

# Langage de Script PHP

## Partie 1 : Les bases du PHP

Yves Fréréric EBOBISSE DJENE

Y.Ebobissedjene@emsi.ma

3ème Année - Ingénierie en Informatique et Réseaux

4 septembre 2023

# Table of Contents

## 1. PHP et HTML

- URLs et serveurs Web
- Programmation côté serveur
- Cycle de vie d'une requête PHP
- Outils et Serveurs
- Envoi de code HTML au client
- Exécution du PHP

## 2. Introduction au langage PHP

- Les variables
- Les types
- Les opérateurs
- Le conditionnelle
- Les boucles
- Les tableaux
- Les fonctions
- Les commentaires

## 3. Envoi de données au serveur

- Requêtes et paramètres
- Formulaire HTML

# Table of Contents

## 1. PHP et HTML

- URLs et serveurs Web
- Programmation côté serveur
- Cycle de vie d'une requête PHP
- Outils et Serveurs
- Envoi de code HTML au client
- Exécution du PHP

## 2. Introduction au langage PHP

- Les variables
- Les types
- Les opérateurs
- Le conditionnelle
- Les boucles
- Les tableaux
- Les fonctions
- Les commentaires

## 3. Envoi de données au serveur

- Requêtes et paramètres
- Formulaire HTML

# PHP et HTML

## URLs et serveurs Web

*<https://server/path/file>*

Quand un client saisit et valide un URL dans le navigateur :

- ❶ L'ordinateur récupère l'adresse IP address du serveur à l'aide d'un **DNS** (Domain Name Server).
- ❷ Le navigateur se connecte à cette adresse IP et demande le fichier.
- ❸ Le serveur (par exemple, Apache) récupère le fichier sur son disque et envoie son contenu au navigateur du client.

# PHP et HTML

## URLs et serveurs Web

Certains URLs spécifient des programmes que le serveur exécute, récupère le résultat de cette exécution et l'envoie au navigateur :

L'URL ci-dessous permet de contacter le serveur **`www.emsi.ma`** et de lui demander d'envoyer le résultat du programme **`formations/certifications/index.php`** après l'avoir exécuté.

<http://www.emsi.ma/formations/certifications/index.php>

# PHP et HTML

## Programmation côté serveur



La programmation côté serveur s'effectue avec des langages de programmation (ou des framework) comme :

- PHP, Java/JSP, Ruby on Rails, ASP.NET, Python, Perl, NodeJS

Les serveurs Web sont capables d'exécuter des programmes écrits dans ces langages et peuvent envoyer les résultats de ces programmes au navigateur du client.

# PHP et HTML

## Programmation côté serveur

Chaque langage/framework a ses avantages et ses inconvénients

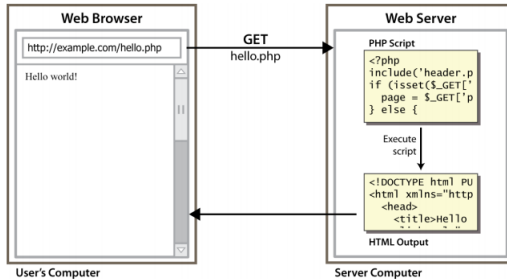
La popularité des langages/frameworks varie avec le temps :

- [Programming Language Popularity](#)
- Google Trends pour les [frameworks](#) populaires et aussi pour [PHP](#).

On utilise PHP dans ce cours à cause de ses avantages : simplicité, popularité, interprété, bien documenté, ...

# PHP et HTML

## Cycle de vie d'une requête PHP



La requête du navigateur est un fichier **.html** (contenu statique) :  
le serveur envoie simplement le contenu du fichier.

La requête du navigateur est un script **.php** (contenu dynamique) :  
le serveur exécute le script et envoie le résultat.



