

PHP : INTRO – HTML – WAMP – PREMIERS PROGRAMMES

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

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

<http://www.w3schools.com/php/>

Open Class Room

SOMMAIRE

Sommaire	1
INTRO-HTML-WAMP-Premiers programmes	3
1. Introduction au PHP	3
Exemples et exercices.....	3
Objectifs du cours	3
Site dynamique et site statique – Rôle de l'utilisateur et de l'administrateur	4
Historique du PHP.....	5
Architecture client serveur	6
Principes du PHP : écrire du code HTML	9
2. Rappels HTML - CSS	12
HTML de base : h1-h6, p, br, img.....	12
Squelette html	12
CSS de base.....	13
Chargez tous les exemples HTML	14
Exemple 1 : une page de base	14
Exemple 2 : une page de base avec un lien vers un formulaire	15
Exemple 3 : une page de base avec un lien vers un formulaire dans un tableau	17
3. Les outils de développement	19
Outils pour le fonctionnement d'un site WEB.....	19
Outils de développement PHP-MySQL.....	20
4. Installation de WAMP.....	22
5. Premiers usages de WAMP	23
1 : Démarrer WAMP	23
2 : Installer l'icône WAMP directement dans la barre d'icône	23
3 : Passer Wamp en français	23
4 : Consulter les serveurs dans le gestionnaire des tâches	24
5 : Tester WAMP : afficher la page WampServer	25
6 : Allez dans le dossier des projets : C:\wamp64\www, créez un dossier « home »	25
7 : Afficher le contenu de « home » dans la page WampServer	25
8 : Accéder au dossier « home » dans la page WampServer	25
6. Premiers programmes en PHP	26
9 : Premier programme php	26
10 : Fichier index.php	27
11 : Deuxième programme php.....	28
12 : Consulter le contenu des dossiers d'un site en ligne !	29
WAMP - version détaillée (on peut passer !)	30

0 : Reprise détaillée du paragraphe précédent : premiers usages - résumé	30
1 : Tester WAMP	30
2 : Travailler avec des fichiers PHP	32
3 : Paramétrages divers	37
4 : PHPMyAdmin	37
5 : Paramétrage des installations : display_errors et display_startup_errors	37
MAMP – version détaillée	39
1 : Principes.....	39
2 : Lancer MAMP	39
3 : Problème d’installation – problème de port	39
4 : Accès aux projets web	39
5 : PHPMyAdmin.....	39
6 : Fichier php.ini	40
7. Trois exemples de l’instruction echo (ou print)	41
Chargez et installez les exemples	41
Exemple 1 : code HTML minimum.....	41
Exemple 2 : code HTML minimum avec un echo php	42
Exemple 3 : page HTML standard avec un echo php	43
Exercice echo	44
Principes généraux du code PHP	45
8. Gestion des erreurs	46
Affichage des erreurs - Rappel	46
Principaux messages d’erreurs.....	46
TP : Gestion des erreurs - variables display_errors et display_startup_errors à On	47
9. Inclusion de fichiers PHP – factorisation du code HTML	48
Rappel sur les balises structurantes HTML.....	48
Le problème	48
Solution PHP : l’inclusion de fichier en PHP	48
Exemple :	49
Solution HTML : l’inclusion de fichier en HTML.....	50
TP : Includes.....	51
10. Documentation	52
Principes	52
Exemple d’usage : php.net	52

Edition : février 2019

INTRO-HTML-WAMP-PREMIERS PROGRAMMES

1. Introduction au PHP

Exemples et exercices

Les exemples du cours sont dans un fichier zip fournis avec l'article du cours : 00-BASES-DU-LANGAGE.zip.

Les exemples sont présentés dans un chapitre en vert.

Chargez ce fichier et mettez-le dans le dossier Partie_2 du répertoire web « www » du serveur WAMP.

Les exercices à faire sont présentés dans un chapitre en jaune.

Objectifs du cours

Des pages web : HTML et aussi...

- Formulaire de contact
- Espace membre avec login et pseudo, forum, actualités, newsletter, etc.
- Back office
- Faire circuler les informations de page en page
- Faire attention aux failles de sécurité,
- Enregistrer les infos dans une BD,
- Structurer son site avec la POO et l'architecture MVC.

Site statique (web 1.0)

L'utilisateur consulte le contenu du site sans fournir d'informations (message, login, etc.).

L'administrateur ajoute de nouveaux éléments dans le site (de nouvelles informations, de nouveaux articles, de nouvelles pages) en revenant dans le code, donc en programmant.

Les sites statiques sont faits uniquement en HTML, CSS et JavaScript : ce sont des langages clients universels.

Site dynamique (web 2.0)

L'utilisateur fournit des informations au site : messages, login, etc. Cette partie du site est publique (même s'il faut un mot de passe pour se connecter) : c'est le **front-office**.

L'administrateur gère le contenu du site (ajout, modification, suppression d'informations, d'articles, de messages, d'utilisateur, modération de messages, etc.) par une partie de site dédiée à l'administration : c'est le **back-office**.

Les sites dynamiques sont faits avec :

- les langages clients : HTML, CSS et JavaScript
- **un langage serveur** : PHP ou Java ou C# ou Ruby ou Python ou JavaScript (Node), etc.
- Ces langages serveurs utilisent en général **une base de données** : MySQL, MariaDB, Oracle, SQL-Server, PostgreSQL, SQLite, etc.

Souvent, on associe une BD à un langage serveur :

- PHP-MySQL,
- Java-Oracle,
- C#-SQL-Server.

On utilise aussi souvent un **framework** qui facilite et accélère le travail :

- Symfony, Laravel, Zend (PHP),
- JEE (Java),
- .Net (c#),
- RubyOnRails (Ruby),
- Django (Python).

Historique du PHP

PHP-MySQL : début du WEB 2.0 grand public.

PHP : langage simple, un peu « sale » (versus Java, langage propre).

Aujourd'hui : PHP objet, langage propre.

Le PHP a intégré les aspects sécurité.

S'utilise souvent avec des frameworks (Laravel, Symfony, Zend, etc.).

<https://fr.wikipedia.org/wiki/PHP>

Langages côté client

HTML, CSS, JavaScript

Applications côté client

Navigateurs : Firefox, Internet Explorer, Safari, Opera, etc.

