



Les bases du Développement web

Sommaire

Introduction

Définitions

Le développement web

- langages
- outils

Les étapes de la création

- réflexion
- mise en œuvre
- création
- mise en place

Aller plus loin

- frameworks
- outils
- CMS

Pour finir

Bibliographie

Licence

Introduction

Le développement web est un métier intéressant qui requiert diverses compétences. Mais en quoi consiste-t-il précisément ? Que faut-il connaître, comment s'y préparer et se former ?

C'est ce que nous allons voir ici. Vous n'allez pas apprendre à développer dans tel ou tel langage mais vous aurez les bases pour démarrer. Ce tutoriel va vous donner un aperçut de ce métier.

Développer pour le web requiert de connaître comme il faut le fonctionnement d'internet, des protocoles réseaux, d'un système client / serveur.

Bien évidemment, il faut savoir développer, et ce dans plusieurs langages différents. Maîtriser une seule technologie ne suffit pas.

Il faut également savoir utiliser les bons outils pour être productif et faciliter son travail. Un IDE complet mais pas trop compliqué pourra grandement vous faciliter la tâche lors de l'apprentissage d'un langage par exemple.

```

43 <body <?php body_content()></div>
44 <div id="fb-root"></div>
45 <script>(function(d, s, id) {
46   var js, fjs = d.getElementsByTagName(s)[0];
47   if (d.getElementById(id)) return;
48   js = d.createElement(s); js.id = id;
49   js.src = "//connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.6&appId=284614643628711";
50   fjs.parentNode.insertBefore(js, fjs);
51 })(document, 'script', 'facebook-jssdk');</script>
52 <div id="page" class="site">
53   <a class="skip-link screen-reader-text" href="#content"><?php esc_html_e( 'Skip to content', 'wordpress' ); ?></a>
54   <header id="masthead" class="site-header" role="banner">
55     <div class="site-branding">
56       <div class="nav-btn pull-left">
57         <?php if(is_home() && $xpanel['homepage-style'] == 1) { ?>
58           <a href="#" id="openMenu"><i class="fa fa-bars fa-3x"></i></a>
59         <?php } else { ?>
60           <a href="#" id="openMenu2"><i class="fa fa-bars fa-3x"></i></a>
61         <?php } ?>
62       </div>
63       <div class="logo pull-left">
64         <a href="<?php echo esc_url( home_url() ) ?>">
65           
67     </div>
68     <div class="search-box hidden-xs hidden-sm pull-left ml-10">
69       <?php get_search_form(); ?>
70     </div>
71     <div class="submit-btn hidden-xs hidden-sm pull-left ml-10">
72       <a href="<?php echo get_page_link($xpanel['submit-link']) ?>" class="header-submit-btn"><i class="fa fa-search fa-3x"></i></a>
73     </div>
74     <div class="user-info pull-right mr-10">
75       <?php
76       if ( is_user_logged_in() ) {

```

Définitions

Internet est un réseau informatique. C'est un réseau de réseau, décentralisé, composé de millions de réseaux publics et privés. Basé sur plusieurs protocoles standardisés, il permet d'échanger des données. Via ces protocoles, des services comme le courrier électronique ou encore le transfert de fichiers en pair à pair ont été créés.

Le **world wide web** est un système hypertexte public. Basé sur internet, il permet, via un navigateur web, de consulter des sites internet.

Le protocole **HTTP** (HyperText Transfer Protocol) a été développé pour le world wide web. C'est un protocole (couche application) de communication client/serveur. Un navigateur internet est un client HTTP.

Le protocole **TCP/IP** (Transmission Control Protocol / Internet Protocol) est l'ensemble des protocoles qui servent pour le transfert de données via internet.

