

**Base de données**

**IFT-2004**

**Connexion à une BD par le développement d'une application (PHP)**

**Marc Philippe Parent**

**Département d’informatique**

**et de génie logiciel**



**Objectifs**

* **Historique**
* **Présentation**
* **Instructions**
* **Structures de contrôle**
* **Classe et objets**
* **HTML**
* **PHP et Oracle.**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **2**



**Histoire et origine de PHP**

* **1994 Rasmus Lerdorf**
* **|9:4| : invente PHP le langage voulait dire personnal homepage.**
* **Trace \_\_visiteurs\_\_\_\_\_**
* **|9:4| : Il voulait une trace des visiteurs sur son site.**
* **1995 publication \_\_PHP/FI\_\_ Personal homepage form interpreter**
* **1997 Andi Gutmans et Zeev Suraski**
* **|9:5| : Ils ont redevelopper le coeur de PHP**
* **1998 PHP 3**
* **PHP 4 \_Zend\_\_ Engine**
* **2010 75%**
* **2016 \_82%\_**
* **Août 2017 PHP ₇.1.8\_\_\_**

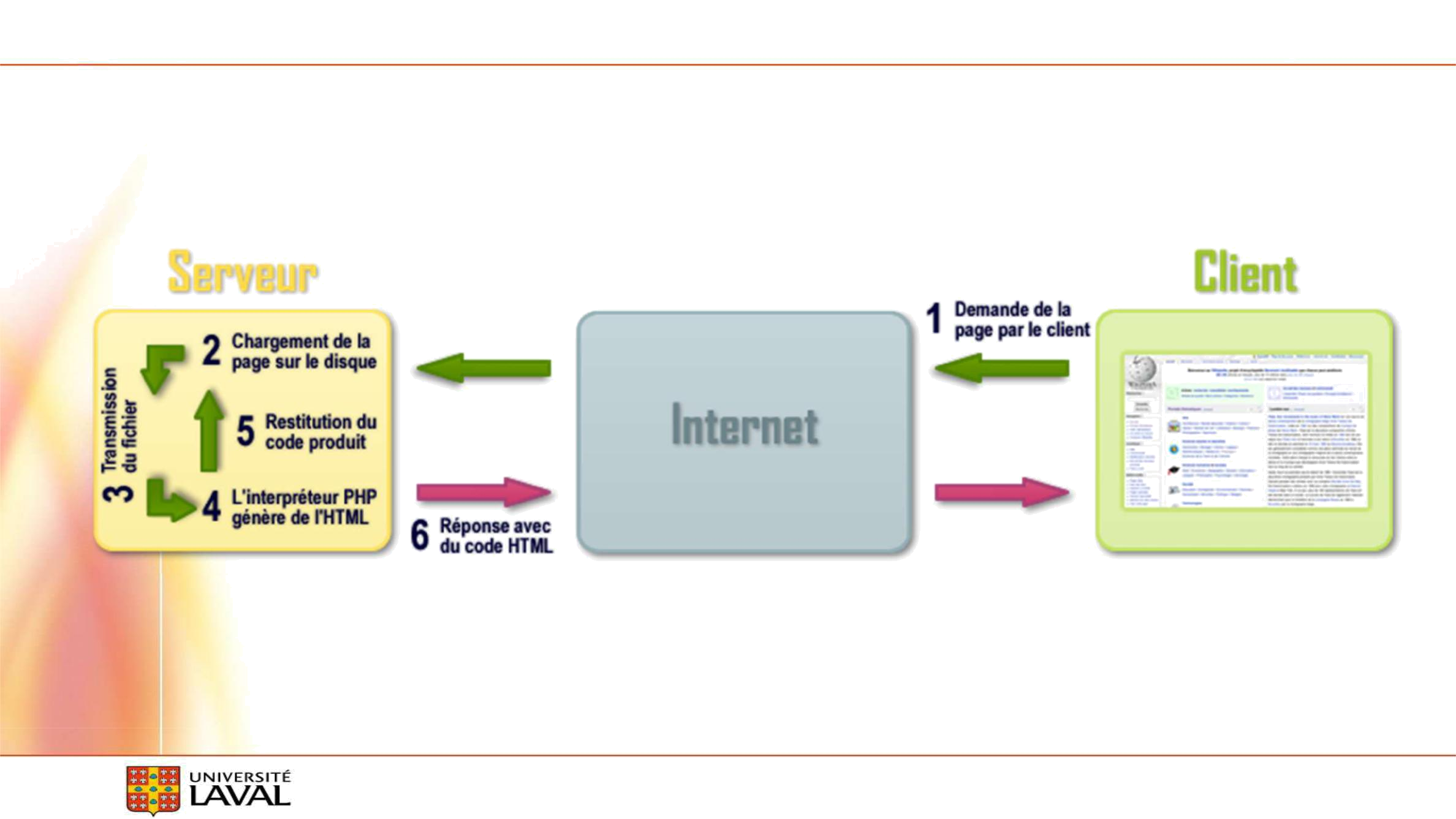
**Département d’informatique et de génie logiciel – Marc Philippe Parent** **3**



**Présentation PHP**

* **Côté serveur**
* **Application web dynamique**
* **Peu typé et souple**
* **Formulaire et BD**
  + 1. **modules BD**
* **\_\_\_\_\_multi-plateforme\_\_\_\_\_\_\_**
* **Libre**
* **Descendant du C**
* **Facilement mixé avec HTML**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **4**



**Fonctionnement**

* **Aussi**
  + **\_\_\_\_\_\_**
  + **\_\_\_\_\_\_\_.**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **5**



