



Les bases du Développement web

Table des matières

Présentation.....	4
HTML.....	5
CSS.....	7
Javascript.....	8
PHP.....	9
MySQL.....	10
Git.....	12
Pour finir.....	13
Bibliographie.....	13

Présentation

Le développement web consiste à développer, créer, une page web (site statique), un site internet (dynamique, qui s'adapte suivant les requêtes de l'utilisateur) ou une application web. On fournit au développeur les spécifications précises des besoins (fonctionnalités désirées, apparence et couleurs) et le développeur transforme cela en code.

Pour faire du développement web, il faut connaître les bases du fonctionnement d'internet et d'un réseau informatique (protocole HTTPS, requêtes POST et GET, etc). Il faut aussi savoir comment marche et se configure un serveur web (Apache ou Nginx) et il faut également maîtriser plusieurs langages.

Les principaux sont le HTML, c'est vraiment la base, avec le CSS. Pour aller plus loin, il faudra apprendre le PHP ainsi que le Javascript pour pouvoir créer un véritable site internet ou une application web.

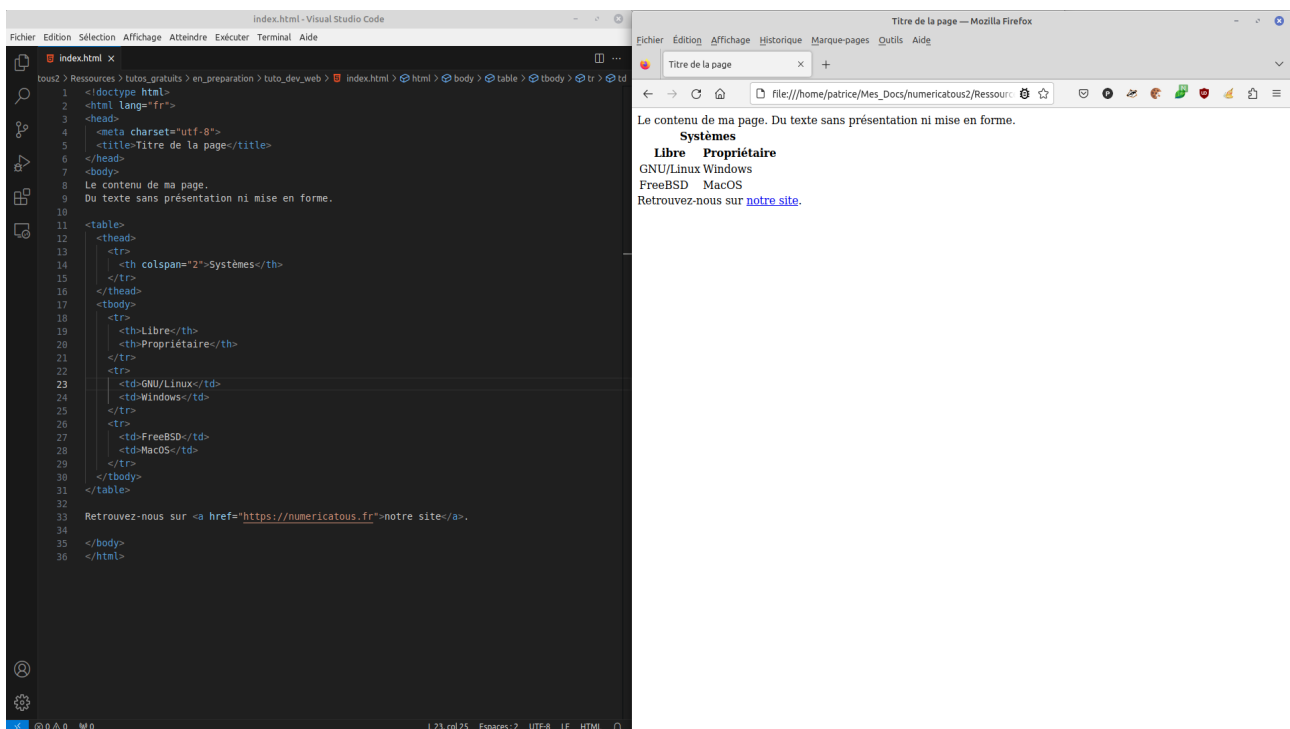
Côté outils, il faudra un serveur web, en local (installé sur votre ordinateur, via un outil de virtualisation comme VirtualBox) ou distant, pour héberger votre code, ainsi qu'un éditeur de code (comme Visual Studio Code). Il faut bien sûr un navigateur web (comme Firefox) pour regarder ce que vous faites. Pour aller plus loin, vous aurez besoin d'un gestionnaire de base de données (MySQL, avec phpmyadmin). Éventuellement un client FTP pour envoyer votre code sur le serveur, et Git pour faire du versionning.

