

<b>NOM :</b>	<b>CLASSE :</b>
<b>PRÉNOM :</b>	

<b>Condition :</b>	<ul style="list-style-type: none"><li>• Travail seul ou en binôme; durée 3 heures</li></ul>
<b>Matériel :</b>	<ul style="list-style-type: none"><li>• un ordinateur sous Ubuntu</li></ul>
<b>Documents :</b>	<ul style="list-style-type: none"><li>• le sujet du TP</li><li>• le cours sur la programmation en php</li></ul>

L'objectif du TP est de découvrir le langage PHP, de l'utiliser pour rendre un site web dynamique ou encore pour commander un Arduino.

## 1. Installation de Lamp

Avant de démarrer la programmation en PHP, l'installation de logiciels est nécessaire pour transformer votre ordinateur en serveur sachant exécuter ce langage. Pour réaliser ce TP, vous aurez donc besoin :

- d'un **éditeur de texte** : **Geany**, déjà installé sur les ordinateurs, fera très bien l'affaire ;
- d'un **navigateur web** : **Firefox**, sur les PC des salles ;
- d'un **serveur web** : **Apache**, c'est lui qui sera chargé de délivrer les pages web aux visiteurs ;
- d'un **interpréteur php** : **PHP**, plug-in pour Apache qui le rend capable de traiter les pages web dynamiques écrites en php ;
- d'un **gestionnaire de base de données** : **MySQL**, que vous utiliserez dans un prochain TP.

Les 3 logiciels Apache, PHP et MySQL ne sont pas installés sur votre ordinateur. Ils constituent ce qu'on appelle un **serveur LAMP**, qui est l'acronyme de **Linux Apache MySQL, PHP**. On retrouve l'équivalent sur Windows, appelé serveur WAMP.



- ✓ Installez ces 3 logiciels en tapant sous la console la commande :

```
sudo apt-get install apache2 php5 mysql-server libapache2-mod-php5 php5-mysql
```

- ✓ Choisir pour mot de passe *sti2dsin* pour le compte root du logiciel MySQL utilisé dans le TP suivant.
- ✓ Les fichiers HTML ou PHP de votre site seront placés dans le dossier `/var/www`. Pour vous donner les droits d'écriture à ce dossier, tapez sous la console les 2 commandes suivantes :

```
sudo chown -R $USER /var/www  
sudo chmod 755 /var/www
```



Attention, si vous travaillez avec une version récente d'Ubuntu (par exemple Ubuntu 14.04) le dossier `/var/www` est remplacé par `/var/www/html`

- ✓ Démarrez Firefox et tapez, dans la barre d'adresse, l'url *localhost*. Le nom *localhost* est associé à l'adresse IP 127.0.0.1 qui correspond à votre machine. Si votre serveur Apache est correctement installé, la page « It works ! » apparaît.
- ✓ Effacez à l'aide de Nautilus le contenu du dossier `/var/www`. Vous êtes prêt pour commencer la programmation en PHP.

## 2. Mon premier programme en PHP

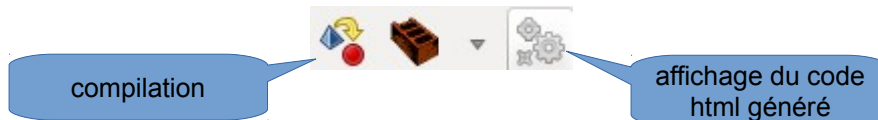
L'objectif de ce premier programme est d'écrire le fameux message « Hello World ! » sur votre page web.

- ✓ Démarrez Geany, et créez un nouveau fichier *index.php* dans le dossier */var/www*.
- ✓ Éditez ce fichier en recopiant le code ci-dessous :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon super site</title>
  </head>
  <body>
    <?php
      echo 'Hello World !';
    ?>
  </body>
</html>
```

entre <?php et ?> se trouve le code écrit en php, le reste est de l'HTML

- ✓ Enregistrez le fichier. Geany est capable de compiler du code écrit en php et donc de vous indiquer d'éventuels erreurs. Vérifiez que votre code est correct, le message « compilation terminée avec succès » doit apparaître dans le bas de l'écran de Geany.



- ✓ Enregistrez le fichier. Ouvrez Firefox et tapez dans la barre d'adresse *localhost*. Le nom *localhost* est associé à l'adresse IP 127.0.0.1 qui correspond à votre machine. Votre page web apparaît.

## 3. Une page web dynamique

La première page créée dans le chapitre précédent n'est pas une page dynamique, le message affiché est toujours le même, et aurait pu être écrite entièrement en HTML.

Nous allons ici écrire une nouvelle page qui ajoute à la précédente l'affichage l'heure et la date.

- ✓ Modifiez la partie php du fichier *index.php* précédent en recopiant le programme ci-dessous. Enregistrez puis vérifiez le résultat sous Firefox. Félicitations, vous venez d'écrire votre première page web dynamique.