**Instructions**

**<?php**

**|9:15| : Le code qui suis est du PHP**

**// forme la plus simple, recommandée**

**echo 'Hello World';**

**<?php**

**echo 'Hello ', 'World';**

**// \_\_\_\_concaténation\_\_\_\_\_ avec un point**

**echo 'Hello ' . 'World';**

**Il est aussi possible d'utiliser la version raccourcie :**

**<?= 'Hello World' ?>**

**Résultat affiché :**

**Hello World**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **6**



**Instructions**

* **\_Balises\_\_\_\_**

**<?php … ?>**

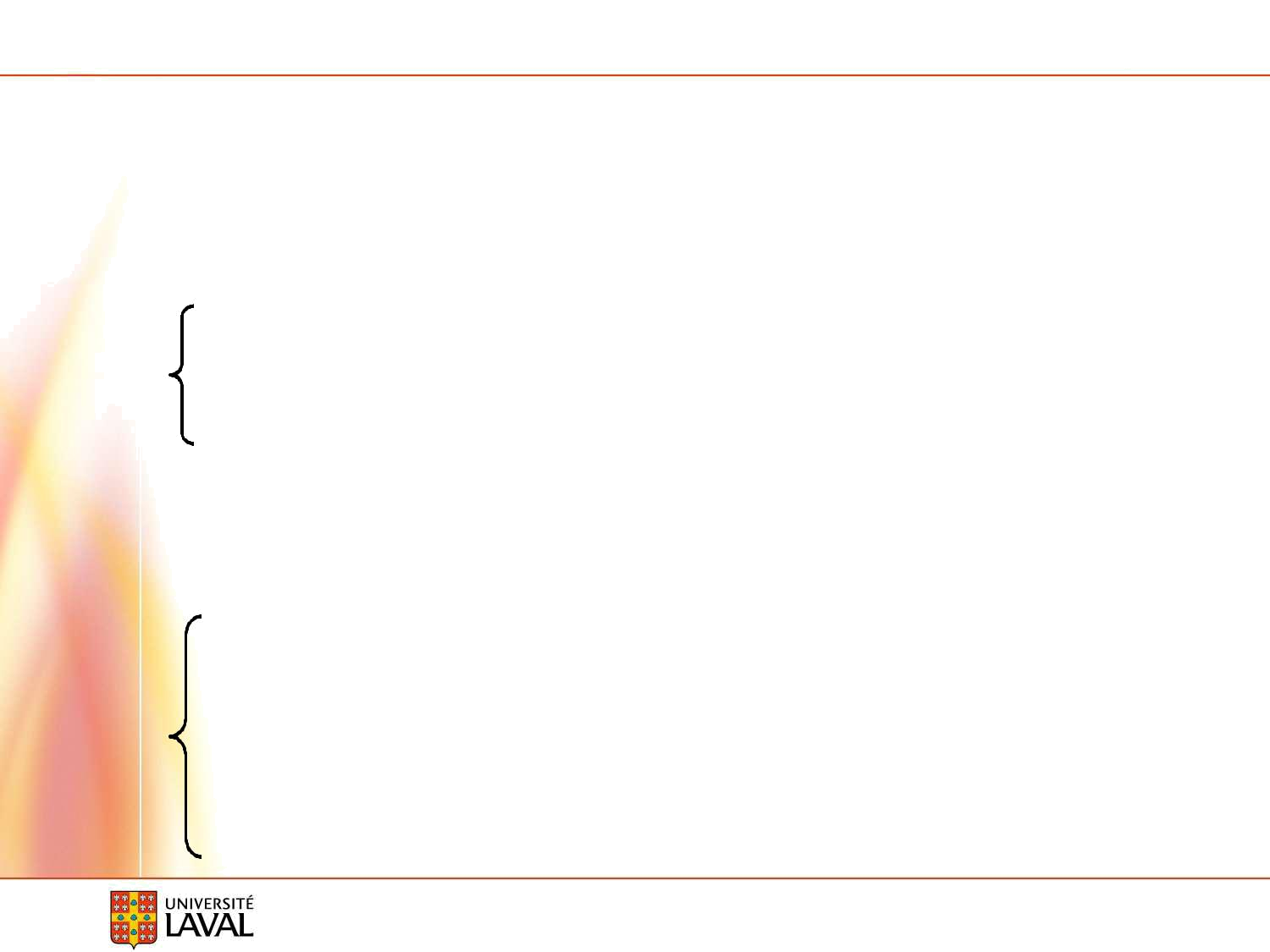
**|9:18| : Identifier le code php a l’intérieur de la page html**

* **Autres**
  + **<? … ?>**
  + **Notation ASP <% … %>**
  + **<script language="php"> … </script>**
* **Séparées par \_ ;\_\_**
* **Générer HTML**

**$nom = 'Albert';**

**echo '<div>Bonjour ' . $nom . ' !</div>';**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **7**



**Structure de contrôle**

* **Conditionnelles**

**If**

**switch**

* **Boucles**

**For**

**While**

**do while**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **8**



**Structures de décision**

**<?php**

* **$\_POST → valeurs du \_\_\_formulaire\_\_\_\_\_**

**$lang = strtolower($\_POST['lang']);**

**|9:38| : il y a un champ langue ou l’utilisateur peut inscire sa langue ex : fr**

**|9:38| : strtolower, mettre en minuscule.**

|9:39| : Idéalement on met des accolades entre chaque condition { } if (ok) {

**if ($lang === 'fr')**

**echo 'Vous parlez français !';**

**elseif ($lang === 'en')**

**|9:39| : différence entre double egal et triple egal, le triple egal ne va pas convertir, si la valeur est null, il ne verra pas la différence. (C’est a tester c’est bizarre.**

**echo 'You speak English!';**

**else**

**echo 'Je ne vois pas quelle est votre langue !';**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **9**



**Structures de décision**

**<?php**

**$nomsAutorises = [ 'Albert', 'Bertrand' ]; $nomEnCours = 'Eve';**

**?>**

**<?php if (in\_array($nomEnCours, $nomsAutorises)): ?>**

**|9:42| : Fonction in\_array, param : valeur a trouver et l’array.**

**<p>Bonjour <?php echo $nomEnCours; ?> !</p>**

**<?php else: ?>**

**<p>Vous n'êtes pas un utilisateur autorisé !</p> <?php endif; ?>**

**|9:43| : Utiliser les accolades c’est mieux.**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **10**



**Structures de répétition**

**\_for\_\_ (exp1; exp2; expr3){**

**instructions**

**}**

**\_while\_\_ (condition) {**

**instructions**

**}**

**\_do\_ {**

**instructions**

**} \_\_while\_\_\_ (condition);**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **11**



**Structure de répétition**

**<?php**

**// exemple for**

**for ($i = 1; $i <= 10; $i++) {**

**echo "numéro " . $i . "<br>";**

**}**

**//exemple while**

**$i = 1;**

**while ($i <= 10) {**

**echo "numéro " . $i++ . "<br>";**

**/\* La valeur affichée est $i avant l'incrémentation (post-incrémentation) \*/**

**}**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **12**



**Objets**

* **Créer classe comme C++ et Java**
* **Héritage \_simple\_\_\_**
* **|9:46| : ll n’y a pas d’héritage multiple comme en C++**

