UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Systèmes d'Information & Informatique

PHP Avancé

Sessions

Orientation à objets

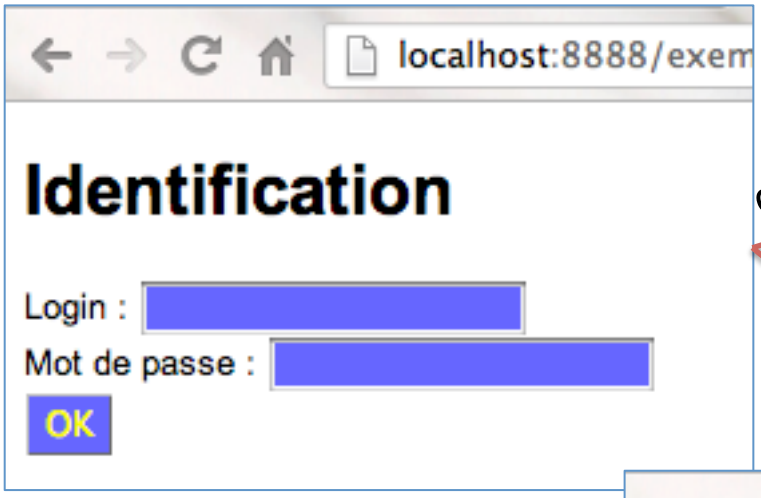
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Sessions

- **Mécanisme de sessions**
  - Chaque visite à un site / page est indépendante
  - Les **sessions** permettent de conserver les informations des visiteurs **entre les pages**
  - Les informations sur les sessions sont stockées sur le **serveur**
- Fonctionnement général
  1) Ouverture de session : **session_start ()**
     - Chaque utilisateur reçoit un identifiant transmis entre les pages
  2) Définition des variables de sessions (données)
     - Les variables de session sont transmises de page à page
     - **$_SESSION["variable"] = valeur ;**
  3) Fermeture de session : **session_destroy()**

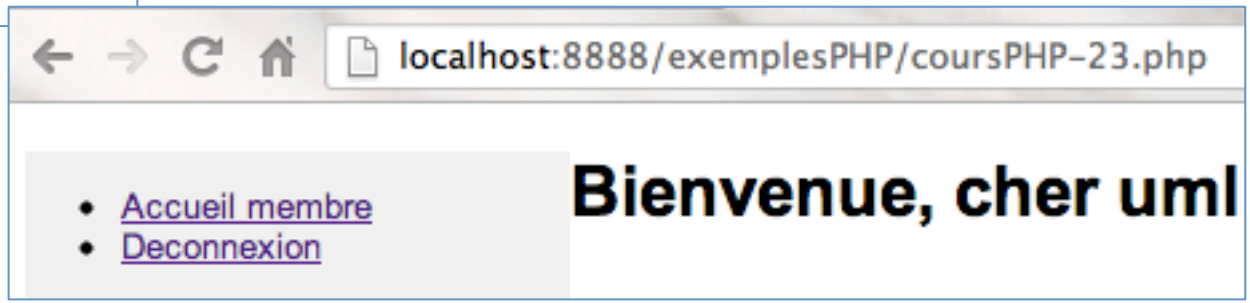
UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Sessions



Login & mdp différents de **uml**

Login & mdp corrects
(**uml /uml**)




```
<form name="…"
 action="coursPHP-23.php"
 method="POST">
   <label >Login : </label>
   <input type="text" name="login" maxlength="15" /> <br/>
   <label >Mot de passe : </label>
   <input type="password" name="mdp" maxlength="15" />
      <br/>
   <input type="submit" value="OK" />
</form>
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Sessions

```
<?php  session_start();  ?>
<html>
  <head> … </head>
  <body>
  <?php
   . . .
      $login = $_POST["login"] ;
      $mdp = $_POST["mdp"];