Langages côté serveur

SQL, PHP, C#, Java, Python, Ruby, etc.,

Applications côté serveur

Serveurs HTTP (ou serveur web) : Apache (PHP), Tomcat (Java, JEE), IIS (C#, .Net)

Serveurs SGBD : MySQL, Oracle, SQL-Server, PostgreSQL, MariaDB, SQLite

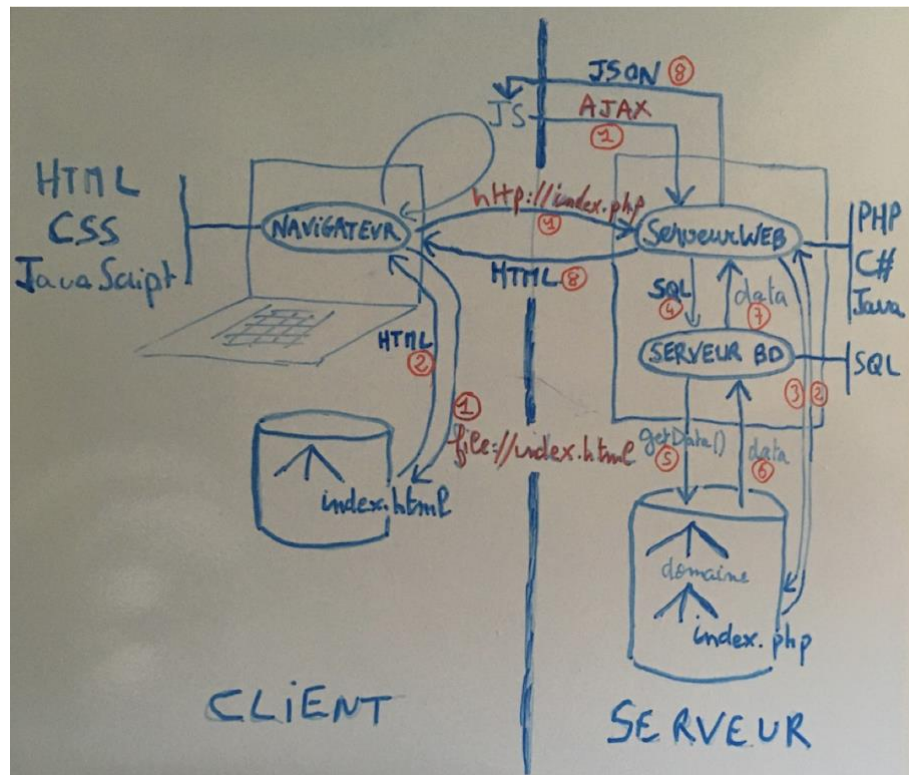
Remarque : un serveur c'est une machine qui fait fonctionner un programme qu'on appelle aussi un serveur (la machine est un serveur parce qu'elle contient un programme serveur). Un programme serveur est un programme qui tourne 24H sur 24 et qui a comme caractéristique de ne pas avoir d'interface utilisateur. Seuls d'autres programmes peuvent communiquer avec lui. Un programme serveur s'appelle **serveur** ou *daemon*.

Le serveur mysql sur PC s'appelle mysqld, « d » pour daemon.

Affichage d'une page HTML dans un navigateur :

➤ 3 cas possibles :

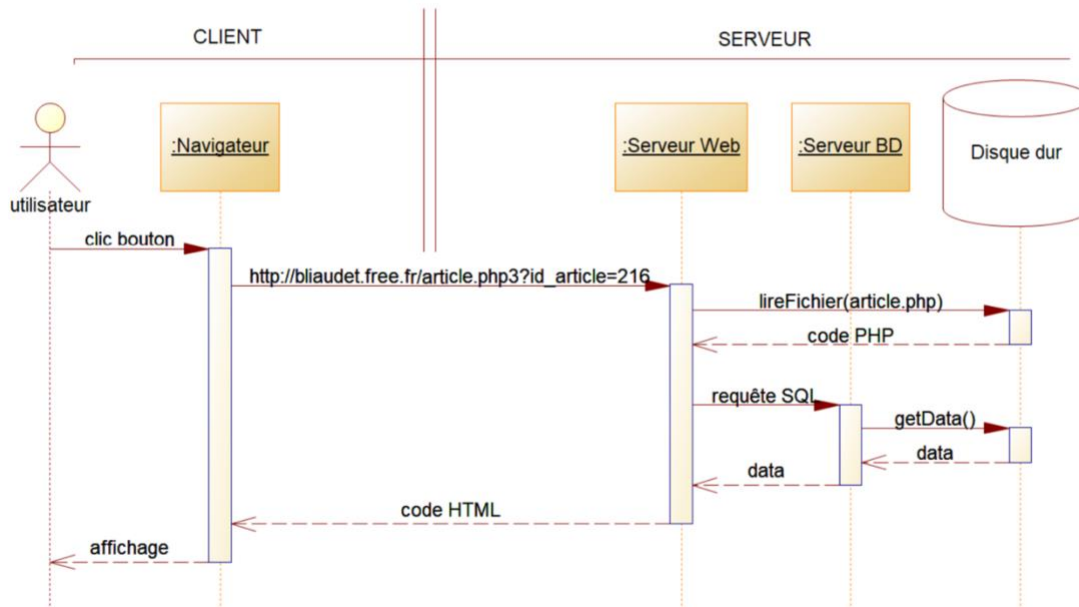
- local,
- langage serveur,
- AJAX



Exemple : comment on accède à une page Web

• Cas 2 : Architecture Client - Serveur

- Comment on accède à la page « introduction web » à partir du site b্লাuidet.free.fr



16

Ecrire du code HTML en PHP

Le PHP sert à écrire du code HTML dynamique, en allant par exemple chercher les informations dans la BD.

Traduction du PHP en HTML

Le navigateur ne comprend que le langage HTML.

Un serveur web permet de traduire du code PHP en page HTML.

Principes de traduction

- 1) Le client demande un fichier HTML ou PHP : une page web, HTML ou PHP.
- 2) Le serveur web récupère cette page HTML ou PHP puis la traduit en HTML (il n'a rien à faire s'il n'y a que du HTML).
- 3) La traduction peut consister à inclure d'autres fichiers PHP et aussi à récupérer des informations dans une base de données ou dans des fichiers.
- 4) La fin de la traduction produit une page entièrement HTML qui est renvoyée au client (au navigateur).

Exemples de traduction

Quand on trouve des instructions PHP, le serveur Apache transforme ces instructions en code HTML

➤ ***Exemple 1 de traduction***

```
<p>
  <?php
    echo "Vous êtes le visiteur n°" . $numVisiteur;
  ?>
</p>
```

La balise <p> est laissée telle quelle. La balise < ?php est traduite.

Le résultat de la traduction donnera, si la variable \$numVisiteur vaut 45 :

```
<p>
  Vous êtes le visiteur n° 45
</p>
```

Il peut aussi y avoir des instructions php qui interrogent la base de données.

➤ **Exemple 2 de traduction**

```
<p>
  <?php
    $reponse = $bdd->query
      ('SELECT prenom FROM utilisateurs WHERE id=235');
    echo "Bonjour " . $reponse;
  ?>
</p>
```

donnera, si le SELECT revoit Bertrand :

```
<p>
  Bonjour Bertrand
</p>
```

Pour fonctionner, cet exemple nécessite qu'on se connecte d'abord à la BD.

Ces pages HTML peuvent être envoyées au navigateur du client qui peut désormais les interpréter et les afficher.

2. Rappels HTML - CSS

HTML de base : h1-h6, p, br, img

<h1> titre </h1> h2-h6

<p>,
,

<a href= « URL » ...