```
<?php
  echo 'Hello World !';
  echo '<br/>';
  $date = date('d-m-Y');
  $heure = date('H:i:s');
  echo 'Nous sommes le '.$date.' et il est '.$heure;
?>
```

- ✓ Pourquoi *date* et *heure* sont-ils précédés du caractère \$ ?
- ✓ La fonction *date* renvoie une chaîne de caractères (String) en fonction des paramètres qu'elle reçoit. Utilisez le site web *php.net*, en faisant une recherche sur cette fonction *date*, afin de modifier votre programme pour afficher :

```
Hello World !
Nous sommes le Sunday 12 October 2014
Il est 21h28m54s
Nous sommes le 284ème jour de l'année 2014
```

Validation prof :

Remarque : pour écrire un **apostrophe** en PHP, vous devez le précéder d'un antislash dans la chaîne de caractères. Par exemple, `echo 'ceci n'est pas une mauvaise chaîne';`

## 4. Les formulaires

Les formulaires sont très présents sur les pages web : ils permettent de transmettre des informations du client vers le serveur.

**Le formulaire s'écrit en HTML**, encadré par les balises `<form>` et `</form>`. Des objets tels que des lignes de texte, des boutons, des cases à cocher sont placés à l'intérieur de ces balises dans le but de recueillir des informations.

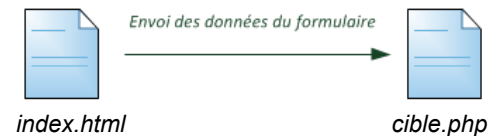
Une fois le formulaire complété, un bouton « Envoyer » permet de transmettre les informations au serveur à l'aide d'une méthode d'envoi, **GET** ou **POST**.

Nous allons créer un formulaire permettant à l'utilisateur de rentrer son nom et son prénom :

Nom:

Prénom :

- ✓ On donne, ci-dessous, les programmes *index.html* et *cible.php*. Le deuxième fichier est appelé par le premier lors de l'appui sur le bouton *Envoyer*. Faites la saisie de ces 2 programmes et testez le fonctionnement sous Firefox.



*index.html*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon super site</title>
  </head>
  <body>
    <form action="cible.php" method="get">
      Nom:
      <input type="text" name="nom" size="20"/><br>
      Prénom :
      <input type="text" name="prenom" size="20"/><br>
      <input type="submit" value="Envoyer"/>
    </form>
  </body>
</html>
```

