

TP02 : navigation entre article

1^{ère} partie

Le but de la séance est de faire une navigation entre article puis entre dossier (catégorie). Les différentes étapes se complètent :

1. Afficher un article
2. Pouvoir changer dans l'url l'article vu
3. Ajouter un lien de navigation vers l'article suivant
4. Ajouter un lien de navigation vers l'article suivant, ce lien a pour texte le nom du fichier cible
5. Ajouter un lien de navigation vers l'article précédent
6. Ajouter un titre pour l'article (le nom du fichier)

sera fait dans la séance suivante

7. Pouvoir changer dans le code php le dossier exploré
8. Naviguer entre dossier : pouvoir changer dans l'url le dossier exploré, ajouter des liens de navigation (équivalent des étapes 3-4-5).

Le sujet est long. L'objectif pour cette séance est d'atteindre l'étape 5.

1. Afficher un article

Créez un dossier TP02 dans votre dossier php (qui est dans votre dossier www). Dans ce dossier, créez un fichier « tp02.php ».

N'oubliez pas la structure du document html à l'intérieur du fichier tp02.php, par exemple :

```
<?php
// genere ici le code html
?>
<!doctype html>
<html>

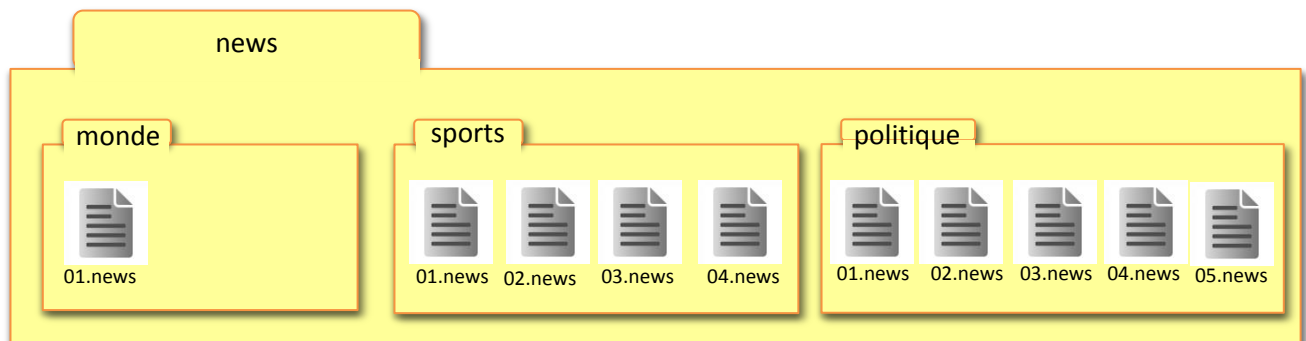
<head>
    <meta charset="utf-8">
    <title>date...</title>
</head>
<body>

<?php
// affichez ici le code html genere
?>

</body>
</html>
```

Ajoutez dans cette page tp02.php [la feuille de style TP02-style.css](#)

Récupérez l'archive [news.zip](#) qui contient un dossier news et des sous-dossiers :



Dans un tableau \$fichiers mémorisez tous les fichiers « .news » du dossier « news/politique ». Dans une variable \$indice définissez l'indice du fichier « .news » (l'article) à afficher. Testez si cette indice est bien dans l'intervalle du tableau \$fichiers :

- Si ce le cas, avec la fonction `file_get_contents` affichez le contenu de cet article dans la page dans une div de classe css « article » .
 - `file_get_contents($filename)`
 - Lecture de tout un fichier, retourne son contenu dans une chaîne
 - <http://www.php.net/manual/fr/function.file-get-contents.php>
- Sinon, affichez le message « article inexistant » (toujours dans une div de classe css « article »).

Essayer votre page en modifiant la valeur de \$indice. N'hésitez pas à mettre des valeurs farfelues comme -1 ; 2412 ; etc.

2. Pouvoir changer dans l'url l'article vu

Plutôt que de changer la variable \$indice dans le code php, passez la valeur dans l'Url (par méthode GET) dans une « variable » nommée indice.

L'url ressemblera à `http://www-mips.unice.fr/~votrelogin/php/tp02/tp02.php?indice=1`

Il faudra une valeur par défaut à \$indice, puis testez si `$_GET["indice"]` existe. Si c'est le cas, affectez à \$indice la valeur ainsi récupérée (en utilisant la fonction `intval` pour la convertir en entier).

Le reste de la page devrait être inchangé.