HTML

Le HTML est un langage de balise qui sert à créer l'ossature de la page, la hiérarchisation des différents éléments et les hyperliens, les liens entre plusieurs documents. C'est donc la base et c'est par lui qu'il faut commencer pour développer son site web.

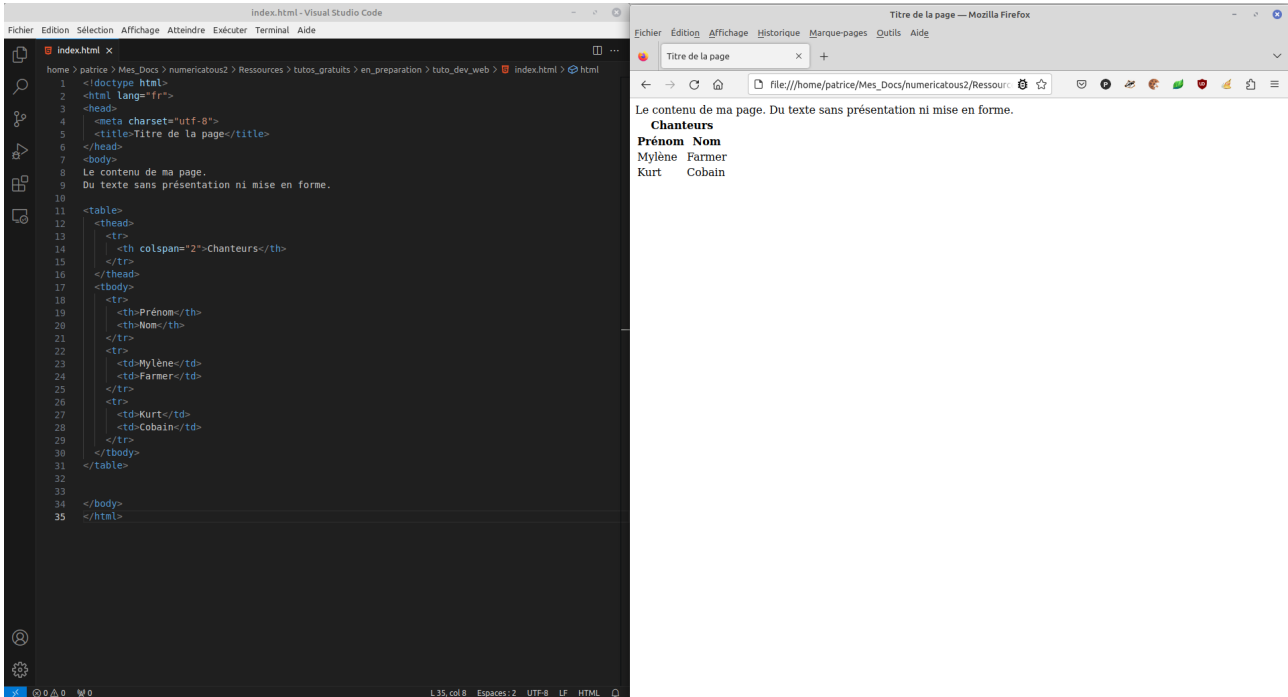
Voici le squelette d'une page web, en HTML. A gauche, c'est le code dans un éditeur de code (Visual Studio Code) et à droite, le rendu dans le navigateur web Firefox.

Pour afficher une page en HTML (même avec du CSS ou du Javascript dedans), pas besoin de serveur web. Faites un simple clic droit dans votre explorateur de fichier sur votre fichier .html puis « Ouvrir avec le navigateur web ». Ce sont des langages qui s'exécutent côté client, c'est-à-dire dans le navigateur.



On voit l'ossature du site, et les balises indispensables pour une page web comme le doctype, qui permet de définir que la page est en HTML. Les balises HTML qui encadrent le contenu, les balises HEAD qui encadrent des balises META qui permettent de donner des informations sur le contenu de cette page (mais qui n'affichent rien sur la page). Enfin, les balises BODY qui elles contiennent le contenu de la page, ce qui est affiché dans le navigateur.