*cible.php*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon super site</title>
  </head>
  <body>
    <?php
      echo 'Bonjour ' . $_GET['prenom'] . ' ' . $_GET['nom'] . ' !<br/>';
      echo 'Pour revenir à la page formulaire <a href="index.html">clique ici</a>';
    ?>
  </body>
</html>
```

Répondez aux questions ci-dessous en analysant le programme *index.html*:

- ✓ Recopiez la ligne permettant de créer un champ de saisie de texte simple appelé prénom.
- ✓ Recopiez la ligne permettant de créer un bouton d'envoi du formulaire.
- ✓ Recopiez la ligne permettant de créer un champ de saisie de texte simple appelé nom.

- ✓ La méthode utilisée ci-dessus est le GET et donc les données du formulaire sont transmises au fichier cible.php par l'URL. Notez ci-dessous l'URL lorsque vous tapez votre nom, votre prénom et que vous validez :
- ✓ Repérez dans cette URL les noms et valeurs des variables transmises.

Nom de la variable			
Valeur			

- ✓ Modifiez le fichier index.html afin d'obtenir le formulaire ci-dessous. Vous utiliserez les sites ci-dessous pour vous aider :  
<http://formation.upyupy.fr/html-xhtml/formulaires-html/>  
<http://cyberzoide.developpez.com/html/form.php3>

Nom:

Prénom :

Sexe :  Masculin  Féminin

Année de naissance :

Vos loisirs favoris :

Sport  Shopping  Cinéma  TV  Jeux vidéos

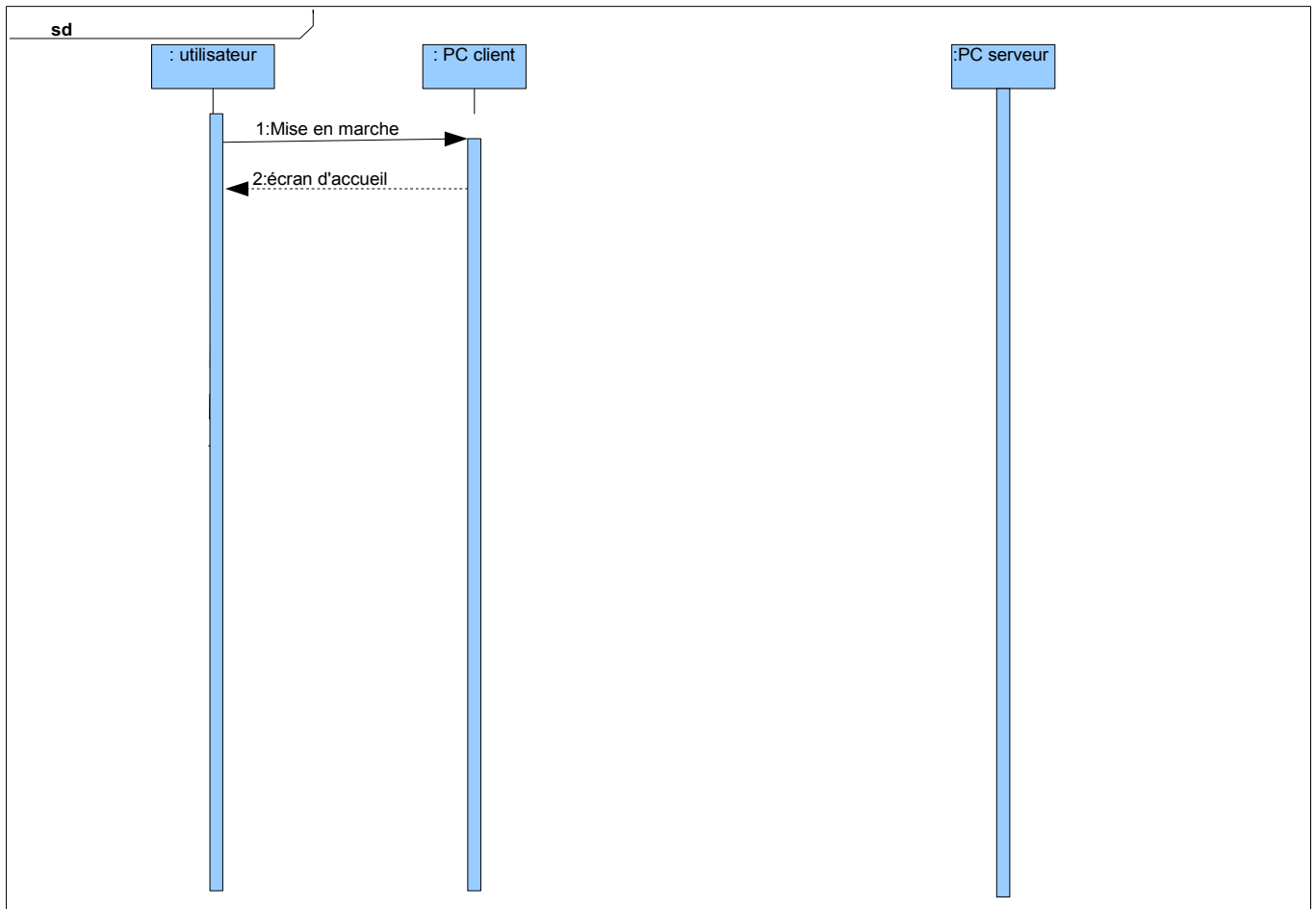
- ✓ Modifiez maintenant le fichier cible.php afin qu'il récupère et affiche toutes les données envoyées sous la forme suivante :

Bonjour pierre Dupond !  
 Vous êtes un homme  
 Vous êtes né en 1997.  
 Vous avez 17 ans.  
 Vos loisirs préférés sont : sport cinema

Pour revenir à la page formulaire, [cliquez ici](#)

Remarque : pour récupérer tous les loisirs du visiteur, vous devez les stocker dans un tableau dans le formulaire et lire le contenu du tableau dans le fichier cible.

- ✓ Démarrez le logiciel Wireshark (*sudo wireshark* sous la console). Faites une capture des trames sur l'interface lo (loopback) lors de l'envoi du formulaire. Appliquez le filtre http afin de ne conserver que les trames HTTP échangées avec votre PC. Complétez alors le diagramme de séquence ci-dessous, montrant les échanges entre l'utilisateur (vous), le PC client et le PC serveur.

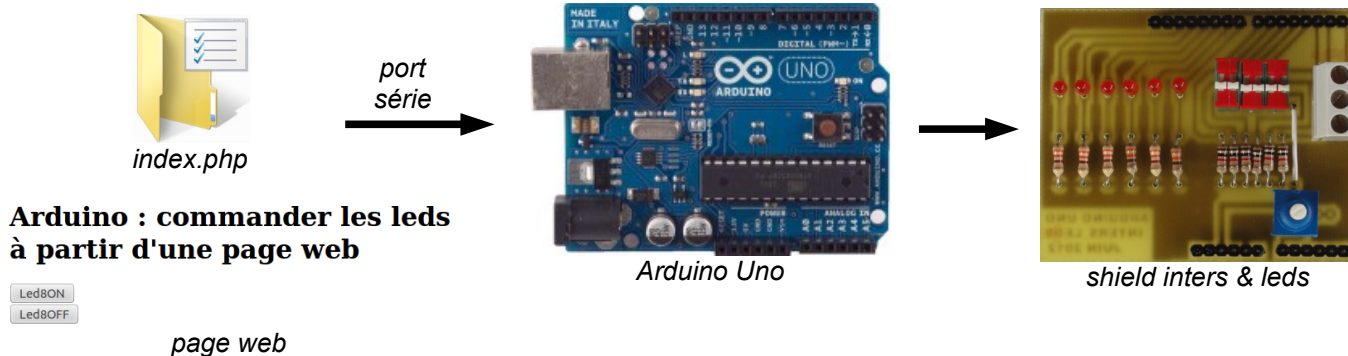


- ✓ Remplacez tous les GET du fichier index.html par des POST. Idem pour le fichier cible.php. Que remarquez-vous dans l'URL quand vous cliquez maintenant sur le bouton Envoyer du formulaire ? Comment sont transmises les informations ?

La méthode POST est préférée à la méthode GET et est généralement utilisée dans les formulaires.

## 5. Communication avec l'Arduino

L'objectif est ici de commander, à partir d'un formulaire, des sorties de la carte Arduino Uno. On utilise un shield inters & leds, les 8 leds étant câblées sur les broches 8 à 13 de l'Arduino.



- ✓ Pour donner accès en PHP au port série, il faut taper les commandes suivantes sous la console :