Une **URL** (Uniform Resource Locator) est une adresse web, qui permet d'identifier précisément une ressource sur le world wide web. Elle se présente comme ceci, et commence toujours par le protocole utilisé, ici HTTPS, donc, sécurisé :

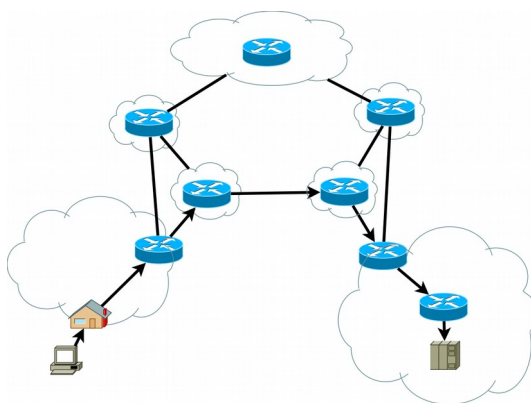
<https://monchernumerique.fr>

Le **FTP** (File Transfer Protocol) est un protocole client/serveur destiné au partage de fichiers entre ordinateurs, et pour envoyer les fichiers de son site sur un serveur web.

Un environnement **client/serveur** désigne un système de communication entre plusieurs programmes. L'un envoie des requêtes, c'est le client, l'autre y répond, c'est le serveur. Le client peut être un ordinateur personnel, une machine virtuelle... Le serveur est généralement un système dédié et très puissant, souvent avec Apache.

MySQL est un système de gestion de bases de données relationnelles, très utilisé dans le développement web. Une alternative totalement libre et compatible est MariaDB.

Voici maintenant un schéma (très simplifié) de fonctionnement du web. Un ordinateur client, à gauche, fait une requête. Elle passe par sa box internet, puis transite via plusieurs serveurs sur le web pour atteindre sa cible, un serveur web. Celui-ci va traiter sa demande et l'a lui renvoyer.



Le développement web

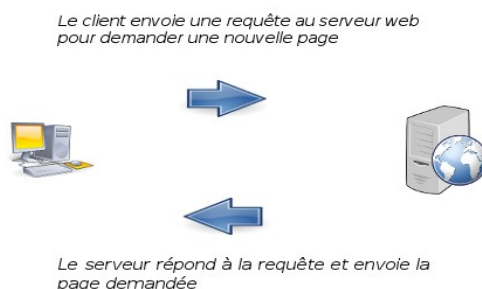
Le développement web est le fait de créer des programmes pour le web, c'est à dire accessible via un navigateur web. Que se soit une simple page web ou un service en ligne.

Pour cela, il faut connaître le fonctionnement d'internet et des réseaux informatiques. Il faut évidemment maîtriser plusieurs langages de programmation. Et disposer des bons outils également, c'est important pour être efficace.

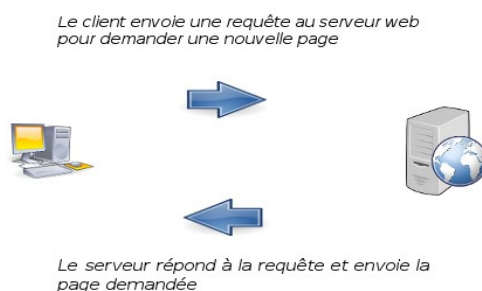
Le développement côté client n'est pas très compliqué. Il s'agit généralement de simples pages web en HTML et CSS que le serveur envoie lorsque l'on clique sur un lien.

Mais si l'on souhaite développer des applications et pages plus complexes, il va falloir maîtriser d'autres techniques. Le client fait une requête au serveur qui traitera celle-ci (un formulaire, une requête sur une base de données) à l'aide de PHP généralement, puis vous renverra le résultat sous forme de page HTML pour que votre navigateur puisse l'afficher.

Site statique. Le fonctionnement d'un site statique est simple. Vous êtes sur le site, vous cliquez sur un lien vers une autre page. Cela envoie une requête du client, le navigateur, vers le serveur qui héberge le site. Ce lien pointe vers une autre page en HTML, et donc le serveur vous envoie simplement la page demandée.



Site dynamique. Sur un site dynamique, ce n'est pas aussi simple. Vous êtes sur le site et vous cliquez sur un lien vers une autre page. Là, le serveur reçoit la requête du client, en PHP, la traite (requête sur la base de données, génération de la page demandée), et renvoie la page demandée spécialement pour le client, qu'il a générée en HTML.



Langages

Pour le développement web, divers langages sont disponibles et nécessaires. Voici les principaux, mais il en existe d'autres.