Ajoutons un peu de contenu avec un tableau et en dessous un lien. On a ici le titre du tableau, des titres de colonnes et le contenu du tableau. Nous n'avons que du HTML, comme vous le voyez, aucune présentation ni style, uniquement l'ossature de notre page.

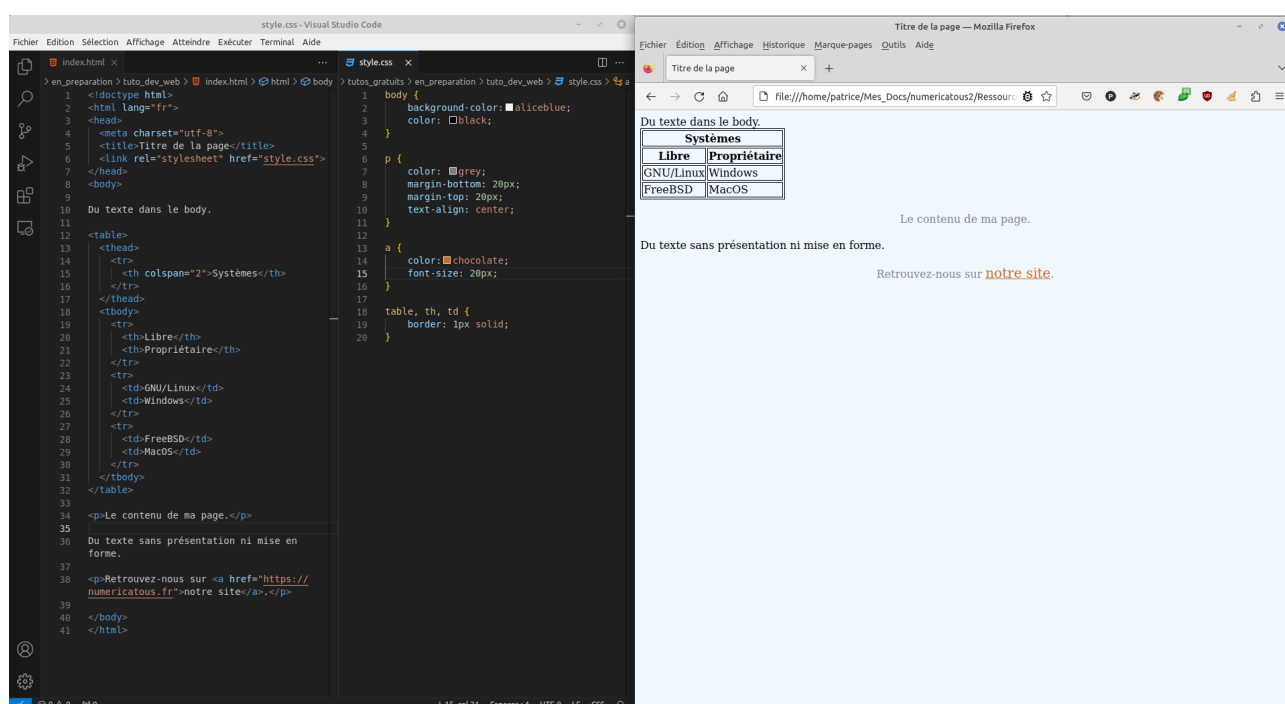


C'est un peu austère comme ceci, mais le HTML ne permet pas de faire plus. Pour cela, il faut passer par les CSS.

CSS

Le CSS (Cascading Style Sheet ou feuille de style en cascade) est un langage du web permettant de décrire la présentation d'un document HTML. Il permet d'ajouter des éléments de style aux balises HTML pour ajouter une couleur, une taille de police d'écriture, un espacement, un type d'alignement. On peut faire énormément de chose aujourd'hui avec du CSS.

Découvrons ici les bases pour améliorer un peu le rendu de notre page HTML. Pour ajouter une feuille de style à une page HTML, il faut la déclarer entre les balises HEAD et donner son chemin complet.