```
sudo adduser www-data dialout
sudo /etc/init.d/apache2 restart
```

Le fichier *index.php* ci-dessous contient à la fois le formulaire et le code PHP gérant les données reçues de ce formulaire.

- ✓ Faites la saisie de ce programme et enregistrez-le, dans /var/www/.

*index.php*

```
<?php
  $comPort = "/dev/ttyACM0"; /*change to correct com port */
  if (isset($_POST["rcmd"])) {
    $rcmd = $_POST["rcmd"];
    switch ($rcmd) {
      case Led8ON:
        $fp = fopen($comPort, "w");
        fwrite($fp, 1); /* this is the number that it will write */
        fclose($fp);
        break;
      case Led8OFF:
        $fp = fopen($comPort, "w");
        fwrite($fp, 2); /* this is the number that it will write */
        fclose($fp);
        break;
      default:
        die("Crap, something went wrong. The page just puked.");
    }
  }
?>
```

nom du port série où est connecté l'Arduino

si un caractère est reçu du formulaire

envoi de l'ordre d'allumage de la led D8 sur la liaison série

envoi de l'ordre d'extinction de la led D8 sur la liaison série

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon super site</title>
  </head>
  <body>
    <h1>Arduino : commander les leds<br/>
    à partir d'une page web</h1>

    <form method="POST" action="<?php echo $PHP_SELF;?>">
      <input type="submit" value="Led8ON" name="rcmd">&nbsp; <br/>
      <input type="submit" value="Led8OFF" name="rcmd">&nbsp; <br />
    </form>
  </body>
</html>
```

création des boutons Led8ON et Led8OFF

Le programme ci-dessous est celui de l'Arduino, qui analyse les codes reçus sur la liaison série et commande, en fonction de ces codes, l'allumage ou l'extinction de la led 8.

- ✓ Faites la saisie et programmez votre Arduino. Testez alors la commande de la led D8 à partir de la page web. N'oubliez pas d'ouvrir le moniteur série de l'Arduino.

ArduinoPHP.ino

```
int usbnumber = 0; //this variable holds what we are currently reading from serial

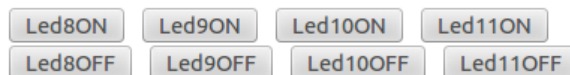
void setup() { //call this once at the beginning
  for (int i=8;i<=11;i++)
    pinMode(i,OUTPUT);
  Serial.begin(9600); //start up serial port
}

void loop() { //main loop
  if (Serial.available() > 0) { //if there is anything on the serial port, read it
    usbnumber = Serial.read(); //store it in the usbnumber variable
    Serial.println(usbnumber);
  }

  if (usbnumber > 0) { //if we read something
    if (usbnumber == 49){ // 49 = '1' (code ASCII)
      digitalWrite(8,HIGH);
    }
    if (usbnumber == 50){ // 50 = '2' (code ASCII)
      digitalWrite(8,LOW);
    }
  }
  usbnumber = 0; //reset
}
```

- ✓ Modifiez alors les programmes *index.php* et *ArduinoPHP.ino* afin de commander les 8 leds à partir de la page web. Testez et faites valider par le professeur.

## Arduino : commander les leds à partir d'une page web



Validation prof :