ISN S9: Les pages dynamiques avec PHP

Objectif :
Créer une page dynamique avec le langage de programmation PHP.

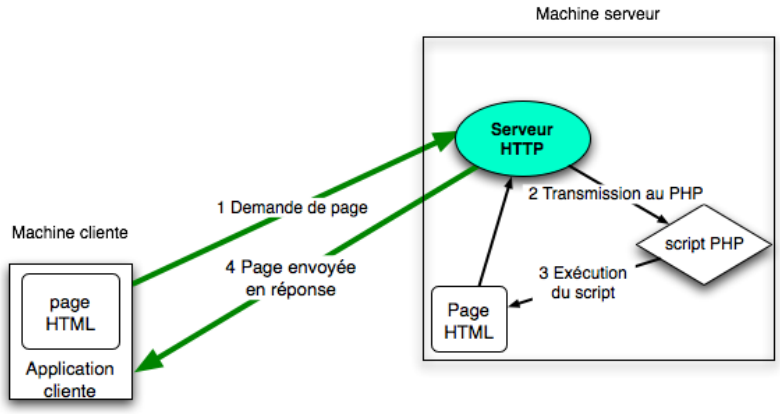
Nous avons pour l'instant réalisé des **pages Web statiques**, c'est-à-dire que leur contenu est figé. Ici nous allons créer des pages Web dont le contenu dépend des choix de l'utilisateur du site, on parle alors de **page dynamiques**.

0. Liens utiles

- Serveur EasyPHP : http://www.easyphp.org
- Toute la documentation sur PHP : http://php.net/manual/fr
- Des bons cours et tutoriels sur PHP : http://php.developpez.com/cours

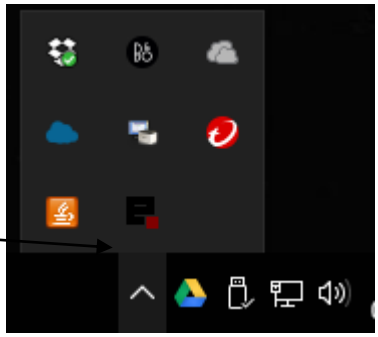
1. Démarrage du serveur Web EasyPhp

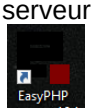
Le code PHP étant exécuté par un serveur pour générer une page HTML, un serveur Web est donc indispensable. Nous allons utiliser EasyPHP.



EasyPHP est un serveur Web capable d'exécuter du code PHP, il est libre et gratuit, c'est un serveur dédié au développement de site Web, c'est-à-dire qu'il est configuré pour afficher les messages d'erreurs et fonctionne sur le poste du développeur.

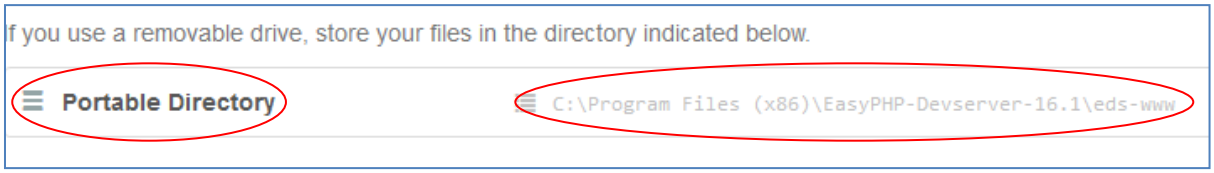
Marche à suivre pour démarrer le serveur :



1. Double-cliquer sur l'icône 

2. Ensuite, une fois le serveur démarré, cliquez sur son icône en bas à droite.
3. Cliquer sur « Open Dashboard ».
4. Démarrer le serveur « DATABASE SERVER » puis le serveur « HTTP SERVER » en cliquant sur les boutons « start ».

En cliquant sur « Portable Directory », vous accédez au site Web local.
En cliquant sur « C:\Program Files (x86)\EasyPHP-Devserver-16.1\eds-www», vous ouvrez l'explorateur de fichier dans le répertoire contenant les pages Web.



Pour toute création ou modification de fichier, utilisez Notepad++ qui est sur votre clé ISN et assurez-vous que les fichiers sont au format utf-8.

