

Développement PHP

Partie 1 : Introduction

DENIS LOEUILLET – IFA – 2017

Source principale :

<https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql>

Partie 1 : Introduction

- Fonctionnement d'un site écrit en PHP
- Historique PHP
- Préparer son environnement de travail
- Syntaxe de base de PHP : écrire son premier script
- Configurer PHP pour visualiser les erreurs

Partie 1 : Introduction

Fonctionnement d'un site écrit en PHP

- Les sites statiques et dynamiques
- Fonctionnement d'un site Web
- Les langages du Web
- Les concurrents de PHP et MySQL

Fonctionnement d'un site écrit en PHP

Les sites statiques et dynamiques

- Les sites statiques
 - Peu complexes
 - HTML, CSS
 - Pas de mise à jour automatique : il faut modifier le code source
 - Sites vitrine uniquement
 - De plus en plus rare
- Les sites dynamiques
 - Plus complexes
 - HTML, CSS + PHP, MySQL par exemple
 - Mise à jour automatique sans intervention du webmaster
 - La plupart des sites web actuels

Fonctionnement d'un site écrit en PHP

Fonctionnement d'un site web

Internet est un réseau composé d'ordinateurs.

Ces ordinateurs peuvent être classés en deux catégories :

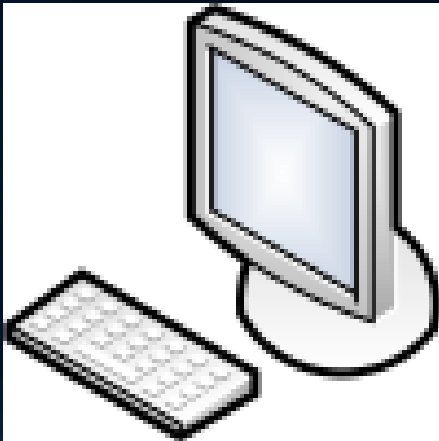
- Les clients
- Les serveurs

Fonctionnement d'un site écrit en PHP

Fonctionnement d'un site web

- Les clients

- Ordinateurs des internautes
- Chaque client est un visiteur d'un site Web



- Les serveurs

- Ordinateurs puissants
- Stockent et délivrent sites Web aux internautes c'est-à-dire aux clients
- Indispensables au bon fonctionnement du Web



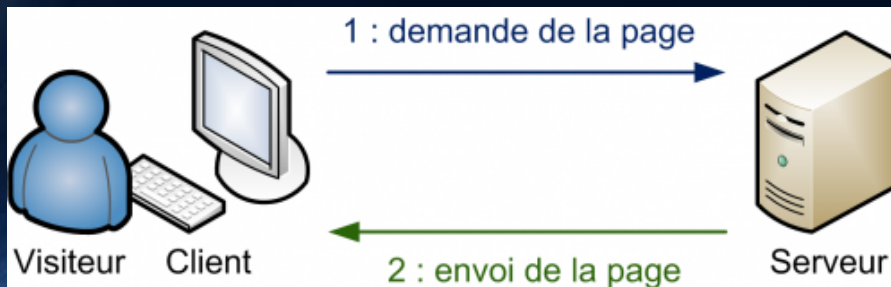
Fonctionnement d'un site écrit en PHP

Fonctionnement d'un site Web

Comment communiquent les clients et les serveurs?

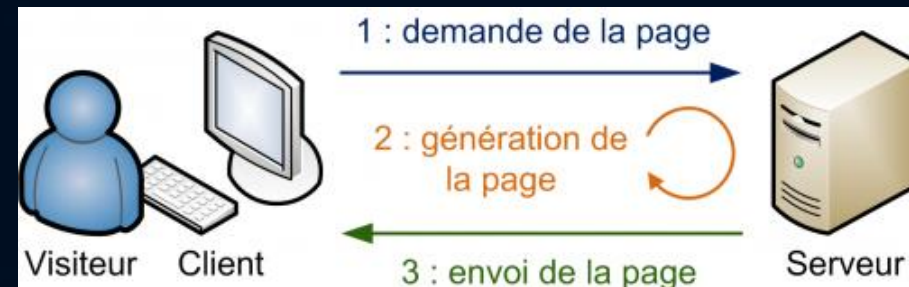
- Sites statiques

- Le client demande une page au serveur
- Le serveur envoie la page



- Sites dynamiques

- Le client demande une page au serveur
- Le serveur prépare la page
- Le serveur envoie la page générée



