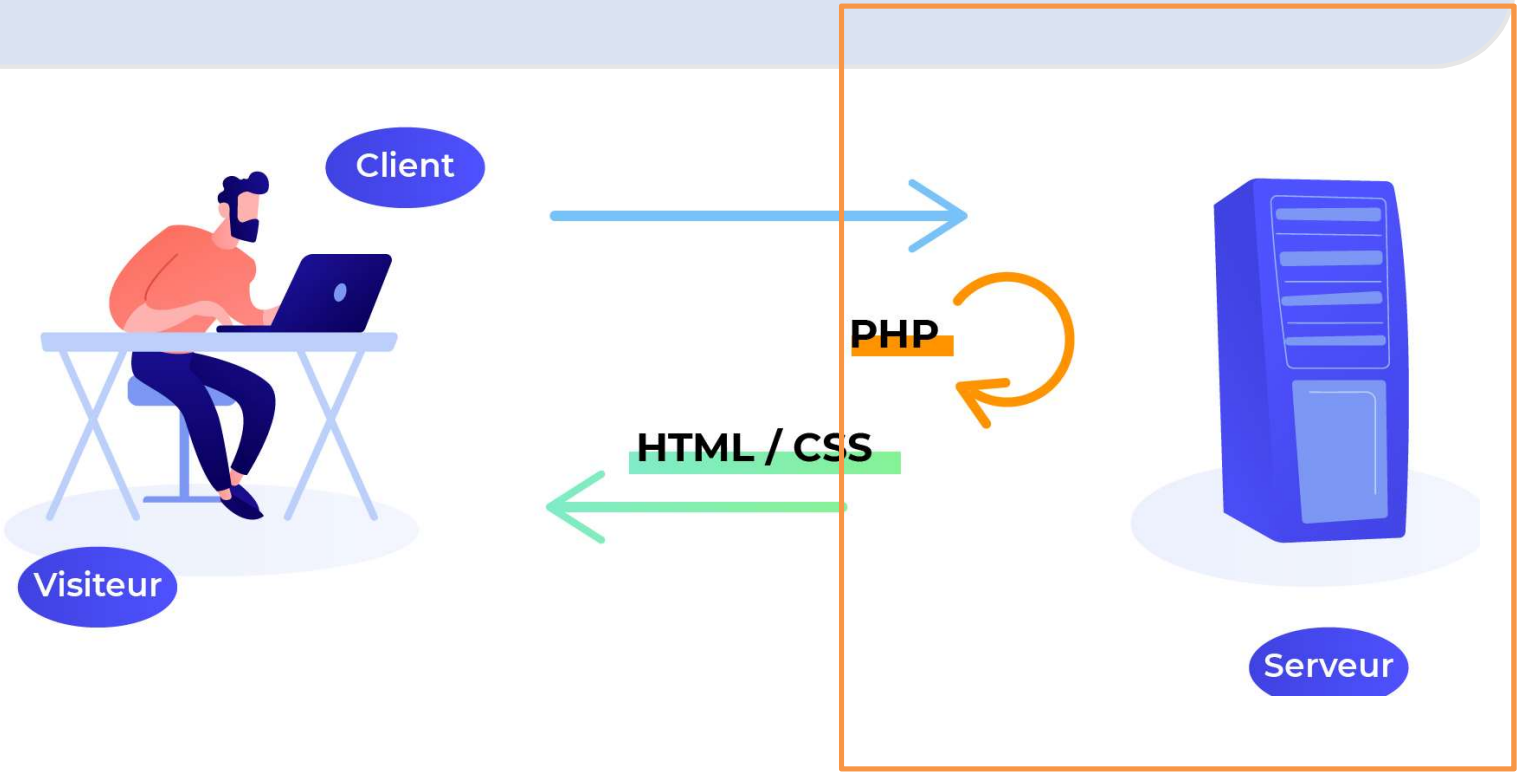
CSCI410
PROGRAMMATION RÉSEAU WEB

Chapitre 2 : Le HTML et CSS
Chapitre 3 : JavaScript
Chapitre 4: PHP : concepts
Chapitre 5 : PHP MySQL
Chapitre 6 : CMS

