

TP Ajax Php SQL (et JSON)

Recherche dynamique d'images dans une base de données

Présentation du projet :

Sur un serveur web, on dispose d'un dossier contenant des images de petites tailles (miniature - Angl : *thumbnail*)

Sur le même serveur, une base de données garde l'inventaire de ces images, avec une liste de mots clés pour les retrouver.

On veut réaliser une interface Web qui affiche les images qui correspondent à un mot clé.

Le mot clé est tapé dans un <input>, les images apparaissent au fur et à mesure que l'on écrit.

C'est un fonctionnement dynamique provoqué par Ajax.

Fichiers fournis :

Partie HTML et CSS :

Pour placer les images dans une <div> et pour obtenir qu'elles se placent de façon régulière, on utilise le principe des *flex-box* du CSS :

```
.deco {
    /*Comportement du contenu*/
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: space-around;
}
```

Plus d'info sur les flex-boxes : <https://www.alsacreation.com/tuto/lire/1493-css3-flexbox-layout-module.html>

On vous fournit aussi le code PHP de connexion au serveur. Vous devez élaborer la requête et l'envoi de la réponse au JavaScript.

METHODE 1 :

Le PHP fabrique le code HTML qui sera inséré dynamiquement dans la <div> par Ajax.

Par un *foreach*, on accumule dans une variable php le texte HTML :

```
$html = "";
foreach ($resul as $ligne)
{
    $html .= "<img src='/tpSqlAjaxImg/' . $ligne['fichier'] . '>";
}
echo $html;
```

- On note la concaténation avec cumul avec le « .= » et ensuite la concaténation simple avec le « . »
- /tpSqlAjaxImg/ est le dossier du serveur où sont stockés les fichiers images.
- \$resul est la variable qui contient le résultat de la requête SQL (A vous de jouer !)

TRAVAIL : Complétez les fichiers PHP et JS pour obtenir le bon fonctionnement de l'application.

Côté JS, l'action Ajax est déclenchée à chaque touche tapée (événement *keyup*)

METHODE 2 :

Cette fois, le PHP se contente d'envoyer le résultat complet de la requête SQL, sous forme de chaîne de caractères **JSON**¹:

```
echo json_encode($resul); // $resul est la variable qui contient le résultat de la requête SQL.
```

Le JavaScript va donc recevoir du JSON, qu'il convertira immédiatement en tableau JavaScript classique :

```
var liste = JSON.parse( e.target.responseText );
```

Après cette opération, **liste** contiendra un tableau à 2 dimensions : [numéro de ligne] ['fichier']

Il vous suffit ensuite de parcourir le tableau **liste** avec un FOR ou en utilisant l'itérateur **liste.forEach** (revoir le module opale http://78.122.136.24/opale/3-%20JavaScript%20-%20Les%20fonctions/co/d-Les_fonctions_Callback.html, au sous-titre « Un Callback de méthode forEach »).

Extrait de code JavaScript pour créer une image et l'insérer dans le HTML quand on connaît son nom et son emplacement :

```
let imag = new Image();  
imag.src = "/tpSqlAjaxImg/" + nomFichier;  
info.appendChild(imag);
```

Le ID de la <div> cible dans la page HTML Nom du dossier commun à toutes les images sur le serveur Variable contenant le nom du fichier image

Il faudra faire ce travail POUR CHAQUE nom d'image reçu.

TRAVAIL : Complétez les fichiers PHP et JS pour obtenir le bon fonctionnement de l'application. Côté JS, l'action Ajax est déclenchée à chaque touche tapée (événement *keyup*)

¹ **JSON** : JavaScript Object Notation : Format d'échange de données universel. Le tableau PHP n'est pas compatible avec le tableau JavaScript. Le passage par un codage/décodage JSON permet de faire communiquer les 2 mondes. https://fr.wikipedia.org/wiki/JavaScript_Object_Notation