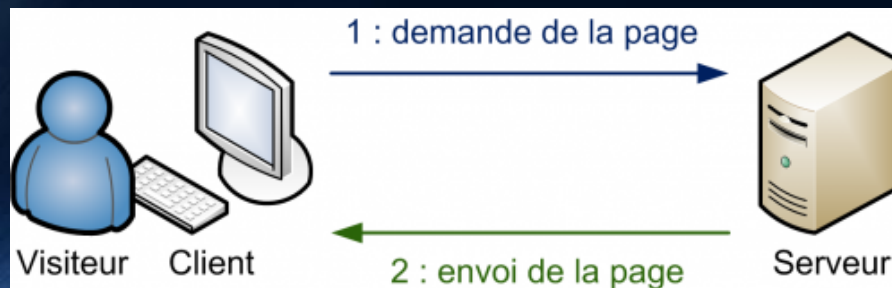
Fonctionnement d'un site écrit en PHP

Fonctionnement d'un site Web

Comment communiquent les clients et les serveurs?

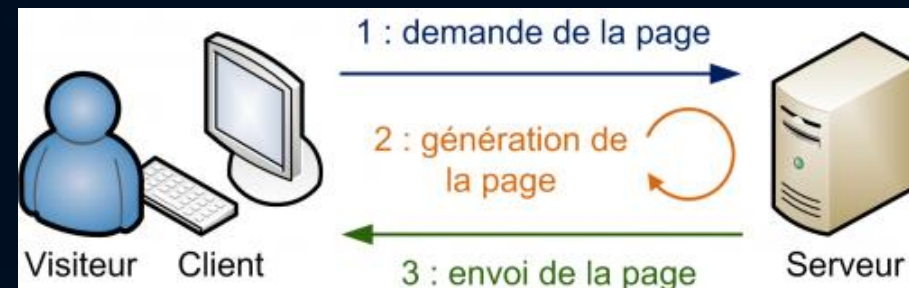
- Sites statiques

- Il ne se passe rien d'autre
- le serveur stocke des pages Web et les envoie aux clients qui les demandent sans les modifier



- Sites dynamiques

- La page est générée à chaque fois qu'un client la réclame
- La page est générée spécialement pour le client
- Le contenu d'une même page peut changer d'un instant à l'autre



Fonctionnement d'un site écrit en PHP

Les langages du Web

La création d'un site Web ➡ connaître plusieurs langages

- Sites statiques

- HTML et CSS : base incontournable

- ✓ HTML : langage de base sites Web
- ✓ CSS : langage de mise en forme
- ✓ Base de tous les sites Web
- ✓ Serveur envoie au client code HTML et CSS



- Sites dynamiques

- HTML et CSS ne suffisent pas : PHP et MySQL

- ✓ PHP :
 - Rend les sites dynamiques
 - Génère la page Web
- ✓ MySQL :
 - ✓ SGBD : Système de Gestion de Base de Données
 - ✓ Rôle : enregistrer et restituer des données



Fonctionnement d'un site écrit en PHP

Les langages du Web

La création d'un site Web  connaître plusieurs langages

- Le langage PHP
 - PHP peut fonctionner seul
 - Compris uniquement par les serveurs
 - Rend les sites Web dynamiques
 - Génère la page Web
 - Génère code HTML
 - Les clients ne comprennent pas PHP, ils connaissent uniquement HTML et CSS
- Le langage MySQL
 - SGBD : Système de Gestion de Base de Données
 - Rôle : enregistrer et restituer des données

Fonctionnement d'un site écrit en PHP

Les concurrents de PHP et MySQL

HTML et CSS  pas de concurrents, ce sont des standards

Par contre beaucoup de concurrence pour PHP et MySQL

Fonctionnement d'un site écrit en PHP

Les concurrents de PHP et MySQL

Les concurrents de PHP

- ASP.NET
 - ✓ Conçu par MICROSOFT
 - ✓ Framework .NET
 - ✓ Langage C#
- Ruby on Rails
 - ✓ Langage Ruby
- Django
 - ✓ Similaire Ruby on Rails mais langage Python
- Java et les JSP (Java Server Pages) appelé JEE ou Java EE :
 - ✓ Utilisé dans le monde professionnel
 - ✓ Mise en place plus lourde
 - ✓ Apprécié des institutions (impôts français par exemple)
- ...

