

*Pôle informatique 2013/2014*

*École Nationale Supérieure des Mines de Saint-Étienne*

# Développement de l'interface Web du projet Hôpital

Antoine Zimmermann

`antoine.zimmermann@emse.fr`

## Le cahier des charges du projet:

<http://www.emse.fr/~zimmermann/Teaching/CSI/projet-csi.html>

# Outils

- EasyPHP, contient:
  - ♦ Un serveur Web (Apache)
  - ♦ Un serveur SQL (MySQL)
  - ♦ Un interpréteur de scripts (PHP)
- Un navigateur Web (de votre choix)
- phpMyAdmin (pour les tests)
- Un éditeur de texte (sous Windows, Notepad++ est bien et il est installé dans les salles machines)

# Éditeur de texte

## ≠ Logiciel de traitement de texte

- Le nom est trompeur mais un **éditeur** de texte est plutôt un éditeur de chaînes de caractères;
- Un logiciel de **traitement** de texte sert à mettre en page un document textuel.
- Un logiciel de traitement de texte ne doit **JAMAIS** être utilisé pour écrire du code de programmation !
- Mais vous pouvez taper la documentation dans Word ou OpenOffice writer

# Architecture

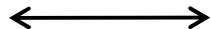
Mon ordinateur



Serveur ex.com



Base de données

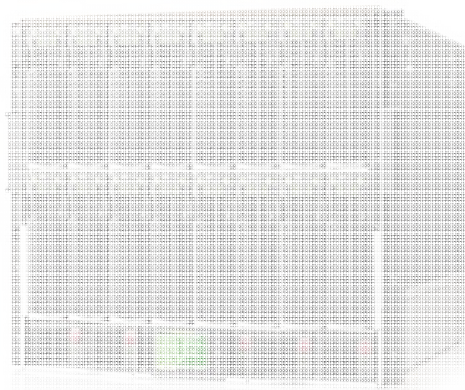


# Fonctionnement

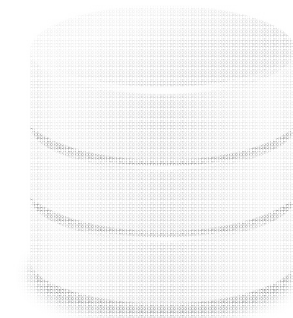
Mon ordinateur



Serveur ex.com



Base de données



Je tape `http://ex.com/truc.php` dans mon navigateur.

# Fonctionnement

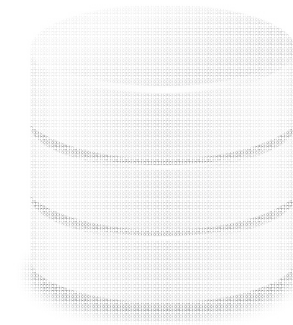
Mon  
ordinateur



Serveur ex.com



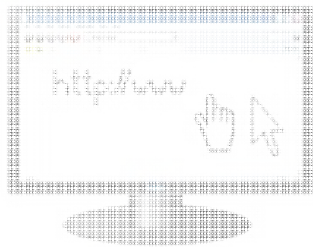
Base de données



`http://ex.com/truc.php` signifie: je demande  
le fichier `truc.php` au serveur `ex.com`

# Fonctionnement

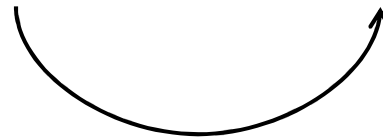
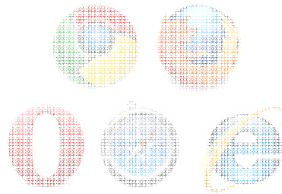
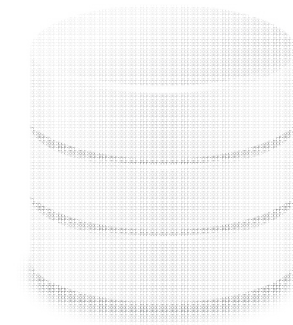
Mon  
ordinateur