Chapitre 4:
PHP

La programmation coté SERVEUR

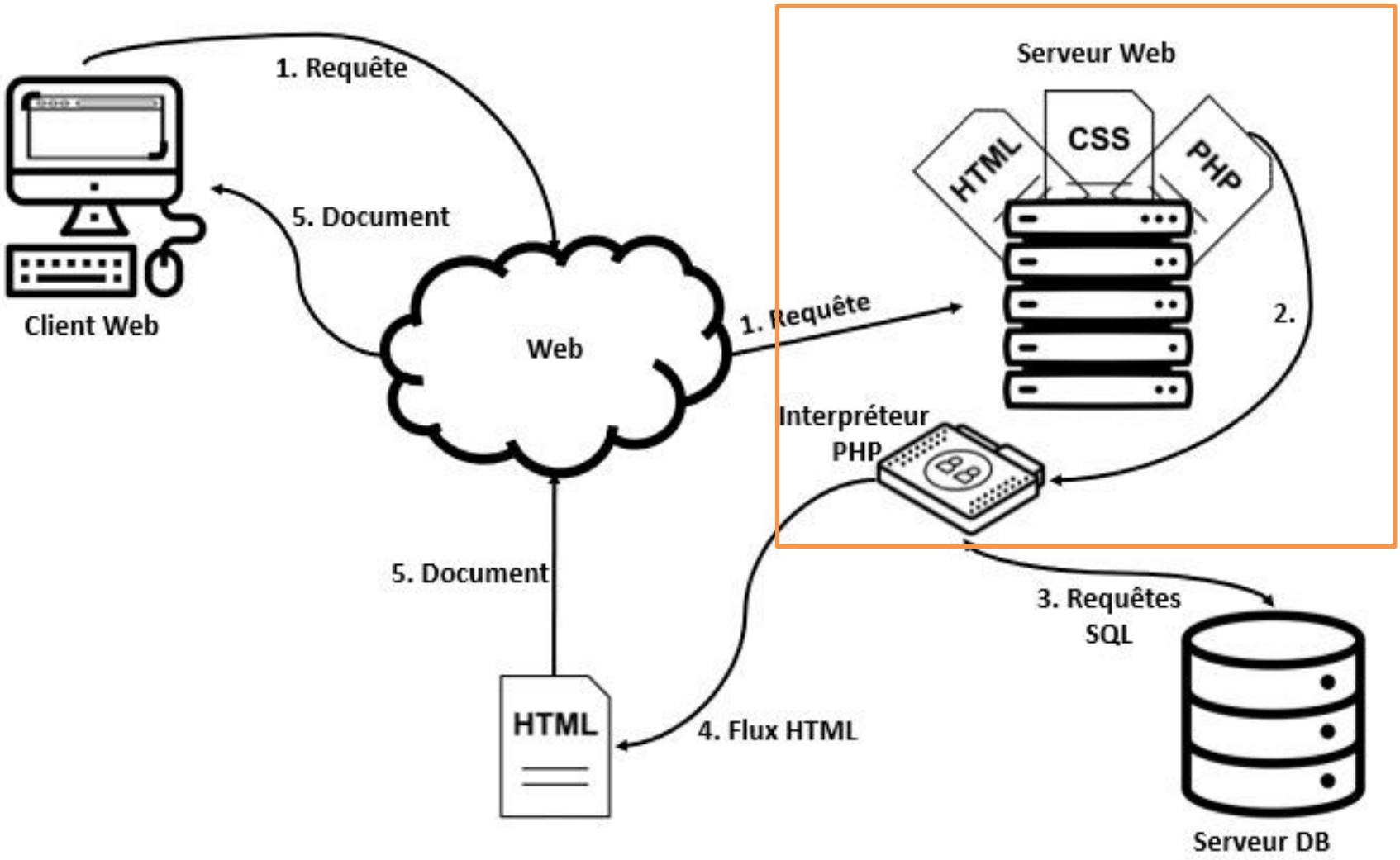


Introduction au PHP

PHP (Hypertext Preprocessor) est un langage de programmation serveur open-source très populaire pour le développement web. Il est largement utilisé pour créer des **sites web dynamiques**, des applications web, et pour interagir avec des bases de données.

Le **PHP est interprété côté serveur**, ce qui signifie que le code PHP est exécuté sur le serveur web, générant du contenu dynamique qui est ensuite envoyé au navigateur de l'utilisateur. Cela permet de créer des pages web interactives et de personnaliser le contenu en fonction des besoins de l'utilisateur.

