

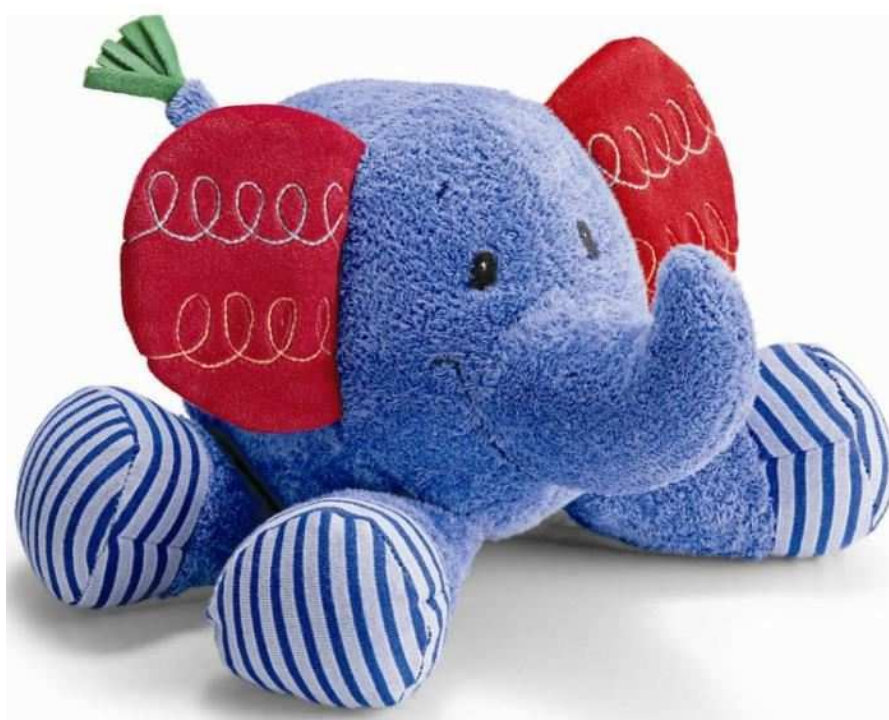
## Niveau 2

TP  
S4.7 - Langages de programmation  
Web dynamique : PHP, JavaScript

# Initiation au PHP

### TABLE DES MATIÈRES

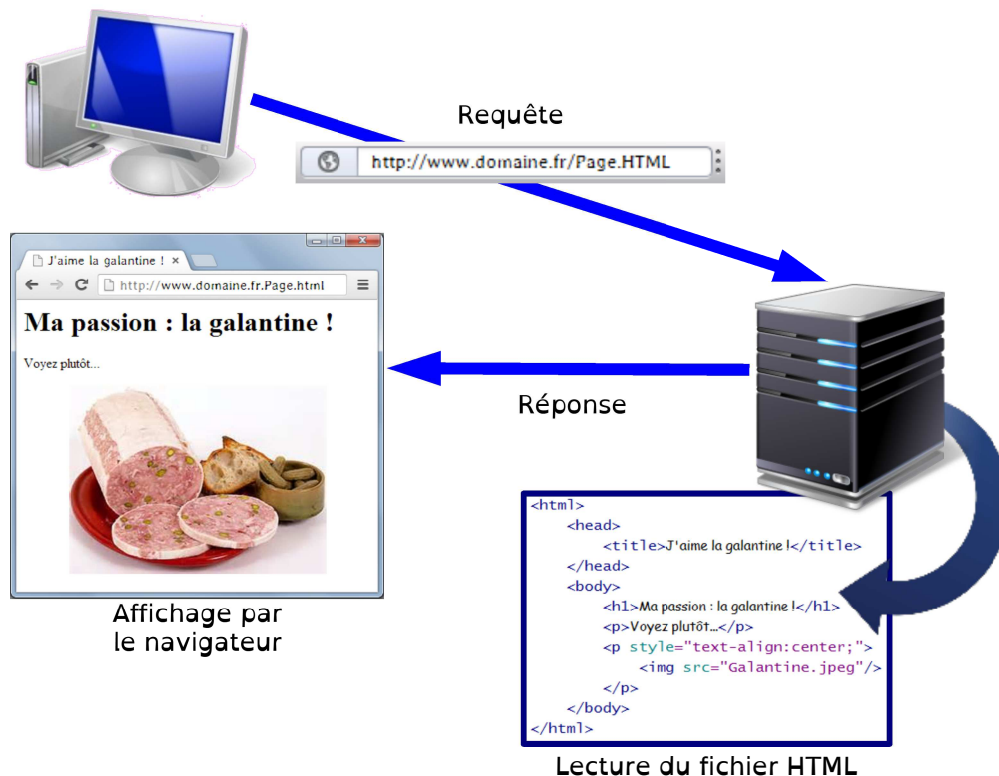
1 - Présentation.....	2
2 - Installation du serveur WAMP.....	3
3 - Travail à réaliser.....	5
3.1 - Je suis bilingue ! (au moins !).....	5
3.1.1 - Mon site s'adapte automatiquement au navigateur.....	5
3.1.2 - Le visiteur peut choisir la langue par boutons.....	6



