Développement WEB PHP - Séance 1

BUT Informatique parcours DACS

Table of contents:

- Présentation de la matière et des objectifs
- 📜 | Introduction à PHP
  - PHP, qu'est-ce que c'est ?
  - Coder en PHP
  - Différences de PHP
    - Echo
    - Debug
    - For each
    - Match
    - Tableaux associatifs
    - Opérateur "." pour ajouter des strings
- 📚 | TD 1 : Premiers scripts
  - Cloner le projet
  - Faire les exercices
- ⚙️ | Fonctions utilitaires & autres outils
  - Utilitaires pour les string (chaînes de caractères)
    - Taille d'une chaîne de caractères
    - String en majuscule ou miniscule
    - Séparer avec explode et regrouper avec implode
    - Supprimer les espaces en trop avec trim
    - Remplacer une partie d'un string avec replace
  - Utilitaires pour les int et float (nombres)
    - Transformer une variable en int ou float
  - Autres utilitaires
    - Checker le type d'une variable
    - Checker si une variable existe (et si elle est vide)
- 📚 | TD 2 : Utilitaires & outils
  - Cloner le projet
  - Faire les exercices

Présentation de la matière et des objectifs

La matière s'oriente et s'organise autour du développement, de la conception et du déploiement d'applications WEB sécurisées. Elle s'imprègne entièrement du parcours DACS (Déploiement d'Applications Communicantes et Sécurisées). À travers ce cours, nous apprendrons PHP, célèbre langage du WEB, ainsi qu'un de ses frameworks "Laravel". Vous serez également sensibilisés à la sécurité WEB ainsi qu'au déploiement autonome d'application.

Plusieurs types de titres indiquant le type de contenu :

- 📜 Cours théorique
- ⚙️ Cours pratique
- 📚 TD/TP
- 📝 Contrôle/QCM

> **INFO**
>
> Des commentaires avec les instructions seront présents tout au long des TD/TP pour vous guider sur les réponses à fournir.
>
> Par exemple :

```php
<?php

$ma_var = "toto";

// Ici, il faut afficher $ma_var
...
```

INFO

Quand le commentaire suivant est présent dans un fichier, **vous devez fournir un lien vers l'endroit où vous avez trouvé la réponse**.

```
/**
 * Lien : https://mon_lien.com
 **/
```

Généralement, vous pouvez trouver la réponse sur 3 sites différents :

- Ce site, en cliquant sur le titre de la partie, vous pourrez récupérer son lien. Par exemple : https://phd.julien-cpsn.com/courses/PHP/seance_1#remplacer-une-partie-dun-string-avec-replace
- Sur la documentation officielle de PHP. Par exemple : https://www.php.net/manual/en/function.str-replace.php
- Un blog, un forum, ou n'importe quel autre site en ligne (donc pas ChatGPT :)

DANGER

Tout commentaire non rempli sera considéré non rendu.

- Si le travail à rendre est **non-noté**, je viendrai vous poser la question.
- Si le travail à rendre est **noté**, la réponse sera comptée comme fausse.

ASTUCE

Vous recevrez la liste des éléments à réviser un peu de temps avant l'examen. Si j'oublie, n'hésitez pas à me le rappeler.

📜 | Introduction à PHP

PHP, qu'est-ce que c'est ?

PHP, acronyme de "PHP: Hypertext Preprocessor", est un langage de script principalement conçu pour le développement web. PHP a vu le jour en 1995 avec sa première version créée par Rasmus Lerdorf, et depuis, il a considérablement évolué pour devenir l'un des langages de script les plus utilisés pour le développement web. Écrit en C, PHP tire parti de la performance et de la flexibilité offertes par ce langage de bas niveau, tout en fournissant une syntaxe plus simple et accessible aux développeurs web. La version actuelle, 8.3.9, intègre de nombreuses améliorations et nouvelles fonctionnalités par rapport à ses prédécesseurs, telles que des optimisations de performance, des améliorations de sécurité, et des fonctionnalités de programmation modernes. Grâce à ces mises à jour régulières, PHP continue de s'adapter aux besoins évolutifs du développement web moderne et reste un choix pertinent pour le développement côté serveur.

Scripting

Contrairement aux langages de programmation traditionnels qui nécessitent la création de programmes complets, un langage de script comme PHP est utilisé pour écrire de petits scripts ou fichiers qui exécutent des actions spécifiques sur un serveur web.

Interprété

PHP est **interprété**, c'est-à-dire que le code est **exécuté ligne par ligne par un interpréteur**, contrairement aux langages compilés (comme C, C++ ou Java) où le code source est transformé en code machine/binaire avant d'être exécuté.