# PHP et HTML

## Outils - Serveur



**Serveur Web**



**Serveur BDD**



**Wampp**



**Windows**



**MAMPP**



Dossier des fichiers `htdocs`

# PHP et HTML

Envoi de code HTML au client : print et echo

## PHP (modèle)

```
print "text";  
echo "text";
```

## PHP (exemple)

```
print "Hello, World!" ;  
print "Escape \"chars\" are the SAME as in Java!" ;  
print "you can have  
line breaks in a string."  
  
print 'A string can use "single-quotes". It's cool!';
```

# PHP et HTML

Envoi de code HTML au client : print et echo

## Résultat

Hello, world !

Escape "chars" are the SAME as in Java !

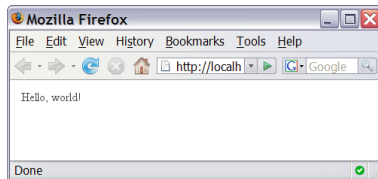
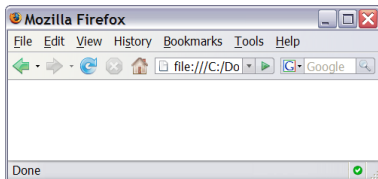
You can have line breaks in a string.

A string can use "single-quotes". It's cool !

Les fonctions **print** et **echo** sont presque identiques.

# PHP et HTML

## Exécuter du PHP



- On ne peut pas visualiser un script **PHP** avec le protocole **file** : rien ne s'affiche ou bien on voit le source du fichier.
- Il faut placer le script **PHP** sur le serveur local (localhost) : on peut alors activer le script via la barre d'adresse du navigateur.

# PHP et HTML

## Squelette de script PHP

### PHP (code)

```
<?php  
//PHP code goes here  
echo "This is a test !";  
?>
```

### Résultat

This is a test !

Le code PHP est toujours contenu dans les balises `<?php` et `?>`

# PHP et HTML

## PHP (code)

HTML content

`<?php`

PHP code

`?>`

HTML content

`<?php`

PHP code

`?>`

HTML content ...

- Le code (PHP) contenu dans `<?php` et `?>` est exécuté et le résultat transmis au client.
- Le code (HTML) non contenu dans `<?php` et `?>` est transmis tel quel au client.

# Table of Contents

## 1. PHP et HTML

- URLs et serveurs Web
- Programmation côté serveur
- Cycle de vie d'une requête PHP
- Outils et Serveurs
- Envoi de code HTML au client
- Exécution du PHP

## 2. Introduction au langage PHP

- Les variables
- Les types
- Les opérateurs
- Le conditionnelle
- Les boucles
- Les tableaux
- Les fonctions
- Les commentaires

## 3. Envoi de données au serveur

- Requêtes et paramètres
- Formulaire HTML

# Introduction au langage PHP

## Variables

### PHP (exemple)

```
$user_name = "Peacetachful";  
$age = 33;  
$age_in_dog_years = $age / 7;  
$this_class_rocks = TRUE;
```

- Les noms de variable sont case-sensitive. On peut utiliser le `_` si le nom est long et composé comme `$user_name`).
- Les noms de variables commencent toujours avec le caractère `$`.
- Les variables sont déclarées implicitement à la première affectation. Une variable peut contenir des valeurs successives de types différents.



# Introduction au langage PHP

## Les types

Les types de base : **integer**, **float**, **boolean**, **string**, **array**, **object**, **NULL**,

La fonction **is\_type** teste le type d'une valeur (i.e **is\_string**).

La fonction **gettype** retourne le type d'une valeur sous la forme d'une chaîne de caractères (pas très utilisée).

PHP **convertit automatiquement** très souvent :

- **string** vers **int** conversion automatique avec **+** pour :  
`("1" + 1 == 2)`.
- **int** vers **float** conversion automatique avec **/** pour :  
`(3 / 2 == 1.5)`.