3. Ajouter un lien de navigation vers l'article suivant

Plutôt que de changer d'article en éditant l'adresse url, vous allez proposer un lien hypertexte (<a>) qui permet de passer à l'article suivant. Quand vous arriverez au dernier article, le suivant sera le premier.

Pour cela, utilisez l'opérateur modulo « % » sur le nombre de fichier.

Ainsi, si l'article (fichier) désigné par la variable indice dans l'url (ou par la valeur par défaut dans la variable php \$indice) existe, et s'il y a bien au moins un autre fichier, alors avant la <div class='article'> contenant l'article, ajoutez une div de classe css « suivant ». Dans cette div, placez un lien « article suivant » vers l'affichage de l'article suivant. Le href de ce lien sera semblable à "?indice=\$indice_suivant". (En effet, si l'adresse n'est pas précisée, l'adresse de la page courante sera ajoutée automatiquement).

4. Ajouter un lien de navigation vers l'article suivant, ce lien a pour texte le nom du fichier cible

L'amélioration sera ici de remplacer le texte du lien « article suivant » par le nom du fichier. Ce nom de fichier est contenu dans le tableau \$fichiers à l'indice calculé (nommé ici \$indice_suivant), mais il contient tout le chemin. Cette valeur ressemble à « news/politique/02.news ». Avec la fonction explode, découpez cette valeur en fonction des « / » et retenez que la dernière valeur pour le texte du lien

5. Ajouter un lien de navigation vers l'article précédent

Répétez les étapes 3 et 4, mais pour rajouter un lien vers l'article précédent. Ce lien ne sera ajouter que : si l'article désigné par l'indice existe et qu'il y ait au moins 2 autres fichiers. Ce lien sera ajouté entre le lien vers l'article suivant et la div de class « article ». Le lien vers l'article précédent sera dans une div de class css « precedent ».

Méfiez-vous de l'opérateur modulo « % » qui peut renvoyer une valeur négative. Si c'est le cas rajoutez la valeur du modulo (ici le nombre de fichier), vous resterez dans la même classe d'équivalence...

6. Ajouter un titre pour l'article

Une étape facile : ajoutez comme titre d'article (<h2>) le nom du fichier .news affiché. Faites comme pour le texte des liens suivant et précédent.

7. Pouvoir changer dans le code php le dossier exploré

Cette étape se traduira par peu de modification du code php. Plutôt que d'afficher les articles du même dossier, il serait intéressant de pouvoir changer de dossier. Pour cela :

- Stockez dans un tableau \$dossiers tous les dossiers contenu dans le dossier « news/ » (utilisez l'option GLOB_ONLYDIR)
- Définissez une variable \$indice_dossier pour désigner le dossier courant, initialisée à 0
- Utilisez une variable \$dossier initialisée à \$dossiers[\$indice_dossier]
- \$fichiers contiendra tous les fichiers .news contenus dans le dossier « \$dossier »

Essayez votre page avec les valeurs 0 puis 1 puis 2 pour \$indice_dossier. Vérifiez qu'avec le dossier « monde », qu'il n'y ait pas de liens de navigation (il n'y a qu'un article).

Créez un dossier « France » dans le dossier news. Re-essayez votre page avec les valeurs 0 ; 1 ; 2 et 3.

Créez deux fichiers .news dans ce dossier « France » et re-essayer à nouveau.

8. Naviguer entre dossier

Finalement, refaites les étapes 3-4-5 et 6, mais au niveau des dossiers. Les classes css sont :

- « dossiersuivant » pour la div qui contiendra le lien vers le dossier suivant
- « dossierprecedent » pour la div qui contiendra le lien vers le dossier précédent
- « dossiercourant » pour la div qui contiendra le nom du dossier « parcouru ».

Les liens vers les articles suivant et précédent devront être mis à jour pour préciser le dossier actuel. Les url ressembleront à :

<http://www-mips.unice.fr/~votrelogin/php/tp02/tp02.php?indice=2&dossier=1>

Les div devront être générés dans l'ordre suivant :

1. Eventuellement la div de class « dossiersuivant » (lien vers le dossier suivant)
2. Eventuellement la div de class « dossierprecedent »
3. La div de classe « dossiercourant » ou si le dossier n'existe pas une div de class « pasdedossier » et contenu un message « Ce dossier n'existe pas »
4. Eventuellement la div de class « suivant » (lien vers l'article suivant)
5. Eventuellement la div de class « precedent »
6. La div de classe « article » ou si l'article n'existe pas une div de class « article » et contenu un message « article inexistant »