- Le premier d'entre eux est le langage **HTML** (HyperText Markup Language). C'est un langage de balises permettant de représenter des pages web et de créer des hyperliens. Il est la base du web.
- Ensuite, afin de pouvoir mettre en page correctement ces pages web, on utilise les **CSS** (Cascading Style Sheets), les feuilles de style en cascade. Elles permettent de séparer le contenu de sa présentation.
- Le **Javascript** est un langage permettant de créer des scripts afin de rendre une page web dynamique et d'interagir avec le visiteur via un formulaire par exemple ou des animations.
- Le **PHP** est un langage de programmation web côté serveur permettant de créer des sites web dynamique (avec forum, espaces membres...) ainsi que des applications web. Il s'utilise avec la base de données **MySQL** principalement.

Voici une page web standard en HTML. On y voit diverses balises (mots encadrés de chevrons), comme `<head>` qui représente l'en-tête de la page, `<title>` pour le titre. Certaines de ces balises possèdent des attributs, qui permettent de préciser certaines valeurs de la balise, comme par exemple `<meta charset= »UTF-8 »>`. Le *charset* précise l'encodage, la façon dont le fichier est enregistré.

La balise `<!DOCTYPE html>` est indispensable car elle précise le langage de la page. `<html>` est la balise principale qui englobe toute la page. On voit la balise fermante `</html>` après `</body>` qui elle représente le corps de la page.