Multi-plateforme

PHP est **multi-plateforme**, ce qui signifie qu'**il peut fonctionner sur différents systèmes d'exploitation**, tout comme Java.

Programmation Orientée Object

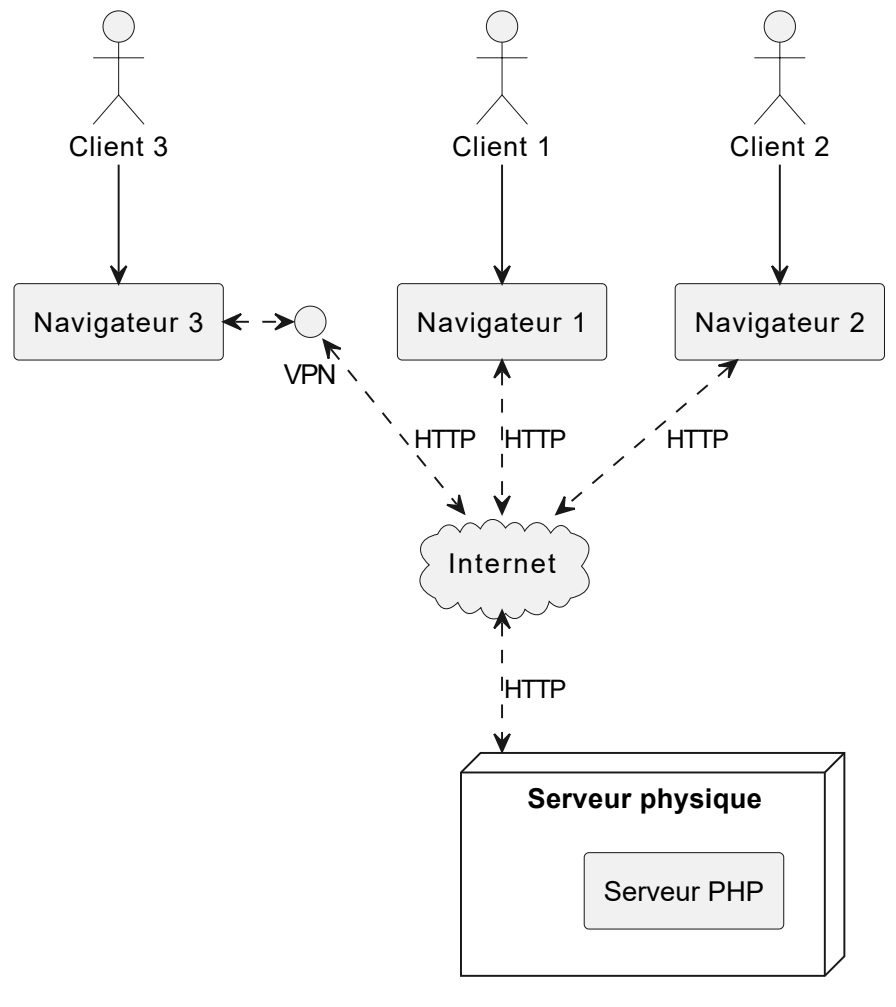
Il prend également en charge la **Programmation Orientée Objet (POO)**, permettant la création de classes, d'objets, et l'utilisation de concepts comme l'héritage et le polymorphisme.

Faiblement typé

PHP est faiblement typé (ou typé de manière souple), ce qui signifie que les types de données des variables ne sont pas strictement définis et peuvent être changés dynamiquement (ex: un int peut devenir un string), facilitant ainsi l'écriture du code mais pouvant parfois conduire à des erreurs subtiles.

Execution côté serveur

À noter que PHP s'exécute uniquement côté serveur : **le code est interprété sur le serveur**, puis le **résultat est renvoyé au client** (navigateur web), ce qui en fait un langage idéal pour la création de pages web dynamiques.



Coder en PHP

> **ASTUCE**
>
> Toujours garder la documentation de PHP près de soi

- Extension ".php"
- Les variables ont toutes un `$`, ex: `$ma_var`
- Point virgule à la fin des lignes
- Convention de nommage snake_case, ex `ma_super_fonction`
- Tous les fichiers commencent avec `<?php`, et peuvent finir par `?>`
- Il est possible de mettre des balises `<?php ... ?>` dans du HTML (toujours dans un fichier `.php`)

```
<div>
    <p>Mon paragraphe</p>
    <div><?php echo $ma_var; ?></div>
</div>
```

Exemple de code PHP :

```php
<?php

$ma_var = 12;

if ($ma_var == 12) {
    echo "Coucou !\n";
}
else {
    $ma_var = null;
}
```