**class Perso {**

**const PV\_INITIAL = 2000;**

**private $pv;**

**Figure**

**perimetre;**

**surface;**

**calculPer( );**

**calculSur( );**

**public function \_\_construct($pv = false) { // Paramètre optionnel if (!is\_numeric($pv) || $pv < 0 || $pv > 100000000)**

**$this->pv = self::PV\_INITIAL;**

**else**

**$this->pv = $pv;**

**}**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **13**

**Objets**

**// \_\_Accesseurs\_\_\_\_\_\_\_\_**

**public function getPv() {**

**return $this->pv;**

**}**

**public function isDead() { return $this->pv == 0;**

**}**

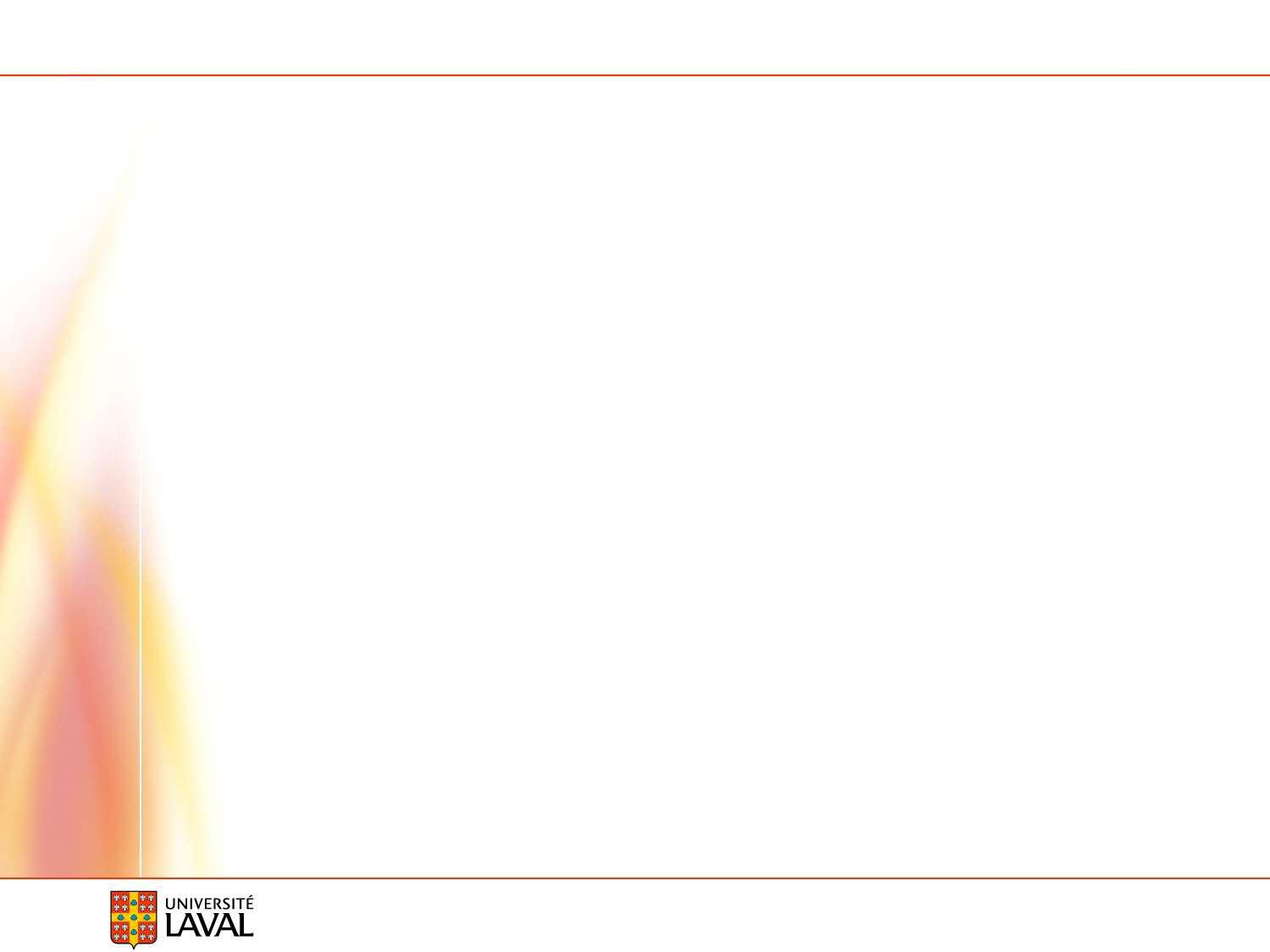
**}**

* **Création d'une classe enfant de Perso**

**class Magicien extends Perso { private $magie;**

**}**

* **Création d'une instance de classe $perso = new Perso(1000);**
* **Utilisation de l'objet**