Introduction au PHP



Configuration Essentielle

- **Créer un dossier SWD dans le dossier htdocs dans XMAPP**
- **Ce dossier va contenir nos pages PHP qui vont être interprété par le serveur web Apache**

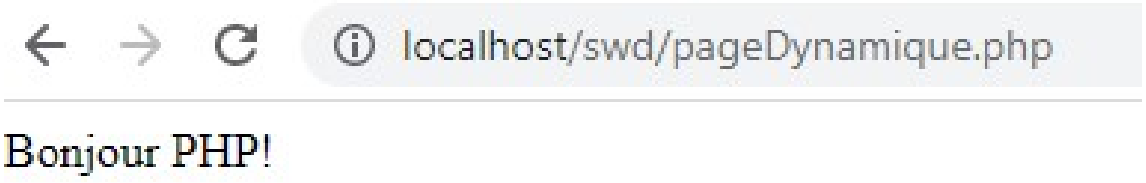
https://github.com/drBenanyMM/Programmation-r-seau-web

Premier programme PHP :

- **Créer votre premier page PHP dans le dossier SWD**
- **pageDynamique.php**

```
<?php
    echo "Bonjour PHP!";
?>
```

- **<?php:** indique le début du code PHP,
- **Echo:** est utilisé pour afficher du texte dans la sortie, et
- **?>:** marque la fin du code PHP.



1. Lancer le serveur Apache
2. Cliquer sur admin
3. Taper le chemin: http://localhost/swd/pageDynamique.php

Programmation en PHP :

- **La base:**
  1. **Déclaration de variables**
  2. **Expressions (vars + operateurs)**
  3. **Instruction (;)**

- **Standards:**
  1. **Structures de contrôles: if , eles, switch**
  2. **Boucles: while, do .. While, for**
  3. **Fonctions**
  4. **Tableaux**
  5. **POO**

Variables en PHP

- En PHP, les variables stocke des données.
- Créer la page variables.php dans le dossier SWD

```
<?php
        $text = "Bonjour, c'est PHP !";
        echo $text;
?>
```

  - Dans cet exemple, nous déclarons une variable $text et y assignons une chaîne de caractères. Ensuite, nous utilisons echo pour afficher la valeur de la variable.

Boucle en PHP :

- Les boucles en PHP vous permettent de répéter des actions.

```
<?php
        for ($i = 1; $i <= 5; $i++) {
                echo "Répétition $i <br>";
        }
?>
```

- Cette boucle affiche "Répétition 1" à "Répétition 5" en utilisant la variable $i pour suivre le nombre d'itérations.

Structures de contrôle en PHP :

- Les structures conditionnelles permet de pour prendre des décisions.

```php
<?php
    $age = 25;

    if ($age >= 18) {
        echo "Vous êtes majeur.";
    } else {
        echo "Vous êtes mineur.";
    }
?>
```

- Ce code vérifie si la variable $age est supérieure ou égale à 18, puis affiche le résultat en conséquence.

Tableaux Numériques en PHP

- Un tableau numérique est un tableau indexé où les clés sont des nombres entiers commençant généralement à zéro.

```
// Création d'un tableau numérique
$fruits = array("Pomme", "Banane", "Orange", "Fraise");
// Accès à un élément du tableau
echo $fruits[0]; // Affiche "Pomme"
// Modification d'un élément du tableau
$fruits[1] = "Kiwi";
// Ajout d'un élément à la fin du tableau
$fruits[] = "Ananas";
// Parcours d'un tableau
foreach ($fruits as $fruit) {
    echo $fruit . "<br>";
}
```

Tableaux Associatifs en PHP

- Un tableau associatif est un tableau où les clés sont des chaînes de caractères. Il vous permet d'associer des valeurs à des clés spécifiques.

```
// Création d'un tableau associatif
$personne = array("nom" => "Ahmed", "prénom" => "Sidi", "âge" => 30);
// Accès à un élément du tableau
echo $personne["nom"]; // Affiche "Ahmed "
// Modification d'un élément du tableau
$personne["âge"] = 35;
// Ajout d'une nouvelle paire clé-valeur
$personne["ville"] = "Nktt";
// Parcours d'un tableau associatif
foreach ($personne as $cle => $valeur) {
    echo $cle . ": " . $valeur . "<br>";
}
```

Manipulation de tableaux en PHP

```php
<?php
// Compter les éléments dans un tableau
$nbFruits = count($fruits); // $nbFruits contient le nombre d'éléments (4)
// Vérifier si un élément existe dans un tableau
if (in_array("Banane", $fruits)) {
    echo "Banane est dans le tableau.";
}
// Supprimer un élément du tableau
unset($fruits[1]); // Supprime "Banane" du tableau
// Réindexer un tableau
$fruits = array_values($fruits); // Réindexe le tableau
?>
```

