

# Développement Web 2

## TP/TD - N°1

### Exercice 1 – Simuler le protocole *http / https*

Même si le client HTTP le plus connu est votre navigateur, il est facile de simuler un client HTTP autrement. La commande *telnet* permet d'envoyer du texte à une machine distante. En envoyant le texte d'une requête HTTP à un serveur HTTP, celui nous envoie sa réponse http normalement.

Exemple :

```
1. > telnet pageperso.lif.univ-mrs.fr 80
2. GET /~luigi.santocanale/teaching/DW/ HTTP/1.1
3. Host: pageperso.lif.univ-mrs.fr
```

Attention, si vous souhaitez interroger de la même manière un serveur sur un port sécurisé (par exemple 443 pour HTTPS), vous ne pouvez pas utiliser cette méthode puisque la requête serait alors envoyée en clair et ne pourrait pas être interprétée par le serveur. Il faudrait donc dans ce cas utiliser l'outil *s\_client* fourni par *openssl*. Vous recevrez immédiatement les informations de certificat et pourrez à la suite envoyer votre requête.

- Utiliser *s\_client* afin d'afficher le code html de la page <https://www.w3schools.com/>

### Avant de poursuivre le TP

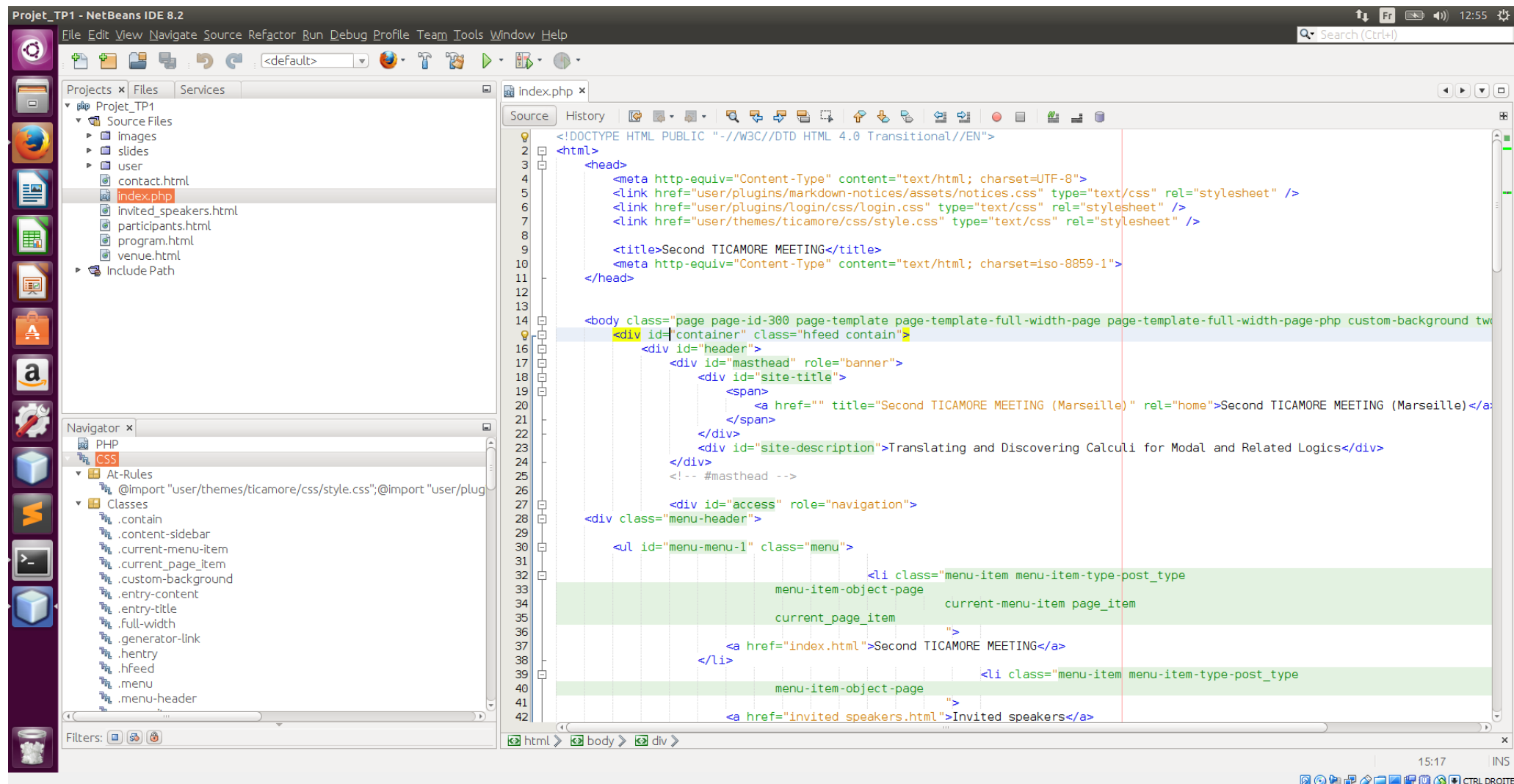
Nous allons commencer par lancer le serveur :

- Ouvrez une console.
- Allez à la racine de votre compte avec la commande *cd*.
- Vérifiez avec la commande *pwd* que vous vous trouvez à la racine de votre compte.
- Créez un répertoire PHP avec la commande *mkdir PHP*.
- Déplacez-vous dans ce répertoire avec la commande *cd PHP*.
- Lancez le serveur avec la commande suivante *php -S localhost:8080*.
- Ne fermez pas cette console car les erreurs PHP vont s'afficher dedans.