Serveur ex.com



Base de données

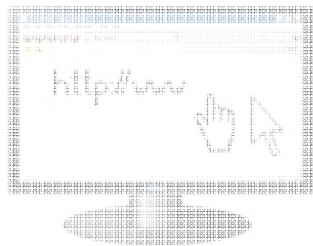


*Apache*: « c'est un fichier PHP, peux-tu me l'interpréter? » *PHP*: « Ok. »



# Fonctionnement

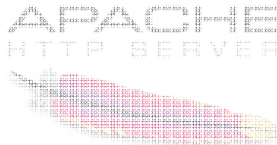
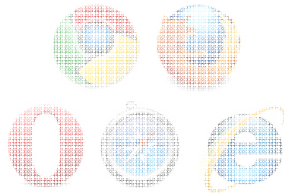
Mon  
ordinateur



Serveur ex.com



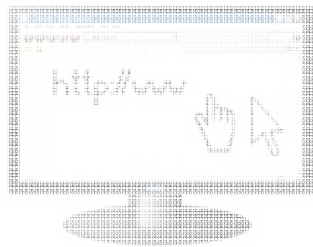
Base de données



*PHP*: « il y a une requête dans ce code, je les envoie à MySQL. »

# Fonctionnement

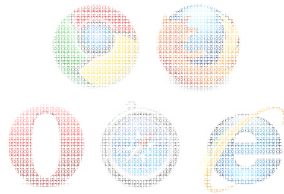
Mon  
ordinateur



Serveur ex.com



Base de données



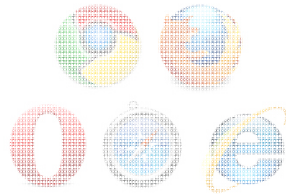
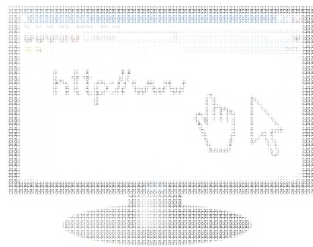
MySQL: « Voici le résultat de ta requête. »

PHP: « Merci, je traite ce résultat et fini

l'interprétation du code de truc.php. »

# Fonctionnement

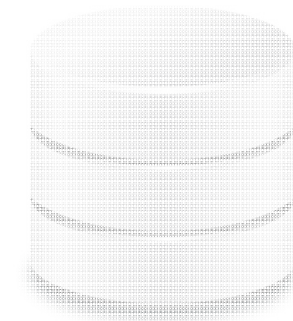
Mon  
ordinateur



Serveur ex.com



Base de données



APACHE  
HTTP SERVER



*PHP*: « Voici le code HTML que le script  
truc.php a généré. »

# Fonctionnement

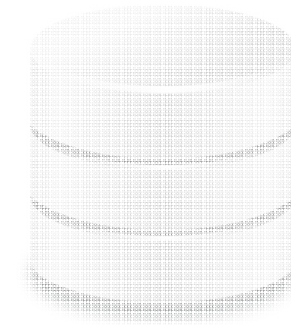
Mon  
ordinateur



Serveur ex.com



Base de données



*Apache*: « Voici la page HTML truc.php »

# Le module PHP

- PHP est un langage de programmation complet avec lequel on peut programmer n'importe quelle application
- Mais ce que PHP « rend » au serveur Web, c'est une page Web générée par le programme PHP et qui sera transmise au navigateur
- Le navigateur ne sait rien du contenu réel du fichier `truc.php`
- PHP ne sait pas exécuter une requête SQL, il se contente de la transmettre au serveur MySQL



# phpMyAdmin

Mon ordinateur



Serveur ex.com



Base de données



`http://127.0.0.1/home/index.php`

# phpMyAdmin

- *phpMyAdmin* est une application Web programmée en PHP
- *phpMyAdmin* n'est qu'une **interface** pour dialoguer facilement avec une base de données MySQL
- *phpMyAdmin* **ne sait pas exécuter une requête SQL**, il se contente de la transmettre au serveur MySQL et d'en afficher la réponse dans une page Web