Différences de PHP

Echo

Print avec `echo` (ne pas oublier d'ajouter un `\n`, car il ne retourne pas à la ligne)

```php
$ma_var = "toto";
echo $ma_var;
echo "Salut c'est $ma_var\n";
```

totoSalut c'est toto

Debug

Debugguer une variable avec `var_dump()`. Cela permet d'afficher le contenu complet de la variable au moment où est appelée la fonction pour régler d'éventuels problèmes.

```php
$ma_var = [1, 2, 5];
var_dump($ma_var);
```

array(3) { [0]=> int(1) [1]=> int(2) [2]=> int(5) }

For each

Le `for each` ("pour chaque")

```php
$mes_valeurs = [1, 2, 3, 4];

foreach ($mes_valeurs as $var) {
    echo "$var\n";
}
```

🖥️ Sortie

```
1
2
3
4
```

Il est aussi possible d'utiliser une clé

```php
$mes_valeurs = [10, 11, 12, 13];

foreach ($mes_valeurs as $index => $var) {
    echo "$index: $var\n";
}
```

🖥️ Sortie

```
0: 10
1: 11
2: 12
3: 13
```

Match

Plus récent, le `match`

```php
$ma_var = 2;

$mon_autre_var = match ($ma_var) {
    1, 2 => "toto",
    $ma_var > 10 => "tata",
    default => "cas par défaut"
};

echo $mon_autre_var;
```

🖥️ Sortie

```
toto
```

Tableaux associatifs

Pour créer des tableaux avec des index personnalisés (comme un dictionnaire)

```php
$ma_var = [
    0 => "toto",
    "mon index 1" => "tata",
    "mon autre index" => "tonton"
];

echo $ma_var["mon autre index"];
```

🖥️ Sortie

```
tonton
```

Opérateur "." pour ajouter des strings

Pour ajouter/concatener des strings en PHP, on utilise l'opérateur `.`

```php
echo "mon string 1" . "mon string 2";
```

🖥️ Sortie

```
mon string 1mon string 2
```

📚 | TD 1 : Premiers scripts

Cloner le projet

Cloner le projet https://github.com/PHP-BUT2-DACS/TP-TD

```
git clone https://github.com/PHP-BUT2-DACS/TP-TD.git
```

> ⚠️ **ATTENTION**
>
> **Ce projet vous servira tout le long du module**. Gardez le précieusement !

Faire les exercices

1. Dans un terminal, exécutez la commande `php -v` (ou `php --version`) pour être informé de la version de votre interpréteur PHP.

2. Ensuite, naviguez jusqu'au dossier du projet `/Séance 1/TD1`. Puis démarrez le serveur PHP avec la commande suivante :

```
php -S localhost:8080
```

> Comprendre la commande
>
> - `php` fait référence à l'interpréteur PHP pré-installé sur votre machine.
> - L'option `-S` (pour **S**erver) permet de **créer un serveur PHP**. Cette option nécessite deux paramètres :
>   - `<addr>`, c'est-à-dire **l'adresse à laquelle on veut utiliser notre serveur** (`localhost`, correspond à l'hôte local)
>   - `<port>` permet de préciser **le port à utiliser** pour l'adresse précédemment fournie. En informatique, il est couramment admis que les ports 808X, 800X, 300X, 500X sont réservés pour le développement.

3. Ouvrir le navigateur à l'URL http://localhost:8080/exo1.php

4. Une fois la commande lancée et l'URL ouverte, votre terminal devrait afficher quelque-chose de similaire :

> **SORTIE**

```
┌─ PHP DACS/Séance 1/TD1 on  main via 
└─$ php -S localhost:8080
[Tue Sep  3 21:24:35 2024] PHP 8.2.20 Development Server (http://localhost:8080) started
[Tue Sep  3 21:24:44 2024] [::1]:46752 Accepted
[Tue Sep  3 21:24:44 2024] [::1]:46752 [200]: GET /exo1.php
[Tue Sep  3 21:24:44 2024] [::1]:46752 Closing
[Tue Sep  3 21:24:44 2024] [::1]:46758 Accepted
[Tue Sep  3 21:24:44 2024] [::1]:46758 [404]: GET /favicon.ico - No such file or directory
[Tue Sep  3 21:24:44 2024] [::1]:46758 Closing
```

5. Répondez aux exercices du TD1. Vous pouvez demander de l'aide si nécessaire.

> **ASTUCE**
>
> La commande `php -h` (ou `php --help`) pourra vous aider si vous êtes perdus avec d'autres commandes de PHP.