Dans chaque balise, on peut mettre des couples : attribut = « valeur »

Squelette html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Ma page</title>
  </head>

  <body>
    Contenu HTML
  </body>
</html>
```

Principes

selecteur { p1 :v1 ; p2 :v2 ; ... ; pn :vn }

Un selecteur qui définit la ou les balises concernées.

Un couple propriété : valeur.

Exemples de sélecteur

* (toutes les balises),

p, h1, h2, etc,

#idCeci : les balises avec un id = « idCeci »

.classeCela : les balises avec un class = « classeCela »

p.classeCela : les balises p avec un class = « classeCela »

[att] : les balises ayant un attribut nommé « att »

[att = petit] : les balises avec un attribut att=« petit »

pseudo classe

Une pseudo-classe est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la déclaration. Il se place après un selecteur simple et sont introduits par « : »

elt : first-child : premier enfant d'une balise

a : link : les a non visités

a : visited : les a visités

elt : hover : les élément survolés

elt : focus : les élément qui ont le focus

Chargez tous les exemples HTML

<http://bliaudet.free.fr/IMG/zip/00-Rappels-HTML.zip>

Exemple 1 : une page de base

Coder une page avec 1 titre, un paragraphe de texte, un sous titre, un texte, une image.

Changer la typo du texte => font-family: Comic ; font-style:italic ; color: blue;

Encadrer le titre => border : solid black 1px;

Centrer l'image => text-align : center ;

Pour mettre du faux texte : chercher « Lorem ipsum dolor » sur google. Un site propose du faux texte.

Pour les couleurs nommées, voir par exemple : <http://stylescss.free.fr/couleurs.php>.

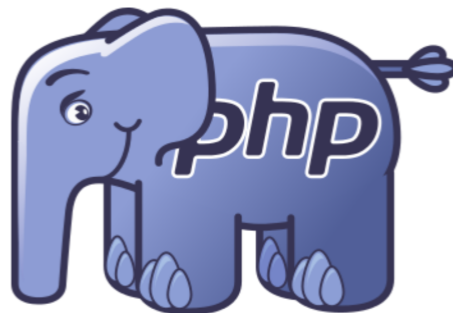
Exemple 1

Titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sous titre

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Rappels théoriques sur le formulaire

➤ *Les balises d'un formulaire*

<form action= « page.php » method=GET> : pour regrouper les input. L'action définit la page appelée quand on valide de formulaire. La méthode est une notion PHP.

<fieldset> : pour mettre un cadre autour des input.

<legend> : pour mettre un texte sur le cadre

<input type = « text » name = « nom »> Type=text pour avoir un champ de saisie. Le « name » permettra d'accéder en PHP à la « value » saisie.

<input type="submit" name="submit" value="Valider"> Type : submit pour avoir un bouton de validation. La value c'est le texte affiché. Quand on clique sur le bouton, la page fait appel à l'action de la balise <form>

<input type="reset" name="reset" value="Annuler"> Type : reset pour annuler la saisie en cours.

<input type="hidden" name="saisie" value="saisie"> Type : hidden faire circuler un name et une value sans que cela concerne l'utilisateur (c'est caché). C'est très utile en programmation PHP.

<select> et **<option>** pour créer un menu déroulant.

<textarea> pour avoir une zone de saisie.

➤ *Autres type de la balise < input>*

En précisant le type de la balise input, on obtient un comportement différent (par exemple, avec un type tel, si on ne saisie pas un numéro de téléphone, le champ de saisie passe en rouge).

type=password : caractères cachés, **email**, **url**, **tel**, **number** (+min, max, step), **range** (+min, max, step avec curseur), **color** (firefox propose une palette colorée), **date**, **search**, **checkbox** (+checked) : boutons à cocher, **radio** : radio bouton (1 seul parmi plusieurs).

➤ *Autres attributs de la balise < input>*

Attribut **maxlength** (input) : taille maximum de la saisie.

Attribut **autofocus** (input) : dans un input place le curseur directement dans le champ.

Attribut **required** (input) : oblige la saisie avant d'envoyer le formulaire (pas dans Safari).

Exemple 2

Le code propose la page de l'exercice 1 avec un lien vers un formulaire. Le formulaire renvoie vers la page de l'exercice 1 quand on le valide.

On a 3 cas possible pour l'âge : <18, 18-25, >25)

Saisie des informations

Nom :

nom

Mot de passe :

motDePasse

Age

inférieur à 18 ans

Observations :

Zone d'observations à saisir

Valider

Annuler

Rappels théoriques sur les tableaux**➤ balises de structures : `<table>` et `<tr>`**

Pas de texte directement dedans.

- `<table>` : Le tableau
- `<tr>` : La ligne

➤ balises de données : `<th>` et `<td>`

Dans chaque ligne, on met les cellules.

- `<th>` : pour le nom de la colonne (la 1^{ère} ligne).
- `<td>` : pour toutes les autres cellules.

➤ Encadrement minimum : `<table border>`

Nombre	Carré	Racine
1	1	1
2	4	1.41421356237
3	9	1.73205080757
4	16	2

- `<table border>` encadre comme suit :

Exemple 3

On peut mettre le formulaire dans un tableau :

On a le même résultat que précédemment, sauf que les zones de saisie sont alignées.

Saisie des informations	
Nom :	<input type="text" value="nom"/>
Mot de passe :	<input type="text" value="motDePasse"/>
Age :	<input type="text" value="inférieur à 18 ans"/>