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de ma page</title>
</head>
<body>
Hello World !
Texte de la page
</body>
</html>
```

Outils

Voici les outils de base pour la création d'un site web. Notez que pour démarrer la création de page HTML, un simple éditeur de texte comme Notepad est suffisant.

[Visual Code Studio](#), éditeur de code.

[FileZilla](#), client FTP pour envoyer ses pages sur le serveur web.

[The Gimp](#), logiciel de retouche photo, de création d'image (pour le logo, la favicon et les illustrations).

[Github](#), gestionnaire de code source en ligne.

Voici comment se présente l'éditeur Visual Code Studio. Il est complet, adaptable et dispose de très nombreux plugins pour ajouter des fonctionnalités.

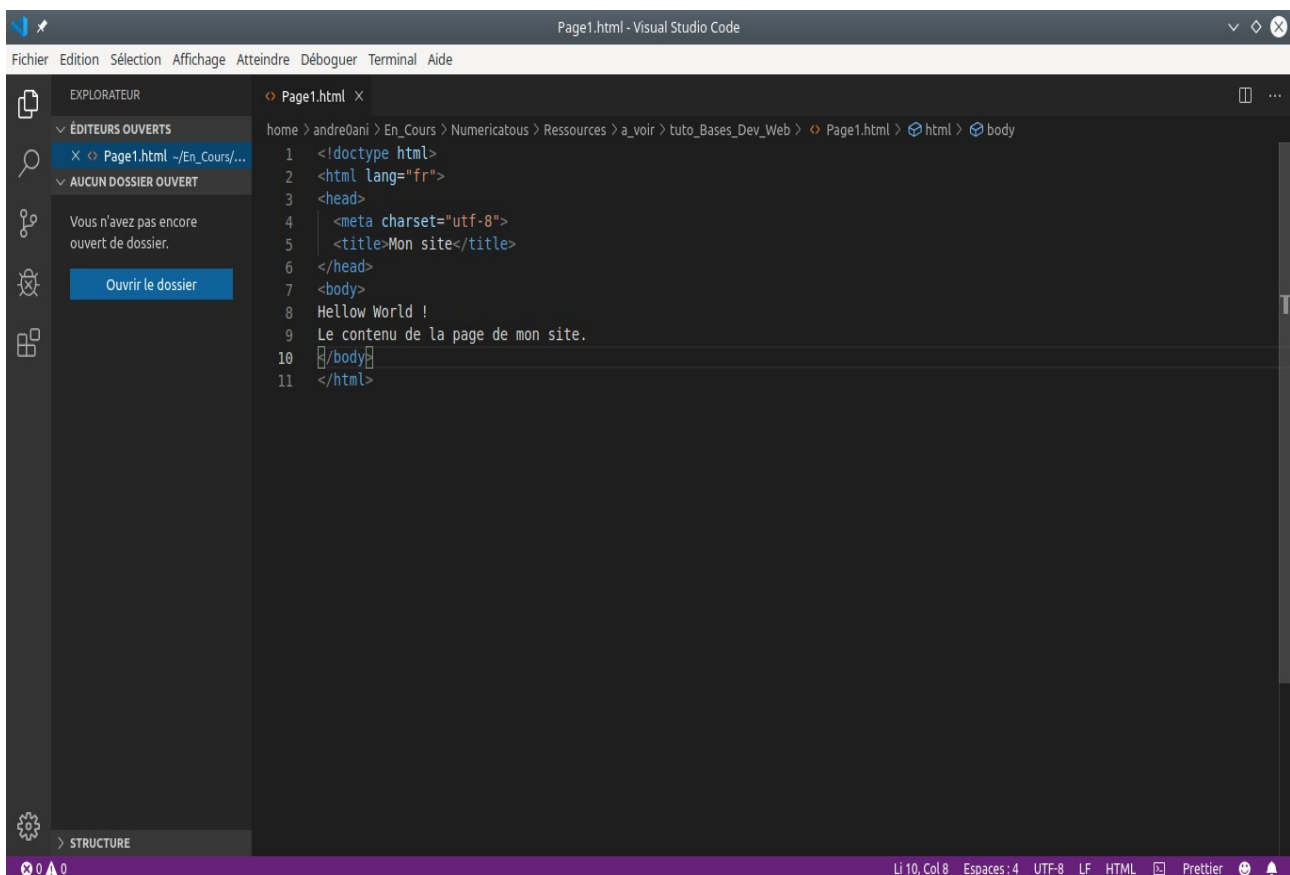


Illustration 1: Présentation de l'interface de Visual Code Studio

Les étapes de la création

Réflexion : Étudier précisément ses besoins et objectifs.

Mise en œuvre : Choisir les technologies à utiliser, un hébergeur, un nom de domaine.

Création : Créer les pages du site.

Mise en place : Mise en ligne sur l'hébergeur et référencement.

Réflexion

Avant de commencer à créer sa première page HTML, il faut d'abord réfléchir et se poser quelques questions :

Quels sont les besoins ? Quel est l'objectif du site, et sa cible ? Quels sont les services souhaités ? De quelle technologie aura-t-on besoin ? Quels sont les moyens à notre disposition ?

Il faut bien définir tout cela afin que le projet soit clair.

Savoir si l'on veut un site statique, simple, ou au contraire un site utilisant une base de données, des scripts PHP / Javascript et diverses fonctionnalités plus avancées.

Est-ce juste une petite vitrine pour présenter une activité de loisir ou est-ce un site professionnel ? Quel est le public visé ?

Cette étape ne doit pas être négligée, car c'est de cette étude que découlera tout le reste.

Mise en œuvre

Une fois décidé des besoins, des objectifs et des moyens, il va falloir choisir les technologies à utiliser pour réaliser cela. Il faudra ensuite chercher un hébergeur. L'hébergeur est là pour stocker votre site web, les pages, les documents et la base de données également, sur ses serveurs. Il en existe des gratuits et des payants.

Pour choisir, il faut comparer ce qu'ils proposent, notamment la place qui vous est octroyée pour votre site (de quelques mégas à plusieurs gigas), les technologies proposées (PHP, compte mail, accès FTP, panneau d'administration).

Pour un petit site, un hébergeur gratuit est bien souvent suffisant. Mais pour un site plus conséquent ou à caractère professionnel, un hébergeur payant est recommandé. Ils proposent généralement divers services, comme des comptes mails, des sous-domaines, la création de newsletter, un nom de domaine inclus, plusieurs bases de données ainsi que des systèmes de statistiques pour surveiller le nombre de visiteurs.

Voici une liste d' [hébergeurs](#) gratuits et payants.

Il faut réfléchir ensuite à son nom de domaine. C'est l'adresse que l'on tape dans le navigateur internet pour accéder au site. Avec un hébergeur gratuit, le nom de domaine que l'on choisi est souvent suivi du nom de son hébergeur. Mais avec un hébergeur payant, on a son propre nom de domaine, ce qui fait plus « professionnel » et qui est également important pour le référencement. Ce nom de domaine est uniquement loué, pour une certaine période, auprès de l'AFNIC (Association Française pour le Nommage Internet en Coopération), qui est le « registrar » ou bureau d'enregistrement en Français, des noms de domaines de premier niveaux en .fr.

