

# Design Pattern MVC

Etude de cas.

Dr. Dellys Hachemi

Cours Technologie et Développement Web (TDW) 2CS

2025-2026

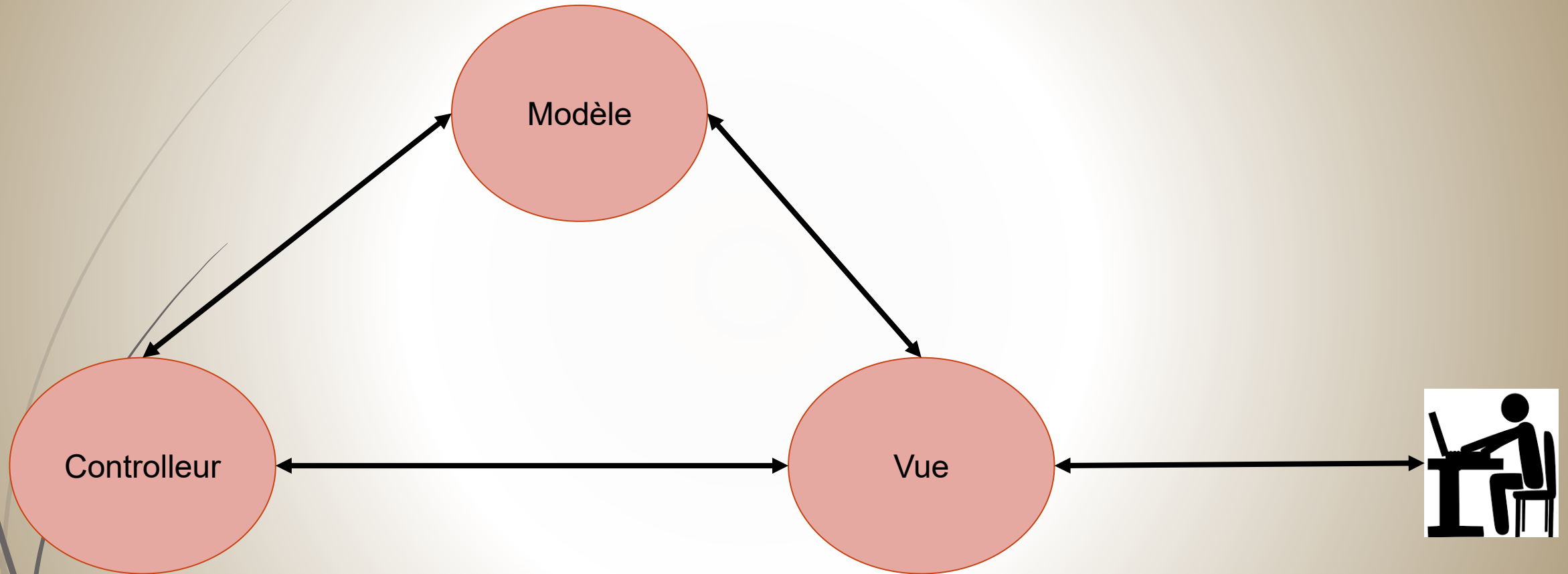
# Introduction

- Un code programmé à l'aide d'un langage web sans modélisation ni organisation peut devenir très rapidement difficilement maintenable.
- L'évolution des outils de développement web génère des codes avec des milliers de lignes et la programmation nécessite une organisation et des règles de programmation.
- Un code professionnel doit être modulaire, bien découpé, bien documenté et facile à comprendre.
- Le patron de conception MVC (Modèle- Vu- Contrôleur) est parfaitement adapté à la programmation WEB et c'est le plus utilisé par les concepteurs d'application WEB.