# HTML

- HTML (HyperText Markup Language) sert à **décrire** la structure d'un document et permet de **lier** un document à d'autres documents par les liens hypertextes
- HTML **n'est pas** un langage de programmation
- HTML est un langage dit « balisé » ou « à balises ». Une « balise » sert à marquer le début et la fin d'un morceau du document, en indiquant dans le nom de la balise de quel type d'élément il s'agit



# HTML (exemple)

`<!-- ceci est un commentaire, il n'est pas affiché par le navigateur -->`

`<html> <!-- balise de début du document -->`

`<head>`

`<title>Page d'accueil</title>`

`</head>`

`<body>`

`<h1>Gestion des patients</h1>`

`<p>Veuillez vous <a href="login.php">authentifier</a>.</p>`

`</body>`

`</html> <!-- balise de fin du document -->`

# HTML (exemple)

`<!-- ceci est un commentaire, il n'est pas affiché par le navigateur -->`

`<html>`

`<head> <!-- balise de début d'en-tête -->`

`<title>Page d'accueil</title>`

`</head> <!-- balise de fin d'en-tête -->`

`<body>`

`<h1>Gestion des patients</h1>`

`<p>Veuillez vous <a href="login.php">authentifier</a>.</p>`

`</body>`

`</html>`

Le navigateur n'affiche pas l'en-tête mais peut l'utiliser pour savoir comment traiter la page

# HTML (exemple)

```
<!-- ceci est un commentaire, il n'est pas affiché par  
le navigateur -->
```

```
<html>
```

```
  <head>
```

```
    <title>Page d'accueil</title> <!-- titre -->
```

```
  </head>
```

```
  <body>
```

```
    <h1>Gestion des patients</h1>
```

```
    <p>Veuillez vous <a  
href="login.php">authentifier</a>.</p>
```

```
  </body>
```

```
</html>
```

Le titre de la page n'apparaît pas sur la page elle-même dans le navigateur

# HTML (exemple)

```
<!-- ceci est un commentaire, il n'est pas affiché par  
le navigateur -->
```

```
<html>
```

```
  <head>
```

```
    <title>Page d'accueil</title>
```

```
  </head>
```

```
  <body> <!-- corps du document -->
```

```
    <h1>Gestion des patients</h1>
```

```
    <p>Veuillez vous <a
```

```
href="login.php">authentifier</a>.</p>
```

```
  </body> <!-- fin du corps du document -->
```

```
</html>
```

Le corps contient toute la structure du document qui sera affiché

# HTML (exemple)

`<!-- ceci est un commentaire, il n'est pas affiché par le navigateur -->`

`<html>`

`<head>`

`<title>Page d'accueil</title>`

`</head>`

`<body>`

`<h1>Gestion des patients</h1> <!-- titre de 1er niveau -->`

`<p>Veuillez vous <a href="login.php">authentifier</a>.</p>`

`</body>`

`</html>`

Il y a des titres de niveau 1 à 6, pour intituler des sections et des sous section. En général, h1 correspond au titre de l'ensemble du document.

# HTML (exemple)

`<!-- ceci est un commentaire, il n'est pas affiché par le navigateur -->`

`<html>`

`<head>`

`<title>Page d'accueil</title>`

`</head>`

`<body>`

`<h1>Gestion des patients</h1>`

`<p>Veuillez vous <a`

`href="login.php">authentifier</a>.</p>`

`<!-- p pour paragraph, a pour anchor -->`

`</body>`

`</html>`

Anchor signifie « ancre ». Le texte à l'intérieur de la balise **a** devient un lien hypertexte qui pointe vers le fichier nommé à côté de **href**.