Création



Illustration 2: Logo CSS et HTML

Une fois que tout les préparatifs sont terminés et que l'on sait précisément ce que l'on souhaite obtenir, on peut commencer la création des pages du site. Voilà une page HTML de base créée dans l'éditeur Visual Code Studio. Comme vous le voyez, il ressemble à un éditeur de texte :

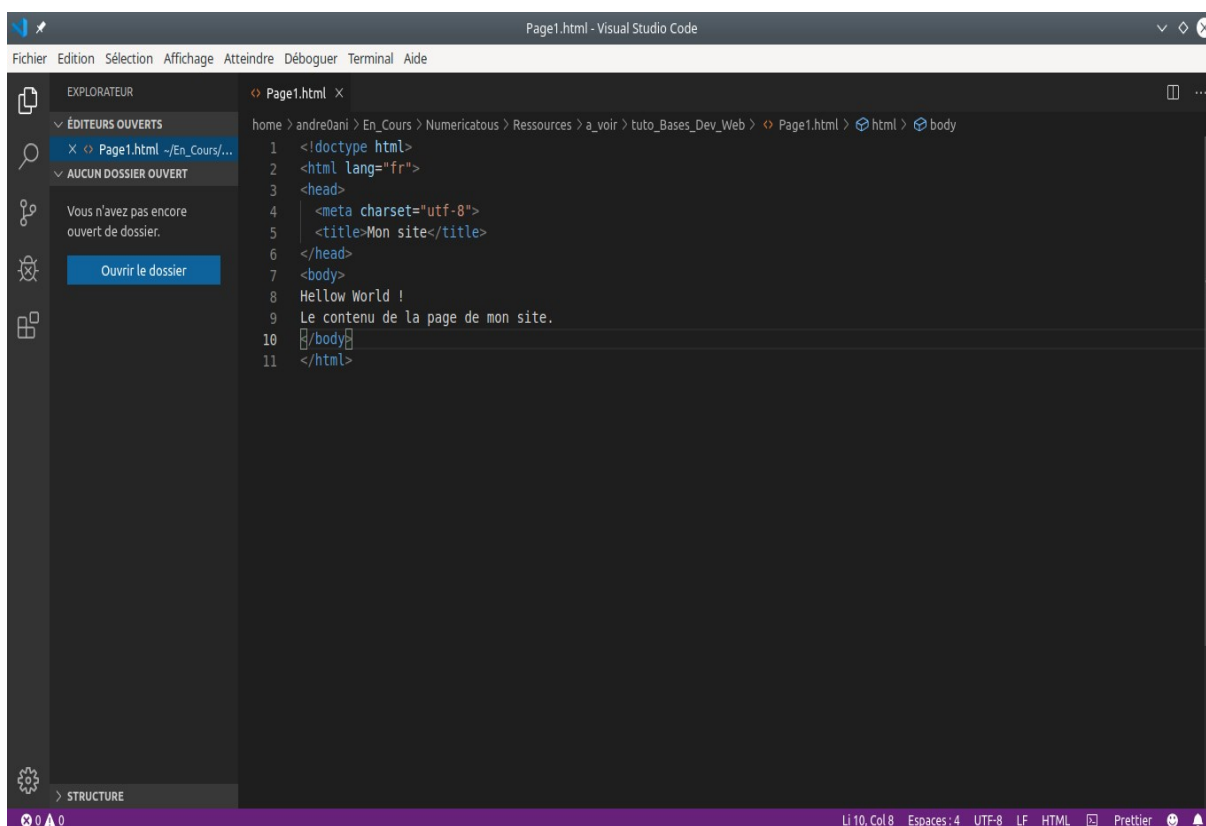


Illustration 3: Interface de l'éditeur Visual Code Studio

L'étape de création est très importante. Il faut bien étudier la présentation du site, la navigation, les couleurs et la taille du texte, tout le design donc. Il faut que le site paraisse clair, ordonné, que les textes soient lisibles et la navigation simple.

Dans l'éditeur, on tape son texte, on choisit sa police et sa taille, on place ses images comme on le souhaite, on crée les liens entre les diverses pages. Il faut que tout cela paraisse homogène et cohérent.

N'oubliez pas de créer une page de contact afin que les visiteurs puissent vous écrire pour vous faire part de leurs remarques, ou vous poser des questions.

Un aspect à ne pas négliger lors de la création, c'est que le site puisse s'afficher de la même manière sur différents navigateurs (qui ne gèrent pas tous les balises *HTML* de la même manière, ou le *Javascript*). Pour cela, il faut utiliser des standards, c'est à dire certaines technologies et protocoles standardisés, reconnu par tous.

Il ne faut surtout pas négliger non plus l'aspect responsive design, c'est à dire le fait que votre site doit être tout aussi accessible et utilisable sur un écran d'ordinateur que sur l'écran d'un smartphone.

Une autre chose très importante à étudier lors de la création de votre site ou application, c'est l'accessibilité. Il faut qu'une personne mal voyante, ou handicapée moteur, puisse surfer et naviguer correctement sur le site. Pour cela, il existe des techniques à mettre en œuvre (balise ALT pour les images, les couleurs...). Rendez-vous sur le site du [W3C](#) pour en savoir plus.

Dernière chose, pour illustrer votre site, ne prenez pas des images n'importe où. Il existe des banques d'images en ligne que vous pouvez utiliser sur votre site gratuitement et légalement, car elles sont sous licence libre ou dans le domaine public. C'est un aspect à ne pas négliger lorsque l'on crée des documents numériques.

Le référencement doit être pris en compte dès la phase de conception d'un site, pensez-y. Utilisez les bons mots-clé et hiérarchisez votre contenu.

Les informations importantes

Les balises meta

Ce sont des informations, placées entre les balises `<head>` et `</head>` d'une page HTML, qui donnent des informations aux navigateurs web sur la page, c'est informations ne s'affichent pas.

Il y a les *meta name* et les *meta http-equiv*, toujours suivi du mot clé *content*, qui permet de donner la valeur de la balise. Un exemple :

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="description" content="Mon Site - Le sujet">
<meta name="keywords" content="mots clés relatifs à mon sujet">
```