2. Premiers programmes PHP :

a) Programme HelloWord.php :

Le code PHP peut s'insérer directement dans une page Web ;

- Le code doit se trouver entre les balises **<?php** et **?>**.
- Le nom du fichier doit avoir l'extension **.php** au lieu de .html.
- Le fichier doit se trouver dans le répertoire « C:\Program Files (x86)\EasyPHP-Devserver-16.1\eds-www ».

I) Créez un fichier nommé « Q1.php » et copiez dedans le contenu suivant :

```
<html>
    <head>
          <meta charset="utf-8"/>
          <title>TP PHP ISN</title>
    </head>
    <body>
          <?php
                echo("Hello Word !");
          ?>
    </body>
</html>
```

II) Après l'avoir mis dans « C:\Program Files (x86)\EasyPHP-Devserver-16.1\eds-www » , ouvrez ce fichier dans un navigateur Web en tapant l'adresse du serveur puis le nom du fichier (127.0.0.1/Q1.php), quel est le résultat ?
Que fait la fonction echo() ?

b) Les constantes et variables :

Comme avec tous les autres langages de programmation, il est possible de créer des constantes et des variables, exemple :

```
<?php
    // déclaration de constante
    define('ISN', 'Informatique et Sciences du Numérique');

    // déclaration de variables
    $nom = "Mettre votre nom ici";
    $prenom = "Mettre votre prénom ici";
    $age = 17; // modifiez si nécessaire
?>

<html>
    <head>
          <meta charset="utf-8"/>
          <title>TP PHP ISN</title>
    </head>
    <body>
          <p>Je m'appelle <?php echo($prenom . " " . $nom); ?>.</p>
          <p>J'ai <?php echo($age); ?> ans.</p>
          <p>Je suis en cours d'<?php echo(ISN); ?>.</p>
    </body>
</html>
```

III) Copiez ce code dans un fichier nommé Q3.php. Placez-le au bon endroit puis ouvrez ce fichier dans un navigateur Web, quel est le résultat ?
Quel est l'opérateur de concaténation de chaine de caractères ?

IV) Ajoutez une ligne dans Q3.php pour afficher une ligne supplémentaire contenant « L'année prochaine, j'aurai 18 ans. » en utilisant la variable $age.

Pour tout savoir sur les variables : http://php.net/manual/fr/language.variables.basics.php

3. Le Pipotron :

Lorsque l'on termine un rapport, un compte-rendu, un exposé,… et que l'on manque d'inspiration pour la conclusion, le Pipotron est là pour nous aider : http://www.lepipotron.com

Il s'agit d'un programme qui génère une phrase en assemblant 8 parties de phrase choisis par l'utilisateur ou aléatoirement. Exemple :

Eu égard à *la dualité de la situation* *de ces derniers temps,* *il serait bon de* *s'intéresser à* *la totalité des voies* *envisageables.*

Pour les 6 premières parties, il y a 10 possibilités, et 5 pour les 2 dernières parties, ce qui nous fait $10^6.5^2$ soit 25 000 000 de conclusions possibles.

Nous allons réaliser un programme identique en PHP.

Dans un premier temps il faut entrer dans un tableau à 2 dimensions les différentes parties de la phrase. C'est ce qui est fait dans le fichier *pipo.php*.

```
<?php
//-------------------------------------------------------
    $pipo[0][0] = "Avec ";
    $pipo[0][1] = "Considérant ";
    ...
//-------------------------------------------------------
    $pipo[1][0] = "la situation ";
    $pipo[1][1] = "la conjoncture ";
```

- Le 1er indice représente la partie de la phrase (de 0 à 7).
- Le 2ème indice représente une des possibilités pour cette partie de la phrase (de 0 à 9).

a) Inclusion et débogage :

- La fonction include() permet d'inclure un autre fichier php, on s'en servira ici pour inclure *pipo.php*.
- La fonction var_dump() permet d'afficher les caractéristiques d'une variables (cela est bien pratique pour déboguer un programme).
- La balise html <pre> permet d'afficher un texte pré-formaté.