# 1 - PRÉSENTATION

Le but de ce TP est la mise en œuvre du langage PHP, qui est dédié à la programmation de pages internet dynamiques avec, ou non interrogation de bases de données.

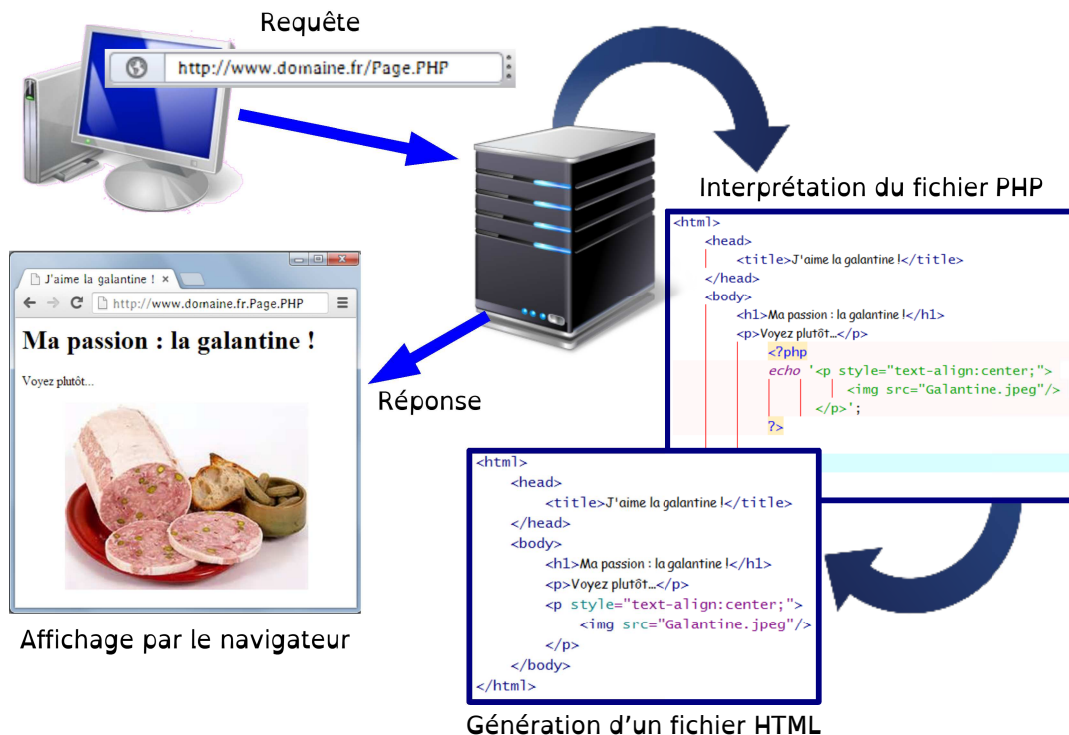
Lorsqu'un utilisateur demande d'afficher une [Page.HTML](#), son navigateur effectue une requête à un des serveurs du domaine.fr. Ce dernier retrouve le fichier correspondant (ainsi que les diverses images...) et le renvoie au navigateur, qui peut alors l'interpréter et l'afficher.