Elles ne sont plus très utiles, en tous cas pour le référencement, mais c'est toujours mieux de les remplir correctement, notamment l'encodage des caractères pour éviter les soucis avec les lettres accentuées.

La langue

Dans une page HTML, il faut préciser la langue du document. Cela se fait au niveau de la première balise HTML, comme ceci :

```
<html lang="fr">
```

Le sitemap

Le sitemap est une « carte » de votre site, qui recense toutes les pages importantes que vous souhaitez référencer. Il est important d'en avoir un, à jour, afin que votre site soit correctement analysé et référencé sur les moteurs de recherche.

La favicon

C'est la petite image qui apparaît dans le navigateur à côté de l'adresse du site généralement. Sa taille est de 16x16 ou 32x32 pixels. Le format utilisé peut être du *png*.

Pour la faire apparaître, on insère cette ligne entre les balises `<head>` et `</head>` de sa page et on place la *favicon* à la racine de son site :

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

ou aussi ceci si le format est *png* :

```
<link rel="icon" type="image/png" href="favicon.png" />
```

Les CSS



Illustration 4: Logo W3C Valid CCS

Les feuilles de style en cascade (*Cascading Style Sheets*) permettent de définir des propriétés pour certains paragraphes, un style pour les différents titres, elle permettent de mettre en place le design du site. Il suffit de définir une fois les caractéristiques que l'on souhaite appliquer à un élément donné, et, dans chaque page, cela sera fait sans avoir à répéter à chaque fois ces propriétés.

Il faut créer un fichier que l'on peut nommer *style.css* par exemple, et que l'on placera à la racine du site ou dans un répertoire dédié. C'est dans ce fichier que l'on mettra en place toutes les règles d'affichage que l'on souhaitera (tel niveau de titre avec telle couleur, telle taille, tel bord...). Puis, il faudra ajouter dans les pages web cette balise pour indiquer d'utiliser notre feuille de style, entre les balises `<head>` de la page :

```
<link rel="stylesheet" href="style.css">
```

Voyons un petit exemple avec un titre (les titres en HTML sont numérotés de 1 à 6, 1 étant le plus important) :

```
H2 {  
    color: #4ac0f0;  
    font-size: 1.6em;  
    text-decoration: underline;  
    text-transform: uppercase;  
    margin-bottom: 10px;  
    border: 10px #0000ee;  
    font-family: 'caesar';  
}
```

Ici, cela définit la propriété de tous les titres de niveau H2. La couleur, la taille de la police, la décoration (ici souligné), la transformation (ici en majuscule), la couleur de fond, la marge du bas, la bordure (sa taille et sa couleur), et pour finir la police. Dans la page web, tout ce qui sera entre les balises `<h2>` et `</h2>` aura ces propriétés.