Fonctionnement d'un site écrit en PHP

Les concurrents de PHP et MySQL

Avantages de PHP

- Importante communauté
- Facile à utiliser
- Idéal pour les débutants et les professionnels : Wikipédia et Facebook fonctionnent grâce à PHP

Fonctionnement d'un site écrit en PHP

Les concurrents de PHP et MySQL

Les concurrents de MySQL

Contrairement à PHP les concurrents de MySQL ne sont pas systématiquement libres et gratuits

➤ Oracle

- ✓ Le SGBD le plus célèbre, complet et puissant
- ✓ payant
- ✓ Plutôt réservé aux entreprises

➤ MarieDB

- ✓ Variante libre de MySQL qui a été racheté par Oracle

➤ Microsoft SQL Server

- ✓ Microsoft
- ✓ Souvent utilisé avec ASP.NET
- ✓ Payant

➤ PostgreSQL

- ✓ Libre et gratuit
- ✓ Fonctionnalités avancées
- ✓ Communauté moins importante que MySQL et Oracle

Fonctionnement d'un site écrit en PHP

Les concurrents de PHP et MySQL

Plusieurs combinaisons possibles

- On peut combiner ces outils entre eux, on peut par exemple utiliser PHP avec une autre base de données que MySQL comme Oracle ou PostgreSQL
- De même, MySQL peut être utilisé avec n'importe quel autre langage : Java, Python, Ruby...
- Cependant, PHP + MySQL est la combinaison la plus courante

Partie 1 : Introduction

Historique de PHP

- Définition
- Histoire
- Utilisations
- Versions
- Historique des versions
- PHP 7

Historique de PHP

Définition

- PHP: Hypertext Preprocessor, plus connu sous son sigle PHP
- langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.
- PHP est un langage orienté objet.
- PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.
- Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

Historique de PHP

Histoire

- PHP fut créé en 1994 par Rasmus Lerdorf pour son site web.
- Rasmus décida en 1995 de publier son code, pour que tout le monde puisse l'utiliser et en profiter.
- PHP s'appelait alors PHP/FI (pour *Personal Home Page Tools/Form Interpreter*).
- En 1997, deux étudiants, Andi Gutmans et Zeev Suraski, redéveloppèrent le cœur de PHP/FI. Ce travail aboutit un an plus tard à la version 3 de PHP, devenu alors *PHP: Hypertext Preprocessor*. Peu de temps après, Andi Gutmans et Zeev Suraski commencèrent la réécriture du moteur interne de PHP. Ce fut ce nouveau moteur, appelé Zend Engine — le mot *Zend* est la contraction de *Zeev* et *Andi* — qui servit de base à la version 4 de PHP.

Historique de PHP

Utilisations

- En 2002, PHP est utilisé par plus de 8 millions de sites Web à travers le monde
- En 2007 par plus de 20 millions
- En 2013 par plus de 244 millions.
- De plus, PHP est devenu le langage de programmation web côté serveur le plus utilisé depuis plusieurs années.
- En 2010, PHP est le langage dont les logiciels open source sont les plus utilisés dans les entreprises, avec 57 % de taux de pénétration.

Année	Part de marché
2010	75%
2013	75%
2016	82%

Historique de PHP

Versions

- La version actuelle est la version 7.1, sortie le 1 décembre 2016. Elle utilise *Zend Engine 2*
- Depuis juin 2011 une mise à jour / an comportant des changements fonctionnels importants, la durée de vie d'une branche est de 3 ans et trois branches stables sont maintenues. Cela signifie que lorsqu'une nouvelle version de PHP 5.x sort, la version 5.x-3 n'est plus supportée.
- Depuis la sortie de PHP 5.6, la configuration est :
 - branche de développement : 7.1 ;
 - branches encore supportées (corrections de bugs et de la sécurité) : 7.0, 5.6 ;
 - toutes les branches < 5.6 ne sont plus supportées.
- Il est à noter qu'historiquement, PHP disposait d'une configuration par défaut privilégiant la souplesse à la sécurité (par exemple register globals, qui a été activé par défaut jusqu'à PHP 4.2). Cette souplesse a permis à de nombreux développeurs d'apprendre PHP mais le revers de la médaille a été que de nombreuses applications PHP étaient mal sécurisées. Le sujet a bien été pris en main par le PHPGroup qui a mis en place des configurations par défaut mettant l'accent sur la sécurité. Il en résultait une réputation de langage peu sécurisé, réputation d'insécurité qui n'a plus de raison d'être

Historique de PHP

Historique des versions

En 2005, le projet de faire de PHP un langage fonctionnant d'origine en Unicode (en vue de faciliter la manipulation de texte dans différentes langues) a été lancé par Andrei Zmievski, ceci en s'appuyant sur la bibliothèque [International Components for Unicode](#) (ICU) et en utilisant UTF-16 pour représenter les chaînes de caractères dans le moteur.