**echo 'Votre personnage a ' .**

**$perso->getPV() . ' PV.';**

**// Constantes de classes**

**echo 'Le PV par défaut attribué à un '**

**. 'nouveau personnage est de ' .**

**Perso::PV\_INITIAL . '.';**

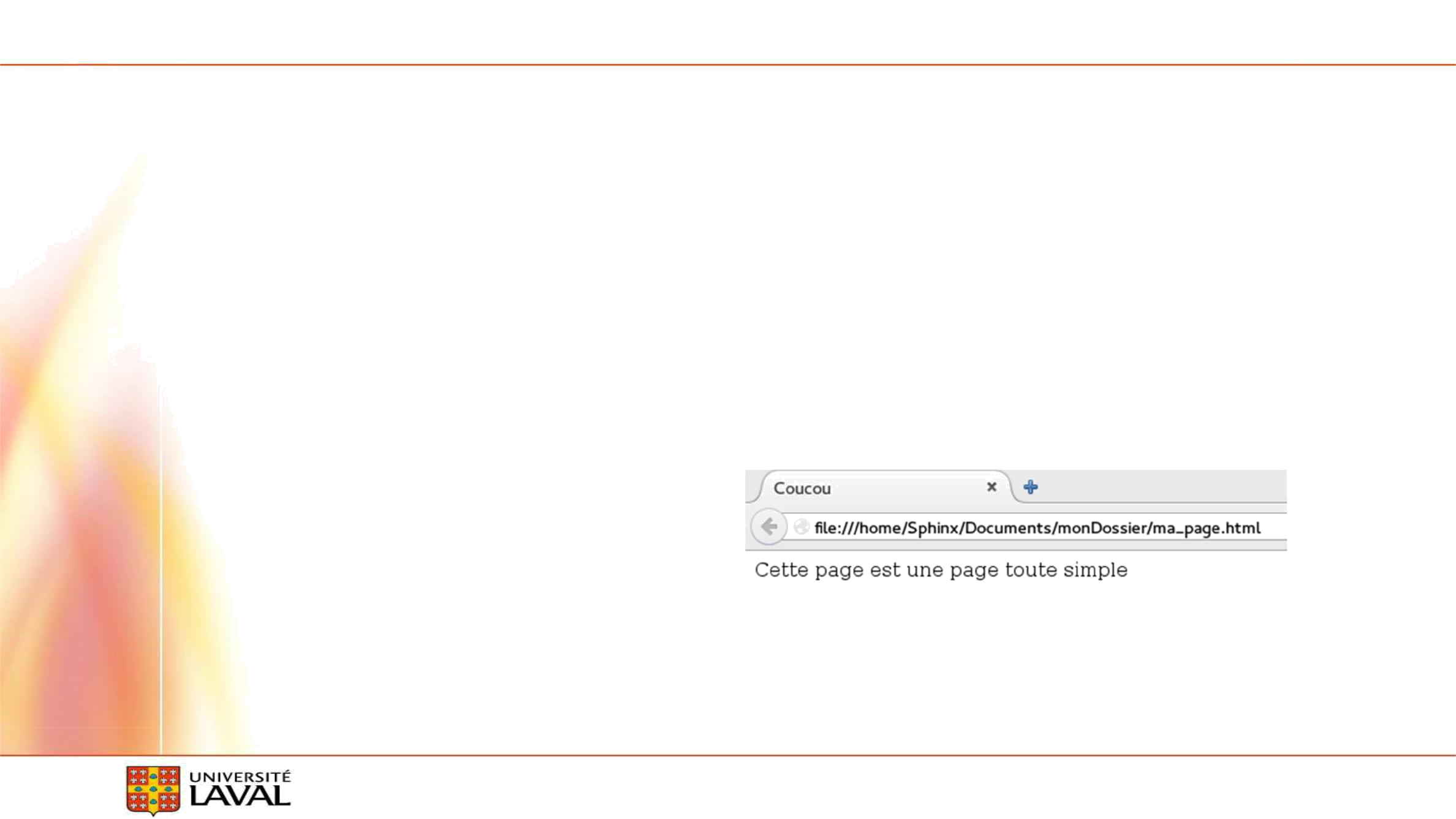
* **Destruction de l'objet unset($perso);**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **14**



**HTML**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **15**



**HTML**

* **Stricte \_minimum\_\_\_\_**
* **Textes, images, hyperliens, styles**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Coucou</title>**