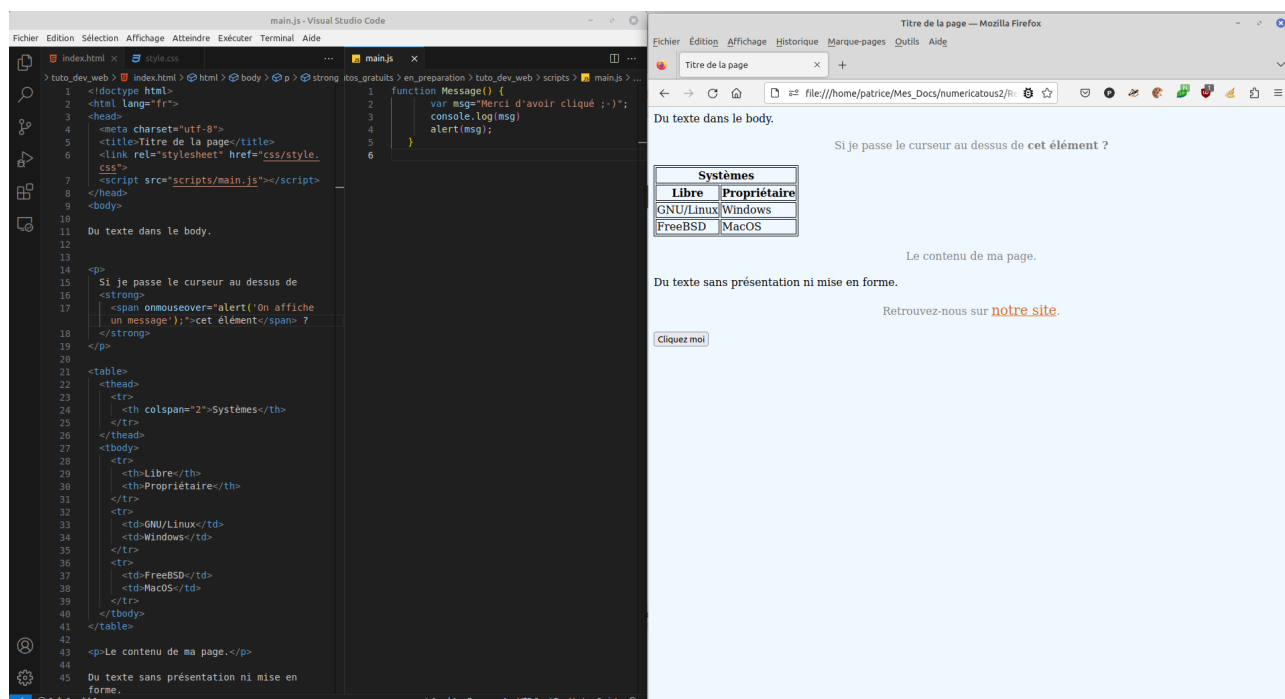


Pour se simplifier la vie, il existe le framework [Bootstrap](#), qui n'est pas le seul, mais certainement le plus utilisé, qui permet d'avoir de nombreux outils CSS pour créer des interfaces soignées et responsive. C'est une grande collection de classes CSS prêtes à l'emploi pour créer des menus, des tableaux et tout ce qu'il faut dans une page web.

Javascript

Le Javascript va permettre d’animer notre page, d’ajouter un peu de dynamisme en réagissant aux actions du visiteur. On va pouvoir agir sur les différents éléments de notre page pour les modifier, par exemple au survol de la souris ou lors d’un clic.

On peut inclure le Javascript directement dans notre page HTML ou dans un fichier séparé, comme pour la feuille de style CSS. C’est mieux pour garder un code propre et organisé d’avoir un dossier pour chaque type d’élément.



Depuis quelques années, de nombreux frameworks Javascript ont vu le jour. Un framework permet de créer tout un projet web avec des fonctions et des outils prédéfinis qui permettent de gérer un grand nombre de cas d’utilisation. Cela permet de faciliter grandement la vie du développeur.

Mais bien sûr, chaque framework a ses spécificités et demande un certain temps d’apprentissage. Il faut d’abord se familiariser avec le langage avant de débiter dans l’utilisation d’un framework.

En voici quelques un, il en existe d’autres :

[ReactJS](#), [Angular](#), [VueJS](#) ou encore [MeteorJS](#).

PHP

Le PHP, contrairement aux langages vu précédemment, est un langage de programmation serveur. Il est donc nécessaire d'avoir un serveur web (Apache ou Nginx) pour pouvoir créer et travailler sur du code PHP. Il permet de faire divers types d'opérations (mathématiques, logiques, etc), il gère différents types de boucle (if .. else), on peut travailler sur des fichiers, des tableaux ainsi que des bases de données évidemment.