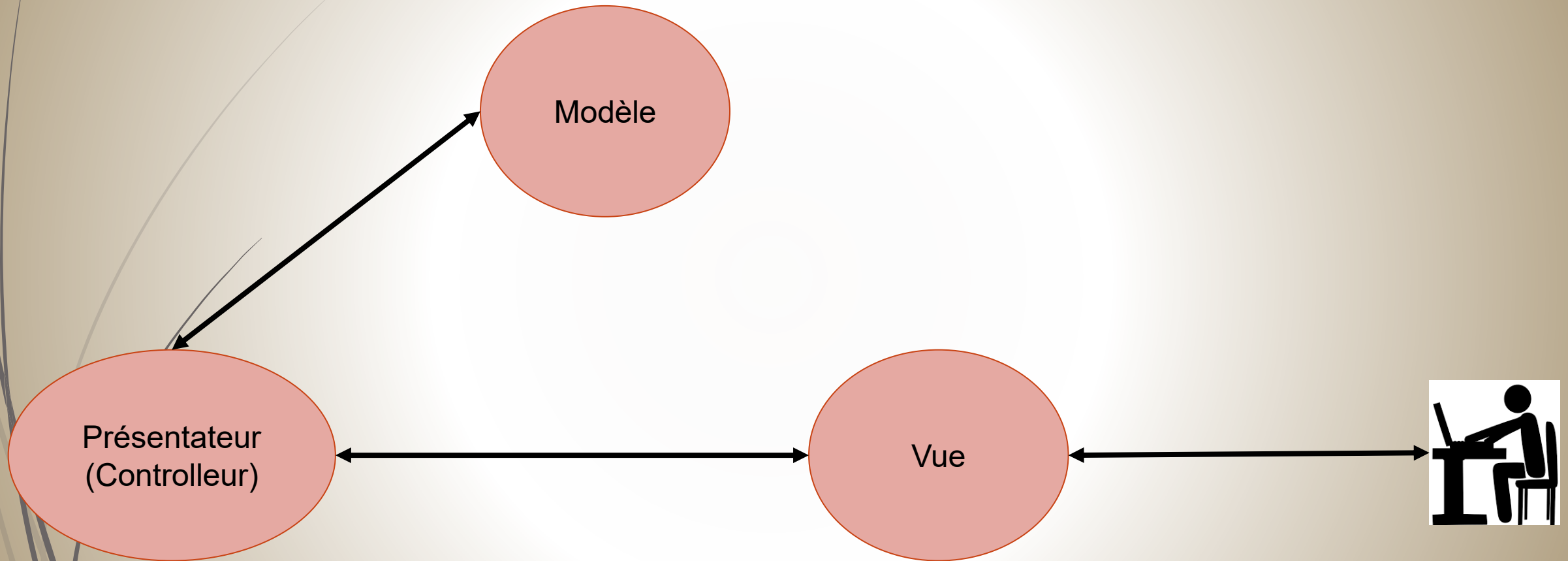
# Design Pattern MVC

- Le pattern MVC est une architecture logicielle très utilisée dans le WEB.
- Le MVC permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.
  - **Modèle** : cette partie gère les *données* du site. Les traitements effectués dans le modèle ne concernent que la récupération des données et leurs sauvegardes dans la source de données (BDD, fichier ...ect.). Le modèle renvoie les données brutes (éventuellement organisées) vers le contrôleur.
  - **Vue** : cette partie gère l'*affichage du site*. Son code se compose essentiellement du HTML, CSS et JavaScript. Les traitements y sont limités au minimum comme quelques boucles et conditions PHP très simples afin de récupérer les données du contrôleur et organiser l'affichage..
  - **Contrôleur** : cette partie gère les traitements et les algorithmes principaux du site. C'est l'intermédiaire entre le modèle et la vue : le contrôleur va demander des données au modèle les traiter et envoyer les informations adéquates à la vue pour les afficher.

# Design Pattern MVC



# Design Pattern MVP



## Organisation d'un code PHP

### ➤ **Cas d'étude:** Menu d'un site d'une école de formation

Nous prenons comme cas d'étude le menu d'un site d'une école de formation fait sans suivre de pattern et nous allons le transformer en un code sous architecture MVC.

### ➤ Nous nous concentrons par la suite sur la gestion du menu en MVC.



# Entête et connexion à la BDD

7

```
<html>
<head>
<meta name="Ecole" content="TDW" />
<meta name="Author" content="Dellys Hachemi" />
<link href="CSS/Exo2.css" rel="stylesheet" type="text/css" />
<script src="JS/jquery-3.3.1.min.js" ></script>
<script src="JS/Exo-JQuery.js" ></script>
<title> Ecole de Formation TDW </title>
</head>
<body>

<?php

$dsn="mysql:dbname=tdw_mvc; host=127.0.0.1;";
try{
$c=new PDO($dsn, "root", "");
}
catch(PDOException $ex){
printf("erreur de connexion à la base de donnée", $ex->getMessage());
exit();
}

?>
```

