

Université Paris Nanterre

Programmation web et Introduction à PHP

L2 MIASHS

M. NAFI

m.nafi@parisnanterre.fr

Année: 2022/2023

Objectifs du cours

❑ Création de **sites web statiques, interactifs**
et **dynamiques** à l'aide de:

- Langage **html**
- Langage **CSS**
- Langage **JavaScript**
- Langage **PHP**

Plan du cours

1. Introduction
2. Langage Html
3. Langage CSS
4. Langage Javascript
5. Langage PHP

Plan du cours

1. Introduction
2. Langage HTML
3. Langage CSS
4. Langage Javascript
5. **Langage PHP**

Introduction

- PHP (**P**HP **H**ypertext **P**reprocessor)
- Il a été créé en 1994 par Rasmus Lerdorf
- Un langage de script utilisé pour le développement de pages web **dynamiques**
- Les **scripts** ou codes PHP sont exécutés sur le **serveur**. Le résultat est retourné au client (navigateur) sous forme de document ou code **HTML**

- Syntaxe du PHP est **proche** de celle du langage C, Perl ou Java
- PHP est **multiplateformes** (Windows, Linux, Mac OS, etc)
- **Compatible** avec la plupart des **serveurs** web (Apache, IIS, etc)

- Supporte un large éventail de **bases de données** (MySQL, MariaDB, PostgreSQL, etc)
- **Open source** et gratuit
- Un fichier PHP porte l'extension **.php** et contient du texte, html, css, javascript et le code php.

Example

- `<!DOCTYPE HTML>`

```
<HTML>
```

```
  <head>
```

```
    <title> titre </title>
```

```
  </head>
```

```
  <body>
```

```
    <h1> <?php echo "Salut";?> </h1>
```

```
  </body>
```

```
</HTML>
```



```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <head>
```

```
    <title> titre
```

```
    </title>
```

```
  </head>
```

```
  <body>
```

```
    <h1> Salut </h1>
```

```
  </body>
```

```
</HTML>
```

Serveur

Client

Installation

❑ Besoins

- Installer un serveur web (Apache)
- Installer l'interpréteur PHP
- Installer un SGBD (MySQL)

❑ Ou installer **XAMPP**(Apache MySQL PHP Perl)

- **WAMP**: Windows
- **LAMP**: Linux
- **MAMP**: Mac OS

Syntaxe

- Un fichier PHP porte l'extension **.php** et contient du texte, html, css, javascript et le code ou script php.
- Un script PHP est délimité par les deux balises: `<script language='php'>` et `</script>` ou `<?php` et `?>`
- Une instruction se termine par `;` (séparateur)

Ex.

```
<?php
    echo " Hello world ";
?>
```

Variables

- Les variables sont des **conteneurs** pour le stockage des informations
- En PHP, une variable commence par le **signe \$**, suivi du **nom** de cette variable
- Le **nom** de la variable doit commencer par une **lettre** ou le caractère de **soulignement** **'_'**.

- Le nom des variables est **sensible** à la casse ($\$x \neq \X)
- Le nom des variables ne doit pas commencer par un nombre
- Le nom d'une variable ne peut contenir que des caractères alphanumériques et des caractères de soulignement ($A-z$, $0-9$ et $_$)
- **Nb**: Donner des noms **significatifs** aux variables ($\$age$ au lieu de $\$a$)

- Ex. `<?php`
 `$x = "PHP!";`
 `$y = 1;`
 `?>`

- Après exécution de ce script PHP, la variable:
 - `$x` contiendra la chaîne de caractère `"PHP!"`
 - `$y` la valeur entière `"1"`

Portée d'une variable

- **Locale**: déclarée et accessible seulement à l'intérieur d'une fonction
- **Globale**: déclarée et accessible en dehors d'une fonction
 - Le mot-clé « **global** » est utilisé pour accéder à une variable globale à l'intérieur d'une fonction
- **Statique**: on utilise le mot clé « **static** » pour **garder** la valeur d'une variable locale (ie elle ne sera pas supprimée à la fin de l'exécution de la fonction)

Variables superglobales

- Toujours accessibles de n'importe où et quel que soit leurs portées
 - `$_POST`
 - `$_GET`
 - `$_REQUEST`
 - `$GLOBALS`
 - `$_SERVER`
 - `$_FILES`
 - `$_ENV`
 - `$_COOKIE`
 - `$_SESSION`