Observations :	<div>Zone d'observations à saisir</div>
<div>Valider Annuler</div>	

3. Les outils de développement

Outils pour le fonctionnement d'un site WEB

Présentation

Pour faire fonctionner un site WEB, il faut des outils, côté client et côté serveur.

Côté Client : le navigateur

Le **navigateur** qui interagit avec l'utilisateur. **Il comprend le HTML, le CSS et le JavaScript**, mais il ne comprend pas le PHP.

Côté Serveur : le serveur web et le serveur de BD¹

Le **serveur web** (ou serveur http) interagit avec le navigateur et le serveur de BD. Le serveur web traduit le code PHP en HTML. **Il comprend donc le PHP.**

Le **serveur de BD** interagit avec le serveur web et un disque dur pour sauvegarder et récupérer des données. **Il comprend le SQL**

¹ Pour la différence entre serveur web et serveur d'application :
<https://www.supinfo.com/articles/single/1156-difference-serveur-web>. Le pdf correspondant si le lien ne marche pas :
<http://biliaudet.free.fr/IMG/pdf/Difference-entre-serveur-web-et-serveur-d-application.pdf>

Présentation

Pour développer un site WEB, il faut des outils qui permettent d'écrire le code et de faire de sa machine un serveur pour le navigateur.

Côté Client – Partie statique

- HTML, CSS : **Editeur de texte (VSCode, SublimeText, brackets, etc.)**
- Affichage de la page : **Navigateur**
- Hébergement du site : **logiciel FTP**

Côté Serveur - Partie dynamique

- PHP : **Editeur de texte**
- SQL : **Editeur de texte**
- Serveur HTTP : **Apache**
- Serveur SGBD : **MySQL**

Solutions

Il existe des solutions tout en un qui fournissent les serveurs HTTP et SGBD ainsi qu'une interface utilisateur pour utiliser ces serveurs et pour tester notre code HTML et PHP sur notre machine de développement qui joue le rôle du serveur pour elle-même en tant que client.

Par exemple : WAMP (Windows, Apache, MySQL, PHP), MAMP sur Mac, LAMPP sous Linux (un P en plus pour Perl), XAMPP. EasyPHP, etc.

Aucune solution n'est parfaite ni radicalement supérieure à une autre. C'est une question d'habitude.

On choisit WAMP. On peut aussi utiliser XAMPP. Ou MAMP si on est sur MAC.

4. Installation de WAMP

http://bliaudet.free.fr/article.php3?id_article=307

Aller dans l'option 3 : WAMP

Vous allez télécharger un fichier d'installation : wampserver3.1.4_x64.exe (selon la version).

Pendant l'installation : choisir le navigateur par défaut (Chrome ou Firefox de préférence, surtout pas internet explorer), choisir un éditeur par défaut (VSCode de préférence ou Sublime Text).

Autoriser le déblocage du pare-feu.

Détails sur l'installation de WAMP : dans l'article :
http://bliaudet.free.fr/article.php3?id_article=277, le poly :

http://bliaudet.free.fr/IMG/pdf/01_Installation_de_WAMP.pdf

5. Premiers usages de WAMP

1 : Démarrer WAMP

Il faut autoriser le pare-feu pour que les 2 serveurs puissent démarrer.

Double clic sur le raccourci sur le bureau : **Ca marche ! On obtient une icône verte.**

Si on a une icône rouge, c'est qu'aucun serveur n'est démarré.

Si on a une icône orange, c'est qu'un seul serveur est démarré.

On peut démarrer, arrêter, redémarrer les services (services = serveurs).

2 : Installer l'icône WAMP directement dans la barre d'icône

Déplacer l'icône pour qu'elle soit directement dans la barre des icônes :



3 : Passer Wamp en français

icône Wamp – bouton gauche - choix langage

4 : Consulter les serveurs dans le gestionnaire des tâches

- On peut regarder les processus qui tournent sur Windows : CTRL-ALT-SUP ou CTRL-SHIFT-ESC.
- Il faut afficher les détails.
- Apache HTTP Server : c'est le serveur web. Il peut y en avoir deux, mais c'est le même !
- mysqld : c'est le serveur de la BD MySQL. Il peut y en avoir deux, mais c'est le même ! Il peut même y en avoir 3 : un de plus pour MariaDB !
- Dans tous les cas, si on arrête les serveurs, il ne doit en rester aucun. Si il en reste il faut les arrêter !

5 : Tester WAMP : afficher la page WampServer

Dans le navigateur : localhost

Ou bien icône Wamp – bouton droit - choix localhost (icône Wamp : icône verte dans la zone de notification : en bas à droite).

La page WampServer apparaît.

6 : Allez dans le dossier des projets : C:\wamp64\www, créez un dossier « home »

icône Wamp – bouton droit - choix répertoire www

Mettez un dossier : « home » dans le répertoire www

7 : Afficher le contenu de « home » dans la page WampServer

Icône Wamp – bouton droit - choix localhost

Le dossier home apparaît dans la zone « Vos Projets » : il n'est pas cliquable.

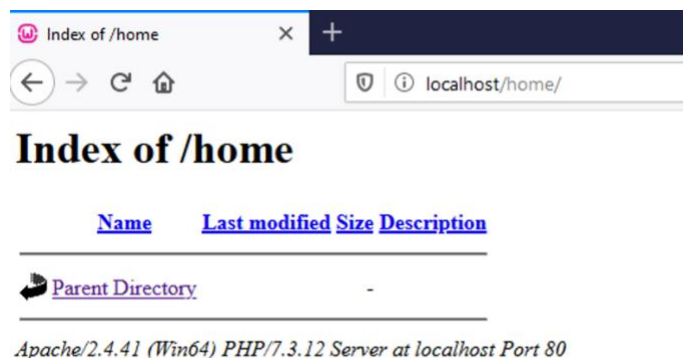
8 : Accéder au dossier « home » dans la page WampServer

Icône Wamp – bouton droit - choix localhost

Dans l'url, on trouve : localhost

Ajoutez /home : l'url est donc : localhost/home

On arrive sur la page web du répertoire home :



6. Premiers programmes en PHP

9 : Premier programme php

➤ *Code*

Dans le dossier « home », ajouter un fichier test_1.php avec le code suivant