      if ( $login == "uml" AND $mdp == "uml") {
      $_SESSION["login"] = $login ;
      …
      echo "<h1>Bienvenue, cher $login </h1>" ;
      }
      else {   echo "<h1>Desolé ! </h1>";
          echo "<p> Page accessible uniquement aux membres. </p>";
       }
?>
</body> </html>
```

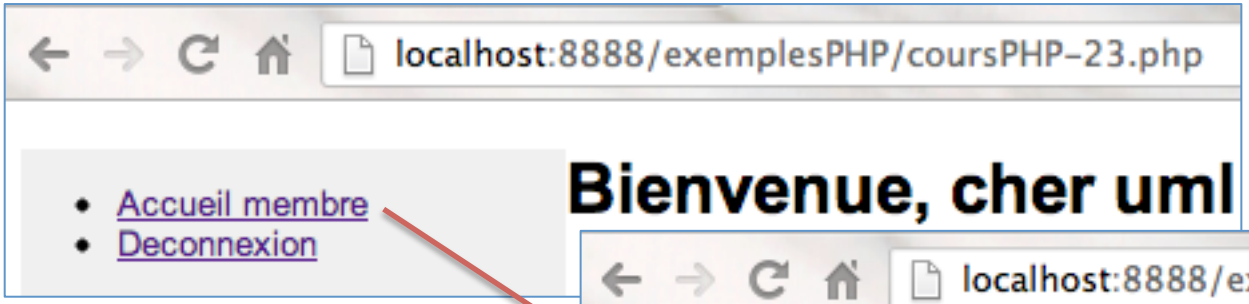
Ouverture d'une session
(au début de **chaque page**)

Définition des variables de session
$_SESSION["var"]

Les variables de session contiennent les informations qui passeront de page en page.

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Sessions

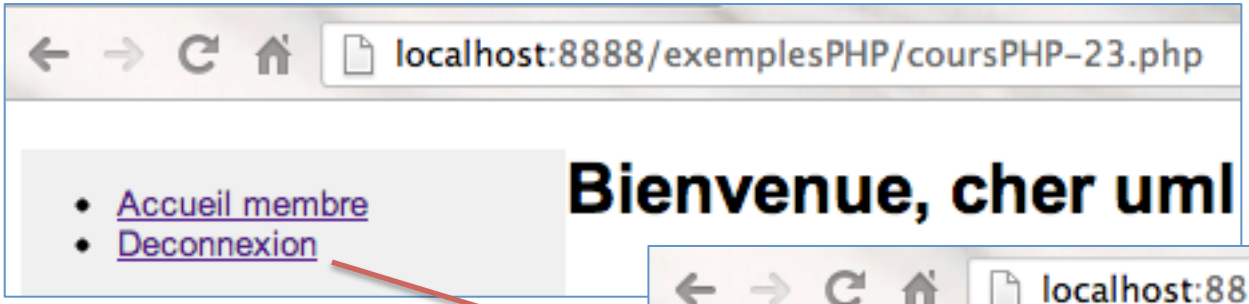


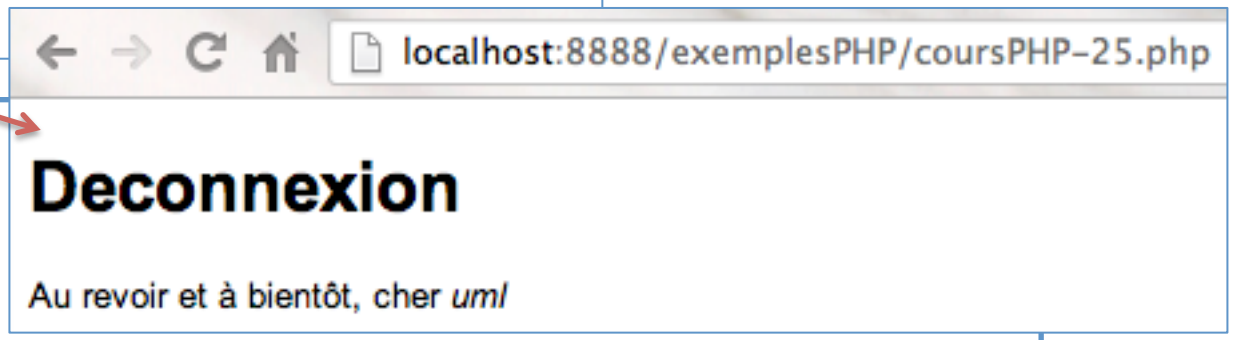

```
<?php session_start(); ?>
<html> <head>… </head>
<body>
<?php
   if ( isset( $_SESSION["login"] ) AND ! empty( $_SESSION["login"] ) ) {
       $login = $_SESSION["login"] ;
…
       echo "<p>Client <b> $login </b> :  Ceci est une
   }
   else {     echo "<h1>Desolé ! </h1>";
        echo "<p> Il s'agit d'une page privée !! Il faut être membre. </p>";
 }
?>
…
```

Usage des variables de session
$_SESSION["var"]

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Sessions