Voyons un exemple un peu plus complet. Une page web est divisée en plusieurs sections, principalement le haut de la page, avec généralement le logo et le menu, le contenu de la page au centre et le pied de page en bas. Pour cela on peut utiliser des balises `<div>` `</div>` pour délimiter ces sections. Elles peuvent s’imbriquer autant de fois que nécessaire, tout dépend de l’organisation de votre page. Veillez bien à imbriquer vos balises comme il faut, comme ceci, en fermant correctement chaque balises dans le bon ordre :

```
<div class="header">
  <h1><em>Le header de la page.</em></h1>
</div>
<div class="contenu">
  <h2><strong>Le contenu de la page.</strong></h2>
</div>
<div class="footer">
  Le footer de la page.
</div>
```

Dans notre fichier `style.css`, nous pourrions avoir ceci, qui permettrait de mettre en place le design de chacune des parties du site. Ici, on a mis en place une couleur de fond, des marges et un alignement au centre.

```
.header {
  background-color: blue;
}

.contenu {
  background-color: green;
  margin-top: 10px;
  margin-bottom: 10px;
}

.footer {
  background-color: blue;
  text-align: center;
}
```

Le PHP

PHP acronyme récursif PHP Hypertext Preprocessor, langage de script open source conçu pour le développement web principalement. Créé en 1994 par Rasmus Lerdorf. Langage orienté objet, web côté serveur le plus utilisé au monde. 2018 : 80 % des sites utilisent du PHP. Syntaxe proche du C.

Langage de script, interprété, le serveur interprète le code PHP pour générer du HTML à renvoyer au navigateur du client (site dynamique contrairement à un site statique où le serveur renvoie directement la réponse, sans avoir à la générer). Fonctionne avec les serveur Apache, Nginx, il est multiplate-forme.

Le PHP va permettre d'ajouter des fonctionnalités au site, comme la gestion d'un formulaire de contact, l'accès à une base de données et le traitement d'informations envoyées par le client.

Pour inclure du PHP dans une page web, on utilise ces balises :

`<?php>` et `</ ?>`

Il faut également modifier l'extension de la page, de html en php.

Voici un simple formulaire :

```
<form method="post" action="cours-form-traitement.php">
  <p>
    <label for="nom">Votre nom :</label>
    <input type="text" name="nom" id="nom" placeholder="nom" required />
  </p>
  <p>
    <label for="age">Votre âge :</label>
    <input type="number" name="age" id="age" />
  </p>
  <input type="submit" value="Envoyer" />
</p>
</form>
```

Comme vous le voyez à la première ligne, action renvoie vers un fichier, c'est lui, sur le serveur, qui traitera les données envoyées par le formulaire lorsque le visiteur cliquera sur le bouton envoyer. Ici, il y a juste le nom et l'âge.

Et voici ce fichier, en PHP donc, qui va traiter ces données puis les afficher. Les lignes commençant par // sont des commentaires, important pour savoir ce que fait votre code. On vérifie que le formulaire est rempli, on récupère les variables envoyées qu'on place dans des variables puis on les affiche (avec la commande echo).

```
<?php
// vérifie que le formulaire est rempli
```

```
if(!empty($_POST)) {  
    // récupère les données  
    $nom = filter_input(INPUT_POST, 'nom', FILTER_SANITIZE_EMAIL);  
    $age = filter_input(INPUT_POST, 'age', FILTER_SANITIZE_ENCODED);  
}  
  
//affiche les données  
echo 'Voilà les informations reçues : ' . '<br>';  
echo 'Votre nom : ' . $nom . '<br>';  
echo 'Votre âge : ' . $age . '<br>';  
?>
```

PHP include

Quand on crée un site, certaines parties de ce site reviennent sur chaque page, comme le menu, ou le bas de page. Et si on doit les modifier, il faut le faire sur chacune des pages. C'est long, et ce n'est pas très pratique.