```
<?php  
    echo «<h1>Hello World </h1> » ;  
?>
```

➤ *Test*

Ce fichier apparaît dans la page WampServer, dans la page de l'url <http://localhost/home/>

En cliquant dessus, on apparaît la page HTML correspondant au code.

On peut visualiser le code HTML avec un MAJ-CTRL-i (pour Firefox).

On voit uniquement le code HTML.

10 : Fichier index.php

Si un répertoire contient un fichier index.php ou index.html, il est automatiquement exécuté quand on rentre dans le dossier.

Dupliquez test_1.php en le nommant index.php

Ajoutez :

```
echo «<h1>Hello World avec index </h1> » ;
```

Vérifiez cette propriété.

Ensuite, renommer index.php en index0.php.

Vérifiez qu'il n'y a plus d'exécution automatique.

11 : Deuxième programme php

➤ *Code*

Dans le dossier « home », ajouter un fichier test_2.php avec le code suivant

```
<?php
// en utilisant l'url :
// localhost/home/helloWorld.php?name=Bertrand
```

```
$name = $_GET['name'];
// le header précise le type de la page
header('Content-Type: text/html');
echo '<html>';
echo '<body>Bonjour '. $name . '</body>';
echo '</html>'
?>
```

➤ *Test*

Ce fichier apparaît dans la page WampServer, dans la page de l'url <http://localhost/home/>

En cliquant dessus, on apparaît la page HTML correspondant au code : mais il y a un bug !

Il faut ajouter dans l'url : ?name=Bertrand pour que ça marche :

http://localhost/home/test_2.php?name=Bertrand

On peut visualiser le code HTML avec un MAJ-CTRL-i (pour Firefox).

On voit uniquement le code HTML.

12 : Consulter le contenu des dossiers d'un site en ligne !

Exemple : <http://www.amberieunatation.fr/spip/IMG/jpg/>

Sur ce site, on constate qu'on peut accéder au contenu du dossier !

On peut remonter le dossier : <http://www.amberieunatation.fr/spip/IMG>

Jusqu'à la racine du site : <http://www.amberieunatation.fr>

La racine contient un fichier index.php : c'est la page d'accueil du site qui s'affiche.

Pour éviter le problème, on peut par exemple mettre un fichier index.php dans tous les répertoires qui renvoie par exemple sur la page d'accueil (fonction header en PHP).

WAMP - version détaillée (on peut passer !)

0 : Reprise détaillée du paragraphe précédent : premiers usages - résumé

Ce chapitre détaille ce qu'on a fait au paragraphe précédent

1 : Tester WAMP

Démarrer WAMP

Il faut autoriser le pare-feu pour que les 2 serveurs puissent démarrer.

Double clic sur le raccourci sur le bureau : **Ca marche ! On obtient une icône verte.**

Passer Wamp en français


icône Wamp – bouton gauche - choix langage

Tester WAMP : afficher la page WampServer

Dans le navigateur : localhost

Ou bien icône Wamp – bouton droit - choix localhost (icône Wamp : icône verte dans la zone de notification : en bas à droite).

La page WampServer apparaît.


WampServer

Version 3.1.0 - 64bit
french
classic

Configuration Serveur

Version Apache : 2.4.27 - [Documentation](#)
Version de PHP : 5.6.31 - [Documentation](#)
Server Software : Apache/2.4.27 (Win64) PHP/5.6.31 - Port défini pour Apache : 80

Extensions Chargées :

✱ apache2handler	✱ bcmath	✱ bz2	✱ calendar	✱ com_dotnet
✱ Core	✱ ctype	✱ curl	✱ date	✱ dom
✱ ereg	✱ exif	✱ fileinfo	✱ filter	✱ ftp
✱ gd	✱ gettext	✱ gmp	✱ hash	✱ iconv
✱ imap	✱ intl	✱ json	✱ ldap	✱ libxml
✱ mbstring	✱ mcrypt	✱ mhash	✱ mysql	✱ mysqli
✱ mysqlnd	✱ odbc	✱ openssl	✱ pcrc	✱ PDO
✱ pdo_mysql	✱ pdo_sqlite	✱ Phar	✱ Reflection	✱ session
✱ SimpleXML	✱ soap	✱ sockets	✱ SPL	✱ sqlite3
✱ standard	✱ tokenizer	✱ wddx	✱ xdebug	✱ xml
✱ xmlreader	✱ xmlrpc	✱ xmlwriter	✱ xsl	✱ Zend OPcache
✱ zip	✱ zlib			

Version de MySQL : 5.7.19 - Port défini pour MySQL : 3306 - [Documentation](#)
Version de MariaDB : 10.2.8 - Port défini pour MariaDB : 3307 - [Documentation](#)

Outils

- [phpinfo\(\)](#)
- [phpmyadmin](#)
- [Ajouter un Virtual Host](#)

Vos Projets

Aucun projet.

Pour en ajouter un nouveau, créez simplement un répertoire dans 'www'.

Vos Alias

- [adminer](#)
- [phpmyadmin](#)
- [phpsysinfo](#)

Vos VirtualHost

- [localhost](#)

La zone « Vos Projets » est vide.

2 : Travailler avec des fichiers PHP

Allez dans le dossier des projets : C:\wamp64\www

Aller dans C:\wamp64\www.

C'est dans ce dossier qu'on mettra tous les dossiers de travail.

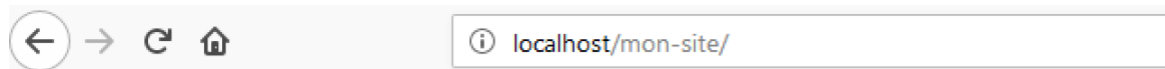
Créer un dossier « mon-site », par exemple.

La zone « Vos Projets » contient le nom du dossier qu'on vient de créer.

The screenshot shows the WampServer control panel. At the top, there's a logo and the text 'WampServer'. Below it, the version is '3.1.0 - 64bit' with language and theme dropdowns set to 'french' and 'classic'. The main section is 'Configuration Serveur', which includes details about Apache (2.4.27), PHP (5.6.31), and MySQL (5.7.19). It lists various loaded extensions like apache2handler, Core, ereg, gd, imap, mbstring, mysqlnd, pdo_mysql, SimpleXML, standard, xmlreader, zip, bcmath, ctype, exif, gettext, intl, mcrypt, odbc, pdo_sqlite, soap, tokenizer, xmlrpc, zlib, bz2, curl, fileinfo, gmp, json, mhash, openssl, Phar, sockets, wddx, xmlwriter, calendar, date, filter, hash, ldap, mysql, pcre, Reflection, SPL, xdebug, xsl, com_dotnet, dom, ftp, iconv, libxml, mysqli, PDO, session, sqlite3, xml, and Zend OPcache. At the bottom, there are four tabs: 'Outils' (with links to phpinfo(), phpmyadmin, and a button to add a virtual host), 'Vos Projets' (showing a folder icon and the name 'mon-site'), 'Vos Alias' (showing links to adminer, phpmyadmin, and phpsysinfo), and 'Vos VirtualHost' (showing a link to localhost).

Accéder au projet : localhost/mon-site

En entrant localhost/mon-site dans un navigateur, on accède au projet :



Index of /mon-site

Name	Last modified	Size	Description
 Parent Directory		-	

Apache/2.4.27 (Win64) PHP/5.6.31 Server at localhost Port 80

Accès direct aux projets

Si on clique directement sur le nom du projet, on obtient une erreur et on n'accède pas au projet.

➤ **Solution 1 :**

Pour remédier à cela : sur l'icône verte Wamp : bouton droit / paramètres Wamp / ajouter localhost dans l'URL.

Ca ajoute un warning dans la zone « Vos Projets ».

Mais on peut désormais accéder aux projet en cliquant sur leur nom.

The screenshot shows the WampServer control panel. At the top, there's a WampServer logo and version information (Version 3.1.0 - 64bit) with language (french) and theme (classic) dropdowns. The main section is titled "Configuration Serveur" and lists various components and their versions: Apache (2.4.27), PHP (5.6.31), MySQL (5.7.19), and MariaDB (10.2.8). Below this, there's a list of "Extensions Chargées" (Loaded Extensions) including apache2handler, Core, ereg, gd, imap, mbstring, mysqlnd, pdo_mysql, SimpleXML, standard, xmlreader, zip, bcmath, ctype, exif, gettext, intl, mcrypt, odbc, pdo_sqlite, soap, tokenizer, xmlrpc, zlib, bz2, curl, fileinfo, gmp, json, mhash, openssl, Phar, sockets, wddx, xmlwriter, calendar, date, filter, hash, ldap, mysql, pcre, Reflection, SPL, xdebug, xsl, com_dotnet, dom, ftp, iconv, libxml, mysqli, PDO, session, sqlite3, xml, and Zend OPcache. At the bottom, there are four sections: "Outils" (Tools) with links to phpinfo(), phpmyadmin, and "Ajouter un Virtual Host"; "Vos Projets" (Your Projects) showing "mon-site" with a warning "Warning: See below"; "Vos Alias" (Your Aliases) with links to adminer, phpmyadmin, and phpsysinfo; and "Vos VirtualHost" (Your Virtual Hosts) showing "localhost". A note at the bottom states: "C'est une mauvaise idée d'ajouter localhost dans les url de lancement des projets. Il est préférable de définir des VirtualHost dans le fichier wamp/bin/apache/apache2.4.27/conf/extra/httpd-vhosts.conf et de ne pas ajouter localhost dans les url."

➤ **Solution 2 :**

Dans le dossier vos projets (C:/wampp64/www) ouvrez le fichier index.php.

Vers la ligne 33, après la ligne :