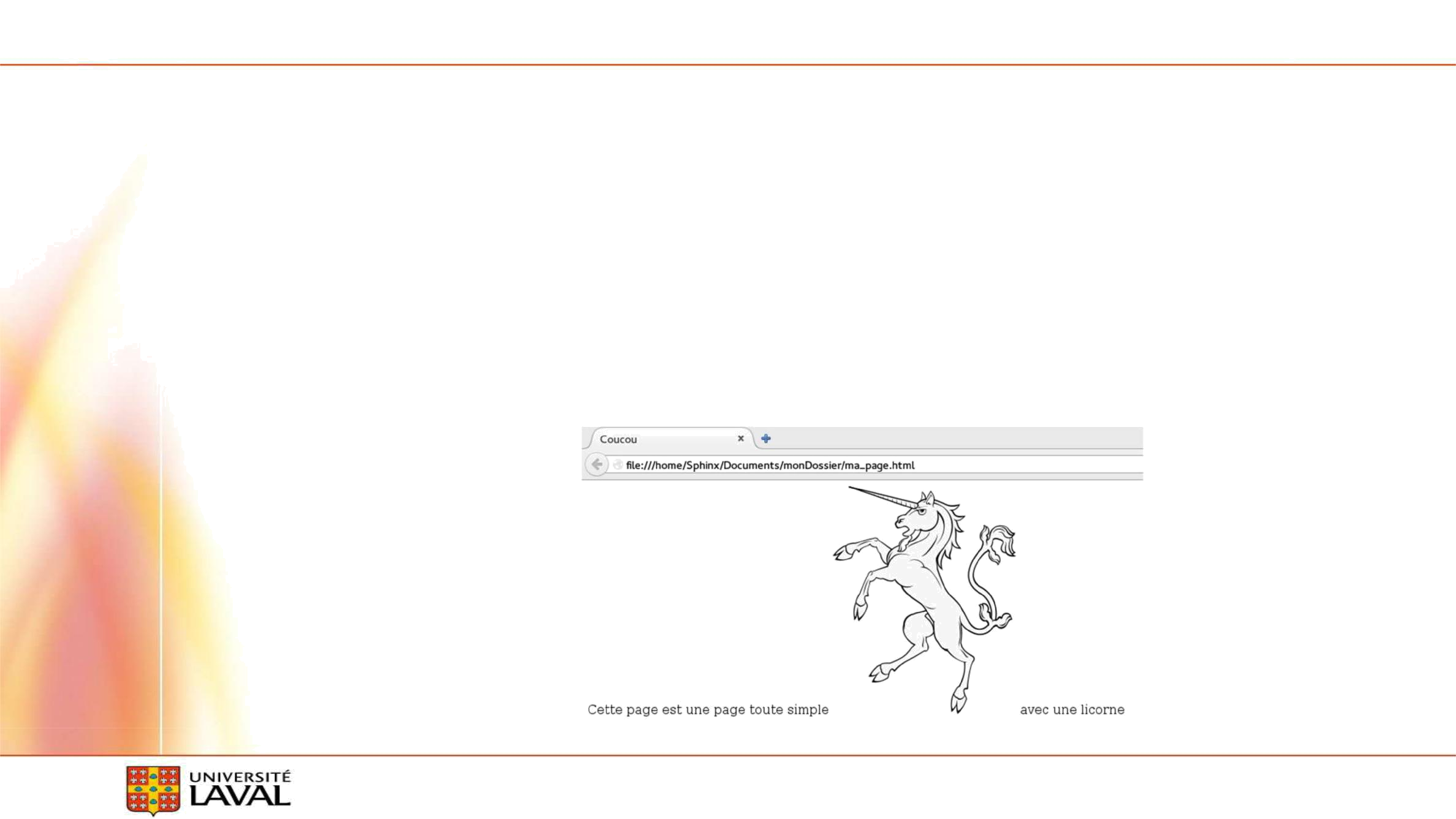
**</head>**

**<body>**

**Cette page est une page toute simple </body>**

**</html>**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **16**



**Image**

**…**

**<body>**

**Cette page est une**

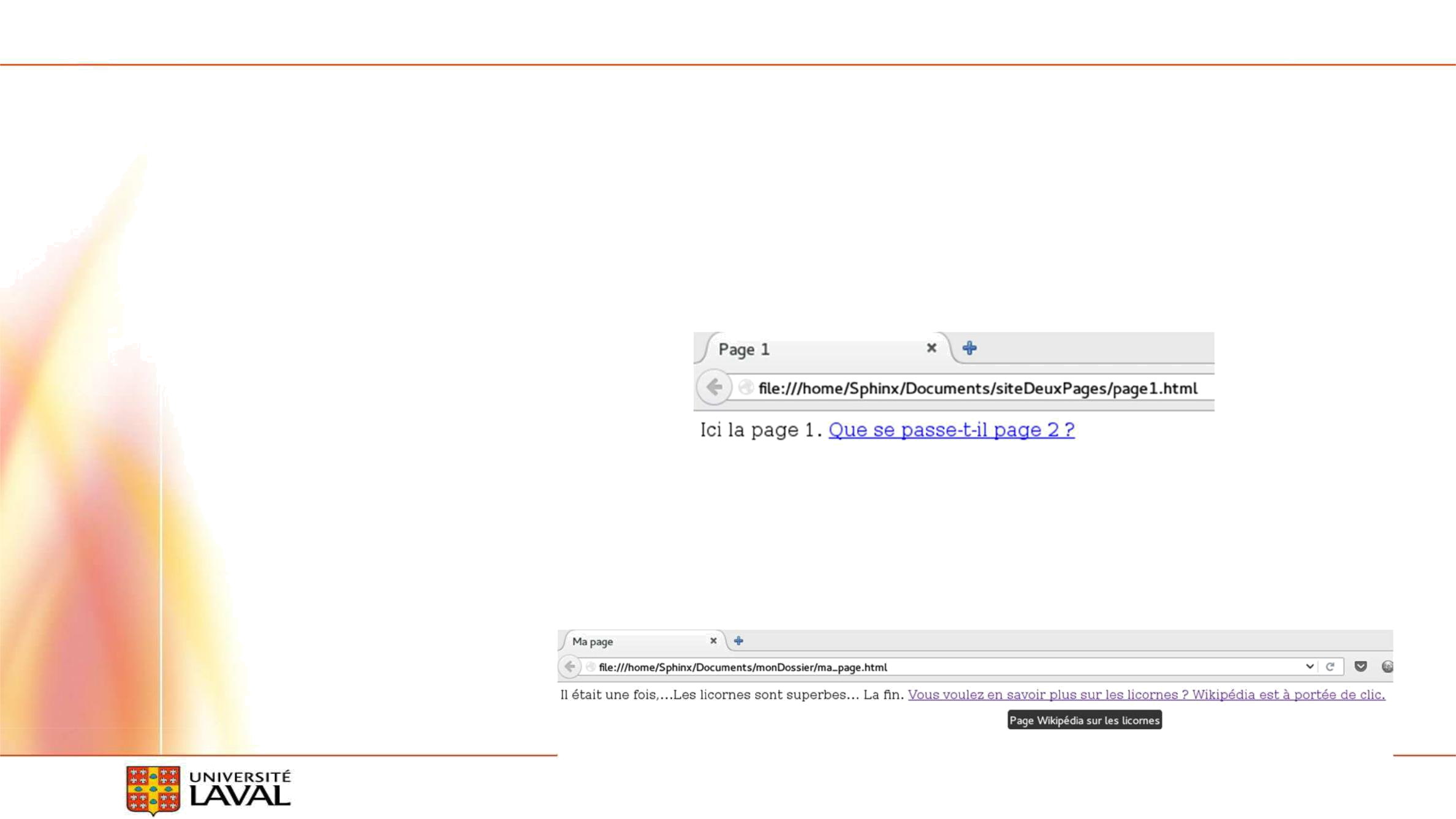
**page toute simple**

**<img \_\_src\_="licorne.png" alt="Une image de licorne :)" /> avec une licorne**

**</body>**

**…**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **17**



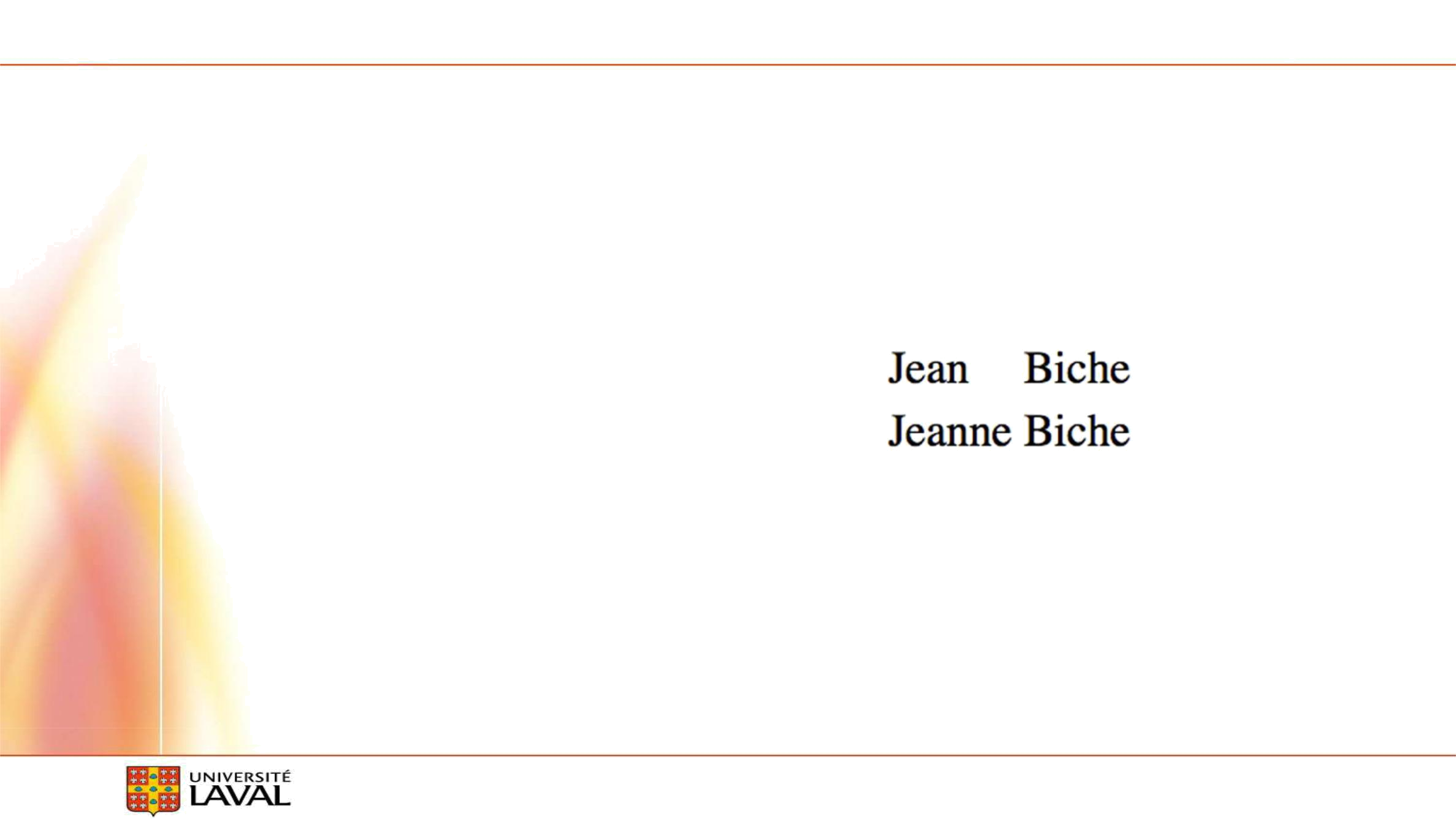
**hyperlien**

**Ici la page 1.**

**<a \_href\_\_\_="page2.html" title="Vers la page 2">Que se passe-t-il page 2 ?</a>**

**<a href="https://fr.wikipedia.org/wiki/Licorne" title="Page Wikipédia sur les licornes">Vous voulez en savoir plus sur les licornes ? Wikipédia est à portée de clic.</a></body>**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **18**



**Tableau**

**<table>**

**<tr>**

**<td>Jean</td>**

**<td>Biche</td>**

**</tr>**

**<tr>**

**<td>Jeanne</td>**

**<td>Biche</td>**

**</tr>**

**</table>**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **19**



**Formulaire**

**<form method="post" action="cible.php">**

**<p>**

**Votre nom: <input type="text" name="nom" value="" />**

**<input type="submit" value="Valider" />**

**</p>**

**</form>**

* **Dans \_\_cible.php\_\_\_\_\_**

**<p>Bonjour <?php echo $\_POST['nom']; ?> !</p>**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **20**



**Exercices**

* **À l'origine que voulait-on faire avec PHP?**
* **A-Construire son CV sur un site**
* **B-Compter les visiteurs d'un site**
* **C-Compter les pages d'un site**

