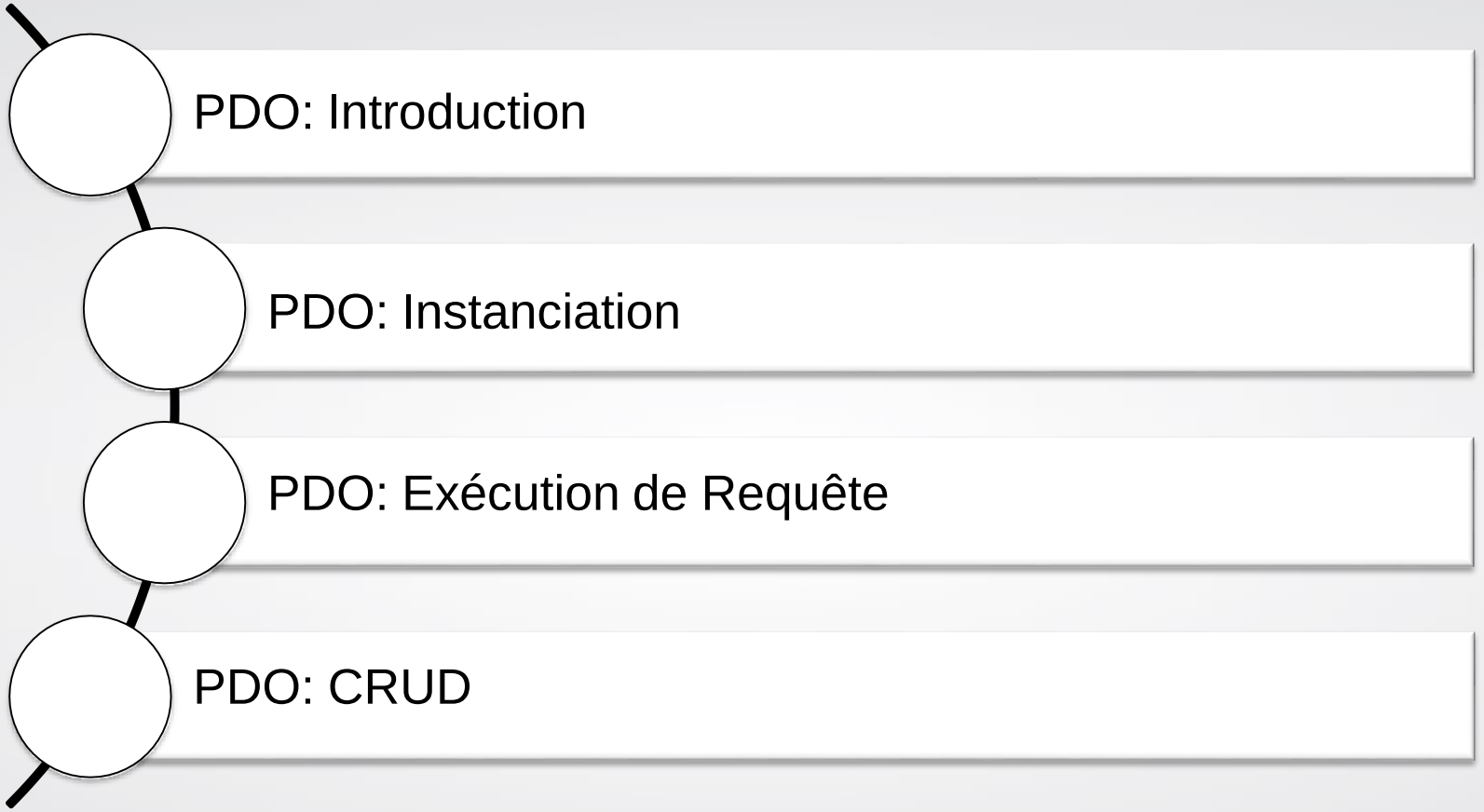
Chapitre 5: PHP

UP Web

AU: 2022/2023

United Nations Educational, Scientific and Cultural Organization
UNESCO Chair "Project-based learning" ESPRIT School of engineering, Tunisia
uniTwin
EUR-ACE® Délivrée par la Commission des Titres d'Ingénieur
CONFÉRENCE DES GRANDES ÉCOLES
cdio CONCEIVE DESIGN IMPLEMENT OPERATE