```
$suppress_localhost = ($wampConf['urlAddLocalhost'] == 'off' ? true : false);
```

Ajoutez :

```
$suppress_localhost= false;
```

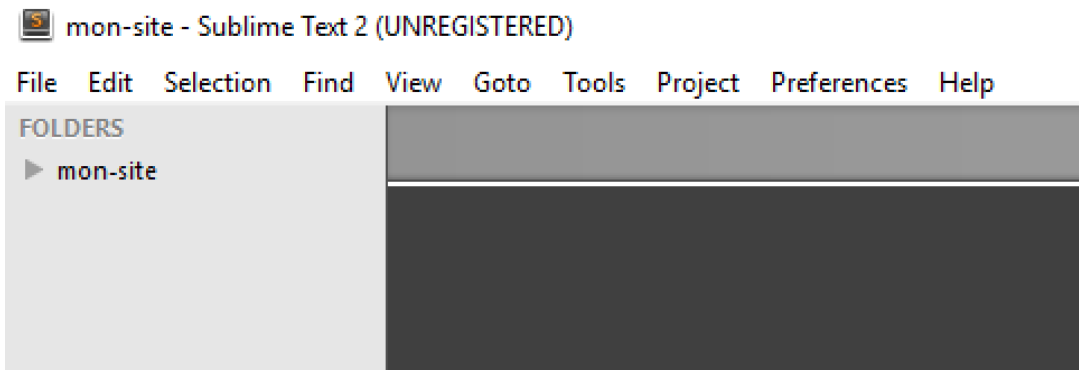
Cette solution donne les mêmes résultats que la précédente, en plus compliqué !

➤ ***Solution 3 : « à la main »***

Dans le navigateur : localhost/mon-site

Création d'un premier fichier PHP

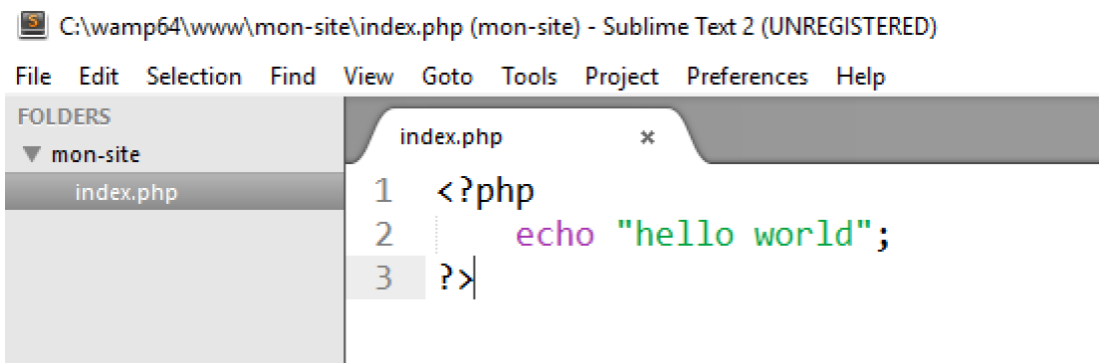
Ouvrir le dossier « mon-site » avec VSCode ou SublimeText :



Sur le nom du dossier, ouvrir un fichier : new file.

Puis save : on l'appelle index.php, il se trouve dans le répertoire mon-site.

Dans le fichier index.php :



Exécuter le fichier PHP

Il suffit de cliquer sur le dossier « mon-site ». Par défaut, le fichier « index.php » est envoyé au serveur apache qui le retourne au navigateur.



3 : Paramétrages divers

Bouton droit sur l'icône verte Wamp

➤ *Langue*

Bouton droit / language/ français

➤ *Paramètres de Wamp*

« Ajouter localhost dans l'URL » : pour l'accès direct au dossier de travail, par exemple.

➤ *Outils*

Pour tester les ports, par exemple

Clic sur l'icône verte Wamp

- Démarrer ou arrêter les serveurs (services) : apache et mysql
- Accès direct à la page localhost
- Accéder à phpmyadmin, l'utilitaire pour gérer la BD.
- Paramétrer Apache, MySQL, PHP (AMP).

4 : PHPMyAdmin

On accède à PHPMyAdmin ainsi :

<http://localhost/phpmyadmin/>

ou

icône Wamp / PHPMyAdmin

5 : Paramétrage des installations : `display_errors` et `display_startup_errors`

Fichier php.ini

Il existe un fichier php.ini qui paramètre l'usage de WAMP pour le php.

On le trouve dans clic / PHP

Sous MAMP, il faut trouver le bon ! Quand on va dans phpInfo, on trouve le configuration file :

/Applications/MAMP/bin/php/php7.0.0/conf

C'est ce fichier qu'il faut modifier.

Variables `display_errors` = On et `display_startup_errors` = On

Les variables `display_errors` et `display_startup_errors` permettent d'afficher des messages d'erreur plus explicites.

Elles doivent être à ON pour que les messages d'erreur s'affichent.

Il faut redémarrer le serveur Apache pour que la prise en compte soit faite.
On peut les voir dans le phpInfo de MAMP.

Alternative pour un site en production

Attention, ces messages d'erreurs ne doivent pas être vus par les utilisateurs.

Pour un site déjà en production, il faut passer par les fichiers de log.

Regarder dans le fichier de conf pour plus d'informations.

MAMP – version détaillée

1 : Principes

On suit la même logique que pour tester WAMP.

Il faut donc suivre le chapitre précédent pour faire la partie pratique et revenir dans ce chapitre pour voir les spécificités de MAMP.

2 : Lancer MAMP

- Il faut autoriser le parefeu pour que les 2 serveurs puissent démarrer.
- Pour démarrer MAMP : Application/MAMP/MAMP.app
Le programme propose d'utiliser MAMP PRO : on évite !
- Préférences : pour gérer les ports des serveurs et le démarrage automatique des serveurs
- Démarrer / Arrêter les serveurs

3 : Problème d'installation – problème de port

Onglet préférence, menu ports :

Apache est sur 8080 par défaut

4 : Accès aux projets web

localhost :8080

Dans un navigateur, taper : « localhost :8080 ».

<http://localhost:8080/>

Une page index of / apparaît. Elle est vide, elle contiendra nos projets.

Cette page correspond au répertoire : /Applications/mamp/htdocs

Ce répertoire est paramétrable dans Préférences /serveur web

Ouvrir la page web start /mon site web

On accède à la page localhost :8080 ainsi

Ouvrir la page web start / mon site web

5 : PHPMyAdmin

On accède à PHPMyAdmin ainsi :

<http://localhost:8080/phpMyAdmin>

ou

6 : Fichier php.ini

Il existe un fichier php.ini qui paramètre l'usage de WAMP pour le php.

On le trouve dans clic / PHP

Sous MAMP, il faut trouver le bon ! Quand on va dans phpInfo, on trouve la configuration file :

/Applications/MAMP/bin/php/php7.0.0/conf

C'est ce fichier qu'il faut modifier.

7. Trois exemples de l'instruction echo (ou print)

Chargez et installez les exemples

Commencez par charger le dossier zippé : 02-premiers-programmes-echo.zip et installez-le dans le répertoire du serveur WAMP (www).

Exemple 1 : code HTML minimum

Objectifs

Simple affichage de texte sans php.

Comparaison entre un fichier .html et .php

Codes dans php

➤ *Fichier Exemple_01_bonjourHTML.html*