Étant donné que cela représentait un changement majeur tant dans le fonctionnement du langage que dans le code PHP créé par ses utilisateurs, il fut décidé d'intégrer cela dans une nouvelle version 6.0 avec d'autres fonctionnalités importantes alors en développement. Toutefois, le manque de développeurs experts en Unicode ainsi que les problèmes de performance résultant de la conversion des chaînes de et vers UTF-16 (rarement utilisé dans un contexte web), ont conduit au report récurrent de la livraison de cette version

- 2009 : 5.3
- 2012 : 5.4
- 2013 : 5.5
- 2014 : 5.6
- Décembre 2015 : 7.0
- 2017 : 7.1

Historique de PHP

Historique des versions

Depuis janvier 2014, l'idée d'une nouvelle version majeure introduisant Unicode mais se basant sur UTF-8 (largement devenu depuis le standard du Web pour l'Unicode) et permettant certains changements pouvant casser la rétro-compatibilité avec du code PHP ancien est de nouveau discutée et les RFC sont maintenant triées selon leur implémentation en 5.x (évolutions ne causant pas ou marginalement de cassure de la rétro-compatibilité) ou dans la future version majeure (évolutions majeures du moteur et évolutions impliquant une non-compatibilité ascendante).

Historique de PHP

PHP 7

- La version finale de PHP 7 est sortie en décembre 2015
- PHP 7 est plus rapide
- La mise à jour des applications vers PHP 7 pourrait engendrer un surcroît de performance de 25% à 70%

Partie 1 : Introduction

Préparer son environnement de travail

- Introduction
- Les programmes nécessaires

Préparer son environnement de travail

Introduction

- PHP s'exécute sur le serveur
- PHP génère des pages web
- Seul un serveur peut lire PHP
- Votre ordinateur n'est pas un serveur
- 2 solutions :
 - Utiliser un serveur distant
 - Transformer votre ordinateur en serveur

Préparer son environnement de travail

Les programmes nécessaires

- Site statique

- Un éditeur de texte pour écrire les programmes
 - ❖ Bloc note
 - ❖ Notepad++
 - ❖ ...
- Un navigateur web pour tester les programmes
 - ❖ Firefox
 - ❖ Internet Explorer
 - ❖ Google Chrome
 - ❖ ...

- Site dynamique

- Un éditeur de texte pour écrire les programmes
- Un navigateur web pour tester les programmes
- Programmes pour interpréter PHP

Préparer son environnement de travail

Les programmes nécessaires

- Site dynamique

- Programmes pour interpréter PHP

- ❖ Apache :

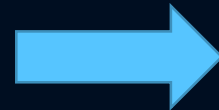


- ✓ C'est serveur Web HTTP : délivre les pages Web
 - ✓ Ne traite que les pages HTML

- ❖ PHP

- ✓ Plug-in pour Apache qui le rend capable de traiter des pages web dynamiques en PHP

- ❖ MySQL : logiciel gestion BDD



- Serveur LAMP ou XAMPP :
paquetage tout prêt pour LINUX

- Linux

- Apache

- MySQL

- PHP/Python/Perl : le langage de programmation associé

Préparer son environnement de travail

Les programmes nécessaires

- Site dynamique
 - Si vous ne disposez pas de serveur web il faut en installer un sur votre propre ordinateur

Préparer son environnement de travail

Les programmes nécessaires

L'éditeur de texte

- Editeur de texte simple
 - Bloc note
 - Suffit à écrire en HTML ou PHP
 - Aucun apport spécifique
- Editeur de texte spécialisé
 - Notepad++, Sublime Text, PHPStorm (payant)
 - Colore le code
 - Numérote les lignes
 - Recherche de texte
 - ...

Préparer son environnement de travail

En résumé

- Pour créer des sites web dynamiques, nous devons disposer d'un serveur Web
- Les principaux outils dont nous avons besoin sont :
 - Apache : le serveur web ;
 - PHP : le programme qui permet au serveur web d'exécuter des pages PHP ;
 - MySQL : le logiciel de gestion de bases de données.
- Il est conseillé d'utiliser un éditeur de texte qui colore le code source comme Sublime Text pour programmer convenablement en PHP.

Partie 1 : Introduction

Syntaxe de base de PHP

- Introduction
- Les balises PHP
- Afficher du texte
- Les commentaires

Syntaxe de base de PHP

Introduction

Dans la première partie du cours nous avons découvert le principe de fonctionnement de PHP maintenant nous allons passer à la partie opérationnelle et écrire notre premier programme PHP.

