

Construire un site dynamique

Partie 2



Sommaire



➤ Composer

- Définition
- Installation
- Commandes de base
- Composer.json et composer.lock
- L'autoload

➤ Pagination

- MySQL, Limit et Offset
- Paramètres d'URI
- Page précédente et page suivante.

composer



- Composer est un gestionnaire de packages pour PHP.
- Il permet de garder à jour les librairies PHP en utilisant un simple fichier .json et une invite de commande.
- Il utilise GIT pour récupérer les librairies et gère les dépendances automatiquement.

<https://getcomposer.org/>

composer

Il faut parfois pour certains packages installer aussi git.

<https://github.com/msysgit/msysgit>

Ensuite il faut ajouter les exécutables au PATH de windows:

- C:\msysgit\bin
- C:\msysgit\mingw\bin

composer



- Pour l'installer il faut modifier le "path" de windows et y ajouter le chemin vers l'interpréteur PHP.
- `php -r "readfile('https://getcomposer.org/installer');" | php`
- `php composer.phar -V` pour vérifier la version installée
- Pour connaître la liste des packages que l'on peut récupérer avec composer:
- <https://packagist.org/>

composer



- `php composer.phar search phpmailer` pour chercher si un package correspond à nos besoins (ici phpMailer).
- `php composer.phar require phpmailer/phpmailer` pour mettre à jour le fichier `composer.json` et télécharger le package.
- `php composer.phar install` pour installer les packages définis dans le fichier `composer.json`.
- `php composer.phar update` pour vérifier si des mises à jour sont disponibles.
- `php composer.phar self-update` pour mettre composer à jour.

composer.json

➤ Composer utilise un fichier composer.json pour connaître la liste des packages à récupérer.

```
{  
    "require":  
    {  
        "wf3/w": "*",  
        "setasign/fpdf": "1.7"  
    }  
}
```

Ensuite lancer `php composer.phar install` (ou `update`)

composer.json

- Le fichier composer.lock indique les packages installés, en précisant le hash précis de la version (hash de commit Git).
- Le fichier composer.lock indique aussi les dépendances des packages installés.

autoload



- composer génère aussi pour nous un fichier autoload.php qui permet d'inclure les classes automatiquement.
- `include_once './vendor/autoload.php';`

```
<?php
```

```
// autoload.php @generated by Composer
```

```
require_once __DIR__ . '/composer' . '/autoload_real.php';
```

```
Return ComposerAutoloaderInit6fb7c2ed2bf09543edcd36606319df5d::getLoader();
```

<http://jolicode.com/blog/composer-gerer-les-dependances-de-son-projet-simplement>

<http://jefaisvotresite.com/blog/article/composer>

Exercice pratique:

Installer composer et récupérer quelques packages.



Pagination



Savoir paginer ses données permet d'éviter d'afficher des trop grands tableaux dans la page:

- Le temps de chargement est fortement amélioré.
- On évite les difficultés de lecture liées à l'utilisation des l'ascenseur.
- On évite les bugs d'affichage.
- Le résultat est plus professionnel.

MySQL LIMIT



La fonction LIMIT permet de récupérer un nombre limité d'enregistrements, et de définir à partir de quel enregistrement on débute. Soit la requête:

```
SELECT nomEleve, prenomEleve from eleve LIMIT 0, 10
```

Cette requête récupère l'ensemble des élèves de la table, mais ne retournera que les 10 premiers enregistrements.

```
SELECT nomEleve, prenomEleve from eleve LIMIT 10, 10
```

Cette requête récupère l'ensemble des élèves de la table, mais ne retournera que les enregistrements 11 à 21.

Paramètres d'URI



Il est possible de se servir de l'url pour passer des paramètres au script PHP. Soit l'url:

<http://localhost/wf3/pagination.php?page=1>

On peut récupérer le numéro de page dans le tableau \$_GET

```
<?php
    echo 'numéro de page: '.$_GET['page'];
?>
```

Paramètres d'URI



<http://localhost/wf3/pagination.php?page=1>

```
<?php
    echo 'numéro de page: ' . $_GET['page'];
    // en connaissant le numéro de page et le nombre d'éléments à
    //afficher on peut créer dynamiquement les paramètres de
    pagination
    $parPage = 10;
    $debut = ...
    $req = "SELECT * from INSEE LIMIT ".$debut.", ".$parPage;
    echo $req;
?>
```

Pages précédentes et suivantes



<http://localhost/wf3/pagination.php?page=1>

```
<?php
    $pageActuelle = $_GET['page'];
    if ($pageActuelle > 1)
    {
        echo '<a href="pagination.php?page=' .
            ($pageActuelle - 1) . '">Précédent</a>';
    }
    echo '<a href="pagination.php?page=' .
        ($pageActuelle + 1) . '">Suivant</a>';
?>
```

Exercice pratique:
Afficher la table INSEE dans un tableau
HTML de 25 enregistrements par page.



Exercices pratiques