Plan


PDO: Introduction
PDO: Instanciation
PDO: Exécution de Requête
PDO: CRUD

Objectifs

- Les architectures du web
- Comprendre la syntaxe PHP
- Appréhender les notions de l'orientée objet
- Se connecter à une BD
- Manipuler les données d'une BD via PHP

Prérequis

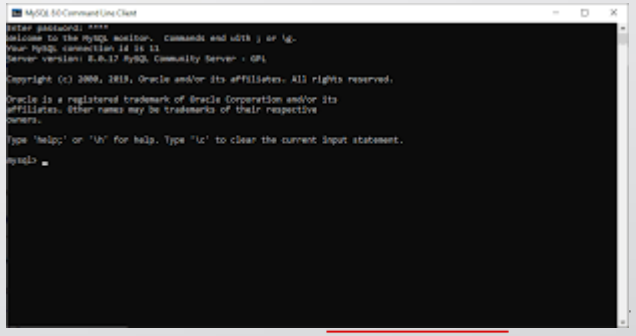
- Langage HTML

Introduction

Il existe plusieurs outils pour administrer une base de données, notamment :

- ❖ Les applications web tels que PhpMyAdmin, MySQL Workbench
- ✓ Une interface conviviale pour interagir avec la base de données via un navigateur web
- ○ Limitée en termes de flexibilité et de portabilité
- ❖ Les logiciels tels que DBeaver,Navicat
- ❖ CLI : Les méthodes de ligne de commande destinées aux administrateurs de bases de données et aux développeurs expérimentés.
- ✓ Une connaissance approfondie des commandes

et des options spécifiques à chaque base de données.



Manipulation de la BD

PDO: PHP Data Objects

- PDO définit une interface pour accéder à une BD depuis PHP.
- PDO supporte plusieurs systèmes de bases de données:
  - MySQL
  - ODBC
  - SQLITE
  - OCI Oracle Call Interface
  - SQLite
  - …

Manipulation de la BD

PHP connexion à la BD

- Pour se connecter au serveur, on peut utiliser le fichier '**connection.php**':

```php
<?php
    $servername = 'localhost';
    $username = 'username';
    $password = 'password';
    $dbname = 'myDB';
    try {
        $pdo = new PDO(
            "mysql:host=$servername;dbname=$dbname",
            $username,
            $password,
            [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
            ]
        );

        echo "Connected successfully";
    }
    catch(PDOException $e) {
        echo "Connection failed: ". $e->getMessage();
    }
?>
```

Manipulation de la BD

PHP connexion à la BD

Cette configuration permet de lancer une exception PDOException en cas d'erreur.

```php
<?php
    $servername = 'localhost';
    $username = 'username';
    $password = 'password';
    $dbname = 'myDB';
    try {
        $pdo = new PDO(
            "mysql:host=$servername;dbname=$dbname",
            $username,
            $password,
            [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
            ]
        );

        echo "Connected successfully";
    }
    catch(PDOException $e) {
        echo "Connection failed: ". $e->getMessage();
    }
?>
```

Manipulation de la BD

PHP connexion à la BD

Cette configuration permet de spécifier la méthode de récupération. Dans se cas, chaque ligne est retournées dans un tableau indexé par le nom des colonnes.

```php
<?php
    $servername = 'localhost';
    $username = 'username';
    $password = 'password';
    $dbname = 'myDB';
    try {
        $pdo = new PDO(
            "mysql:host=$servername;dbname=$dbname",
            $username,
            $password,
            [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
            ]
        );

        echo "Connected successfully";
    }
    catch(PDOException $e) {
        echo "Connection failed: ". $e->getMessage();
    }
?>
```

Manipulation de la BD

PHP connexion à la BD

- Ou en utilisant la syntaxe suivante:

```php
<?php
    $servername = 'localhost';
    $username = 'username';
    $password = 'password';
    $dbname = 'myDB';
    try {
        $pdo = new PDO(
            "mysql:host=$servername;dbname=$dbname",
            $username,
            $password
        );
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
        echo "Connected successfully";
    }
    catch(PDOException $e) {
        echo "Connection failed: ". $e->getMessage();
    }
?>
```

Manipulation de la BD

PHP connexion à la BD

- La première étape consiste à créer un objet PDO:

```php
$conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
```

1 2 3

1. **dsn ou Data Source Name**: représente les informations nécessaires pour se connecter à la base.
2. **Username** (optionnel pour certains pilote tel que sqlite)
3. **Password**
4. **Options**: un tableau contenant les options spécifiques à la connexion. Ce paramètre est optionnel.