```
bonjour sans balise dans un fichier .php  
<h3>bonjour avec balise dans un fichier .php</h3>
```

Résultats :

```
bonjour sans balise dans un fichier .php  
bonjour avec balise dans un fichier .php
```

➤ *Fichier Exemple_01_bonjourHTML.php*

```
bonjour sans balise dans un fichier .php  
<h3>bonjour avec balise dans un fichier .php</h3>
```

Résultats :

```
bonjour sans balise dans un fichier .php  
bonjour avec balise dans un fichier .php
```

Explications

Le résultat est le même dans les deux cas.

Le fichier .html est fourni directement au navigateur. Le texte s'affiche.

Le fichier .php est traité par le serveur Apache. Il ne contient pas de balise php : le serveur web n'effectue aucune transformation. Le résultat est transmis au navigateur. Le texte s'affiche.

On évite d'utiliser des fichiers html dans programmes côté serveur.

Exemple 2 : code HTML minimum avec un echo php

Objectifs

- Simple affichage de texte avec du php
- Comparaison entre un fichier .html et .php

Fichier php :

```
<?php echo "bonjour avec balise php"; ?>  
<?php echo "<h3>bonjour avec balises HTML et php</h3>"; ?>
```

Résultats dans le navigateur :

```
bonjour sans balise php  
bonjour avec balises HTML et php
```

Page HTML générée :

```
bonjour sans balise php  
<h3>bonjour avec balises HTML et php</h3>
```

Explications

- Le fichier .php est traité par le serveur Apache.
- La ligne de PHP se trouve entre : < ?php et ?>
- Echo est une instruction. Chaque instruction sera terminée par un ;
- Le serveur Apache traite les instructions situées entre < ?php et ?>. echo est une instruction qui permet d'écrire ce qui se trouve entre les guillemets.
- Le résultat est un fichier HTML qui est transmis au navigateur. Le texte s'affiche.

Remarques

Faire un echo en PHP plutôt que d'afficher directement le texte en HTML, ça ne sert à rien.

Mais le PHP est un langage de programmation qui permettra de récupérer les informations des formulaires HTML, de faire des calculs complexes, d'enregistrer des informations dans la BD ou dans un fichier, de récupérer des informations dans la BD ou dans un fichier, de remplir des tableaux avec des données de la BD, etc.

Exemple 3 : page HTML standard avec un echo php

Objectifs

Page HTML classique avec un simple affichage de texte php.

Comparaison entre un fichier .html et .php

Fichier php :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Exemple 1: echo</title>
  </head>

  <body>
    <?php echo "<h1>Affichage de texte HTML et PHP</h1>" ?>
    <?php
      echo "<p> PHP :";
      echo "<span style=\"color: red;\">
        Ligne écrite avec un \"echo\" <strong>PHP</strong>
      </span></p>";
    ?>
  </body>
</html>
```

Résultats dans le navigateur :

Affichage de texte HTML et PHP

PHP : Ligne écrite avec un "echo" PHP.

Page HTML générée :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Exemple 1: echo</title>
  </head>

  <body>
    <h1>Affichage de texte HTML et PHP</h1>
    <p> PHP :
      <span style="color: red;">
        Ligne écrite avec un "echo" <strong>PHP</strong>
      </span>
    </p>
  </body>
</html>
```

Explications

Le h1 est mis dans un echo et dans un < ?php séparé (ça ne sert à rien, c'est pour le test !).

Le < ?php suivant fait :

un echo qui ouvre un <p>

puis un autre qui ouvre un avec du style

et un strong qui ferme le mais aussi le</p> ouvert dans le précédent echo.

Pour mettre un guillemet dans le texte d'un echo (donc pour afficher un guillemet dans le code HTML), on met \"

Exercice echo

Coder toute la page précédente avec des échos. Combien faut-il en mettre au minimum ?

Principes généraux du code PHP

Règles

- Le code PHP commence par `< ?php` (sans espace) et finit par `?>`. On va parler de « balise PHP ».
- On peut placer une balise PHP n'importe où dans le code HTML (pas dans du code CSS !)
- Dans une balise PHP on peut mettre autant d'instructions et de lignes PHP qu'on veut.
- On doit mettre un `;` à la fin de chaque instruction PHP.
- On peut mettre des commentaires entre `/*` et `*/` dans le code PHP. Ces commentaires peuvent s'étaler sur plusieurs lignes. Ces commentaires disparaîtront dans le code HTML.
- On peut mettre des commentaires de fin de ligne avec `//`

Erreurs courantes

- Parse error : quand on oublie le `;` à la fin d'une instruction (le `;` est facultatif sur la dernière instruction)
- Un fichier `.html` à la place d'un fichier `.php`

8. Gestion des erreurs

Affichage des erreurs - Rappel

Variables display_errors = On et display_startup_errors = On

Les variables display_errors et display_startup_errors permettent d'afficher des messages d'erreur plus explicites.

Par défaut, les erreurs s'affichent sur PC.

Pour voir ces variables, ouvrez le fichier php.ini : WAMP / PHP / php.ini

Cherchez les variables display_errors et display_startup_errors

Vous trouvez les lignes : (sans ; au début, sinon c'est un commentaire)

display_startup_errors = On

display_errors = On

Il faut redémarrer le serveur Apache pour que la prise en compte soit faite.

Les erreurs s'affichent

On peut voir les variables dans le phpInfo de Mamp.

Alternative pour un site en production

Attention, ces messages d'erreurs ne doivent pas être vus par les utilisateurs.

Pour un site déjà en production, il faut passer par les fichiers de log.

Regarder dans le fichier de conf pour plus d'informations.

Principaux messages d'erreurs

Parse error

Est indiqué le fichier, la ligne, souvent la cause.

L'erreur peut être sur la ligne ou sur la ligne précédente.

Ça peut correspondre à :

- Oubli de ;
- Apostrophe ou guillemet non fermé
- Concaténation à problème

Undefined function

La fonction n'est pas reconnue quand on s'en sert : il faut probablement un include.

Wrong parameter count

Une fonction n'a pas le bon nombre de paramètres.

Erreur autour de la BD

Les erreurs SQL sont plus ou moins explicites. Il faut vérifier que le code SQL fonctionne.

headers already sent by...

Si on a du code HTML avant un header ou une session

Maximum execution time exceeded

Boucle sans fin

TP : Gestion des erreurs - variables `display_errors` et `display_startup_errors` à On

Chargez le zip « 04-exercice-gestion-erreur.zip »

Essayez le programme `TP_gestion_erreur.php` et corrigez les erreurs.

Affichage des erreurs sur MAC

Par défaut, les erreurs ne s'affichent pas sur MAC.

Il faut modifier les variables `display_errors` et `display_startup_errors` dans le fichier `php.ini`

Pour cela : Démarrez MAMP : Ouvrir la page web start / `phpinfo`

Cherchez `php.ini`

Vous trouvez pour le Configuration File (`php.ini`) Path la valeur :
`/Applications/MAMP/bin/php/php7.0.0/conf`

Allez dans ce répertoire et ouvrez le fichier `php.ini`

Passez les deux variables à On :

`display_startup_errors = On`

`display_errors = On`

Relancez le serveur : MAMP / Arrêter les serveurs – MAMP / Démarrer les serveurs

Affichage des erreurs sur PC

Par défaut, les erreurs s'affichent sur PC.

Pour voir ces variables, ouvrez le fichier `php.ini` : WAMP / PHP / `php.ini`

Cherchez les variables `display_errors` et `display_startup_errors`

Vous trouvez les lignes : (sans ; au début, sinon c'est un commentaire)

`display_startup_errors = On`

`display_errors = On`

Les erreurs s'affichent

9. Inclusion de fichiers PHP – factorisation du code HTML

Rappel sur les balises structurantes HTML

HTML 5 propose des balises de structuration des pages :

- header, nav, footer comme balise plutôt unique dans la page qui correspondent à des éléments qu'on retrouve sur toutes les pages du site
- section, article, aside comme balises éventuellement multiples dans la page et différentes selon les pages
- div comme balise de structuration personnelle.

Le problème

header, nav et footer vont en général se retrouver sur toutes les pages.

Comment éviter de répéter le même code sur toutes les pages et donc de devoir faire des mises à jour sur toutes les pages ?

Solution PHP : l'inclusion de fichier en PHP

Tout le code de la balise header peut être mis dans un fichier header.php

Ensuite, l'instruction « include » permet de faire appel à ce fichier et de l'inclure dans le fichier appelant.

include (« fichier.php »)

Avec l'instruction suivante :