Manipulation de chaînes en PHP

- Un tableau associatif est un tableau où les clés sont des chaînes de caractères. Il vous permet d'associer des valeurs à des clés spécifiques.

```php
<?php
// Concaténation de chaînes
$nom = "Ahmed";
$prenom = "Sidi";
$nomComplet = $prenom . " " . $nom; // $nomComplet contient "Ahmed Sidi"
// Longueur d'une chaîne
$phrase = "Ceci est une phrase.";
$longueur = strlen($phrase); // $longueur contient la longueur de la phrase
// Rechercher et remplacer dans une chaîne
$chaine = "Le chat est sur le tapis.";
$nouvelleChaine = str_replace("chat", "chien", $chaine); // Remplace "chat" par "chien"
```

Manipulation de chaînes en PHP

- Un tableau associatif est un tableau où les clés sont des chaînes de caractères. Il vous permet d'associer des valeurs à des clés spécifiques.

```
// Découper une chaîne en tableau
$chaine = "rouge,vert,bleu";
$tableauCouleurs = explode(",", $chaine); // $tableauCouleurs est un tableau ["rouge", "vert", "bleu"]

// Convertir en majuscules ou minuscules
$texte = "Ceci est un Exemple";
$texteEnMajuscules = strtoupper($texte); // Convertit en majuscules
$texteEnMinuscules = strtolower($texte); // Convertit en minuscules
?>
```

Manipulation de fichiers en PHP

```php
<?php
// Lire un fichier
$contenu = file_get_contents("monfichier.txt");
echo $contenu;
// Écrire dans un fichier
$donnees = "Données à écrire dans le fichier";
file_put_contents("nouveaufichier.txt", $donnees);
// Ouvrir un fichier en mode lecture
$handle = fopen("autrefichier.txt", "r");
if ($handle) {
    while (($ligne = fgets($handle)) !== false) {
        echo $ligne;
    }
    fclose($handle);
}
```

```php
// Ajouter du contenu à un fichier
$ajout = "Nouvelles données à ajouter";
file_put_contents("monfichier.txt", $ajout, FILE_APPEND);
// Supprimer un fichier
unlink("fichierasupprimer.txt");
?>
```

Fonctions et procédures en PHP

- Les fonctions et les procédures sont utilisées pour encapsuler du code réutilisable.

```
<?php
// Fonction qui ajoute deux nombres et retourne le résultat
    function addition($a, $b) {
        $somme = $a + $b;
        return $somme;
    }

    $resultat = addition(5, 3);
    echo "5 + 3 = " . $resultat; // Affiche "5 + 3 = 8"
?>
```

Fonctions et procédures en PHP

- Les fonctions et les procédures sont utilisées pour encapsuler du code réutilisable.

```
<?php
// Procédure qui affiche un message personnalisé
    function afficherMessage($prenom, $nom) {
        echo "Bonjour, " . $prenom . " " . $nom . " !";
    }

    afficherMessage("Ahmed", "Sidi"); // Appel de la procédure
    // Affiche "Bonjour, Ahmed Sidi !"
?>
```

Fonctions et procédures en PHP

- Fonction sans retour

```
<?php
// Fonction qui calcule la somme de trois nombres et affiche le résultat
    function calculerSomme($a, $b, $c) {
        $somme = $a + $b + $c;
        echo "La somme est : " . $somme;
    }

    calculerSomme(2, 4, 6); // Appel de la fonction
    // Affiche "La somme est : 12"
?>
```

Les superglobales en PHP

- Les superglobales en PHP sont des variables prédéfinies qui sont disponibles dans tout le script PHP sans avoir besoin d'être déclarées.
- Elles contiennent des informations importantes sur le serveur, la session, l'utilisateur, etc.

Les superglobales en PHP

1. $_GET : Cette superglobale est utilisée pour récupérer des données provenant de paramètres passés dans l'URL, généralement à partir de liens ou de formulaires avec la méthode GET.
2. $_POST : Cette superglobale est utilisée pour récupérer des données soumises via un formulaire HTML avec la méthode POST.
3. $_REQUEST : Cette superglobale est utilisée pour récupérer des données de l'utilisateur, à la fois à partir de GET et POST.
4. $_SESSION : Cette superglobale est utilisée pour stocker des données de session. Vous pouvez stocker des informations sur l'utilisateur et les retrouver sur plusieurs pages.
5. $_COOKIE : Cette superglobale est utilisée pour récupérer les cookies stockés sur le navigateur de l'utilisateur.
6. $_SERVER : Cette superglobale contient des informations sur le serveur et l'environnement d'exécution.
7. $_FILES : Cette superglobale est utilisée pour gérer les fichiers téléchargés via un formulaire.