On peut inclure du PHP directement dans une page HTML ou alors inclure un fichier ne contenant que PHP. Pour inclure du PHP dans une page HTML, il faut indiquer que le code est du PHP, ce qui se fait à l'aide des balises `<?php` et `?>` comme ceci :

```
<?php
    $f = fopen("/sys/class/thermal/thermal_zone0/temp", "r");
    $temp = fgets($f);
    echo round($temp / 1000) . 'C°';
    fclose($f);
?>
```

Pour inclure un fichier PHP, on va utiliser un mot clé, *include* ou *require*, et indiquer le chemin (relatif ou absolu) du fichier.

Ici, PHP va nous permettre de lancer une commande du système et de faire un test, une comparaison, sur le résultat obtenu :

```
<?php
    $sshs = exec("systemctl is-active sshd");
    if(strpos($sshs, "inactive") === false){
        echo "Actif";
    } else{
        echo "Inactif";
    }
?>
```

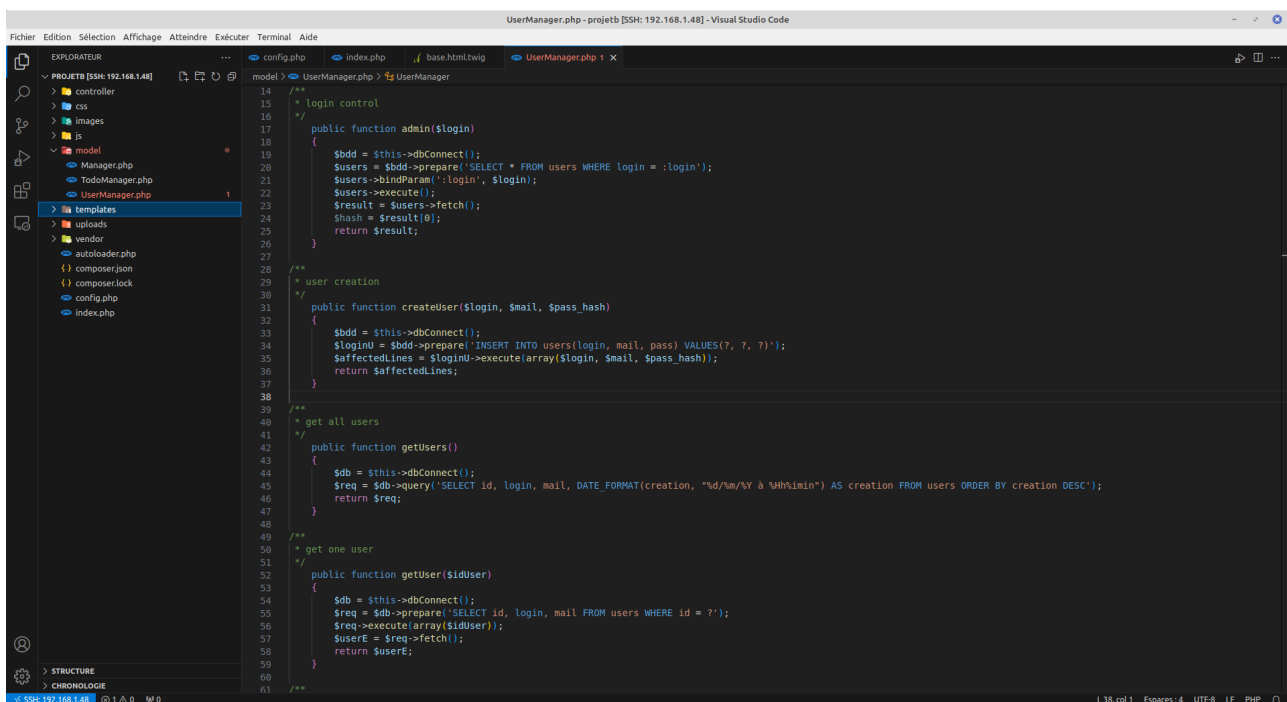
PHP dispose lui aussi de divers framework. Le plus connu est certainement [Symfony](#) mais il en existe d'autres, comme [Laravel](#), [CodeIgniter](#), [CakePHP](#).

