

PHP : Introduction

L2 MPCIE - UE Développement Web

David Lesaint
david.lesaint@univ-angers.fr



Janvier 2019

Développer pour le Web suppose de ...

Maîtriser différents langages et paradigmes de programmation

- Contenu et structure des pages web : HTML5/XHTML, DOM, SVG ...
- Style des pages web : CSS.
- Traitements côté client : JavaScript, VBScript/JScript (Microsoft), ActionScript (Adobe), ...
- Traitements côté serveur : PHP, Python, Ruby, JSP, ASP, ...
- Echange de données : XML, JSON, MySQL, ...

Développer pour le Web suppose de ...

Se conformer aux pratiques du génie logiciel

- Méthodes de conception objet (patrons).
- Méthodes agiles (développement piloté par les tests, intégration continue, ...).

S'adapter à un ensemble d'outils en constante évolution

- Bibliothèques (Bootstrap, jQuery, AngularJS, ...).
- Cadriciels (Symfony, CodeIgniter, ...).
- Outils de développement (PHPEclipse, PHPUnit, PHPDocumentor, ...).

Langages et outils abordés

- PHP
- JavaScript
- XML, JSON, AJAX (Asynchronous Javascript + XML)
- jQuery (bibliothèque JavaScript)
- Bootstrap (framework front-end HTML/CSS)

Objectifs du cours PHP

Maîtriser les bases du langage PHP **version 7** :

- Variables, constantes, types
- Instructions de contrôle
- Chaînes de caractères
- Tableaux
- Formulaires
- Fonctions

- Dates
- Programmation objet
- Images dynamiques
- Fichiers
- Cookies, sessions, emails
- Bases de données
- XML

Le langage PHP

Origines

- PHP (acronyme pour Php Hypertext Processor) est un langage de scripts interprété.
- Créé en 1994 par Rasmus Lerdorf.
- Utilisé pour produire “facilement” des pages Web dynamiques.

Un langage impératif, orienté objet, fonctionnel

- Structure proche du langage C.
- Typage dynamique des variables.
- Tableaux associatifs, objets, ressources, ...

Site et documentation officiels

<http://www.php.net/manual/fr>

Quelques versions choisies

- 4.1 : variables superglobales.
- 4.3 : CLI (Common Line Interface) en supplément de CGI (Common Gateway Interface).
- 5.0 : modèle objet.
- 5.3 : espaces de noms, résolution statique à la volée, fermetures, fonctions anonymes.
- 5.4 : traits.

PHP 7.0 (déc. 2015) - 7.1 (déc. 2016) - 7.2 (nov. 2017)

- Nouveau moteur : Zend engine.
- Prise en charge cohérente du 64 bits.
- Gain de performances en temps CPU et consommation mémoire (facteur 4 sur certains benchmarks).
- Amélioration de la hiérarchie des exceptions.
- De nombreuses erreurs fatales converties en exceptions.
- Un générateur de nombres aléatoires sécurisé.
- Suppression des interfaces SAPI et des extensions obsolètes ou non-maintenues.

PHP 7.0 (déc. 2015) - 7.1 (déc. 2016) - 7.2 (nov. 2017)

- Typage des déclarations de fonction : retours et scalaires.
- Nouvel opérateur null coalescing ??
- Nouvel opérateur de comparaison <=>
- Classes anonymes, assertions à coût nul, ...

Plus d'informations

[http://linuxfr.org/news/
sortie-de-php-7-0-un-nouveau-depart](http://linuxfr.org/news/sortie-de-php-7-0-un-nouveau-depart)

Exemples d'éditeurs de HTML/PHP

- Bacs à sable : [PHPSandbox](#), [Online PHP Functions](#).
- Editeurs légers : [HTML Kit](#), [Geany](#).
- EDI : [PHP Development Tools \(PDT\)](#) pour Eclipse, [Zend Studio](#).

Installation d'un serveur local

Elements d'un serveur local :

- Serveur Apache.
- Interpréteur PHP.
- Bases de données pour MySQL, pour SQLite, ...
- Utilitaires PHPMyAdmin pour créer et gérer bases et tables de données MySQL, SQLiteManager pour SQLite, ...

Outils d'installation

- LAMP pour Linux.
- WAMP pour Windows.
- MAMP pour Mac.

Organisation de PHP

Organisation modulaire

- Module standard : accès aux types, instructions et fonctions élémentaires.
- Modules additionnels : ajout de fonctionnalités particulières (eg. accès et gestion de diverses bases de données).

Pour connaître la liste des modules disponibles :

- pour PHP en ligne de commande (CLI) :

```
$ php -m
```
- pour PHP déployé sur votre serveur, chargez le script :

```
<?php phpinfo(); ?>
```

Structure des fichiers HTML5

Une page dynamique PHP est un document HTML envoyé au client par le serveur.