- **\$_POST**: utilisée pour collecter les données de formulaire soumises avec la méthode 'Post'
- **\$_GET**: utilisée pour collecter les données de formulaire soumises avec la méthode 'Get'
- **\$_REQUEST**: utilisée pour collecter les données après la soumission d'un formulaire HTML
- **\$GLOBALS**: utilisé pour accéder aux variables globales depuis n'importe quel endroit du script PHP
- **\$_SERVER**: contient des informations sur les en-têtes, les chemins et les emplacements des scripts

Types de données

- PHP attribue **automatiquement** un type de données à la variable, en fonction de sa **valeur**
- Les variables peuvent stocker des données de **différents** types tels que: **Integer, float, string, array, boolean, object, Null, resource**

Integer

- Nombre entier **négatif** ou **positif**

Ex. `$x = 20;`

- Quelques fonctions sur les entiers
 - `is_int()`
 - `is_integer()`
 - `is_long()`

Float

- Nombre décimal ou à virgule flottante
- **Ex.** `$x = 20.5;`
- Quelques **fonctions** sur les nombres réels
 - `is_float()`
 - `is_double()`

String

- Une **séquence** ou une **chaîne** de caractères
- Texte entre **guillemets**

Ex. `$x= "Apprendre PHP"`

- Voici quelques fonctions couramment utilisées pour manipuler les chaînes de caractères:
 - **strlen()** retourne la **taille** d'une chaîne de caractères
 - **strrev()** retourne l'**inverse** d'une chaîne de caractères
 - **strpos()** retourne la **position** d'une sous chaîne
 - **str_replace()** **remplace** certains caractères par d'autres

Boolean

- **Deux** valeurs ou états: **vrai** ou **faux**

Ex. `$x=true;`

`$y=false;`

Array

- Variable contenant plusieurs valeurs

Ex.

```
$couleur = array ("rouge", "orange", "vert");
```

Object

- Concept de la programmation **orientée objet**
- Une **instance** d'une classe
- Un objet hérite toutes les **propriétés** et tous les **comportements** de la classe

Null

- Aucune valeur ou une seule valeur: **null**

Fonction vardump()

- Fonction `var_dump()` est utilisée pour connaître le **type** et la **valeur** d'une variable.

- Ex.

```
<?php  
$x=10;  
echo var_dump($x);  
?>
```

Résultat: `int(10)`

Opérateurs

- Affectation ($=$, $+=$, $-=$, $*=$, $/=$, $\%=$)
- Arithmétiques ($+$, $-$, $*$, $/$, $\%$)
- Concaténation ($.$)
- Logiques ($\&\&$ (et), $||$ (ou), $!$ (non))
- Comparaison ($==$, $!=$, $<$, $<=$, $>$, $>=$)

Structures de contrôle (1/2)

☐ Instructions conditionnelles

- If () else
- If () elseif () else
- Switch

Structures de contrôle (2/2)

❑ Boucles

- **While**: Tant que la condition est satisfaite, un bloc de code est exécuté.
- **Do ...while**: Répéter un bloc de code tant que la condition est vérifiée (le bloc est exécuté au **moins** une **seule** fois)
- **For**: exécuter un bloc de code un nombre de fois **spécifique**
- **Foreach**: utilisé pour les array

While

- Syntaxe

```
while (condition est vraie) {  
    code à executer;  
}
```

Ex.

```
<?php  
    $x = 1;  
    while($x <= 10) {  
        echo "$x <br>";  
        $x++;  
    }  
?>
```

Do ... while

- Syntaxe

```
Do {  
    code à executer;  
} while (condition est vraie)
```

- Ex.

```
<?php  
    $x = 1;  
    Do {  
        echo "$x <br>";  
        $x++;  
    } while($x <= 10)  
?>
```

For

- Syntaxe

```
for (init; test; increment)  
{  
    Code à executer;  
}
```

- Ex.

```
<?php  
    $x = 1;  
    for ($x=1;$x <= 10;$x++) {  
        echo "$x <br>";  
    }  
?>
```

Foreach

- Syntaxe

```
foreach ($array as $valeur) {  
    code à executer;  
}
```

- Ex.

```
<?php  
    $x = array("1","2","3");  
    foreach ($x as $valeur) {  
        echo "$valeur <br>;  
    }  
?>
```