```
< ?php include (« header.php ») ; ?>
```

le code de la page est limité à ce qu'elle contient de spécifique. Si on modifie le header, on le modifie pour toutes les pages.

On peut faire la même chose avec le footer et le nav. Si le nav est dans le header, on peut mettre l'inclusion dans le fichier header.php

Syntaxe

include est une fonction avec des parenthèses à laquelle on fournit le nom du fichier qu'on veut charger, entre guillemets, à la différence de echo qui ne prend pas de parenthèses.

Exemple :

body avec include du header et du footer

```
<body>
  <?php include("header.php");?>
  <section>
  </section>
  <?php include("footer.php"); ?>
</body>
```

header qui inclut le nav

```
<header>
  <h1>Mon SUPER SITE</h1>
  <?php include("nav.php"); ?>
</header>
```

L'ancienne solution : inclusion de fichier en HTML : balise frame (obsolète)

Les sites qui n'utilisent pas de langage serveur utilisait la balise <frame> pour factoriser les pages.

Cette balise n'est plus supportée par le HTML 5.

Elle est remplacée par la balise <iframe>

http://www.w3schools.com/tags/tag_frame.asp

La balise frame ne se ferme pas.

Elle s'utilise à l'intérieur d'une balise <frameset>

La balise <iframe> : inclusion de fichier en HTML (déprécié)

Cette balise sert surtout à inclure des morceaux qu'un autre site permet de partager.

A cette adresse, le site fournit un iframe dans le bouton exporter :

<https://www.franceculture.fr/emissions/les-series-musicales-dete/les-series-musicales-dete-lundi-24-juillet-2017>

Cet usage est toutefois déprécié.

```
<!DOCTYPE html>
<html>
<body>

<iframe src="https://www.franceculture.fr/player/export-
reecouter?content=9c19db61-0761-4b50-b4ae-907f0dbf3259"
width="481" frameborder="0" scrolling="no" height="137">
    <p>Your browser does not support iframes.</p>

</iframe>

</body>
</html>
```

Widget

Les widgets tendent à remplacer les iframe. Ce sont de simples div à copier dans son site.

Un widget de météo ici :

<http://www.meteovista.fr/Widgets-Meteo/4266446/0>

TP : Includes

Chargez le zip <05-exercice-include.zip >

Il contient un dossier avec une page HTML.

1)

Réécrivez-la en sortant le header et le footer dans des fichiers séparés et en utilisant des include.

Testez le résultat.

2)

Ensuite, mettez les fichiers header et footer dans un dossier « include » et le fichier css dans in dossier « css ». Le but est de n'avoir que le fichier index1.php à la racine.

3)

Ajoutez une page qui contient l'en-tête et le footer et une section qui contient uniquement un lien permettant de revenir à l'index de la page principale (index1.php).

Dans index1.php, un des menus permettra d'accéder à cette nouvelle page qu'on appelle index2.php.

10. Documentation

Principes

Les références

Il existe plusieurs sites de référence pour avoir de la documentation :

php.net : <http://php.net/docs.php>

w3school : <http://www.w3schools.com/php/>

Et des cours :

Code academy : <https://www.codecademy.com/learn/php>

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

Et bien sûr : google !

Les bons usages

Il faut toujours essayer de passer par une documentation « officielle » pour comprendre le mieux possible sans se contenter d'un exemple qui marche.

Il faut donc apprendre à se servir de la documentation

Exemple d'usage : php.net

Page d'accueil en français

<http://php.net/manual/fr/> : on y trouve le sommaire

Liste des fonctions

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

Notez :

Extensions sur les mathématiques : contient Math : fonctions mathématiques

Traitement de texte : contient Chaîne de caractères : fonction de traitement de chaînes.

Accès direct à une fonction : php.net/nomFonction

php.net/date : j'arrive directement sur la fonction date :

<http://php.net/manual/fr/function.date.php>

Si je remonte le menu de droite, je reviendrai au niveau de la liste des fonctions.

Liste des thématiques de fonctions

- audio

- identification
- ligne de commande
- archivage et compression
- cartes de crédit
- cryptographie
- bases de données
- dates et aux heures
- systèmes de fichiers
- langage humain et encodage de caractères
- génération et traitement des images
- emails
- mathématiques
- données non-textuelles
- contrôle des processus
- moteurs de recherche
- serveurs
- sessions
- traitement du texte
- variables et aux types
- services Web
- Windows
- XML