V) En utilisant ces 2 fonctions, complétez le programme pipotronQ5.php pour qu'il affiche la variable $pipo. Tester dans le navigateur (n'oubliez pas de placer aussi le fichier pipo.php dans le répertoire eds-ww).

b) La conclusion N° 00000000 :

On appelle la conclusion N° 00000000 celle qui utilise la 1ère possibilité (indice 0) pour chaque partie de la phrase. Cette conclusion est : « *Avec la situation présente, il convient d'étudier toutes les solutions envisageables.* »

VI) Compléter le programme pipotronQ6.php pour qu'il affiche cette conclusion.
Vous noterez un petit problème de liaison (*de étudier*), nous réglerons de problème plus tard.

c) La boucle for(;;) :

Plutôt que de répéter 8 fois la même commande, nous allons utiliser la boucle for(;;) pour afficher les 8 parties de la conclusion N° 00000000.
Voir http://php.net/manual/fr/control-structures.for.php

VII) Compléter le programme pipotronQ7.php pour qu'il affiche cette conclusion N° 00000000.

Mais notre page est toujours statique ! Un petit peu d'aléatoire va rendre la page dynamique.

d) La fonction rand() :

Au lieu d'utiliser la valeur 0 pour le 2ème indice du tableau $pipo, nous allons créer un nombre aléatoire entre 0 et 9.
La fonction rand() génère et renvoie un nombre aléatoire, voir http://php.net/manual/fr/function.rand.php

VIII) Compléter le programme pipotronQ8.php pour qu'il affiche une conclusion aléatoire.
Vérifiez que le résultat paraisse bien aléatoire.

e) Un peu d'ordre maintenant !

Pour l'instant le code PHP est incrusté un peu partout dans le code HTML. Or il est préférable de procéder d'abord aux traitements et aux calculs, et ensuite afficher le résultat. Le programme précédent va prendre la structure suivante :

```
<?php
    include("pipo.php");

    $conclusion = "";
    for ($i = ...          ) {
          $alea = ...
          $conclusion = ...
    }
?>

<htmp>
    <head>
          <title>Le pipotron</title>
    </head>
    <body>
          <h1>Le pipotron</h1>
          <?php echo($conclusion); ?>
    <body>
<html>
```

Partie « traitement » en PHP

Partie « affichage » en HTML.

Dans la partie HTML, on utilisera du code PHP uniquement pour afficher des variables.

IX) Compléter le programme pipotronQ9.php pour qu'il affiche une conclusion aléatoire en respectant la structure ci-dessus.

f) Les fonctions

Nous avions constaté à la question VI un problème de liaison. Nous allons créer une fonction qui résout le problème.

Tout d'abord, d'où vient le problème ? Le problème survient lorsque la 4ème partie de phrase se termine par « de » et que la partie suivante commence par une voyelle. Exemple :
« Vu l'inertie actuelle, on se doit de étudier toutes les issues possibles. »

Quatre cas peuvent se produire ici :

- « de e » qu'il faudra remplacer par « d'e »
- « de é » qu'il faudra remplacer par « d'é »
- « de i » qu'il faudra remplacer par « d'i »
- « de a » qu'il faudra remplacer par « d'a »

Pour cela, nous pouvons utiliser 4 fois la fonction str_replace(), voir http://php.net/manual/fr/function.str-replace.php

Le programme aura la structure suivante :

```
<?php
    include("pipo.php");

    function adapterLiaison($str)
    {
          //
          // ici on modifie $str, à compléter
          //
          return $str;
    }

    $conclusion = "";
    for ($i ...              ) {
          $alea = ...
          $conclusion = ...
    }

    // on utilise la fonction adapterLiaison() pour supprimer
    // le problème de liaison
    $conclusion = ...
?>

<htmp>
    <head>
          <title>Le pipotron</title>
    </head>
    <body>
          <h1>Le pipotron</h1>
          <?php echo($conclusion); ?>
    <body>
<html>
```

X) Compléter le programme pipotronQ10.php pour qu'il affiche une conclusion aléatoire sans problème de liaisons.

g) Les formulaires :

Nous allons maintenant demander à l'utilisateur le N° de la conclusion qu'il souhaite obtenir, ce numéro est un nombre de 8 chiffres, chaque chiffre servira d'indice pour choisir une variante d'une partie de la phrase, exemple :

N° de conclusion	Conclusion
00000000	*Avec la situation présente, il convient d'étudier toutes les solutions envisageables.*
12345678	*Considérant la crise qui est la nôtre, il serait intéressant d'imaginer la totalité des problématiques s'offrant à nous.*
99999999	*Tant que durera la baisse de confiance de ces derniers temps, il faut de toute urgence se remémorer certaines alternatives de bon sens.*

Pour cela il nous faut insérer dans la page un formulaire. Voir http://php.net/manual/fr/tutorial.forms.php.