```
<?php session_start(); ?>
<html> <head>… </head>
<body>
<?php
   if ( isset( $_SESSION["login"] ) AND ! empty( $_SESSION["login"] ) ) {
…
        unset($_SESSION["login"]);
        session_destroy();
   }
   else {     echo "<h1>Desolé ! </h1>";
        echo "<p> Pas de connexion active. </p>";
  }
?>
…
```

Fermeture de la session
session_destroy()

Ne pas oublier de vider les variables de session
unset($_SESSION["var"])

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Exemple Sessions & BdD

- **Comment relier l'exemple à une vraie BdD ?**
  - Réaliser l'authentification à partir des informations (login/mot de passe) qui sont dans la Base de Données
- **Ajouter les étapes pour consulter une BdD**
  - on établit la **connexion**
  - on **cherche** un utilisateur avec ce mot de passe dans la BdD (**requête**)
  - on analyse des **résultats** obtenus
  - on **ferme** de la **connexion**

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Exemple Sessions & BdD

- Accueil membre
- Deconnexion

Bienvenue, cher toto@gmail.com

On ajuste la session en fonction du résultat

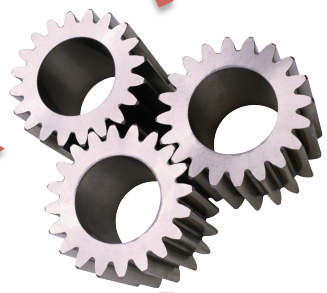
Exemple Session | MAM

Identification

Login : toto@gmail.com

Mot de passe : ••••

OK



PHP
authentification.php

On récupère les informations de la BdD

+ Options

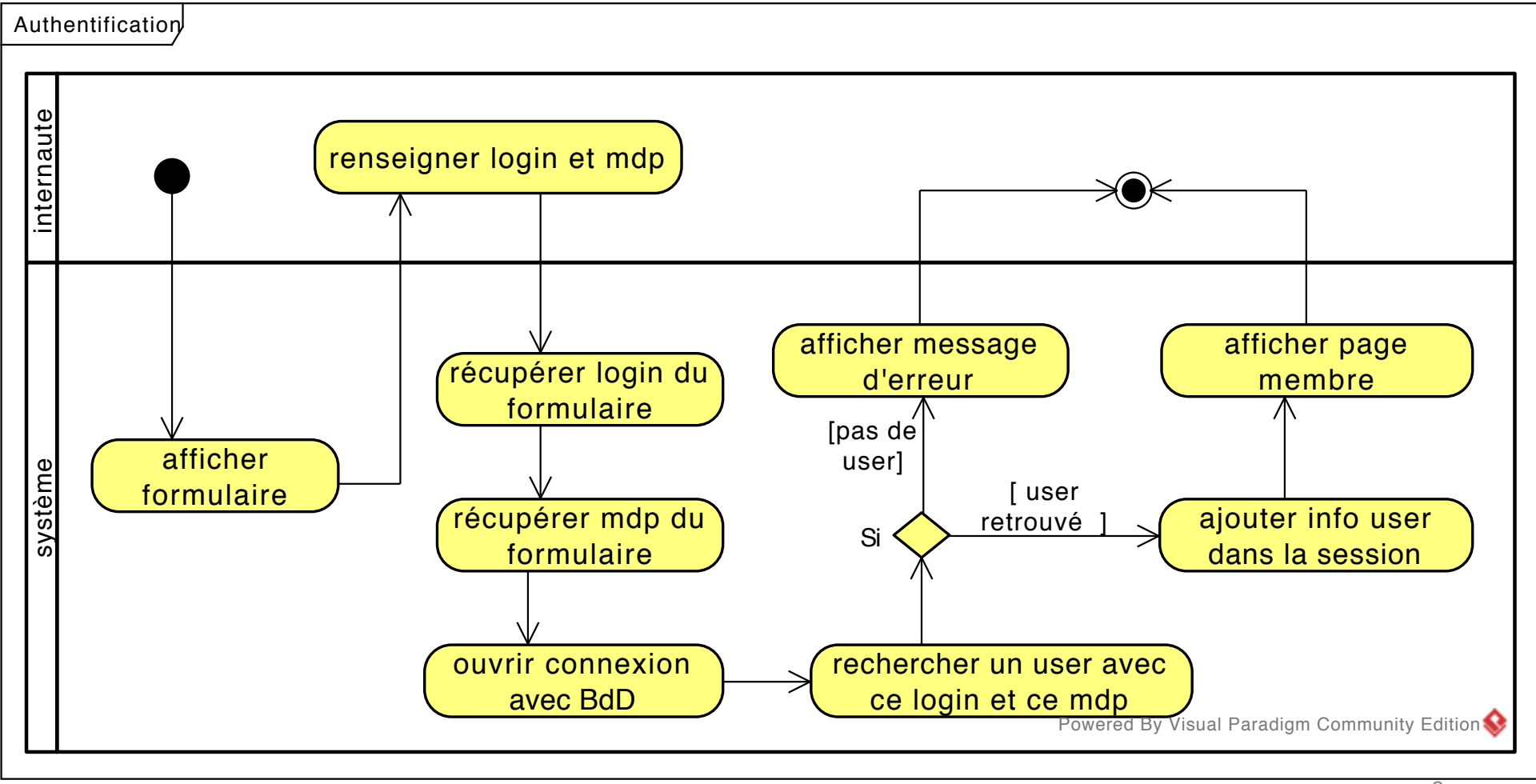
	email	nom	mdp	adresse
Modifier Copier Effacer	toto@gmail.com	Toto	toto	paris

Tout cocher *Pour la sélection :* Modifier Effacer Exporter

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

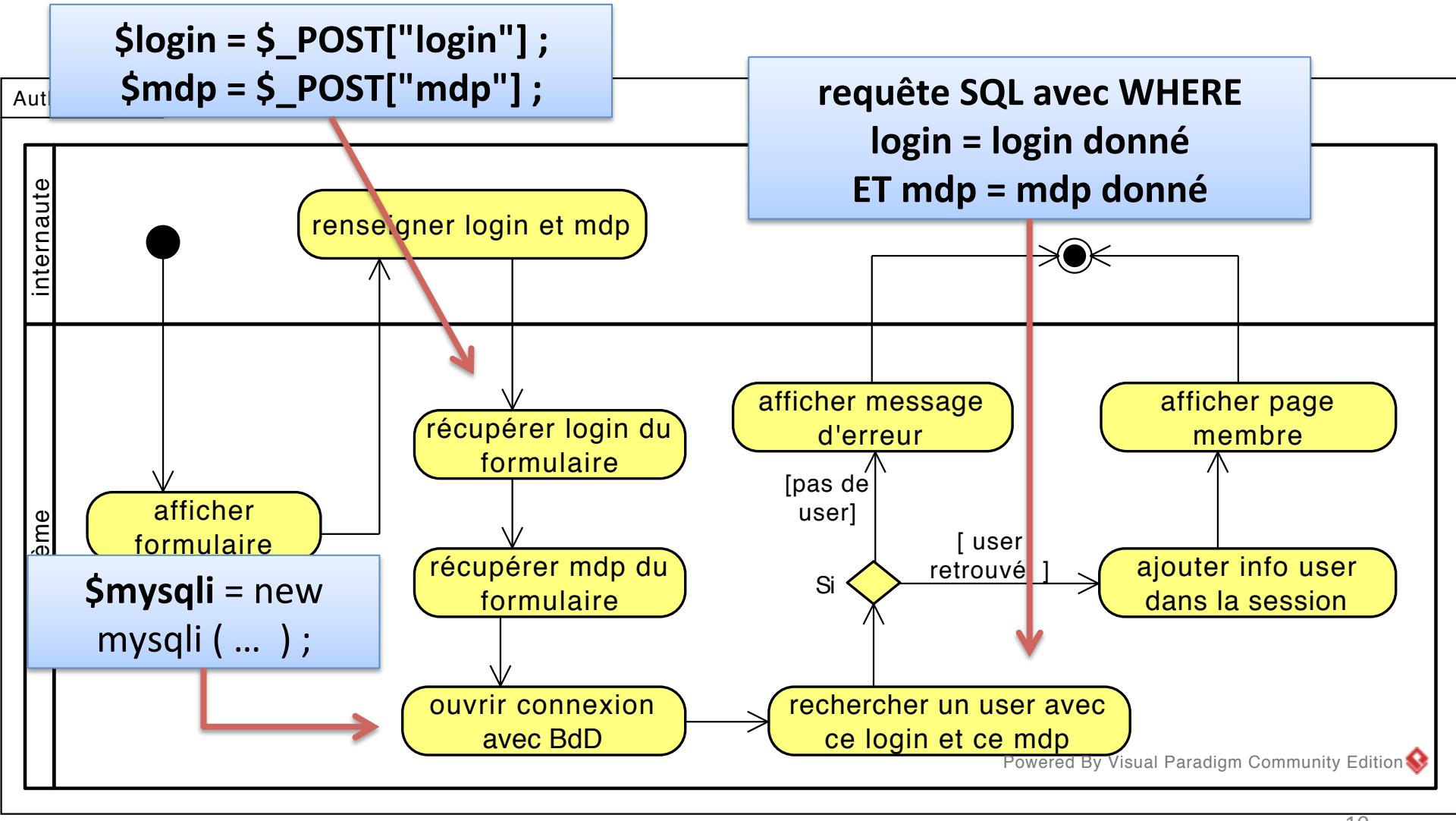
Exemple Sessions & BdD

- Algorithme « authentification »



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Exemple Sessions & BdD



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Exemple Sessions & BdD