**Conversion explicite** avec (type) : `$age = (int) "21";`

# Introduction au langage PHP

## Les opérateurs

- Opérateurs Arithmétiques : `+, -, *, /, %`
- Opérateurs d'Incrémentation & Décrémentation : `++, --`
- Opérateurs d'Affectation : `=, +=, -=, *=, /=, %=, . =`
- Opérateurs de concaténation : `.`

Beaucoup d'opérateurs convertissent automatiquement leurs opérandes : `5 + "7"` donne `12`.

# Introduction au langage PHP

## Le type string

### PHP

```
$favorite_food = "Asian";  
print $favorite_food[2];
```

# i

C'est identique à un tableau de caractères, utilisable avec les `[]`

L'opérateur de concatenation est `.` (point) et non `+`

- `5 + "2 turtle doves"` retourne 7.
- `5 . "2 turtle doves"` retourne "52 turtle doves".

Peut être délimitée par `" "` ou `' '`.

# Introduction au langage PHP

## Fonctions sur le type string

### PHP

```
$name = "Emsi - 3iir"
$length = strlen($name);           # 11
$cmp = strcmp($name, "Emsi");      # > 0
$index = strpos($name, "i");       # 3
$last = substr($name, 8, 4);       # "3iir"
$name = strtoupper($name);        # "EMSI - 3IIR"
```

# Introduction au langage PHP

## Fonctions sur le type string

Nom	Fonctions équivalentes en Java
<code>strlen</code>	<code>length</code>
<code>strpos</code>	<code>indexOf</code>
<code>substr</code>	<code>substring</code>
<code>strtolower</code> , <code>strtoupper</code>	<code>toLowerCase</code> , <code>toUpperCase</code>
<code>trim</code>	<code>trim</code>
<code>explode</code> , <code>implode</code>	<code>split</code> , <code>join</code>

# Introduction au langage PHP

## Chaînes et interpretation des variables

### PHP (exemple)

```
$age = 16;  
print "You are " . $age . " years old."  
print "You are $age years old." ;           # You are 16 years old.
```

- Les variables dans les chaînes délimitées par " " sont interprétées.
- Les variables dans ces chaînes sont remplacées par leur valeur.

# Introduction au langage PHP

## Chaînes et interpretation des variables

- Les variables dans les chaînes délimitées par `' '` ne sont pas interprétées :

### PHP (exemple)

```
print 'You are $age years old.';           # You are $age years old.
```

- Pour lever toute ambiguïté, on peut placer le nom de la variable entre :

### PHP (exemple)

```
print "Today is your $age th birthday."; # age th not found  
print "This's your {$age} th birthday."; # This's your 16th birthday.
```

# Introduction au langage PHP

## Le type Boolean

### PHP

```
$feels_like_summer = FALSE;  
$php_is_rad = TRUE;  
$student_count = 217;  
$nonzero = (bool) $student_count; # TRUE
```

Les valeurs suivantes sont vues comme **FALSE** (toutes les autres valeurs sont vues comme **TRUE**) :

- 0, 0.0 et Tableaux avec 0 element.
- " ", "0", et **NULL** (y compris les variables non initialisées).

**(bool)** permet de convertir explicitement une valeur vers un booléen.

**print(FALSE)** ne produit rien, **print(TRUE)** produit 1.



# Introduction au langage PHP

## La forme conditionnelle if/else

### PHP (Modèle)

```
if (condition) {  
    statements;  
}  
else if (condition) {  
    statements;  
}  
else {  
    statements;  
}
```

On peut utiliser **elseif** (sans espace) au lieu de **else if**.

# Introduction au langage PHP

## La boucle for

### PHP (Modèle)

```
for (initialization ; condition ; update) {  
    statements ;  
}
```

### PHP (exemple)

```
for ($i = 0 ; $i < 10 ; $i++) {  
    print "$i squared is " . $i * $i . " <br>" ;  
}
```

# Introduction au langage PHP

## La boucle while

### PHP (Modèle 1)

```
while (condition) {  
    statements ;  
}
```

### PHP (modèle 2)

```
do {  
    statements ;  
} while (condition) ;
```

Les instructions **break** et **continue** existent et fonctionnent comme en Java.