Le PHP (**Hypertext Preprocessor**) est un langage de scripts. Ces scripts sont intégrés dans des pages HTML grâce à un attribut de balise spécifique. Lors de la consultation d'une [Page.PHP](#), le navigateur effectue une requête au serveur qui héberge la page désirée. Ce dernier analyse la page demandée grâce à son préprocesseur PHP. Le script est exécuté par l'interpréteur associé. Le résultat est alors inséré (ou retourné) sous la forme d'un code HTML que le navigateur interprète à son tour pour l'afficher.

L'exécution d'un code PHP peut être automatique, lorsqu'il s'agit d'une page d'extension [.PHP](#), ou manuelle, par le clic d'un bouton de formulaire par exemple.

Dans tous les cas, le code du script PHP n'est **jamais** visible sur le navigateur : seul le code HTML résultant de son interprétation par le serveur peut être affiché.



L'aide-mémoire PHP fourni vous donnera les notions de bases d'écriture de scripts et la syntaxe des commandes essentielles.

Pour plus d'informations : <http://php.net/manual/fr/>

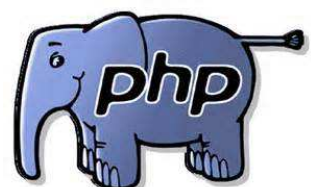
## 2 - INSTALLATION DU SERVEUR WAMP

Pour le besoin de ce TP, nous allons utiliser un système portable de serveurs intégrés.

Pour cela, **WAMP Server** va parfaitement convenir : c'est un package fonctionnant sous **Windows**, qui intègre un serveur **Apache** (serveur Web HTTP), **MySQL** (serveur de Base de Données, SGBD) et **PHP** (langage de scripts adapté au développement Web, notamment pour réaliser des pages dynamiquement).



**Apache**



Contrairement à un serveur de production, un système **WAMP** est destiné au développement et à la mise au point de sites, de bases de données et/ou de scripts PHP, et donc n'est pas prévu pour tenir une charge importante de requêtes extérieures.

- 1) Après avoir récupéré le fichier d'installation de **WAMP**, vous en ferez l'installation, sur votre ordinateur. Il vous est demandé de rédiger ici, un manuel d'installation précisant toutes les étapes nécessaires ainsi que les diverses valeurs à fournir et illustré de captures des écrans rencontrés. Ce manuel sera à fournir finalement au format **PDF**.
- 2) Lancez **WAMP** et validez l'installation. Si vous rencontrez des problèmes lors de ce premier lancement, ajoutez une rubrique au manuel d'installation « Problèmes possibles lors du premier lancement » que vous complétez avec un listing de ces problèmes et la solution à apporter à chacun d'entre eux.
- 3) Validez le fonctionnement du serveur **HTTP**, **Apache**, à partir d'un navigateur en ouvrant la page de configuration située en <http://localhost/>. Normalement vous devriez obtenir un écran de ce style :



Votre serveur HTTP local est donc fonctionnel !

**Bravo !**

Vous pouvez dès à présent ajouter à votre site personnel le CV numérique que vous avez réalisé lors d'un précédent TP. Pour cela, vous disposez d'un directory spécifique dans le répertoire de **WAMP** : **WAMP\www** !

Toutes les pages que vous mettrez dans ce dossier seront accessibles à partir de l'adresse locale de l'ordinateur : **127.0.0.1**. Par exemple **127.0.0.1/NOM\_Prenom\_Site/HTML/Page\_Personnelle.html**.

Pour qu'une page soit la page d'accueil de votre site, et donc soit accessible directement sans avoir à ajouter le nom de la page dans l'**URL**, elle doit porter le nom **index.html**. Elle sera affichée directement en entrant l'adresse locale du serveur.

Les éventuels **Warnings** sont générés par les serveurs eux-mêmes. Il est possible de les supprimer en remplaçant la ligne  
`error_reporting = E_ALL`  
par  
`error_reporting = E_ALL & ~E_NOTICE | E_STRICT`  
dans le fichier `php.ini`.

## 3 - TRAVAIL À RÉALISER

Nous allons nous servir du serveur **WAMP**, car ce dernier propose un interpréteur PHP intégré.