```
<?php  session_start();  ?>
<html>
  <head> … </head>
  <body>
  <?php
   . . .
      $login = $_POST["login"] ;
      $mdp = $_POST["mdp"];

     $user="uml";
     $host="localhost";
    $password="uml";
     $database="Produits";

     $mysqli = new mysqli ( $host, $user, $password, $database ) ;

     if ($mysqli->connect_errno) {
          die ("Echec lors de la connexion à MySQL : " . $mysqli->connect_error);
     }
```

Début de la page reste le même

Avant de définir le **$_SESSION,**
on va **ouvrir la connexion** avec la **BdD**
et rechercher un user avec ce login

1ère étape : **ouverture de connexion**

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

Exemple Sessions & BdD

```
. . .
    $sql = "SELECT * FROM client " .
            " WHERE email='".$login . "' AND  mdp='".$mdp."'" ;

     $result = $mysqli->query($sql);


     if ( $result->num_rows > 0 ) {
        $_SESSION["login"] = $login ;
         ...
        echo "<h1>Bienvenue, cher $login </h1>" ;
     }
     else {   echo "<h1>Desolé ! </h1>";
          echo "<p> Page accessible uniquement aux mer
      }

    $mysqli->close();
?>
```

2ème étape : **la requête SQL**

attention aux « ' » **avant** et **après** les **valeurs**

3ème étape : **les résultats**

Si on a des résultats, on a notre user…
un user avec ce login et ce mdp

4ème étape : **fermer la connexion**

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Orientation à objets

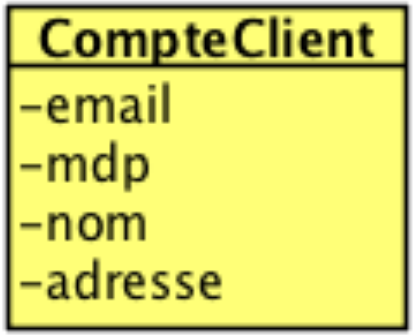
- Comment mieux organiser son code ?
  - Les mêmes lignes de code se répètent sur plusieurs pages PHP
    - connexion
    - afficher les résultats
    - panier

Créer des classes !

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Orientation à objets

- **PHP est un langage « orienté objets »**



Définition de classe
class CompteClient

Visibilité :
private *$nom*

```
<?php
 class CompteClient {
     private $email;
     private $mdp;
     private $nom;
     private $adresse;

  public function __Set($attr, $val) {
         $this->$attr = $val ;
  }
  public function __Get($attr) {
         return $this->$attr ;
  ...
}
```

Définition d'un attribut

Opération :
public function…

get/set magiques
__Set
__Get

Retourner une valeur
return valeur ;

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Orientation à objets

- **Classes & objets**

On importe la définition de la classe **CompteClient**

$user est un **objet CompteClient**

On peut accéder à ses **attributs** (set / get « magiques »)

On peut invoquer les **opérations** de la classe **CompteClient**

```
<?php
  . . .
   require "CompteClient.php";

   $user = new CompteClient();

   $user->email = $_POST["login"];
   $user->mdp = $_POST["mdp"];

   if (  $user->chercheUserMdP() > 0 ) {
       $_SESSION["login"] = $user->email  ;
      …
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Orientation à objets

```php
<?php
 class CompteClient {
  . . .
 function chercheUserMdP () {
        include "connexionOO.php";

       $mysqli = new mysqli($host, $user, $password, $database);
       . . .
        if ( $result->num_rows > 0 ) {
             $ligne = $result->fetch_object();
             $this->nom = $ligne->nom;
             $this->adresse = $ligne->adresse;
          }

        return $result->num_rows ;
  }
?>
```

```php
<?php
    $host = "localhost";
    $user = "uml";
    $password = "uml";
    $database = "Produits";
?>
```

On isole dans la classe **CompteClient** tout l'accès aux comptes clients

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier

- **Mécanisme de sessions et les paniers**
  - Base pour la gestion de **panier** dans les sites de e-commerce
  - Les produits choisis par le client sont enregistrés en tant que **variables de session**
  - On peut y garder des **objets SIMPLES**

```
class LigneProduit {
    public $nom ;
    public $qte ;

    /* constructeur */
    function __construct( $nom ) {
        $this->nom = $nom;
        $this->qte = 1;
    }
}
```

Contenu du panier est gardé dans les variables de session.
Tableau contenant des objets LigneProduit.
Chaque **$_SESSION[$produit]** contient un objet.

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier

```
function ajouterProduit($produit) {
   $qte = 0;

   if ( ! isset ( $_SESSION[$produit] )  ) {
        $_SESSION[$produit] = new LigneProduit($produit);
        $qte = $_SESSION[$produit]->qte
   }

   else { // produit déjà là, augmenter alors sa quantité
        $objet = $_SESSION[$produit] ;
        $objet->qte = $objet->qte + 1;
        $qte = $objet->qte ;
   }
     return $qte;
}
```

Chaque produit choisi est identifié par un « id » (ici le nom).
$_SESSION[$produit]
va contenir un objet **LigneProduit**

S'il n'y a aucun **$_SESSION[$produit] ,** on va créer un nouveau objet **LigneProduit**