Syntaxe de base de PHP

Les balises PHP

HTML : composé de balises

- ``, `<a>`, `<p>`...

Le code PHP vient s'insérer au milieu du code HTML. On va progressivement placer dans nos pages Web des morceaux de code PHP à l'intérieur du HTML.

Ces bouts de codes PHP seront les parties dynamiques de la page, c'est-à-dire les parties qui peuvent changer toutes seules.

Syntaxe de base de PHP

Les balises PHP : la forme d'une balise PHP

Pour utiliser PHP on introduit une nouvelle balise :

- Elle commence par `<?php`
- Elle se termine par `?>`
- Le code PHP se trouve à l'intérieur de cette balise

Syntaxe de base de PHP

Les balises PHP : la forme d'une balise PHP

Balise PHP vide :

- `<?php ?>`

On insère le code PHP à l'intérieur de la balise

- `<?php /* Le code PHP se met ici */ ?>`

Le code PHP peut aussi s'écrire sur plusieurs lignes

- `<?php
/* Le code PHP se met ici
Et ici
Et encore ici */
?>`

Syntaxe de base de PHP

Les balises PHP : insérer une balise PHP au milieu du code HTML

La balise PHP s'insère au milieu du code HTML :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est une page de test avec des balises PHP</title>
5     <meta charset="utf-8" />
6   </head>
7   <body>
8     <h2>Page de test</h2>
9
10    <p>
11      Cette page contient du code HTML avec des balises PHP.<br />
12      <?php /* Insérer du code PHP ici */ ?>
13      Voici quelques petits tests :
14    </p>
15
16    <ul>
17      <li style="color: blue;">Texte en bleu</li>
18      <li style="color: red;">Texte en rouge</li>
19      <li style="color: green;">Texte en vert</li>
20    </ul>
21
22    <?php
23      /* Encore du PHP
24      Toujours du PHP */
25    ?>
26  </body>
27 </html>
```

Syntaxe de base de PHP

Les balises PHP : insérer une balise PHP au milieu du code HTML

Peut-on placer une balise PHP n'importe où ?

Oui, n'importe où :

- Dans la page
- Dans le header de la page

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est une page de test <?php /* Code PHP */ ?></title>
5     <meta charset="utf-8" />
6   </head>
```

Syntaxe de base de PHP

Les balises PHP : insérer une balise PHP au milieu du code HTML

Peut-on placer une balise PHP n'importe où ?

Oui, n'importe où :

- Au milieu d'une balise HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ceci est une page de test</title>
5     <meta <?php /* Code PHP */ ?> charset="utf-8" />
6   </head>
```

Syntaxe de base de PHP

Les balises PHP : insérer une balise PHP au milieu du code HTML

À quoi ça sert?

Comment ça fonctionne? Comment PHP génère du code HTML?

Syntaxe de base de PHP

Les balises PHP : l'instruction « echo »

- PHP est un langage de programmation
- HTML est un langage de description car il permet de décrire une page Web
- Tout langage de programmation contient des instructions
- Chaque instruction se termine par un « ; »
- Une instruction commande à l'ordinateur d'effectuer une action précise
- L'instruction « echo » permet d'insérer du texte dans la page Web

```
1 <?php echo "Ceci est du texte"; ?>
```


Syntaxe de base de PHP

Les balises PHP : l'instruction « echo »

```
1 <?php echo "Ceci est du texte"; ?>
```

- On écrit l'instruction « echo » suivie du texte à afficher entre guillemets.
- Les guillemets permettent de délimiter le début et la fin du texte
- L'instruction se termine par un « ; » ce qui signifie fin de l'instruction
- On peut afficher des balises

```
1 <?php echo "Ceci est du <strong>texte</strong>"; ?>
```

Que va-t-il s'afficher?

Syntaxe de base de PHP

Les balises PHP : l'instruction « echo »

Comment afficher des guillemets?

- Lorsque l'ordinateur rencontre un guillemet il comprend que le texte à afficher débute ou finit là
- La solution consiste à faire précéder le guillemet d'un antislash « \ »

```
1 <?php echo "Cette ligne a été écrite \"uniquement\" en PHP."; ?>
```

Syntaxe de base de PHP

Les balises PHP : l'instruction « echo »

Exemple :

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title>Notre première instruction : echo</title>
5         <meta charset="utf-8" />
6     </head>
7     <body>
8         <h2>Affichage de texte avec PHP</h2>
9
10        <p>
11            Cette ligne a été écrite entièrement en HTML.<br />
12            <?php echo "Celle-ci a été écrite entièrement en PHP."; ?>
13        </p>
14    </body>
15 </html>
```