Les pages réalisées seront placées dans le répertoire à créer :

`www\Site_Bilingue`.

### 3.1 - JE SUIS BILINGUE ! (AU MOINS !)

#### 3.1.1 - MON SITE S'ADAPTE AUTOMATIQUEMENT AU NAVIGATEUR

- 4) À partir de la page « Centres d'intérêts » de votre CV numérique, créez une nouvelle page rédigée en anglais. Pour faciliter la suite, les deux pages portent un nom identique suivi soit de `_fr` soit de `_en` ; par exemple `Loisirs_fr.html` et `Loisirs_en.html`. Ces deux pages se trouvent dans le même directory. Finalement, changez l'extension de ces fichiers en `.PHP`.
- 5) La variable globale `$_SERVER` contient un grand nombre d'informations relatives à l'environnement de travail du serveur. Certaines d'entre elles sont relatives à la configuration du navigateur. Pour utiliser une de ces données, il suffit de connaître son intitulé PHP, de l'invoquer dans le tableau `$_SERVER` et de la stocker dans une variable. Par exemple, l'instruction `$Script_FileName= $_SERVER['SCRIPT_FILENAME']` place dans la variable `$Script_FileName` le chemin absolu du fichier script en cours d'exécution.  
Quel est l'intitulé à utiliser afin de connaître les langues acceptées par défaut par votre navigateur pour les pages ? Rédigez, dans un nouveau fichier `index_1.php`, contenant exclusivement du script PHP, le code permettant de récupérer ces valeurs puis de les afficher dans le navigateur.
- 6) Les données précédentes sont indiquées dans une unique chaîne de caractères. Le PHP propose diverses fonctions de traitement des chaînes de caractères. Il est notamment possible d'en extraire une partie (`substr`), de rechercher des caractères spécifiques pour la découper (`explode`, `strtok`), de modifier la case (`strtolower`, `strtoupper`)... Selon la chaîne que vous avez affichée lors de la question précédente, modifiez votre script de façon à ce que seule la première langue ne soit affichée. Elle sera aussi mémorisée dans une variable `$langue`.
- 7) Avec cette variable, vous initialiserez une nouvelle variable `$fichier`, qui doit mémoriser le nom du fichier HTML associé à la langue par défaut du navigateur. Par exemple si c'est le français, le fichier associé sera `Loisirs_fr.php`. Vous utiliserez cette variable pour afficher cette page au moyen de la commande adéquate, à trouver dans l'aide-mémoire.

### 3.1.2 - LE VISITEUR PEUT CHOISIR LA LANGUE PAR BOUTONS

- 8) Dans un nouveau fichier `index_2.php`, de structure HTML standard, ajoutez dans un en-tête deux mini-icônes correspondant au français et à l'anglais. Il vous sera toujours possible d'ajouter d'autres langues par la suite ! Transformez ces images en boutons cliquables au moyen de la balise HTML `BUTTON`. Chaque bouton porte le même nom `langue` mais ont une valeur différente, `fr` pour l'un, `en` pour l'autre. Un clic sur ces boutons permettra de soumettre au script `Select.php` ces valeurs selon la méthode `POST`. Pour qu'un bouton soit efficient, il est nécessaire qu'il soit placé dans un formulaire HTML. La balise `Form` est utilisée à cette fin. Vous aurez des informations précises sur la structure de ces balises et de leurs attributs sur [http://www.w3schools.com/tags/tag\\_button.asp](http://www.w3schools.com/tags/tag_button.asp).
- 9) Créez un nouveau fichier `Select.php`, de structure PHP pure, permettant de sélectionner le fichier à afficher selon la langue cliquée dans la page `index_2.php`. Le script permettra de récupérer le paramètre `langue` passé selon la méthode `POST` et accessible au travers de la variable pré-définie `$_POST`. À partir de cette valeur, il affiche le fichier `Loisirs_fr.php` ou `Loisirs_en.php`.
- 10) Déportez la gestion des icônes de langue dans un nouveau fichier `Header.php`. Modifiez les fichiers `Loisirs_fr.php` et `Loisirs_en.php` de façon à y intégrer cette partie.