S'il y a déjà un **$_SESSION[$produit] ,** on va juste augmenter la valeur de l'attribut « qte » dans l'objet **LigneProduit**

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier

```
function supprimerProduit($produit) {
   $qte = 0 ;

   if ( isset( $_SESSION[$produit] ) ) {
        $objet = $_SESSION[$produit] ;
        $objet->qte = $objet->qte - 1;
       $qte = $objet->qte;

        if ( $qte <= 0) { //on supprime le produit
          unset($_SESSION[$produit]);
        }
    }

   return $qte;
}
```

Lorsqu'on veut supprimer un produit, on va réduire sa quantité dans l'objet **LigneProduit**

On récupère l'objet **LigneProduit** gardé dans **$_SESSION[$produit]**

On réduit sa quantité d'une unité

S'il n'en reste plus (la **quantité** a atteint **0 unités**), on **supprime** le produit de la session

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier

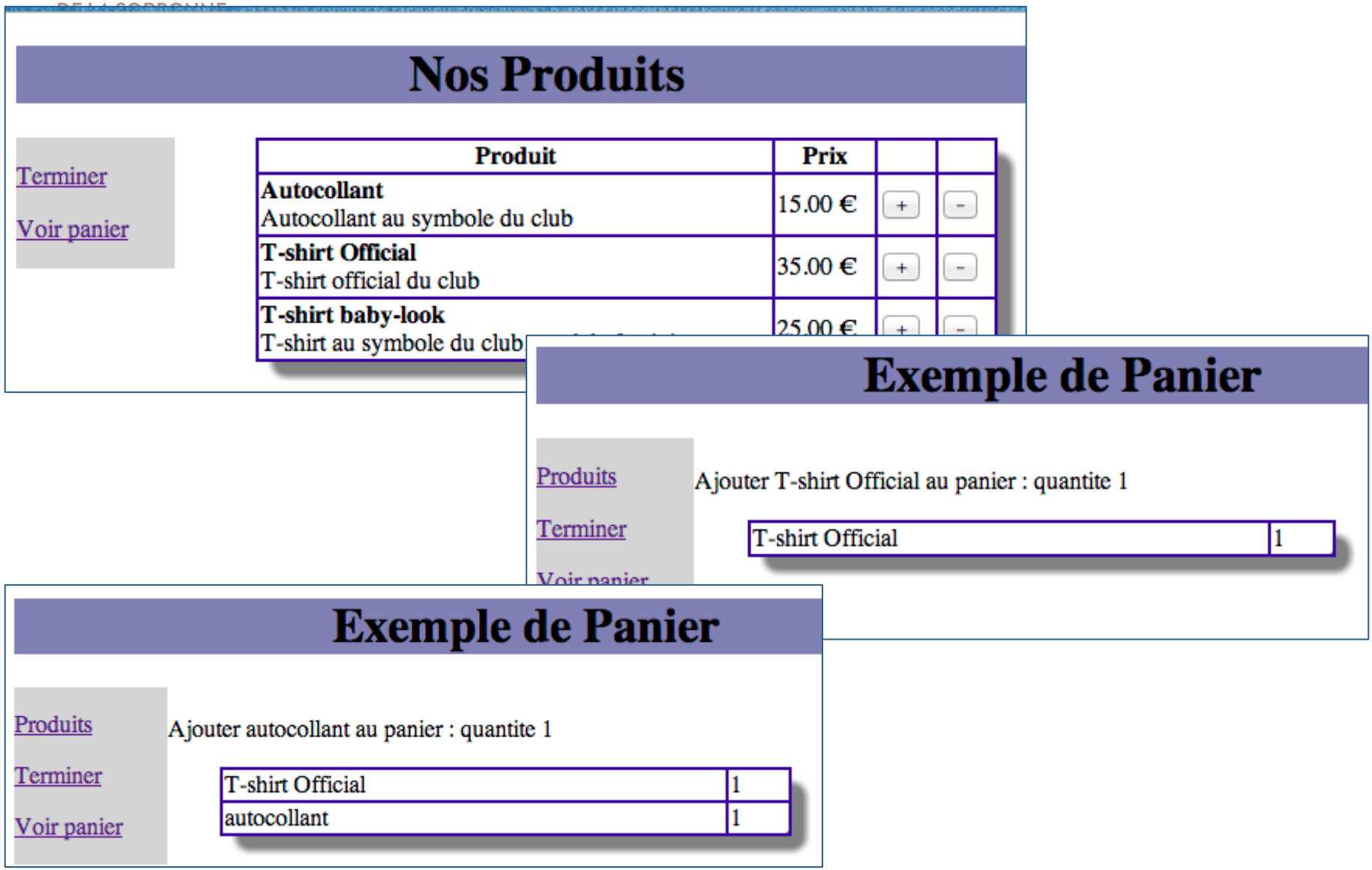
On peut récupérer le contenu du panier en récupérant le contenu de la variable de session **$_SESSION**