Syntaxe de base de PHP

Enregistrer une page PHP

- Enregistrez la page avec l'extension .php
- Enregistrez la page dans votre dossier de travail
- testez

Syntaxe de base de PHP

Les balises PHP : l'instruction « echo »

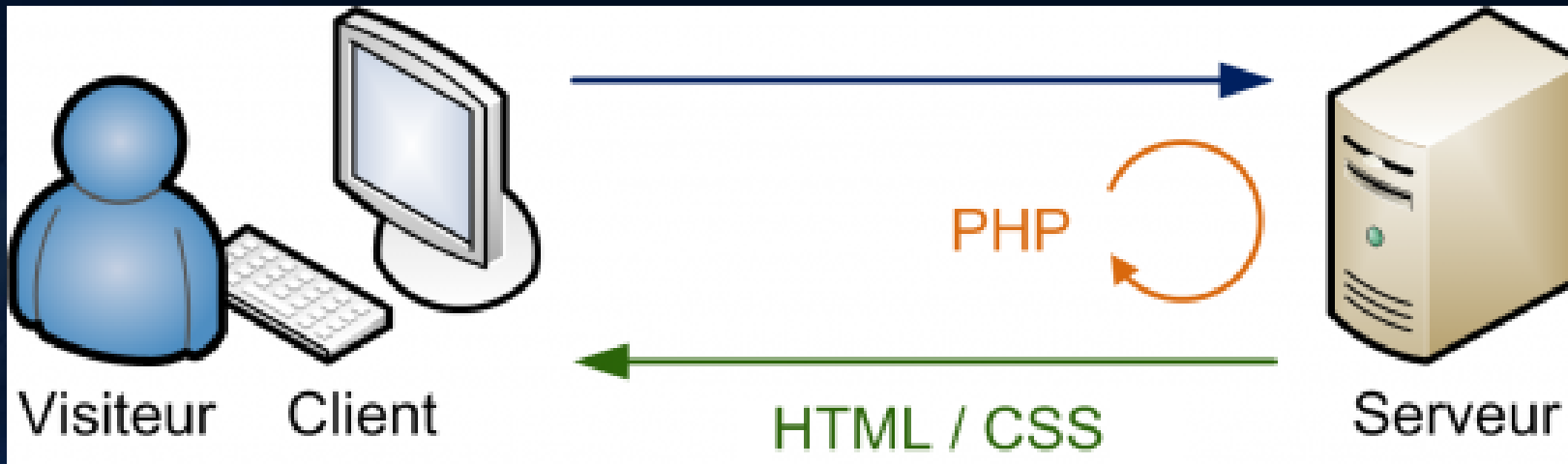
Autre exemple :