# HTML dans le projet

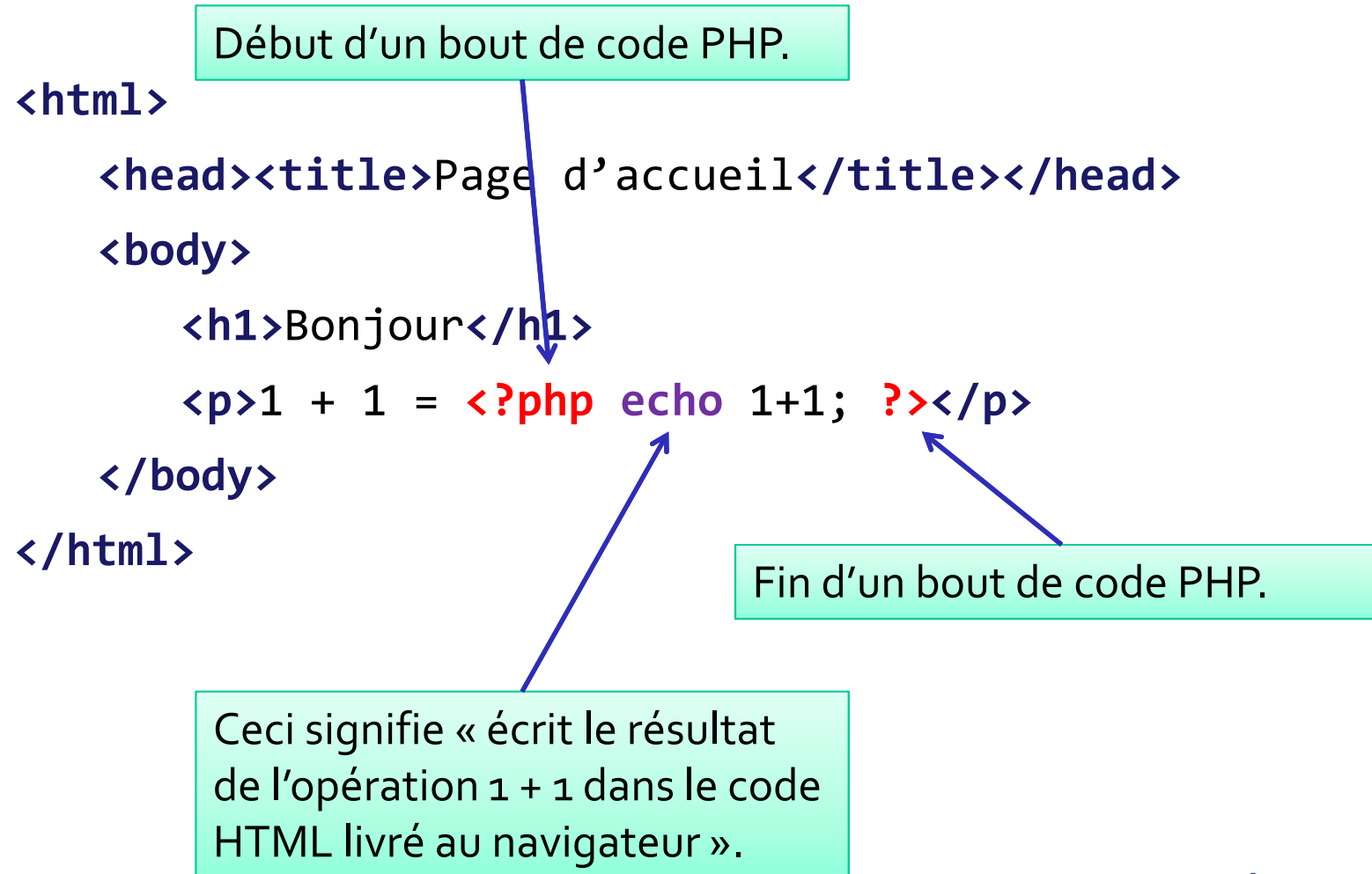
- On ne vous demande pas d'apprendre tout HTML
- Ce que vous devez retenir pour le projet:
  - ♦ Une balise ouverte doit être fermée
  - ♦ `<a href="nom_de_fichier">du texte</a>` pour créer un lien hypertexte vers un autre fichier
  - ♦ Quelques balises simples (h1, p, ul, li) pour présenter vos pages
  - ♦ Les balises de formulaires