**D-Créer un projet étudiant E-Aucune de ces réponses**

* **Qu'elles sont les similitudes avec C++?**

**A-structures conditionnelles et répétitives, héritage**

**B-Déclarations variables, paramètres, structures conditionnelles et répétitives**

**C-Déclarations variables, syntaxe générale, paramètres**

**D-Même famille, héritage, structures conditionnelles et répétitives**

**|10:12| : en C++ on peut faire héritage multiple mais pas en PHP**

**E-Même famille, classes et objets, structures conditionnelles et répétitives**

**F-Toutes ces réponses**

**G-Aucune de ces réponses**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **21**



**PHP + SQL**

* **Exécuter SQL dans PHP**
* **Fonctionnement**

**1-Établir \_\_\_connexion\_\_\_\_ à la BD**

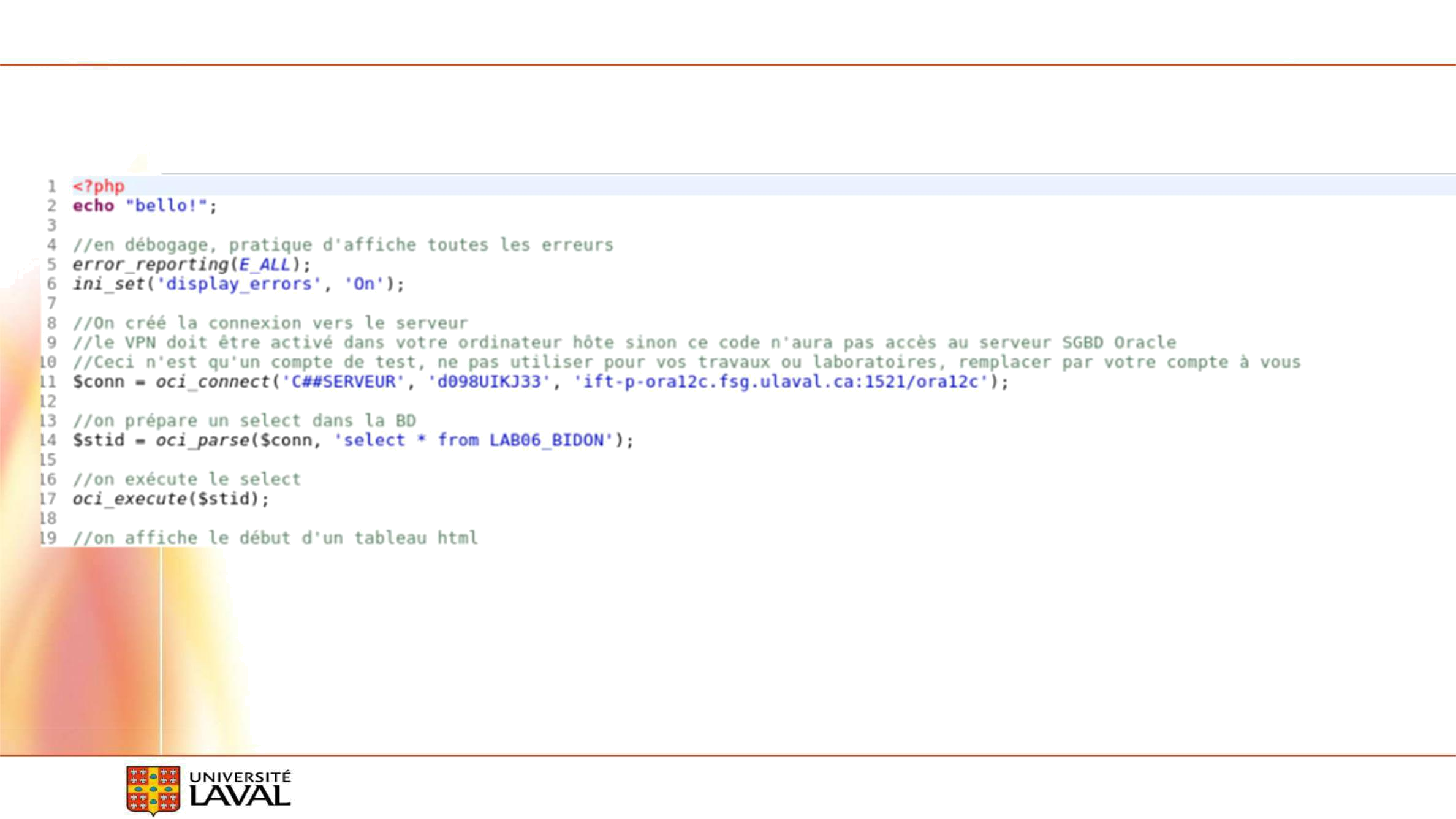
**2-Envoyer instructions \_SQL\_**

**3-Traiter \_\_\_résultat\_\_\_**

**//Répéter 2 et 3 au besoin**

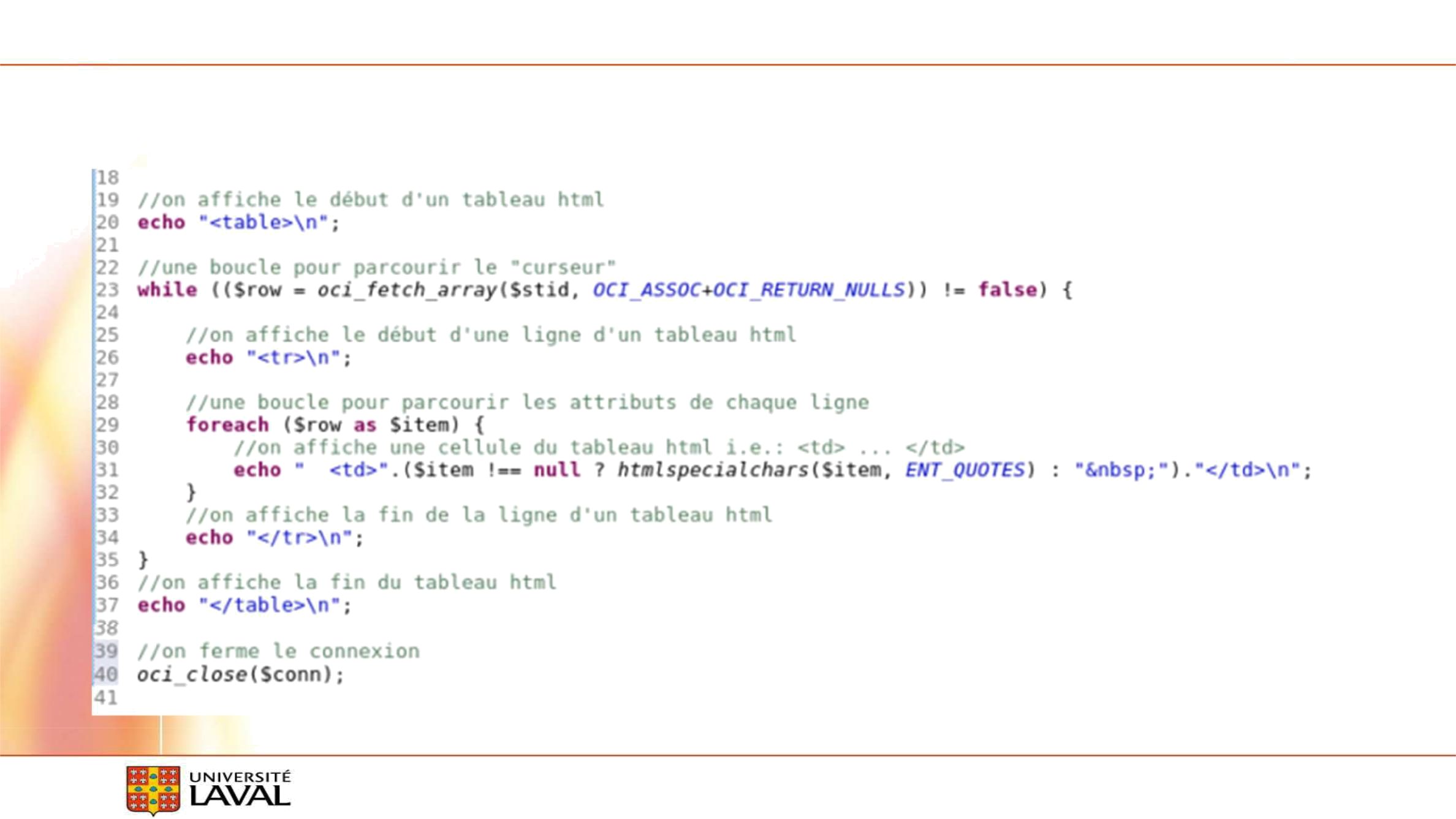
**4-\_\_Fermer\_\_ connexion.**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **22**