Voici le code HTML du formulaire que nous allons utiliser :

```
<form method="post" action="pipotronQ11.php">
    <label>Entrez un nombre de 8 chiffres</label>
    <input name="conclusion_no" type="number" min="0" max ="99999999"/>
    <input type="submit" value="Générer" />
</form>
```

Explications :

- La 1ère ligne contient la balise <form> avec 2 attributs :
  - method : qui indique la méthode utilisée pour envoyer les données du formulaire au serveur. Il y en a 2 : GET et POST.
  - action : indique quel fichier doit être appelé quand on cliquera sur le bouton « Générer ».
- La 2ème ligne contient un <label>, c'est juste du texte à afficher.
- La 3ème ligne contient une entrée du formulaire, on indique son **nom** et son **type**, en option on peut indiquer les limites pour le type « number ».
- La 4ème ligne contient le bouton qui va provoquer l'envoie des données du formulaire au fichier spécifié plus haut (`action="pipotronQ11.php"`).

XI) Insérez ce code dans le fichier précédent juste après le titre <h1>…</h1>et enregistrez sous le nom pipotronQ11.php
Quel est le résultat ?

Ensuite il faut récupérer les valeurs du formulaire, elles se trouvent dans la variables $_POST.

XII) Avec la fonction var_dump(), ajouter une ligne en début de programme pour afficher cette variables,.
Quel est le résultat ?

Voici ce que nous devons faire maintenant :

- Récupérer la valeur transmise par le formulaire où créer une valeur si le formulaire est vide. On utilisera une variable $str_no.
- Ajouter des zéro à droite pour que cette $str_no contienne 8 caractères.
- Initialiser la variable $conclusion
- Pour chaque partie de la onclusion (il y en a 8) :
  - Utiliser le 1er caractère de $str_no pour choisir le texte de cette partie
  - Supprimer ce 1er caractère de $str_no
  - Ajouter à $conclusion la partie de conclusion sélectionnée.
- Adapter les liasons dans $conclusion.

Voici le pseudo-code correspondant :

```
DEBUT
      SI $_POST['conclusion_no'] est définie
      ALORS
            $str_no ← $_POST['conclusion_no']
      SINON
            $str_no ← "00000000"
      FIN SI

      TANT QUE longueur de $str_no < 8
            Ajouter le caractère '0' au début $str_no
      FIN TANT QUE

      $conclusion ← ""
      POUR $i = 0 à 7
            $j ← premier caractère de $str_no converti en nombre (de 0 à 9)
            Supprimer le 1er caractère de $str_no
            $conclusion ← $conclusion + $pipo[$i][$j]
      FIN POUR

      $conclusion ← adapterLiaison($conclusion)
FIN
```

Pour réaliser ce programme on pourra utiliser les fonctions suivantes :

- isset() : http://php.net/manual/fr/function.isset.php
- strlen() : http://php.net/manual/fr/function.strlen.php
- intval() : http://php.net/manual/fr/function.intval.php
- substr() : http://php.net/manual/fr/function.substr.php

XIII) Implémentez ce pseudo-code dans le fichier pipotronQ11.php et enregistrez le sous le nom pipotronQ13.php (pensez à modifier l'attribut « action » de la balise « form »). Testez.

Dans le même genre d'idée, on peut réaliser une page Web qui affiche un des 100 000 000 000 000 de poèmes de Raymond QUENEAU.

https://fr.wikipedia.org/wiki/Cent_mille_milliards_de_po%C3%A8mes

http://emusicale.free.fr/HISTOIRE_DES_ARTS/hda-litterature/QUENEAU-cent_mille_milliards_de_poemes/_cent_mille_milliards.php