MySQL

MySQL est un système de gestion de base de données relationnelles. On s'en sert principalement pour stocker des informations pour pouvoir les utiliser et les traiter, avec le langage PHP. Ces informations sont enregistrées sur le serveur. Il faut pour cela créer une base de données puis des tables qui accepteront chacune un certain type de données (du texte, des chiffres, des dates). Pour gérer les bases de données, on peut utiliser Phpmyadmin ou Dbeaver.

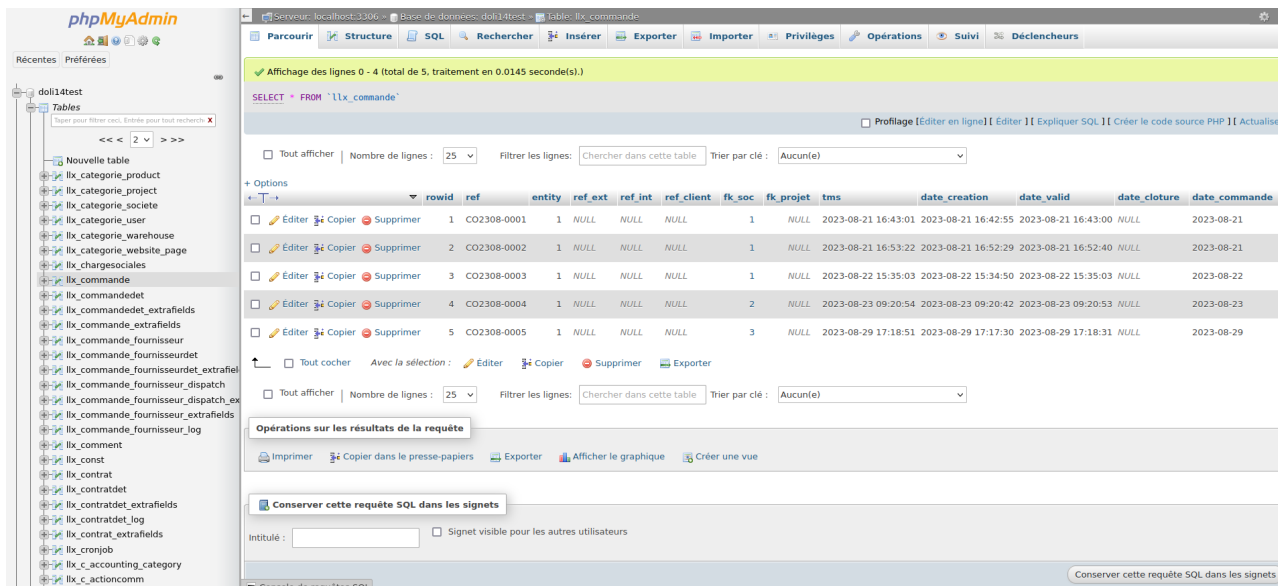
Par exemple, cette requête SQL permet de récupérer l'id, le login, le mail ainsi que la date de création d'un utilisateur dans la table « users » et de trier le résultat par date.

```
$db->query('SELECT id, login, mail, DATE_FORMAT(creation, "%d/%m/%Y à %Hh%imin") AS creation FROM users ORDER BY creation DESC')
```



```
14 /**
15  * login control
16  */
17 public function admin($login)
18 {
19     $bdd = $this->dbConnect();
20     $users = $bdd->prepare('SELECT * FROM users WHERE login = :login');
21     $users->bindParam(':login', $login);
22     $users->execute();
23     $result = $users->fetch();
24     $pass = $result[0];
25     return $result;
26 }
27
28 /**
29  * user creation
30  */
31 public function createUser($login, $mail, $pass_hash)
32 {
33     $bdd = $this->dbConnect();
34     $loginU = $bdd->prepare('INSERT INTO users(login, mail, pass) VALUES(?, ?, ?)');
35     $affectedLines = $loginU->execute(array($login, $mail, $pass_hash));
36     return $affectedLines;
37 }
38
39 /**
40  * get all users
41  */
42 public function getUsers()
43 {
44     $db = $this->dbConnect();
45     $req = $db->query('SELECT id, login, mail, DATE_FORMAT(creation, "%d/%m/%Y à %Hh%imin") AS creation FROM users ORDER BY creation DESC');
46     return $req;
47 }
48
49 /**
50  * get one user
51  */
52 public function getUser($idUser)
53 {
54     $db = $this->dbConnect();
55     $req = $db->prepare('SELECT id, login, mail FROM users WHERE id = ?');
56     $req->execute(array($idUser));
57     $userE = $req->fetch();
58     return $userE;
59 }
60
61 /**
```

Voici un exemple de base de données dans Phpmyadmin. On voit une table (llx_commande), qui fait partie d'une base de données (doli14test), et les différentes informations stockées dedans :



A l'aide d'une requête SQL réalisée via PHP, on va pouvoir ajouter des informations dans cette table, en supprimer ou les modifier.