La méthode GET

La méthode GET permet de transmettre des données à une page PHP,

Dans le dossier SWD, créer les eux pages suivantes:

1. Créer la page HTML **contact.html** avec un formulaire pour entrer des informations
2. Créer une page PHP **traitement.php** qui recevra et traitera ces données.

contact.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Formulaire avec méthode GET</title>
</head>
<body>
   <h1>Formulaire avec méthode GET</h1>
   <form action="traitement.php" method="get">
      <label for="nom">Nom :</label>
      <input type="text" id="nom" name="nom" required>
      <br>
      <label for="prenom">Prénom :</label>
      <input type="text" id="prenom" name="prenom" required>
      <br>
      <input type="submit" value="Soumettre">
   </form>
</body>
</html>
```

La méthode GET

traitement.php

```
<!DOCTYPE html>
<html>
<head>
   <title>Traitement du formulaire</title>
</head>
<body>
   <h1>Résultat du formulaire</h1>

   <?php
   if ($_SERVER["REQUEST_METHOD"] == "GET") {
      $nom = $_GET["nom"];
      $prenom = $_GET["prenom"];

      echo "Nom : " . $nom . "<br>";
      echo "Prénom : " . $prenom . "<br>";
   }
   ?>

   <p><a href="index.html">Retour au formulaire</a></p>
</body>
</html>
```

Exercice

Créer les deux pages suivantes avec l'information associée

contact.html **traitement.php**

Les fonctions prédéfinies

- La fonction **isset()** permet de vérifier si une variable existe et si elle n'est pas définie comme nulle (**null**).
- Elle retourne **true** si la variable est définie et **false** sinon.

```
$nom = "Ahmed";
if (isset($nom)) {
   echo "La variable \$nom existe.";
} else {
   echo "La variable \$nom n'existe pas.";
}
```

Les fonctions prédéfinies

- La fonction print_r() est utilisée pour afficher des informations sur une variable, un tableau ou un objet de manière lisible.
- Elle est principalement utilisée à des fins de débogage.

```
$fruits = array("Pomme", "Banane", "Orange");
print_r($fruits);
```

Les fonctions prédéfinies

- De plus, la fonction **var_dump()** pour obtenir des informations plus détaillées sur la variable, notamment son type et sa taille :

```
$fruits = array("Pomme", "Banane", "Orange");
var_dump($fruits);
```

Les fonctions prédéfinies

- La fonction **empty()** permet de vérifier si une variable est vide.
- Elle renverra true si la variable est vide, false sinon.
- Une variable est considérée comme vide si elle n'est pas définie, si sa valeur est égale à false, 0, ou une chaîne vide

```
$nom = "";
if (empty($nom)) {
    echo "La variable \$nom est vide.";
}
```

Les fonctions prédéfinies

- La fonction count() est utilisée pour compter le nombre d'éléments dans un tableau.
- Elle renvoie le nombre d'éléments dans le tableau.

```
$fruits = array("Pomme", "Banane", "Orange");
$nombreFruits = count($fruits);
echo "Il y a $nombreFruits fruits dans le tableau.";
```

Les fonctions prédéfinies

- is_array() et is_object() : Ces fonctions permettent de vérifier si une variable est un tableau ou un objet respectivement.
- Elles renvoient true si la variable est du type spécifié, sinon false.

```
$fruits = array("Pomme", "Banane", "Orange");
if (is_array($fruits)) {
    echo "\$fruits est un tableau.";
}

$personne = new stdClass();
if (is_object($personne)) {
    echo "\$personne est un objet.";
}
```

Les fonctions prédéfinies

- implode() et explode() : Ces fonctions sont utilisées pour la conversion entre chaînes et tableaux.
- implode() permet de fusionner les éléments d'un tableau en une seule chaîne,
- tandis que explode() permet de diviser une chaîne en un tableau en utilisant un délimiteur.

```
$fruits = array("Pomme", "Banane", "Orange");
$chaine = implode(", ", $fruits); // Fusionne les fruits en une chaîne

$chaine = "Pomme, Banane, Orange";
$fruits = explode(", ", $chaine); // Divise la chaîne en un tableau de fruits
```

SUIVANT
Chapitre 5 : PHP MySQL

Interaction entre le server web et le serveur BDD