# Le langage PHP

- PHP = PHP Hypertext Preprocessor
- Ressemble au langage C (utilisation des accolades pour les blocks, noms des types de données primitifs similaires, etc)
- Mais il y a de grandes différences:
  - On ne déclare pas les variables
  - Il y a de nombreuses fonctions complexes disponibles sans importer quoi que ce soit
  - Sert *a priori* et avant tout à générer des pages Web
  - Peut être utilisé au milieu de code HTML



# Exemple de code PHP



# Exemple de code PHP: les variables

```
<html>
```

```
...
```

```
<?php
```

```
// variable initialisée (pas de déclaration préalable)
```

```
$salut = "Hello";
```

```
$monde = "world";
```

```
$coucou = $salut . " " . $monde . "!"; // concaténation
```

```
echo "<p>" . $coucou . "</p>"; // génère un paragraphe
```

```
?>
```

```
...
```

```
</html>
```

# Exemple de code PHP: « absence » de type

```
<html>
```

```
...
```

```
<?php
```

```
/* les variables ne sont pas typées
```

```
le code suivant affiche simplement 64 */
```

```
$salut = "Hello";
```

```
$salut = 42 + 12;
```

```
echo "<p>" . $salut . "</p>";
```

```
?>
```

```
...
```

```
</html>
```

# Exemple de code PHP: tableaux associatifs (array)

```
<html>
```

```
... <!-- un exemple de tableau associatif, et une  
autre forme de chaine de caractères -->
```

```
<?php
```

```
$utilisateur['nom'] = "Durand";
```

```
$utilisateur["prenom"] = 'Bernard';
```

```
echo '<p>Je m'appelle <a href="profile.php">' .
```

```
    $utilisateur["prenom"] . " " .
```

```
    $utilisateur['nom'] . '</a>.</p>';
```

```
?>
```

```
...</html>
```

# Exemple de code PHP : connexion Bdd

Ceci est obligatoire  
avant d'interagir  
avec une Bdd

```
<html> ... <!-- interaction avec la Bdd -->
```

```
<?php
```

```
$base = mysql_connect('ex.com', 'admin', 'passw');  
mysql_select_db('ma_base', $base);
```

```
/*
```

```
des traitements divers et interaction avec la base
```

```
*/
```

```
mysql_close();
```

```
?>
```

```
...</html>
```

Ceci est obligatoire  
après avoir interagi  
avec une Bdd

# Exemple de code PHP

## ...interaction avec le SGBD

**<?php**

```
$base = mysql_connect('ex.com', 'admin', 'passw');  
mysql_select_db('ma_base', $base);  
$requete = 'INSERT INTO table VALUES (' . $var . ')';  
// on envoie une _chaîne de caractère_ au serveur MySQL  
$result = mysql_query($requete);  
if (!$result) {  
    /* si la commande mysql_query renvoie 0 (donc faux),  
    il y a eu un problème, on traite l'erreur  
};  
mysql_close(); ?> ...
```

# Trois variables spéciales

- `$_POST` contient des données de formulaires (voir TP<sub>4</sub>)
- `$_GET` contient des données indiquées dans l'adresse Web.

p.ex.: `http://ex.com/truc.php?param=val`

Alors `$_GET["param"]` a la valeur `"val"`

# Trois variables spéciales

- `$_SESSION` permet de stocker en mémoire des valeurs qui existeront tant que l'utilisateur est connecté

Exemple de fichier utilisant la variable `$_SESSION`



# ATTENTION!

- Ne pas confondre:
  1. Demander un fichier à un serveur Web à partir de son navigateur
  2. Ouvrir un fichier avec son navigateur
- Il faut toujours mettre les fichiers PHP dans le répertoire `www` du serveur Web
- Il faut toujours tester les fichiers PHP en tapant l'adresse du serveur Web (`http://127.0.0.1` dans le projet)