Les fichiers du répertoire PHP sont maintenant accessibles via l'URL <http://localhost:8080/>. Le nombre 8080 correspond au port d'écoute du serveur. Nous ne pouvons pas utiliser le port classique 80 car nous ne sommes pas administrateur de l'ordinateur.

- Créez un fichier *HelloWorld.php* dans le répertoire PHP avec le contenu de votre choix puis ouvrez un navigateur et consultez l'adresse <http://localhost:8080/HelloWorld.php>. Vous devriez voir le résultat de l'interprétation du fichier *HelloWorld.php*.

## Présentation de NetBeans



- Dans la fenêtre New Project, choisissez 'PHP' et 'PHP application With Existing Sources'.
- Choisissez le répertoire où se trouve le projet, donnez un nom à votre projet et choisissez son encodage.
- Une dernière configuration, et le projet est créé.
- Notre projet est maintenant accessible dans le 'gestionnaire de projet'.

## Exercice 2 – Premier code PHP

Créez un répertoire *Exercice1* dans le répertoire *PHP*. Dans le répertoire *Exercice1*, créez un fichier *index.php* qui génère une page HTML affichant le jour et l'heure du serveur. Pour cela, utilisez la fonction PHP *date*. Une description complète de cette fonction est proposée sur le site [php.net](http://php.net). Essayez les différents formats d'affichage proposés par cette fonction. Consultez ensuite le code généré par votre page via l'URL <http://localhost:8080/Exercice1>. Notez qu'il n'est pas nécessaire de mettre *index.php* dans l'URL car ce fichier correspond au point d'entrée par défaut d'un répertoire.

## Exercice 3 – Manipuler le PHP

L'objectif de cet exercice est d'apprendre les conventions lexicales de PHP, de manipuler les tableaux associatifs en PHP et de comprendre la portée des variables en PHP (Source : <http://www.funhtml.com/php7/index.html>)

- Parmi les variables suivantes, lesquelles ont un nom valide : *mavar*, *\$mavar*, *\$var5*, *\$\_mavar*, *\$\_5var*, *\$\_élément1*, *\$hotel4 \* ?*
- Écrivez un tableau multidimensionnel associatif dont les clés sont des noms de personne et les valeurs des tableaux indicés contenant le prénom, la ville de résidence et l'âge de la personne.
- Utilisez une boucle *foreach* pour afficher le contenu du tableau précédent.
- Considérez le script suivant :

```
1. <?php
2. $config['development'] = true;
3. function myVarDump($var){
4.     if(isset($config['development']) && $config['development'] === true){
5.         var_dump($var);
6.         exit ;
7.     }
8.     return;
9. }
10. $array = [1,2,3];
11. myVarDump($array);
```

- Que s'affiche à l'écran lors de l'exécution du script tel quel ?
- Comment modifier le script afin qu'il s'exécute comme prévu ?

## Exercice 4 – Galerie de portraits

Créez un répertoire Exercice2 dans le répertoire PHP. Créez un fichier *index.php* dans le répertoire que vous venez de créer et insérez les lignes suivantes à l'intérieur :

Afin de faire fonctionner cette page, vous devez ensuite écrire la fonction *generateGallery(\$data)* dans le *portraits.php*. Cette fonction doit générer le code HTML d'une galerie de portraits à partir d'informations présentes dans la variable *\$data*. Votre fonction doit s'attendre à trouver un tableau de tableaux associatifs dans le paramètre *\$data*. Le nombre de tableaux associatifs présents dans le tableau *\$data* est égal aux nombres de portraits de la galerie. Chaque tableau associatif *\$portrait* contenu dans le tableau *\$data* contient le nom, le prénom et un lien vers le portrait du personnage. Ces trois données sont respectivement associées aux clés *name*, *firstname* et *portrait*.