Remarque: La base de données (myDB) peut être créer en utilisant l'application phpMyAdmin.

Manipulation de la BD

PHP connexion à la BD

- La connexion sera fermée automatiquement lorsque le script se termine. Pour fermer la connexion avant, utilisez la commande suivante:

```
$conn = null;
```

- Comment je peux inclure le fichier de « **connection.php** »?

Récrire le code connection.php, utiliser les fonctions :

- require ()
- require-once()
- include()
- include-once()

Manipulation de la BD

PDO: exec(), query(), execute(), prepare()

Fonction	Explication
PDO::exec(string $sql): int	Permet d'exécuter une requête. Renvoie le nombre de ligne affectées. À utiliser avec: Insert, Update et Delete.
PDO::query(string $sql): PDOStatement	Exécute une requête. Retourne le résultat en un objet PDOStatement
PDO::prepare(string $sql): PDOStatement	Prépare la requête sans l'exécuter. Renvoie un objet PDOStatement.
PDOStatement::execute([array $param]): bool	Exécute une requête préparée.

Manipulation de la BD

PDO: CRUD

CRUD: ‘Create’ des données

L'instruction INSERT INTO est utilisée pour ajouter de nouveaux enregistrements à une table.

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'INSERT INTO personne (Nom, Prenom, Age)
                VALUES (:nom, :prenom, :age)'
        );
        $query->execute([
            'nom' => 'Ben Mohamed',
            'prenom' => 'Sarra',
            'age' => 22
        ]);
    } catch (PDOException $e) {
        $e->getMessage();
    }
?>
```

Manipulation de la BD

PDO: CRUD

CRUD: ‘Create’ des données

L'instruction INSERT INTO est utilisée pour ajouter de nouveaux enregistrements à une table.

PDOStatement::bindValue(
$param,
$value,
[$data_type]
) : bool

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'INSERT INTO personne (Nom, Prenom, Age)
                VALUES (:nom, :prenom, :age)'
        );
        $query->bindValue('nom', 'Doe');
        $query->bindValue('prenom', 'Jane');
        $query->bindValue('age', 20, PDO::PARAM_INT);
        $query->execute();
    } catch (PDOException $e) {
        $e->getMessage();
    }
?>
```

Manipulation de la BD

PDO: CRUD

CRUD: ‘Create’ des données

L'instruction INSERT INTO est utilisée pour ajouter de nouveaux enregistrements à une table.

PDOStatement::bindParam
(
$param,
$value,
[$data_type],
[$length]
) : bool

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'INSERT INTO personne (Nom, Prenom, Age)
                VALUES (?, ?, ?)'
        );
        $nom = "Doe";
        $prenom = 'John';
        $age = 30;
        $query->bindParam(1, $nom);
        $query->bindParam(2, $prenom, PDO::PARAM_STR, 12);
        $query->bindParam(3, $age, PDO::PARAM_INT);
        $query->execute();
    } catch (PDOException $e) {
        $e->getMessage();
    }
?>
```

Manipulation de la BD

PDO: CRUD

CRUD: ‘Read’ des données
L'instruction SELECT est utilisée pour sélectionner les données à partir d'une table.

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'SELECT * FROM personne'
        );
        $query->execute();
        $result = $query->fetchAll();
    } catch (PDOException $e) {
        $e->getMessage();
    }
    foreach ($result as $row) {
        echo $row['Nom'] . ' ' . $row['Prenom'];
    }
?>
```

Manipulation de la BD

PDO: CRUD

CR**U**D: ‘Update’ des données
L'instruction UPDATE est utilisée pour modifier les données d'une table.

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'UPDATE personne SET nom = :nom,
            prenom = :prenom where id = :id'
        );
        $query->execute([
            'nom' => 'Doe',
            'prenom' => 'Jane',
            'id' => 1
        ]);
        echo $query->rowCount() . " records UPDATED
        successfully";
    } catch (PDOException $e) {
        $e->getMessage();
    }

?>
```

Manipulation de la BD

PDO: CRUD

CRU**D**: ‘Delete’ des données
L'instruction DELETE est utilisée pour supprimer des données d'une table.

```php
<?php
    require 'connection.php';

    try {
        $query = $pdo->prepare(
            'DELETE FROM personne where id = :id'
        );
        $query->execute([
            'id' => 1
        ]);
        echo $query->rowCount() . " records DELETED
        successfully";
    } catch (PDOException $e) {
        $e->getMessage();
    }

?>
```

Merci de votre attention