Pour chaque **objet LigneProduit** gardé dans **$_SESSION**

```
function afficherPanier() {
    echo "<table>" ;
    foreach($_SESSION as $objet) {
        echo "<tr><td> " . $objet->nom . " </td> <td> "
                . $objet->qte . " </td> </tr> ";
    }
    echo "</table>" ;
}
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT

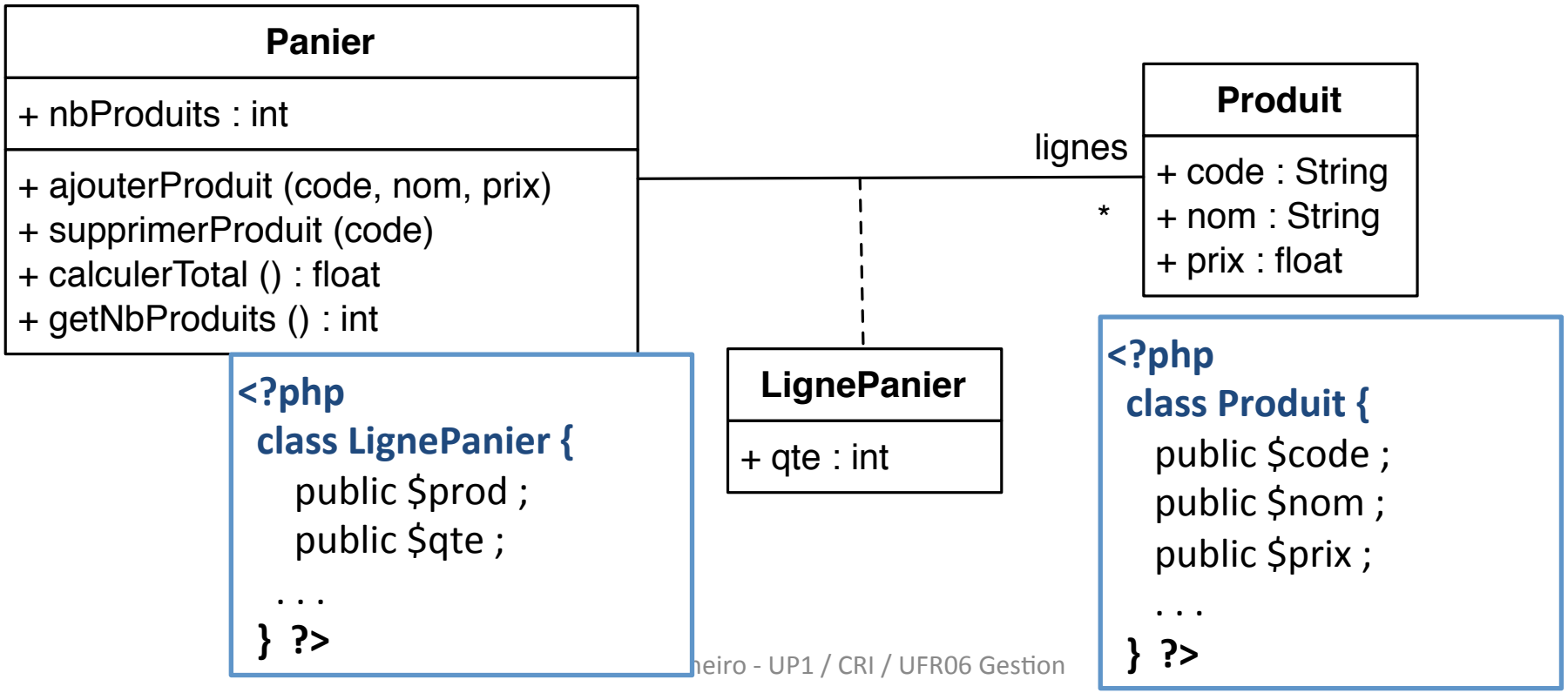
PHP : Panier



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

- Voici un exemple avancé de Panier qui utilise les classes en PHP et la notion de session
- Le panier est modélisé par une **classe Panier**



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

```
class Panier {
  public $lignes ;
  public $nbProduits ;

  function __construct() {
     $this->nbProduits = 0 ;
   }

...
```

Chaque Ligne de Panier est gardée dans un tableau associatif
$this->lignes[$code] => $LignePanier

On commence avec zéro produits dans le panier

On va créer le tableau lors de l'ajout du premier produit au panier