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8" />
5         <title>Ma page web</title>
6     </head>
7     <body>
8         <h1>Ma page web</h1>
9         <p>Aujourd'hui nous sommes le <?php echo date('d/m/Y h:i:s'); ?>.</p>
10    </body>
11 </html>
```

Syntaxe de base de PHP

Comment PHP génère du code HTML

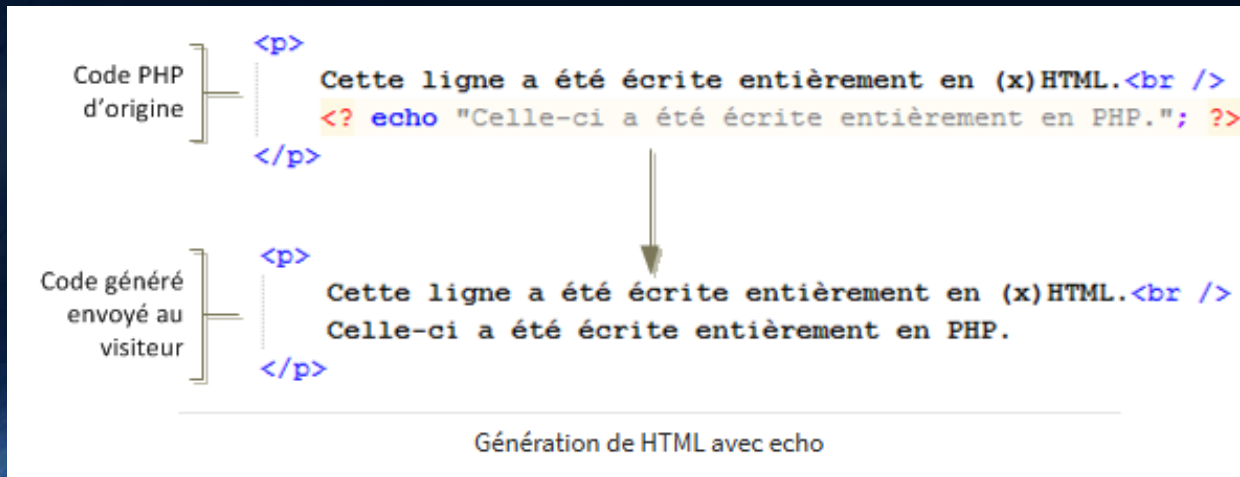
- L'instruction « echo » demande à PHP d'insérer à cet endroit le texte que vous demandez.
- Rappel : PHP génère du HTML et renvoie au client uniquement HTML + CSS



Syntaxe de base de PHP

Comment PHP génère du code HTML

- Le code PHP est exécuté en premier et l'ordinateur fait ce qu'on lui demande. Ici on lui a dit « Affiche ce texte ici ».
- Une fois toutes les instructions PHP exécutées, la page qui sort est une page qui ne contient que du HTML ! C'est cette page de « résultat » qui est envoyée au visiteur, car celui-ci ne sait lire que le HTML.



Syntaxe de base de PHP

Les commentaires

- Très important en PHP comme dans tous les langages de programmation
- Un commentaire est un texte que vous mettez pour vous dans le code PHP. Ce texte est ignoré, c'est-à-dire qu'il disparaît complètement lors de la génération de la page. Il n'y a que vous qui voyez ce texte.
- Cela permet de vous y retrouver dans votre code PHP, parce que si vous n'y touchez pas pendant des semaines et que vous y revenez, vous risquez d'être un peu perdus.
- Vous pouvez écrire tout et n'importe quoi, le tout est de s'en servir à bon escient.

Syntaxe de base de PHP

Les commentaires

- 2 types de commentaires
 - Les commentaires monoligne
 - Les commentaires multilignes

Syntaxe de base de PHP

Les commentaires

- Les commentaires monoligne

Pour indiquer que vous écrivez un commentaire sur une seule ligne, vous devez taper deux slashes : «//». Tapez ensuite votre commentaire.

```
1 <?php
2 echo "J'habite en Chine."; // Cette ligne indique où j'habite
3
4 // La ligne suivante indique mon âge
5 echo "J'ai 92 ans.";
6 ?>
```

Syntaxe de base de PHP

Les commentaires

- Les commentaires multilignes

Il faut commencer par écrire/* puis refermer par*/

```
1 <?php
2 /* La ligne suivante indique mon âge
3 Si vous ne me croyez pas...
4 ... vous avez raison ;o) */
5 echo "J'ai 92 ans.";
6 ?>
```

Syntaxe de base de PHP

Résumé

- Les pages web contenant du PHP ont l'extension.php.
- Une page PHP est en fait une simple page HTML qui contient des instructions en langage PHP.
- Les instructions PHP sont placées dans une balise<?php ?>.
- Pour afficher du texte en PHP, on utilise l'instruction « echo ».
- Il est possible d'ajouter des commentaires en PHP pour décrire le fonctionnement du code. On utilise pour cela les symboles//ou/* */.

Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Configurer PHP pour afficher les erreurs
- Tester l'affichage des erreurs

Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Il va vous arriver de faire des erreurs lors de votre programmation
- Lorsqu'un script PHP plante ➡ page blanche
- Pour faciliter le débogage ➡ afficher les erreurs PHP

➡ changer configuration PHP

Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Configurer PHP pour afficher les erreurs
 - Effectivement PHP est configurable
 - Par défaut, PHP n'affiche pas les erreurs
 - La configuration de PHP se fait dans un fichier appelé « php.ini »
 - Où se trouve ce fichier?

Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Configurer PHP pour afficher les erreurs

➤ Localiser php.ini du serveur

❖ Il existe une commande PHP « `phpinfo()` » (c'est une fonction), nous allons l'utiliser pour localiser le fichier

❖ Créez un fichier PHP avec le code suivant :

