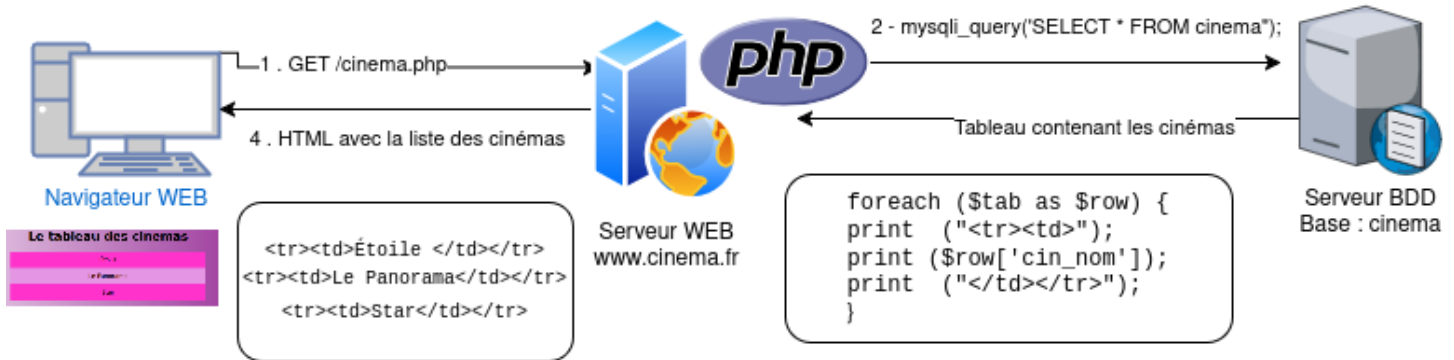


## 1 Architecture 3 tiers :



PHP (Hypertext Preprocessor) est un langage de programmation côté serveur utilisé principalement pour créer des sites web dynamiques. PHP s'exécute sur un serveur web et génère du HTML. PHP interroge les données stockées dans la base de données.

## 2 Syntaxe PHP versus Python

Le nom des variables commencent par \$ , des points virgules ; à la fin des instructions.

L'indentation est non obligatoire un bloc d'instruction commence par une accolade { et se termine par une accolade }.

Les tests des blocs **if** ou **while** sont entre parenthèses (**\$var < 10** ) dans l'instruction **print** les variables sont dans les guillemets "**\$var**"

### Python

```
age = int(input("Votre Age ?"))
if age < 18:
    print("Tu as ",age," ans ")
    print("Tu es mineur")
else:
    print("majeur")

if age >=14 and age < 16:
    print("groupe des cadets")

if age >= 65 or age <10:
    print("Demi tarif")
```

### PHP

```
$age = $_POST['age'];
if ($age < 18){
    print("Tu as $age ans ");
    print("Tu es mineur");
}else{
    print("majeur") ;
}

if ($age >=14 and $age < 16){
    print("groupe des cadets") ;
}

if ($age >= 65 or $age <10) {
    print("Demi tarif");
}
```

```
i = 1;
while i <= 10 :
    print (" nb : ",i)
    i+= 1 ;
}

for i in range (1,11) :
    print(i);
```

```
$i = 1;
while ($i <= 10) {
    print "<p> nb : $i </p>";
    $i+= 1 ;
}
for ($i = 1; $i <= 10; $i++)
{
    print $i;
}
```

### 3 Gestion des formulaires en PHP

Les pages PHP sont nommées .php, elles contiennent du code HTML, une balise particulière **<?php** permet d'indiquer le début du code PHP et la balise **?>** indique la fin du code PHP.

Les valeurs envoyées dans un formulaire POST se retrouveront dans la variable PHP :

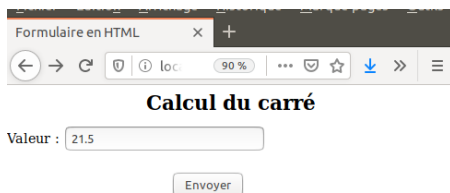
`$_POST[" nom"]` et les valeurs envoyées dans un formulaire GET se retrouveront dans la variable PHP : `$_GET[" nom"]`

`isset($_POST["Nombre"])` : permet de savoir si la variable existe.

`is_numeric($_POST["Nombre"])` : permet de savoir si la variable contient un nombre.

#### saisieNombre.html

```
<body>
<h2>Calcul du carré</h2>
<form method="post" action="calcul.php">
    Valeur :
    <input type="text" name="Nombre">
    <input type="submit" value="Envoyer">
</form>
</body>
</html>
```



#### calcul.php

```
<body> <h2>Résultat</h2>
<?php
if (isset($_POST["Nombre"]) and
is_numeric($_POST["Nombre"])){
    $nombre = $_POST["Nombre"];
    $resultat = $nombre*$nombre;
    print "<p>Le carré de <strong> $nombre
</strong> est <strong> $resultat </strong>
</p>";
} else {
    print "<p>Valeur invalide !</p>";
}
?>
</body>
```



**saisiePrix.html**

```
<form method="get" action = "calculPrix.php">
Prix: <input type="text" name="prix">
<br><br>
Remise: <input type="text" name="remise">%
<br><br>
<input type="submit" value="Envoyer">
</form>
```

Formulaire en HTML

**Calcul du prix après remise**

Prix:

Remise:  %

**calculPrix.php**

```
<?php
if (isset($_GET["prix"]) and
is_numeric($_GET["prix"]) and
isset($_GET["remise"]) and
is_numeric($_GET["remise"])) {
    $prix = $_GET["prix"];
    $remise = $_GET["remise"];
    $prixRemise = $prix - $prix * $remise / 100;
    print "<p>Le prix de <strong>$prix €
    </strong> après remise est de
    <strong>$prixRemise €</strong></p>";
}else{
    print "<p>Valeur invalide !</p>";}
?>
```

**Prix après remise**

Le prix de 40 € après remise est de 34 €

**4 Interrogation d'une base de données**

En PHP, mysqli est une extension qui permet d'interagir avec une base de données MySQL

**4.1 Connexion à la base de données**

Pour commencer, vous devez vous connecter à votre base de données MySQL

Voici comment vous pouvez le faire avec **mysqli** :

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "database_name";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Vérifier la connexion
if (!$conn) {
    die("Erreur de connexion : " . mysqli_connect_error());
}
echo "Connexion BDD : OK ";
?>
```

## 4.2 Exécuter une requête SQL

Une fois connecté, vous pouvez exécuter des requêtes SQL

Voici un exemple pour récupérer des données :

```
<?php
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Sortir les données de chaque ligne
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " .
        $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
?>
```

## 4.3 Insérer des données en SQL

Pour insérer des données dans la base de données :

```
<?php
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
?>
```

## 4.4 Utiliser des requêtes préparées

Pour éviter les injections SQL, vous devez utiliser des requêtes préparées :

```
<?php
$stmt = mysqli_prepare($conn, "INSERT INTO MyGuests (firstname, lastname, email) VALUES
(?, ?, ?)");
mysqli_stmt_bind_param($stmt, "sss", $firstname, $lastname, $email);

// Définir les paramètres et exécuter
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
mysqli_stmt_execute($stmt);

echo "New records created successfully";

mysqli_stmt_close($stmt);
mysqli_close($conn);
?>
```