

# Développement et Architecture Web

## PHP

R.Gosswiller

- 1 L'exécution côté serveur
- 2 Principes du PHP
- 3 Génération du HTML
- 4 Programmation en PHP
- 5 Formulaires et sessions

## L'exécution côté serveur

# Navigation sur internet

## Lien typique

`https :// outlook.office.com /mail/inbox`

- `https ://` : protocole utilisé
- `outlook.office.com` : adresse IP du serveur
- `/mail/inbox` : page demandée

# Adresse IP

## Principe

"outlook.office.com" est un nom de domaine (52.97.233.66)

La transformation d'un nom de domaine en IP se fait par un DNS (Domain Name Service)

13 DNS centraux contrôlent internet (officiellement, et en 2019), avec plus de 600 auxiliaires

## Adresse de test

http ://localhost :8080/

# Page demandée

## Principe

Comme toute autre machine, un serveur possède une arborescence de fichiers.

## Pages classiques

- Homepage, qui s'affiche si aucune page en particulier n'est demandée, aussi appelée index
- Page 404, qui s'affiche si une page spécifique est demandée, mais qu'elle n'existe pas

## Dossier par défaut pour ce cours

C :/ProgramFiles/EasyPHP/data/localweb/var/www/html

# Architecture serveur

## Travail d'un serveur

### Réponse à une requête

- Réception d'une requête
- Interprétation PHP
- Communication avec la BDD (éventuelle)
- Construction de page
- Envoi d'une réponse à la requête

# Principes du PHP

---



# Exemple

## PHP

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>PHP "Hello, World!" program</title>
5      </head>
6      <body>
7          <?php
8              echo '<p>Hello, World!</p>';
9          ?>
10     </body>
11 </html>
```

# Le PHP

## Définition

PHP : Hypertext Preprocessor (acronyme récursif)

# L'exécution côté serveur

## Principe

Contrairement au HTML et au CSS, le PHP est un langage qui s'exécute côté serveur

- Facile à modifier
- Facile à debugger

## Objectifs

- Générer du contenu HTML
- Analyse des flux et requêtes
- Lien Client-Webserver-BDD

# Syntaxe de base

Déclarer une section PHP

Balises `<?php` et `?>`

Instructions suivies de ;

Fichiers `.php`

Déclarer une variable

`$name`

Types génériques

## Déclarations

```
1 $variable = 1;
2 $variable = "toto";
3 $variable = "<br />";
```

# La fonction echo

## Rôle

Affichage/Génération du contenu

Encapsuler les données HTML dans une variable PHP

## Syntaxe

```
1 echo "toto";  
2  
3 echo "<h1>Title </h1 >";  
4  
5 echo $variable;
```

# La fonction var\_dump

## Principe

Affiche les informations liées à la structure d'une variable

Contenu, type, structure

## Syntaxe

```
1 void var_dump( $expr );  
2  
3 $arr = array('a','b','c');  
4 var_dump($arr);  
5 // Affiche array (3) { [0]=> string (1) "a"  
6 // [1]=> string (1) "b" [2]=> string (1) "c" }
```

# Génération du HTML

# Générer du HTML

## Objectif

Générer du HTML, depuis le serveur et pour le client

## Méthode

Tout fichier HTML peut être remplacé par un fichier PHP !

Fichier par défaut : index.html, puis index.php



# Le rôle du PHP

## Principe

PHP est un langage serveur chargé de générer du contenu formaté en HTML. Ce contenu aura pour objectif d'être interprété et affiché dans un navigateur web.

- Gestion des balises HTML
- Affichage du contenu avec echo

## Exemple

```
1 <?php
2     $variable = "Hello World";
3     echo "\$variable <br />";
4 ?>
```

# Interpretation

## Attention

' (simple quote) : Affichage direct

" (double quote) : Interprétation du contenu

## Exemple

```
1 <?php
2     $variable = "Robin";
3     echo "My name is : $variable ";
4         // Affiche My name is : Robin
5     echo 'My name is : $variable ';
6         // Affiche My name is : $variable
7 ?>
```

# Programmation en PHP

---

# Bases du PHP

## If / else conditions

```
1  if($password == "pass") {  
2      echo "Success !<br />";  
3  }  
4  else{  
5      echo " Connection error";  
6  }
```

# Bases du PHP

## Loops

Boucles for, while, do/while

Procédures itératives, génération de contenu

## Syntaxes

```
1  for ($i=0; $i<$limit ; $i++) {}  
2  while ($condition) {}  
3  do {} while ($condition);
```

# Fonctions PHP

## Principe

Mot-clé function (pas de type de retour)

Définition/Appel comme en C

## Syntaxe

```
1 function myFunction ($arg1 , $arg2) {}
```

# Fonctions PHP

## Exemple

```
1 function connection ($login , $pass) {  
2     if($login == "login" && $pass == "pass") {  
3         return true;  
4     }  
5     else{  
6         return false;  
7     }  
8 }  
9 $resultConnect = connection ("login", "password ");
```

# Inclusion de fichiers

## Principe

Fonction include

Transmission de variables

Copie directe du code inclus

Chemin relatif/absolu

## Syntaxe

```
1 include("file.php");
```



# Programmation en PHP

Deux approches

Programmation impérative

Programmation objet

# Formulaires et sessions

# Formulaires

## Principe

Echange de données clients - serveur

Récupération d'informations depuis un formulaire HTML

Méthode : get/post

# Formulaires : Etape 1

## Concevoir le formulaire HTML

```
1 <form method ="post" action ="page.php">
2 <input type="text" name="mytext " />
3 <input type="submit " value="Validate" />
4 </form >
```

## Formulaires : Etape 2

### Récupérer les données d'un formulaire en PHP

```
1 $texte = $_POST ["mytext "];  
2      // Ou  
3 $texte = $_GET ["mytext "];
```

# Sessions

## Définition

Une session est une zone mémoire, verrouillée par une clé  
Il est possible d'y stocker des ensembles de données ou variables

## Rôles

Stocker des données verrouillées par clé  
Données semi-permanentes  
Variables globales (partagées entre les pages)  
Espace utilisateur

# Démarrer une session

## Syntaxe

Function `session_start`

## Principe

Ouvrir une connexion à la session courante

En début de page

Filtrer les connexions directes

## Syntaxe

```
1 session_start();
```

# Gestion de sessions

## Utilisation

Création/gestion de variables Lecture en zone de session

## Syntaxe

```
1  $_SESSION ["variable "] = $var;  
2  if($_SESSION ["variable "] == $value ) {}
```



# Securiser une session

## Clé de session

```
1 session_id([ string $id ]);
```

## Principe

Récupérer l'identifiant de session courant

# Fermer une session

Fermer la session

Ne détruit pas les variables globales définies

Réinitialiser le cookie de session (setCookie)

## Syntaxe

```
1 session_destroy();
```

# Conclusion

## PHP

- Langage serveur
- Gestion des données et requêtes
- Génération de HTML
- Organisation par pages