# Introduction au langage PHP

## Les tableaux

### PHP (Modèle)

```
$name = array();           # Create  
$name = array(value0, ..., valueN);  
$name[index]              # Get element value  
$name[index] = value;     # Set element value  
$name[] = value;          # Append PHP
```

### PHP (Exemple)

```
$a = array();              # Empty array (length 0)  
$a[0] = 23;                # Stores 23 at index 0 (length 1)  
$a2 = array("some", "strings", "in", "an", "array");  
$a2[] = "Ooh !";          # Add string to end (at index 5)
```

**NB : Un tableau peut contenir des éléments de différents types.**

# Introduction au langage PHP

## Fonctions sur les tableaux

Nom	Description
<code>count</code>	Nombre d'éléments dans un tableau
<code>print_r</code>	Affiche le contenu d'un tableau
<code>array_pop</code> , <code>array_push</code> , <code>array_shift</code> , <code>array_unshift</code>	Pour utiliser un tableau comme une pile/file
<code>in_array</code> , <code>array_search</code> , <code>array_search</code> , <code>sort</code> , <code>rsort</code> , <code>shuffle</code>	Recherche et tri
<code>array_fill</code> , <code>array_merge</code> , <code>array_intersect</code> , <code>array_diff</code> , <code>array_slice</code> , <code>range</code>	Création, initialisation et manipulation
<code>array_sum</code> , <code>array_product</code> , <code>array_unique</code> , <code>array_filter</code> , <code>array_reduce</code>	Traitement des éléments

# Introduction au langage PHP

## Exemple de fonctions sur les tableaux

### PHP (Exemple)

```
$tas = array("CA", "MJ", "MG", "SK", "KC");
for ($i = 0; $i < count($tas); $i++) {
    $tas[$i] = strtolower($tas[$i]);
}                                     # ("ca", "mj", "mg", "sk", "kc")

$conner = array_shift($tas);         # ("mj", "mg", "sk", "kc")
array_pop($tas);                     # ("mj", "mg", "sk")
array_push($tas, "kt");               # ("mj", "mg", "sk", "kt")
array_reverse($tas);                 # ("kt", "sk", "mg", "mj")
sort($tas);                          # ("kt", "mg", "mj", "sk")
$ks = array_slice($tas, 1, 2);       # ("mg", "mj")
```

Les tableaux en PHP remplacent la plupart des structures de données de Java : **list**, **stack**, **queue**, **set**, **map**, ...

# Introduction au langage PHP

## La boucle foreach

### PHP (Modèle)

```
foreach ($array as $variableName) {  
    statements;  
}
```

### PHP (exemple)

```
$stooges = array ("Larray", "Moe", "Curly", "Shemp");  
foreach ($stooges as $stooge) {  
    print "Moe slaps $stooge\n";           # Even himself  
}
```

La boucle **foreach** permet d'itérer sur les éléments d'un tableau sans utiliser les indices.

# Introduction au langage PHP

## Les fonctions

### PHP (Modèle)

```
function name(parameter1, ..., parameterN) {  
    statements;  
}
```

### PHP (exemple)

```
function bmi($weight, $height) {  
    $result = 703 * $weight / $height / $height;  
    return $result;  
}
```

- Les types des paramètres et de la valeur de retour ne sont pas spécifiés.



# Introduction au langage PHP

## Appel de la fonction

### PHP (Modèle)

```
name(expression1,..., expressionN);
```

### PHP (exemple)

```
$w = 163;           # pounds  
$h = 70;            # inches  
$my_bmi = bmi($w, $h);
```

### Erreur

Appeler une fonction avec le mauvais nombre d'argument provoque une erreur.

# Introduction au langage PHP

## Portée des variables

### PHP (Exemple)

```
$school = "EMSI"; # global
...
function downgrade() {
    global $school;
    $suffix = "(3IIR)"; # local

    $school = "$school $suffix";
    print "$school\n"; }
```

- Les variables déclarées dans une fonction sont **locales** à cette fonction.
- Les variables déclarées à l'extérieur d'une fonction sont **globales** pour cette fonction.