6. Push les modifications

> **ATTENTION**
>
> N'oubliez pas de push les modifications sur votre repository GitHub

Il faudra créer un repository dans l'organisation https://github.com/PHP-BUT2-DACS, portant le nom `TP-TD-<MON-NOM>`. Par exemple : `TP-TD-CAPOSIENA`.

Il faudra ensuite changer l'URL de mon clone local vers le repository fraichement créé. Cela peut être fait depuis le repository `TD-TD` cloné précédemment via la commande suivante :

```
git remote set-url origin https://github.com/PHP-BUT2-DACS/TP-TD-<MON-NOM>.git
```

Ensuite,

- `git add -A`
- `git commit -m "mon message"`
- `git push`

⚙️ | Fonctions utilitaires & autres outils

Utilitaires pour les string (chaînes de caractères)

Lien vers la documentation de PHP

Taille d'une chaîne de caractères

Calcule la taille d'une chaîne de caractères.

```php
$mon_string = "test";
echo strlen($mon_string);
```

🖥️ Sortie

```
4
```

String en majuscule ou miniscule

Renvoie un string en majuscules ou minuscules.

```php
$ma_var = "toto";
echo strtoupper($ma_var) . "\n";

$ma_var = "TATA";
echo strtolower($ma_var);
```

🖥️ Sortie

```
TOTO
tata
```

Séparer avec explode et regrouper avec implode

Scinde our regroupe une chaîne de caractères en segments

```php
$ma_var = "toto;tata;tonton";

// Séparer un string vers un tableau
$mon_tableau = explode(";", $ma_var);
var_dump($mon_tableau);

// Regrouper des éléments d'un tableau vers un string
$mon_autre_var = implode(";", $mon_tableau);
echo $mon_autre_var;
```

🖥️ Sortie

```
["toto", "tata", "tonton"]
toto;tata;tonton
```

Supprimer les espaces en trop avec trim

```php
$ma_var = "   toto    ";
$mon_autre_var = trim($ma_var);
echo $ma_var;
```

🖥️ Sortie

```
toto
```

Il existe aussi ltrim (left trim) pour supprimer les espaces seulement au début, et rtrim (right trim) pour supprimer les espaces seulement à la fin.

Remplacer une partie d'un string avec replace

```php
$ma_var = "toto";
$mon_autre_var = str_replace("o", "a", $ma_var);
echo $ma_var;
```

🖥️ Sortie

```
tata
```

Utilitaires pour les int et float (nombres)

Transformer une variable en int ou float

```php
$ma_var = "10";
$mon_int = intval($ma_var);
var_dump($ma_var);
```

🖥️ Sortie

```
int(10)
```

Même principe avec `floatval()`

Autres utilitaires

Checker le type d'une variable

- `is_array()` — Détermine si une variable est un tableau
- `is_bool()` — Détermine si une variable est un booléen
- `is_callable()` — Détermine si une valeur peut être appelé comme une fonction dans la portée courante.
- `is_countable()` — Vérifie si le contenu de la variable est une valeur dénombrable
- `is_float()` — Détermine si une variable est de type nombre décimal
- `is_int()` — Détermine si une variable est de type nombre entier
- `is_numeric()` — Détermine si une variable est un nombre ou une chaîne numérique

```php
var_dump(is_int(23));
var_dump(is_int("23"));
```

🖥️ Sortie

```
bool(true)
bool(false)
```

Checker si une variable existe (et si elle est vide)

`isset()` détermine si une variable est considérée définie, ceci signifie qu'elle est déclarée et est différente de `null`.

```php
$est_ce_que_ma_var_existe = isset($ma_var)
var_dump($est_ce_que_ma_var_existe)
```

🖥️ Sortie

```
bool(false)
```

`empty()` détermine si une variable est considérée définie et si sa valeur équivaut à `false`.

```php
$mon_tableau = ["toto", "tata", false]

$est_ce_que_toto_existe = empty($mon_tableau[0])
$est_ce_que_tonton_existe = empty($mon_tableau[2])
var_dump($est_ce_que_toto_existe)
var_dump($est_ce_que_tonton_existe)
```

🖥️ Sortie

```
bool(false)
bool(true)
```

📚 | TD 2 : Utilitaires & outils

Cloner le projet

Si ce n'est pas déjà fait, cloner le projet https://github.com/PHP-BUT2-DACS/TP-TD

```
git clone https://github.com/PHP-BUT2-DACS/TP-TD.git
```

Faire les exercices

1. Lancer le serveur PHP dans le dossier `/Séance 1/TD2`

```
php -S localhost:8080
```

2. Répondez aux exercices du TD2. Vous pouvez demander de l'aide si nécessaire.