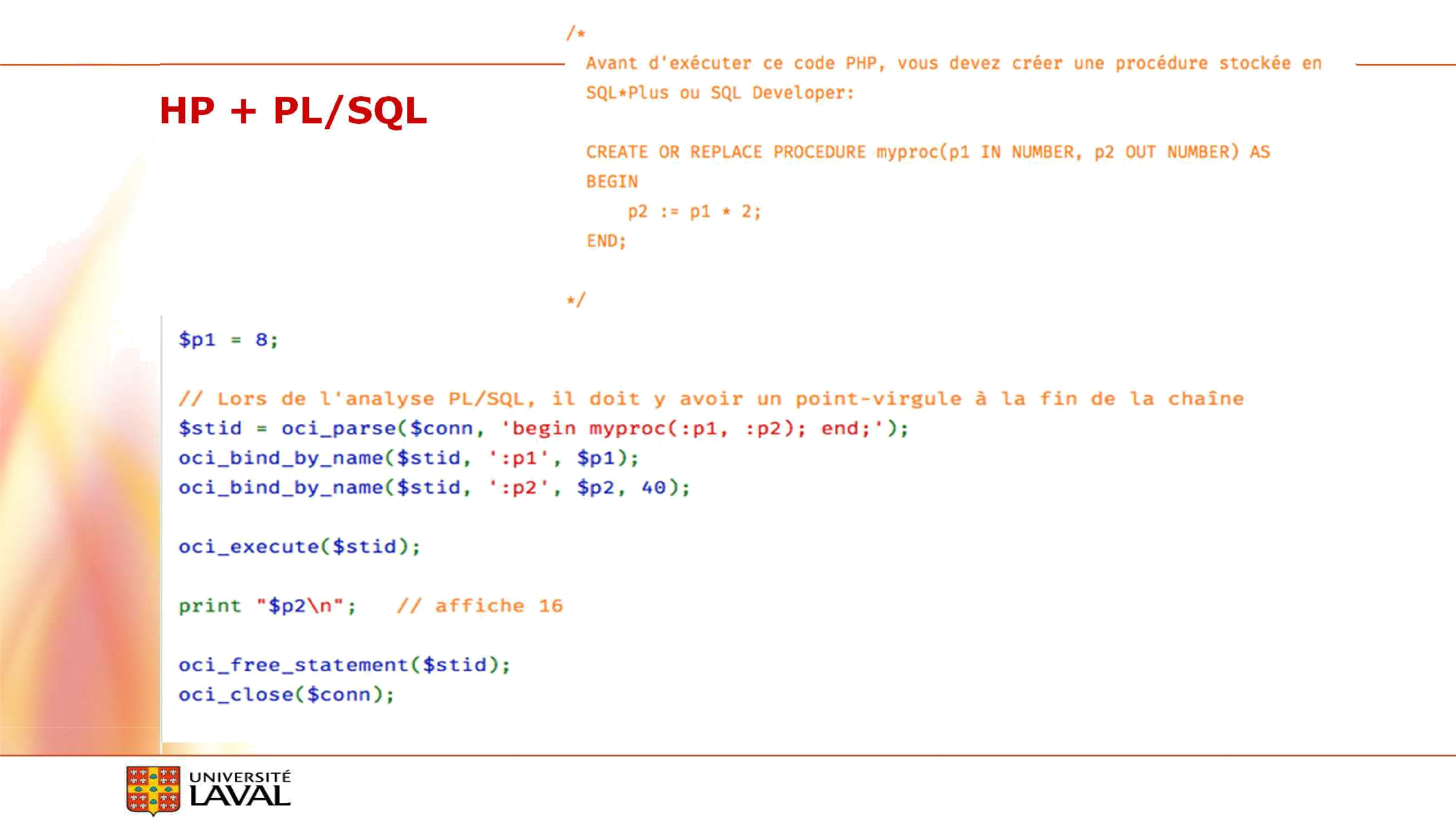
**PHP + SQL**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **23**



**PHP + SQL**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **24**



Activer la validation automatique dans sqldevelopper

**PHP + PL/SQL**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **25**



**Machine virtuelle**

* **\_VPN\_ : Important d’être connecté.**

**Dans le shell de linux :**

* **service httpd start : Va démarrer le serveur de BD.**
* **chkconfig --level 2345 httpd on : Va redémarrer le serveur de bd.**
* **Ouvrir le port 80. firewall-cmd –zone=public –add-port=80/tcp – permanent**
* **$ firewall-cmd – reload**
* **Eclipse \_\_\_\_\_\_**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **26**



**Exercices**

* **Quels sont les 4 étapes d'interaction PHP et base de données?**

**A-Établir connexion, répéter(envoyer instructions, traiter résultats), Fermer connexion**

**B-Établir connexion, répéter(envoyer résultats, traiter instructions), Fermer connexion**

**C-répéter(Établir connexion, envoyer instructions, traiter résultats, Fermer connexion)**

**D-Établir connexion, répéter(envoyer instructions, traiter résultats, Fermer connexion)**

**E-Aucune de ces réponses**

* **Quel commande PHP permettent de préparer des commandes SQL et/ou PL/SQL? A-oci\_connect**

**B-oci\_fetch\_array C-oci\_bind\_by\_name D-oci\_execute E-oci\_parse**

**F-Aucune de ces réponses G-Toutes ces réponses**

**Département d’informatique et de génie logiciel – Marc Philippe Parent** **27**