# Titre, diaporama et description

8

```
<h1 align="center"> Ecole de formation <abbr title="Ecole de formation TDW"> TDW </abbr> </h1></br>

<div id="APdiapo">
<div class="diapo">




</div>
</div>
</br><br></br></br></br>
<p class="description">TDW est une école de formation offrant diverses formations professionnelles dans
```



# Affichage du menu

9

```
<nav>
<ol class="Menu">
<?php

$qtff="select * from type_formation";
foreach($c->query($qtff) as $rowtf){
echo '<li class="type_formation"><a href="pages/' . $rowtf["Nom_type_formation"] . '.html">' .
$rowtf["Nom_type_formation"] . '</a>';
$qf="select * from formation where Id_type_formation=" . $rowtf['Id_type_formation'] . " ";

echo '<ul>';
foreach($c->query($qf) as $rowf){
echo '<li class="formation"><a href="pages/' . $rowtf["Nom_type_formation"] . '.html">' .
$rowf["Nom_formation"] . '</a></li>';
}
echo '</ul></li>';
}

?>
</ol>
</nav>
```

# Résultat

10

## ***Ecole de formation TDW***



TDW est une école de formation offrant diverses formations professionnelles dans les domaines informatique, finance et langues.

Bureautique

Infographie

Langue

Management

Comptabilité

Power point

Photoshop

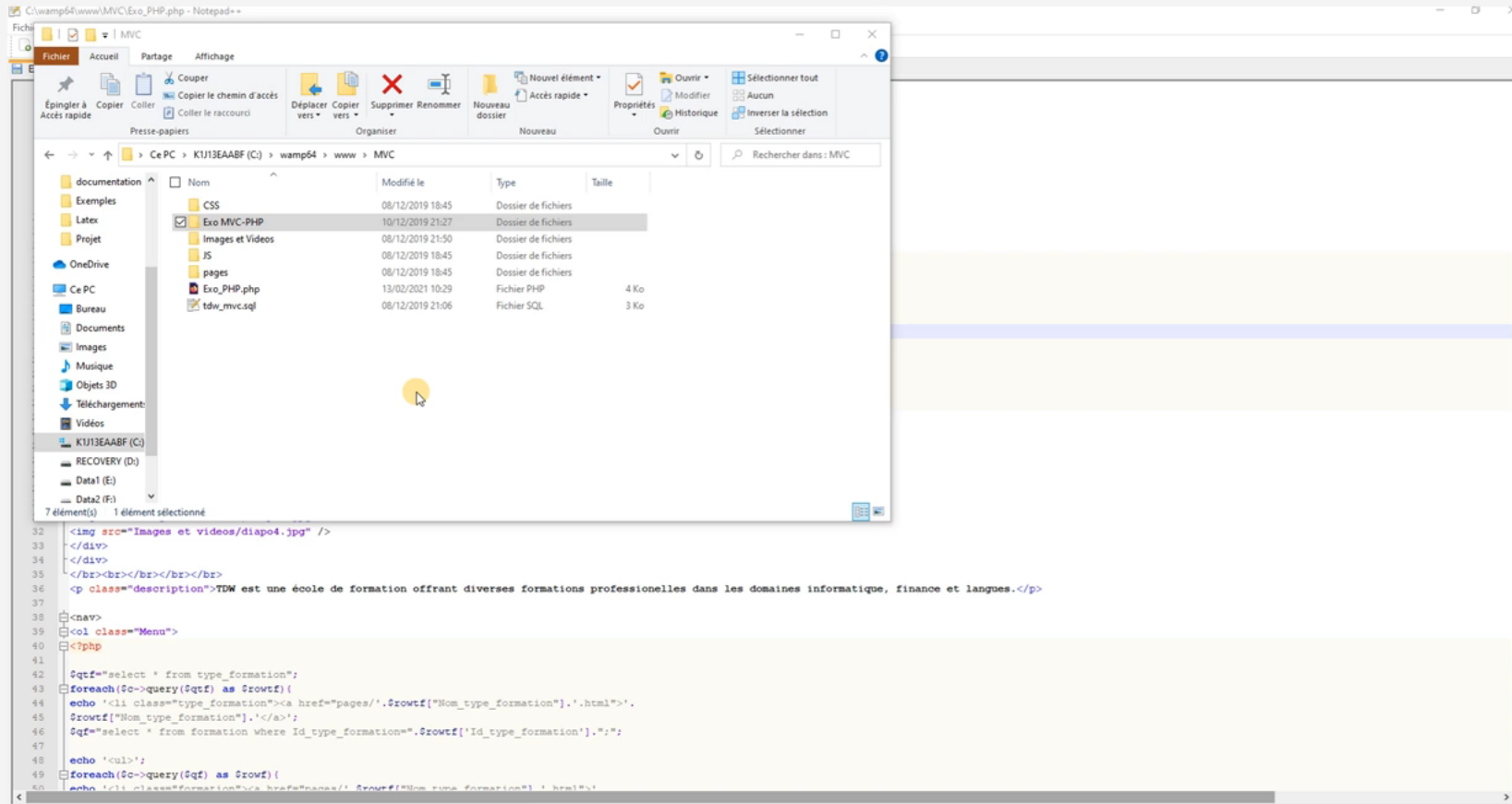
Illustrator

Liste des tarifs:



# Screen Recording

Select this object and click the Screen Recording button to edit



## Passage à une architecture de type MVC

- Pour implémenter un site sous format MVC, il est conseillé de créer au moins une classe pour chacun des concepts:
  1. Une classe modèle.
  2. Une classe vue.
  3. Une classe contrôleur
- Chaque fonctionnalité du cahier des charges devra se transformer en une ou plusieurs méthodes les plus atomiques possible.
- Chaque méthode se décline par la suite sur une ou plusieurs méthodes selon la nécessité

## Passage à une architecture de type MVC

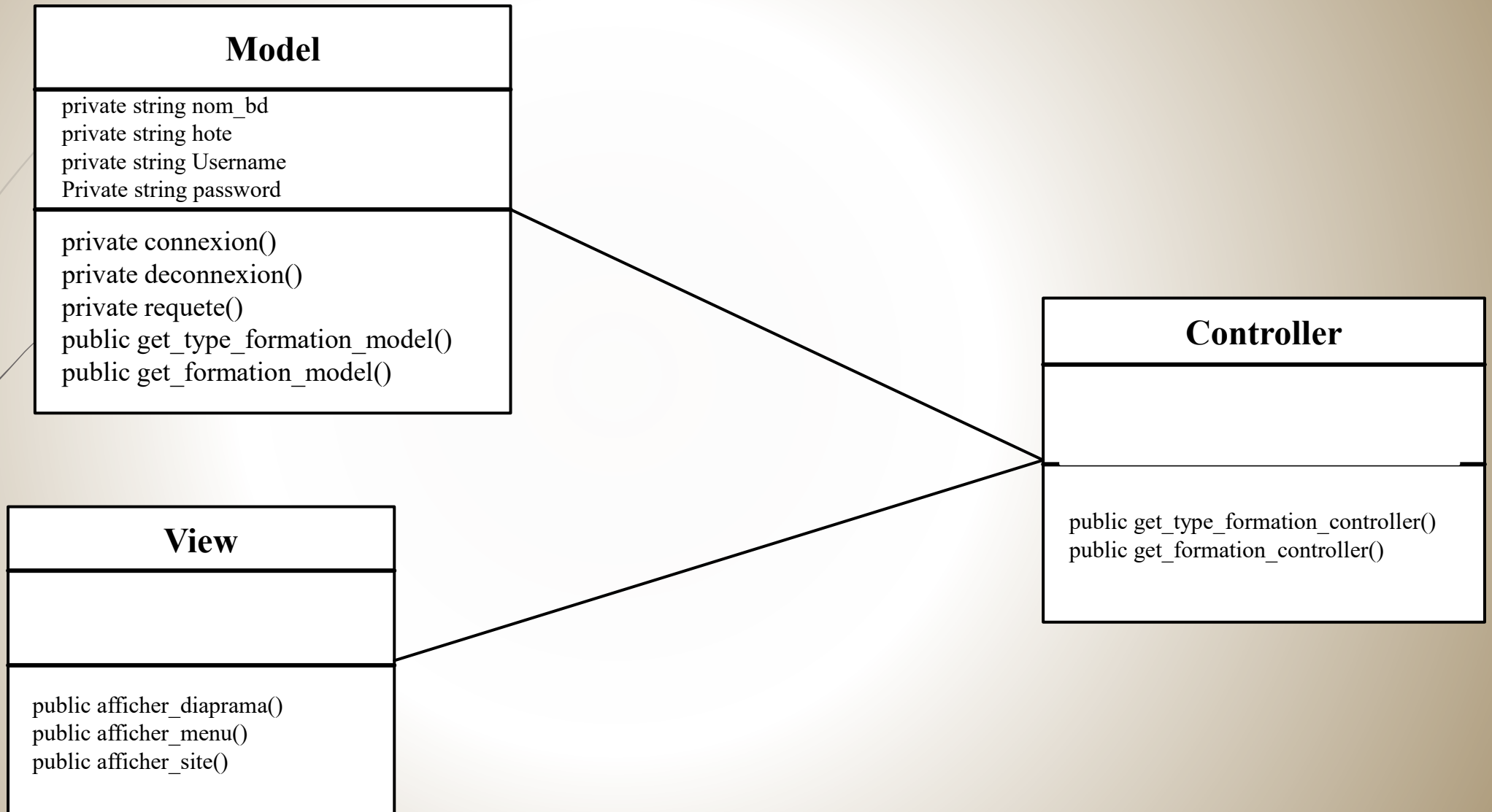
### ➤ Exemple:

1. Méthode `titre_model()`: récupère le titre de la page de la base de données
  2. Méthode `titre_controller()`: invoque `titre_model()`, fais les traitements nécessaires sur les données récupérées et renvoie le résultat obtenu.
  3. Méthode `titre_view()`: invoque `titre_controller()` et affiche les données récupérées à l'aide des langages HTML, CSS et JavaScript.
- 
- ### ➤ Si une fonctionnalité n'a pas besoin d'accéder aux données, si elle ne contient que des langages d'interfaces ou bien si elle est une méthode privée locale, alors celle-ci peut ne pas avoir de déclinaison dans toutes les classes.



# Conception d'un projet en MVC

14





# Les modèles

15

- Les modèles sont programmés uniquement en PHP (ou langage du serveur), elles n'ont pour objectif que l'accès aux données depuis ou vers la source de données (Base de données MySQL dans notre cas).
- Dans ce qui suit, nous définissons la classe `tdw_model` et les méthodes atomisées nécessaires à l'affichage du menu:

# Les modèles

16

```
<?php
require_once('controller.php');

class tdw_model{

    private $dbname="tdw_mvc";
    private $host="127.0.0.1";
    private $user="root";
    private $password="";

    private function connexion($dbname, $host, $user, $password){

        $dsn="mysql:dbname=$dbname; host=$host;";
        try{
            $c=new PDO($dsn,$user,$password);
        }
        catch(PDOException $ex){
            printf("erreur de connexion à la base de donnée", $ex->getMessage());
            exit();
        }
        return $c;
    }
}
```

# Modèle

17

```
private function deconnexion(&$c) {  
    $c=null;  
}  
  
private function requete($c, $r) {  
    return $c->query($r);  
}  
  
public function get_type_formation_model() {  
  
    $c=$this->connexion($this->dbname, $this->host, $this->user, $this->password);  
    $qtf="select * from type_formation";  
    $r=$this->requete($c,$qtf);  
    $this->deconnexion($c);  
    return $r;  
}
```

# Modèle

18

```
public function get_formation_model($v) {  
  
    $c=$this->connexion("tdw_mvc","127.0.0.1","root","");  
  
    if (isset($v)) {  
        $qf="select * from formation where Id_type_formation=" . $v . " ;";  
    } else {  
        $qf="select * from formation";  
    }  
    $r=$this->requete($c,$qf);  
    $this->deconnexion($c);  
    return $r;  
}
```

# Les contrôleurs

19

- Les contrôleurs invoquent les modèles et les vues afin de récupérer des données qu'ils vont traiter et gérer.
- Dans notre exemple, le contrôleur récupère les formations du modèle afin de les envoyer vers la vue qui les affiche dans le menu.
- Le contrôleur invoque aussi la vue lorsqu'il reçoit une requête d'accès au site afin d'afficher ce dernier:

# Les contrôleurs

20

```
<?php
require_once('model.php');
require_once('view.php');

class tdw_controller {

public function get_type_formation_controller(){
    $mtf=new tdw_model();
    $r=$mtf->get_type_formation_model();

    return $r;
}

public function get_formation_controller($v){
    $mf=new tdw_model();
    $r=$mf->get_formation_model($v);
    return $r;
}

public function afficher_site(){
    $v=new tdw_view();
    $v->afficher_site();
}
}
```



# Les vues

21

- La vue récupère des données prêtes du contrôleur et les affiche dans l'interface souhaitée.
- Dans notre exemple, les formations sont reçues à partir du contrôleur puis sont affichées à l'aide de la méthode menu:

# Les vues

22

```
Class tdw_view{  
  
public function entete_page() {  
    ?>  
    <head>  
        <meta name="Ecole" content="TDW" />  
        <meta name="Author" content="Dellys Hachemi" />  
        <link href="CSS/Exo2.css" rel="stylesheet" type="text/css" />  
        <script src="JS/jquery-3.3.1.min.js" ></script>  
        <script src="JS/Exo-JQuery.js" ></script>  
        <title> Ecole de Formation TDW </title>  
    </head>  
    <?php  
}  
  
private function titre_page() {  
    ?>  
    <h1 align="center"> Ecole de formation <abbr title="Ecole de formation TDW"> TDW </abbr> </h1></br>  
    <?php  
}
```

# Vue

23

```
private function titre_page() {
    ?>
    <h1 align="center"> Ecole de formation <abbr title="Ecole de formation TDW"> TDW </abbr> </h1></br>
    <?php
    }
private function diaporama() {
    ?>
    <div id="APdiapo">
    <div class="diapo">
    
    
    
    
    </div>
    </div>
    <?php
    }
private function description() {
    ?>
    <p class="description">TDW est une école de formation offrant diverses formations professionnelles </p>
    <?php
    }
```

```
private function menu() {  
    ?>  
    <nav><ol class="Menu">  
    <?php  
    try{  
        $cf=new tdw_controller();  
        $qtf=$cf->get_type_formation_controller();  
        foreach($qtf as $rowtf){  
            echo '<li class="type_formation"><a href="pages/' . $rowtf["Nom_type_formation"] . '.html">' . $rowtf["Nom_type_formation"] . '</a>';  
  
            $qf=$cf->get_formation_controller($rowtf['Id_type_formation']);  
            echo '<ul>';  
            foreach($qf as $rowf){  
                echo '<li class="formation"><a href="pages/' . $rowtf["Nom_type_formation"] . '.html">' . $rowf["Nom_formation"] . '</a></li>';  
            }  
            echo '</ul></li>';  
        }  
    } catch (Exception $e) { echo 'erreur' . $e->getMessage(); }  
    ?>  
    </ol></nav>  
    <?php  
}
```

# Les Routeurs

- Les routeurs sont une simulation du programme principale (le main) dans les applications web.
- Les routeurs sont généralement les pages principales des sites web et n'invoquent qu'une méthode d'un contrôleur ou d'une vue pour afficher le reste du site.

# Les Routeurs

```
<?php  
require_once('controller.php');  
  
$c=new tdw_controller();  
$c->afficher_site();
```



## ***Ecole de formation TDW***



Bureautique

Infographie

Langue

Management

Comptabilité

Anglais

Turque

Chinois

TDW est une école de formation offrant diverses formations professionnelles dans les domaines informatique, finance et langues.



# Screen Recording

Select this object and click the Screen Recording button to edit

TDW est une école de formation offrant diverses formations professionnelles dans les domaines informatique, finance et langues.

Bureautique   Infographie   Langue   Management   Comptabilite

**Liste des tarifs:**

Formations	Volumes horaires (H)	Taxe (%)	Prix HT (DA)	Prix TTC (DA)
Bureautique	17	0	20000	23400
Infographie	17	0	20000	23400
Langues	120	0	10000	
Management	19	0	25000	30000
Comptabilite	19	0	25000	29750

**Formations TDW teaser**

## Objectif 9

- Transformez le menu de l'application web du laboratoire pour qu'il respecte une architecture en MVC.
- Les items du menu devront être récupérer de la Base de données

30

# Merci

[h\\_dellys@esi.dz](mailto:h_dellys@esi.dz)