# Introduction au langage PHP

## Paramètres par défaut

### PHP (Modèle)

```
name (parameter1=value, ..., parameterN=value) {  
    statements;  
}
```

### PHP (exemple)

```
function print_separated($str, $separator=", ") {  
    for ($i = 1; $i < strlen($str); $i++) {  
        print $separator . $str[$i];  
    }  
}  
  
print_separated("hello");  
print_separated("hello", "-");
```

# h, e, l, l, o  
# h-e-l-l-o

# Introduction au langage PHP

## Les commentaires

### PHP (Modèle)

# commentaire sur une ligne

// commentaire sur une ligne

```
/*  
block de commentaires  
block de commentaires
```

...

```
*/
```

Comme en Java, avec # en plus :

- Beaucoup de développeurs PHP utilisent # au lieu de //

# Table of Contents

## 1. PHP et HTML

- URLs et serveurs Web
- Programmation côté serveur
- Cycle de vie d'une requête PHP
- Outils et Serveurs
- Envoi de code HTML au client
- Exécution du PHP

## 2. Introduction au langage PHP

- Les variables
- Les types
- Les opérateurs
- Le conditionnelle
- Les boucles
- Les tableaux
- Les fonctions
- Les commentaires

## 3. Envoi de données au serveur

- Requêtes et paramètres
- Formulaire HTML

# Envoi de données au serveur

## Requêtes et paramètres

### PHP (exemple)

URL ?name1=value1&name2=value2...

http ://www.google.com/search?q=Aya

http ://example.com/student\_login.php?username=aya&id=1234

- Il est possible de passer des paramètres lors d'une requête à un serveur :
  - Des couples (paramètre,valeur) placés à la fin de l'URL, par exemple, ( username , aya ), et ( id , 1234 ).
- Un script PHP peut lire et utiliser ces paramètres.
- Les paramètres sont toujours des chaînes de caractères.

# Envoi de données au serveur

## Requêtes GET et POST

- Deux types de requête vers le serveur :
  - 1 Une requête GET est utilisée pour obtenir des données (HTML) du serveur sans changer son état. La requête peut avoir des paramètres.
  - 2 Une requête POST est utilisée pour transmettre des données au serveur et éventuellement changer son état. La requête peut avoir des paramètres.

# Envoi de données au serveur

## Les tableaux `$_GET` et `$_POST`

### PHP (exemple)

```
$user_name = $_GET["username"];  
$id_number = (int) $_GET["id"];  
$seats_meat = FALSE;  
if (isset($_GET["meat"])) {  
    $seats_meat = TRUE;  
}
```

- `$_GET["parameter"]` ou `$_POST["parameter"]` contient la valeur d'un paramètre **GET** ou **POST** (une chaîne de caractères).
- les paramètres passés directement dans l'URL sont des paramètres **GET** :

```
http://domain/script.php?name1=value1&name2=value2
```



# Envoi de données au serveur

## Exemple Puissance.php

### PHP (script)

```
$base = $_GET["base"];  
$exp = $_GET["exponent"];  
$result = pow($base, $exp);  
print "$base ^ $exp = $result";
```

### PHP (URL)

puissance.php?base=3&exponent=4

### PHP (Résultat)

3 ^ 4 = 81

# Envoi de données au serveur

## Formulaire HTML

### PHP (Modèle)

```
<form action="destination URL" method="get OR post">  
    form controls  
</form>
```

- L'élément de base en HTML pour transmettre des données au serveur.
- L'attribut **action** contient l'URL du scrip (PHP) qui va traiter les données du formulaire.
- Quand tous les éléments de formulaires ont été renseignés, le formulaire est soumis et les données contenues dans ces éléments sont transmises au script.
- Si l'attribut **action** n'est pas présent, le script qui va traiter les données est le *même* qui contient ce formulaire !