Le code HTML que vous devez générer doit ressembler à celui-ci :

```
1. <!DOCTYPE html>
2. <html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">
3. <head>
4. </head>
5. <body>
6. <?php
7.     include 'portraits.php';
8.     $portrait1 = array('name'=>'Victor', 'firstname'=>'Hugo',
9.         'portrait'=>'http://upload.wikimedia.org/wikipedia/commons/5/5a/Bonnat_Hugo001z.jpg' );
10.    $portrait2 = array('name'=>'Jean', 'firstname'=>'de La Fontaine',
11.        'portrait'=>'http://upload.wikimedia.org/wikipedia/commons/e/e1/La_Fontaine_par_Rigaud.jpg' );
12.    $portrait3 = array('name'=>'Pierre', 'firstname'=>'Corneille',
13.        'portrait'=>'http://upload.wikimedia.org/wikipedia/commons/2/2a/Pierre_Corneille_2.jpg' );
14.    $portrait4 = array('name'=>'Jean', 'firstname'=>'Racine',
15.        'portrait'=>'http://upload.wikimedia.org/wikipedia/commons/d/d5/Jean_racine.jpg' );
16.    $data = array($portrait1, $portrait2, $portrait3, $portrait4);
17.    generateGallery($data);
18. ?>
19. </body>
20. </html>
```

Afin de faire fonctionner cette page, vous devez ensuite écrire la fonction *generateGallery(\$data)* dans le *portraits.php*. Cette fonction doit générer le code HTML d'une galerie de portraits à partir d'informations présentes dans la variable *\$data*. Votre fonction doit s'attendre à trouver un tableau de tableaux associatifs dans le paramètre *\$data*. Le nombre de tableaux associatifs présents dans le tableau *\$data* est égal aux nombres de portraits de la galerie. Chaque tableau associatif *\$portrait* contenu dans le tableau *\$data* contient le nom, le prénom et un lien vers le portrait du personnage. Ces trois données sont respectivement associées aux clés *name*, *firstname* et *portrait*.

Le code HTML que vous devez générer doit ressembler à celui-ci :

```

1. <table>
2. <thead>
3. <tr>
4. <th>Prénom</th>
5. <th>Nom</th>
6. <th>Portrait</th>
7. </tr>
8. </thead>
9. <tbody>
10. <tr>
11. <td>Victor</td>
12. <td>Hugo</td>
13. <td><img style="width:50px;"
14. <td>src="http://upload.wikimedia.org/wikipedia/commons/5/5a/Bonnat_Hugo001z.jpg" /></td>
15. </tr>
16. <tr>
17. <td>Jean</td>
18. <td>de La Fontaine</td>
19. <td><img style="width:50px;"
20. <td>src="http://upload.wikimedia.org/wikipedia/commons/e/e1/La_Fontaine_par_Rigaud.jpg" ></td>
21. </tr>
22. <tr>
23. <td>Pierre</td>
24. <td>Corneille</td>
25. <td><img style="width:50px;"
26. <td>src="http://upload.wikimedia.org/wikipedia/commons/2/2a/Pierre_Corneille_2.jpg" ></td>
27. </tr>
28. <tr>
29. <td>Jean</td>
30. <td>Racine</td>
31. <td><img style="width:50px;"
32. <td>src="http://upload.wikimedia.org/wikipedia/commons/d/d5/Jean_racine.jpg" ></td>
33. </tr>
34. </tbody>
35. </table>

```

Pour vous aider, voici une version de la méthode *generateGallery()* qui génère un tableau avec les noms des personnes :

```

1. <?php function generateGallery($data) { ?>
2. <table>
3. <?php foreach ($data as $portrait) { ?>
4. <tr><td><?=$portrait['name']?></td></tr>
5. <?php } ?>
6. </table>
7. <?php } ?>

```

## Exercice 5 – Structure arborescente d'un document HTML

L'objectif de cet exercice est d'effectuer un rappel sur la structure d'un document HTML en tant que structure arborescente.

- Installez sur Chrome l'extension [HTML Tree Generator](#)
- Consultez avec Chrome votre galerie de portrait *index.php* réalisé précédemment puis inspectez l'arbre de cette page avec l'extension HTG.
- Ajouter à votre page d'autres portraits et inspectez là de nouveau avec HTG.
- Inspectez avec HTG une page de votre choix.

## Exercice 6 – Aspirer un site

Le but de cet exercice est de réussir à mettre en œuvre rapidement un site web en utilisant un site web donné préexistant via les commandes *wget* ou *curl*. Ces derniers sont des utilitaires très pratiques puisqu'ils permettent de télécharger des fichiers en ligne de commande.

- Faites une copie complète du site web <https://ticamore.lis-lab.fr/>

Avec *wget* :

```
1. wget -r -k -np --user-agent=Firefox https://ticamore.lis-lab.fr/
```

-r signifie que le téléchargement sera récursif (télécharge aussi les liens de la page).

-k permet de reconstituer le site localement (les liens sont modifiés pour pointer localement).

-np empêche de remonter dans le répertoire parent.

Exécutez la commande *man wget* pour vous renseigner davantage sur les options les plus avancées

- Modifiez et factorisez le code *html* téléchargé en utilisant *PHP* selon l'exemple vu en cours.

## Pour aller plus loin – Utilisation de *Bootstrap*

Le but de cet exercice est de réorganiser le site produit pour l'exercice 6 en utilisant *Bootstrap*.

La documentation de *Bootstrap* peut être consulté ici : <https://getbootstrap.com/docs/3.3/css/>