```
1 <?php
2
3 phpinfo();
```

❖ Enregistrez le sous le nom `info.php`


❖ Affichez la page

phpinfo()

localhost:8888/info.php

Applications My Tasks in open...


PHP Version 7.0.12



System	Darwin PORT.local 16.3.0 Darwin Kernel Version 16.3.0: Thu Nov 17 20:23:58 PST 2016; root:xnu-3789.31.2~1/RELEASE_ARM64_T8020
Build Date	Oct 24 2016 18:22:06
Configure Command	'./configure' '--with-apxs2=/Applications/MAMP/Library/bin/apxs' '--with-gd' '--with-jpeg-dir=/Applications/MAMP/Library' '--with-png-dir=/Applications/MAMP/Library' '--with-zlib' '--with-zlib-dir=/Applications/MAMP/Library' '--with-freetype-dir=/Applications/MAMP/Library' '--prefix=/Applications/MAMP/bin/php/php7.0.12' '--exec-prefix=/Applications/MAMP/bin/php/php7.0.12' '--sysconfdir=/Applications/MAMP/bin/php/php7.0.12/conf' '--with-config-file-path=/Applications/MAMP/bin/php/php7.0.12/conf' '--enable-ftp' '--enable-gd-native-ttf' '--with-bz2=/Applications/MAMP/Library' '--with-ldap' '--with-mysql=mysqlnd' '--enable-mbstring=all' '--with-curl=/Applications/MAMP/Library' '--enable-sockets' '--enable-bcmath' '--with-imap=shared,/Applications/MAMP/Library/lib/imap-2007f' '--with-imap-ssl=/Applications/MAMP/Library' '--enable-soap' '--with-kerberos' '--enable-calendar' '--with-pgsql=shared,/Applications/MAMP/Library/pg' '--enable-xml' '--with-libxml-dir=/Applications/MAMP/Library' '--with-gettext=shared,/Applications/MAMP/Library' '--with-xsl=/Applications/MAMP/Library' '--with-pdo-mysql=mysqlnd' '--with-pdo-pgsql=shared,/Applications/MAMP/Library/pg' '--with-mcrypt=shared,/Applications/MAMP/Library' '--with-openssl=/Applications/MAMP/Library' '--enable-zip' '--with-iconv=/Applications/MAMP/Library' '--enable-opcache' '--enable-intl' '--with-ldap=shared' '--with-icu-dir=/Applications/MAMP/Library' '--enable-wddx' '--with-libexpat-dir=/Applications/MAMP/Library' '--with-readline' '--with-mhash' 'YACC=/Applications/MAMP/Library/bin/bison'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MAMP/bin/php/php7.0.12/conf
Loaded Configuration File	/Applications/MAMP/bin/php/php7.0.12/conf/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, mcrypt.*, mdecrypt.*

This program makes use of the Zend Scripting Language Engine:

Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies



Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Configurer PHP pour afficher les erreurs
 - Cette page contient de multiples informations
 - ❖ Version PHP
 - ❖ Type de serveur Web
 - ❖ Localisation du ou des fichiers de configuration PHP
 - ✓ Retrouvez la ligne « Loaded Configuration File » et notez la valeur
 - ✓ Ouvrez le fichier et modifier les clés suivantes :
 - `Error_reporting = E_ALL`
 - `Display_errors = On`
 - ✓ Enregistrez le fichier et relancez le serveur

Partie 1 : Introduction

Configurer PHP pour visualiser les erreurs

- Tester l'affichage des erreurs
 - Créez une erreur dans votre script PHP
 - Affichez la page

Partie 1 : Introduction

Récapitulatif

Qu'est-ce qui est indispensable au bon fonctionnement d'un site web écrit avec PHP afin qu'il s'affiche correctement dans un navigateur ?

1. Un serveur web (Apache, Nginx...)
2. PHP
3. Un navigateur

Partie 1 : Introduction

Récapitulatif

PHP permet de créer des sites...

1. Statiques
2. Dynamiques

Partie 1 : Introduction

Récapitulatif

Quelles sont les technologies concurrentes de PHP?

1. MySQL
2. Django
3. HTML
4. Ruby on Rails
5. ASP.NET

Partie 1 : Introduction

Récapitulatif

Quelles sont éditeurs de texte dans lesquels on peut écrire du code PHP :

1. Apache
2. Sublime Text
3. PHPStorm
4. Google Chrome

Partie 1 : Introduction

Récapitulatif

A quoi ressemble une balise PHP ?

1. `<?/ ?>`
2. `<php ?>`
3. `<?php ?>`

Partie 1 : Introduction

Récapitulatif

Pourquoi faut-il configurer PHP pour afficher des erreurs sur sa machine ?

1. Cela nous permet de comprendre nos erreurs et de les résoudre plus vite
2. Cela accélère le temps d'affichage d'une page
3. Cela nous permet d'éviter un bug de PHP qui empêche les pages de s'afficher correctement