```
function ajouterProduit($code, $nom, $prix) {
      if ( $this->nbProduits == 0) {
          $prod = new Produit($code, $nom, $prix);
          $lp = new LignePanier($prod);
          $this->lignes[$code] = $lp;
          $this->nbProduits = 1;
      }
 . . .
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

```
function ajouterProduit($code, $nom, $prix) {
    if ( $this->nbProduits == 0) { . . . }
 else {
    if ( isset ( $this->lignes[$code] ) ) {
        $lp =  $this->lignes[$code] ;
        $qte = $lp->qte;
        $lp->qte = $qte + 1;
    }
    else {
        $prod = new Produit($code, $nom, $prix);
        $lp = new LignePanier($prod);

        $this->lignes[$code] = $lp;
        $this->nbProduits = $this->nbProduits + 1;
    }
}
```

Pour ajouter, on vérifie si le produit est déjà dans le panier

S'il y est, on le **récupère** et on met à jours la **quantité**

S'il n'y est pas, on va y **ajouter** une **nouvelle** ligne de panier

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

```
function ajouterProduit($code, $nom, $prix) {

    if ( isset ( $this->lignes[$code] ) ) {

        $lp = $this->lignes[$code] ;
        $lp->qte = $lp->qte - 1 ;

        if (  $lp->qte < 1) {
            unset($this->lignes[$code]);
            $this->nbProduits = $this->nbProduits - 1;
        }
    }
}
```

On ne supprime que si le produit est dans le panier

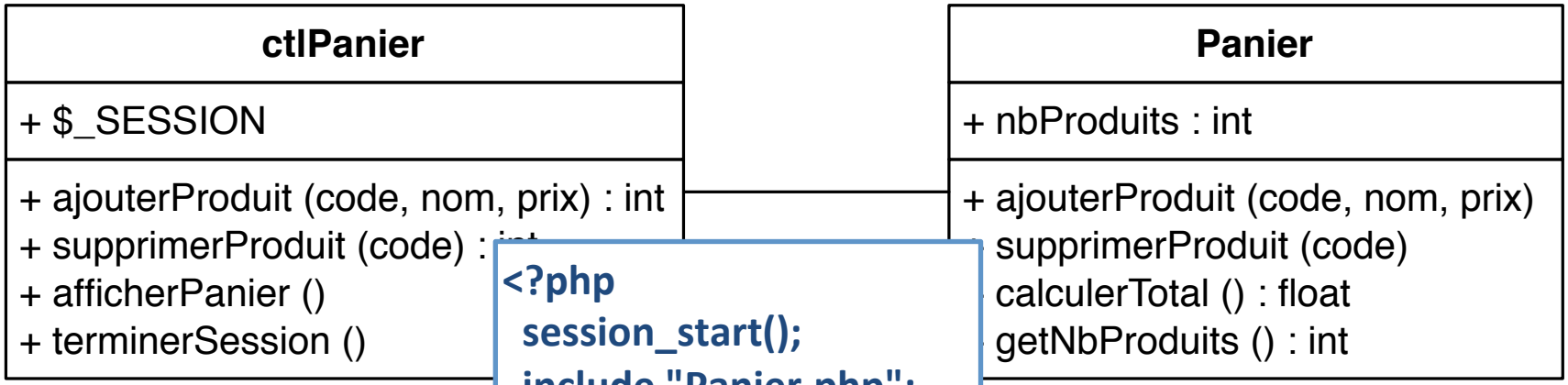
S'il y est, on met à jour la quantité, en **supprimant 1 unité**

Par contre, s'il ne reste plus rien (qte < 1), on **supprime la ligne de panier** du tableau

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

- C'est un objet **Panier** que notre site va manipuler
- Une page « **ctlPanier.php** » va ainsi gérer le panier
- Pour cela, elle va devoir garder un objet **Panier** dans **$_SESSION**




```
<?php
  session_start();
  include "Panier.php";
    . . .
 ?>
<html> . . . </html>
```

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE
ÉCOLE DE MANAGEMENT
DE LA SORBONNE

PHP : Panier avancé

- Or un objet **Panier** est un objet **complexe**
- Pour le garder dans **$_SESSION**, il va falloir le « compacter » : c'est la **sérialisation**
  - $_SESSION["panier"] = **serialize(**$panier**)**
  - **unserialize (** $_SESSION["panier"] **)**

```
function ajouterProduit($produit, $nom, $prix) {
. . .
  $panier = unserialize($_SESSION["panier"]);
  $panier->ajouterProduit($produit, $nom, $prix);
  $_SESSION["panier"] = serialize($panier);
. . .
```

Pour ajouter ou supprimer un produit au panier, on va le **récupérer**, le **modifier** puis le **remettre** dans la session

```
function supprimerProduit($produit) {
. . .
  $panier = unserialize($_SESSION["panier"]);
  $panier->supprimerProduit($produit);
  $_SESSION["panier"] = serialize($panier);
. . .
```