Structure de document HTML 5 (pagehtml.html)

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Titre de la page</title>
6   </head>
7   <body>
8     <h2>Bienvenue sur le site PHP 7 </h2>
9   </body>
10 </html>
```

On pourrait aussi utiliser le suffixe `.htm` ou `.php`.

Première page PHP

codephp.php

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Une page PHP</title>
6   </head>
7   <body>
8     <?php
9       echo "<h3> Aujourd'hui le ". date('d / M / Y
H:m:s') . "</h3><hr />";
10      echo "<h2>Bienvenue sur le site PHP 7</h2>";
11    ?>
12  </body>
13 </html>
```

- Les marqueurs <?php et ?> délimitent un script PHP.
- On peut inclure autant de scripts PHP que l'on veut dans un document HTML.

Cycle de vie d'une page PHP

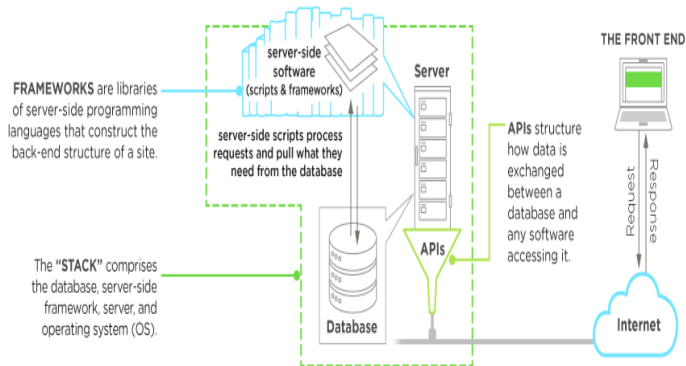
Trois étapes :

- 1 Envoi d'une requête HTTP par le client (eg. navigateur) du type
`http://www.server.com/codephp.php`
- 2 Interprétation par le serveur du code PHP contenu dans la page demandée (ici `codephp.php`)
 - l'interpréteur renvoie le code HTML après évaluation du code PHP rencontré
- 3 Envoi par le serveur d'un fichier dont le contenu n'est que du HTML (+ CSS + JavaScript)
 - voir l'onglet `Code source` de la page sous Firefox

Vue globale : le “back-end”

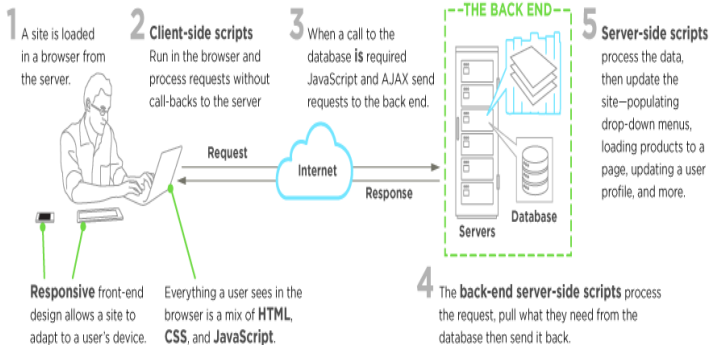
BACK-END DEVELOPMENT & FRAMEWORKS IN SERVER SIDE SOFTWARE

Upwork



Vue globale : le “front-end”

FRONT-END DEVELOPMENT



Inclusion de fichiers externes

Séparer le HTML du PHP pour plus de modularité/réutilisabilité

Pour pouvoir utiliser dans un script `a.php` des variables/fonctions/... stockées dans un script `b.php`, on utilise l'une des instructions suivantes :

- `include("b.php")` : importe le contenu de `b.php` dans `a.php` sans générer d'erreur si `b.php` n'existe pas.
- `require("b.php")` : idem mais génère une erreur fatale et met fin au script `a.php` en cas d'absence de `b.php`.
- `include_once` et `require_once` se comportent comme `include` et `require` respectivement mais importent une seule fois le fichier demandé.

`principal.php`

Extension de fichiers PHP externes

On peut utiliser pour extension de fichiers PHP :

- `.inc` : le navigateur affichera le contenu intégral du fichier.
- `.inc.php` : si le fichier ne contient que des affectations de variables (eg, paramètres sensibles), le serveur ne renverra rien au navigateur.

test.inc

```
1 $password = "WILL show in browser";
```

test.inc.php

```
1 $password = "WON'T show in browser";
```

Plusieurs types de commentaires :

- `//` sur une seule ligne
- `#` sur une seule ligne
- `/*` sur
plusieurs
lignes `*/`
- `/**`
* sur plusieurs lignes
* pour génération automatique
* de documentation (PHPDoc)
*/