# Envoi de données au serveur

Element du formulaire : input

## PHP (Modèle)

```
<input type="text" name="q" value="type in here" >  
<input type="submit" value="Send" >
```

- L'élément **input** permet de créer beaucoup d'éléments de formulaire différents.
- L'attribut **name** est le nom du paramètre envoyé au serveur.
- L'attribut **type** détermine le type d'élément : **button**, **file**, **checkbox**, **hidden**, **password**, **radio**, **reset**, **submit**, **text**, ...
- L'attribut **value** contient le texte initial de l'élément.

# Envoi de données au serveur

Element du formulaire : textarea

## PHP (Modèle)

```
<textarea rows="4" cols="20">
```

Type your comments here.

```
</textarea>
```

- L'élément input permet de créer beaucoup d'éléments de formulaire différents.
- Une zone de saisie de texte sur plusieurs lignes.
- Les attributs **rows** et **cols** spécifient la hauteur et la largeur en nombre de caractères.
- L'attribut **readonly** rend le contenu de l'élément non modifiable.

# Envoi de données au serveur

Element du formulaire : checkbox

## PHP (Exemple)

```
<input type="checkbox" name="lettuce"> Lettuce  
<input <type="checkbox" name="tomato" checked="checked">  
Tomato  
<input <type="checkbox" name="pickles" checked="checked">  
Pickles
```

- 0, 1, ou plusieurs checkboxes peuvent être sélectionnées en même temps.

# Envoi de données au serveur

## Element du formulaire : checkbox

- Lors de l'envoi au serveur :
  - La valeur de l'attribut **name** de chaque checkbox sélectionnée est le nom d'un paramètre transmis au serveur, et la valeur de ce paramètre est **on**.

`http://domain/script.php?tomato=on&pickles=on`

### PHP (Exemple)

```
<input type="checkbox" name="lettuce" checked="checked" >
```

- L'attribut **checked** permet de sélectionner des checkboxes par défaut :

# Envoi de données au serveur

Element du formulaire : button radio

## PHP (Exemple)

```
<input type="radio" name="cc" value="visa"
checked="checked"> Visa
<input type="radio" name="cc" value="mastercard">
MasterCard
<input type="radio" name="cc" value="amex"> American
Express
```

- Choix mutuellement exclusifs.
- Un groupe est déterminé par la même valeur de l'attribut **name**.

# Envoi de données au serveur

Element du formulaire : button radio

- Lors de l'envoi au serveur :
  - La valeur de l'attribut **name** d'un groupe est le nom du paramètre transmis au serveur.
  - La valeur de l'attribut **value** du bouton sélectionné dans ce groupe est la valeur de ce paramètre.

`http ://domain/script.php?cc=visa`



# Envoi de données au serveur

Element du formulaire : input caché

## PHP (Modèle)

```
<input type="text" name="username"> Name <br>
<input type="text" name="sid"> SID <br>
<input type="hidden" name="school" value="EMSI">
<input type="hidden" name="year" value="2019">
```

- Paramètre invisible à l'utilisateur dont la valeur (de l'attribut **value**) est transmise au serveur lorsque le formulaire est soumis.
- Très utile pour échanger des données entre plusieurs appels à des scripts.
- Du point de vue de PHP, identique à un **input** de type **text**.

# Envoi de données au serveur

Element du formulaire : select

## PHP (Modèle)

```
<select name="favoritecharacter">
  <option value="brad">Brad Pitt</option>
  <option value="george">George Clooney</option>
  <option value="johnny" selected>Johnny Depp </option>
</select>
```

- Le nom du paramètre transmis au serveur est la valeur de l'attribut **name** de l'élément **select**.
- La valeur de l'attribut **value** de l'élément **option** sélectionné est la valeur du paramètre transmis au serveur.
- **select** possède les attributs : **disabled**, **multiple**, **size**.
- L'attribut **selected** permet de sélectionner un élément **option** par défaut.