Par exemple, puisqu'ici on est dans la table « commande » de Dolibarr, on pourra faire une requête qui mettra à jour la commande lorsqu'elle sera validée, ou si le contenu de la commande est modifié.

Git

[Git](#) est un système de gestion de version décentralisé. Il permet de suivre les modifications de code sur un projet. C'est un outil incontournable pour le développement.

Vous travaillez sur un projet, vous arrivez à une version stable, vous allez publier cette version sur git, cela créera un instantané de votre code. Puis, vous pourrez créer une branche (qui vous permettra de travailler sur de nouvelles modifications de votre code, sans risquer de casser quelque chose. Lorsque ce code sera prêt et fonctionnel, vous pourrez publier cette nouvelle branche. Si tout est bon dans le code, vous pourrez ensuite inclure cette branche dans votre code de départ.

Git indexe tous vos fichiers et tous les changements faits entre deux publications. Vous pouvez ainsi, si besoin, revenir en arrière, ou inclure une branche dans une autre. C'est vraiment très pratique et très utile. De plus cela permet aussi de travailler à plusieurs sur un même projet.

Pour utiliser Git, il vous faudra un compte sur [Gitlab](#) (ou [Github](#), qui appartient à Microsoft). Ensuite, vous pourrez créer un nouveau projet et envoyer votre code dessus.

Typiquement, les principales commandes sont :

Git init → *initie un dépôt git en local.*

Git add . → *cela permet d'inclure tous vos fichiers dans le projet.*

Git commit -m « explication des changements » → *vous faites un commit quand vous êtes prêt à envoyer votre code.*

Git push origin votre_branche → *vous publiez votre nouveau code sur Git.*

Quelques commandes utiles (il y en a beaucoup d'autres) :

Git checkout nom_branche → *pour changer de branche.*

Git checkout -b nom_branche → *pour créer une nouvelle branche.*

Git pull origin nom_branche → *pour mettre à jour en local votre branche.*

Git stash → *pour annuler votre dernier commit.*

Git fetch → *pour récupérer les infos de votre projet.*

Pour finir

Voilà pour cette rapide découverte du développement web. Les bases donc, c'est le HTML et CSS, des langages clients, pour démarrer. Ensuite, le PHP, indispensable. Ce langage serveur est très puissant et permet de faire beaucoup de choses. Et le Javascript, pour la partie dynamique.

Lorsque vous maîtriserez un peu tout cela, vous pourrez commencer à regarder du côté des frameworks PHP et Javascript. Ils permettent vraiment de gagner du temps dans le développement et de se concentrer sur son projet, mais il faut apprendre à les utiliser correctement.

Bibliographie

https://fr.wikipedia.org/wiki/D%C3%A9veloppeur_web
<https://www.grandecolenumerique.fr/metiers-davenir/les-metiers-du-developpement/developpement-web-et-mobile>
<https://developer.mozilla.org/fr/docs/Learn>
https://fr.wikipedia.org/wiki/Framework_Web
https://developer.mozilla.org/fr/docs/Learn/Server-side/First_steps/Web_frameworks
<https://www.codeur.com/blog/frameworks-php/>
<https://www.codeur.com/blog/choisir-framework-javascript/>
<https://www.hostinger.fr/tutoriels/outils-developpement-web>
<https://kinsta.com/fr/blog/outils-developpement-web/>
<https://www.blogdumoderateur.com/outils-developpeurs-web-debutants/>
https://developer.mozilla.org/fr/docs/Learn/Tools_and_testing/GitHub
<https://www.atlassian.com/fr/git/tutorials/what-is-git>
<https://perso.liris.cnrs.fr/pierre-antoine.champin/enseignement/intro-git/>
<https://datascientest.com/git-tutoriel-sur-le-systeme-de-controle-de-versions-du-developpeur>

Ce document est publié sous les termes de la licence Attribution-NonCommercial-ShareAlike 4.0 International. Pour voir une copie de cette licence, rendez-vous sur :

<http://creativecommons.org/licenses/by-nc-sa/4.0/>