Il existe une solution très simple à mettre en œuvre pour se faciliter la vie, l'utilisation de la directive *PHP include*.

On va commencer par créer une page d'index, qui servira de modèle pour le site, et qui ne contiendra pour le moment que le strict minimum d'une page HTML.

Puis, on va créer des « bouts » de page, avec juste le menu, le pied de page ou la partie contact. Ces pages ne seront pas des pages HTML, il n'y aura dedans que ce que nous souhaitons afficher. Les fichiers auront l'extension *.php* Voici un exemple :

```
<div id="sous_footer">  
<br>  
2013 – DUPONT Jean – Site personnel<br><br>  
Contact : contact@ma-boutique.fr - Paris<br><br>  
</div>
```

Ici, cette partie de page, qui aura les caractéristiques CSS *sous_footer* (défini grâce à la balise *div id*) n'affiche que la partie contact de la page. On enregistre cette page comme *contact.php*, et on la mettra dans un dossier à part, *template* par exemple, pour ne pas mélanger les pages HTML et les pages PHP includes.

Ensuite, on va dans notre page d'index servant de modèle, et on ajoute, où on le souhaite, la partie contact, comme ceci :

```
<?php include("template/contact.php"); ?>
```

Et on fait de même pour chaque partie des pages qui reviennent à chaque fois.

Voici maintenant à quoi peut ressembler une page index pour un site utilisant cette technique. On voit la balise *include* qui appelle une page PHP contenant le haut de page et le pied de page.

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de ma page</title>
</head>
<body>
<?php include("template/header.php"); ?>
Hello World !
Texte de la page
<?php include("template/footer.php"); ?>
</body>
</html>
```

Le Javascript

Le Javascript est un langage de script qui permet de rendre un site plus dynamique et y ajouter de nombreuses fonctionnalités. Créé en 1995, ce langage a beaucoup évolué et s'est largement étoffé. Le gestionnaire de dépendances NPM y est largement pour quelque chose.

De plus, depuis quelques années, plusieurs frameworks Javascript ont vu le jour (citons React, Angular principalement), ce qui a largement contribué à augmenter l'utilisation de ce langage.

Pour inclure un script Javascript dans une page web, il faut ajouter cette ligne entre les balises `<head>`, en spécifiant le chemin vers le fichier ainsi que le nom complet :

```
<script src="script.js"></script>
```

Voici un petit script très simple qui va simplement permettre d'afficher la date dans une DIV précise de la page :

```
<script>
  var d = new Date();
  document.getElementById("GFG").innerHTML = d.getTime();
</script>
```

Quelques conseils

Quelques conseils pour votre code. Pensez à commenter régulièrement votre code afin que quelqu'un relisant celui-ci comprenne facilement ce que chaque partie fait, c'est important. Également nécessaire, une bonne indentation de chacun de vos fichiers, pour rendre le code plus clair et plus lisible. Et n'hésitez pas à aller lire la documentation en ligne des outils que vous utilisez, à chercher des informations en ligne sur les balises et commandes, posez des questions dans des forums.

Encore 3 choses importantes à prendre en compte, et ceci dès la conception du site.

La vitesse d'affichage de vos pages. Une page trop lente fera fuir l'internaute. Vous pouvez tester votre site sur cet [outil](#) en ligne.

Il vous indique un score, et surtout, vous donne les informations nécessaires afin d'améliorer ce score et d'augmenter la vitesse de vos pages. Principalement, il y a les fichiers CSS ou les *Javascript* que l'on peut minimiser, activer la compression *GZIP* sur le serveur (à l'aide du fichier *htaccess*), optimiser les images et spécifier leur taille dans les fichiers *HTML*.

Également à prendre en compte, c'est la validité de votre code. Une page avec des erreurs va rebuter l'utilisateur, l'affichage sera différent d'un navigateur à l'autre, et cela n'est pas bon pour le référencement non plus.

Rendez-vous sur le validateur du [W3C](#) (World Wide Web Consortium) afin de vérifier chacune de vos pages ainsi que vos feuilles de styles).

Et aussi le *responsive design*, c'est à dire le fait qu'un site s'affiche aussi bien sur un ordinateur de bureau que sur une tablette ou sur le petit écran d'un smartphone. Il existe des *frameworks* prêt à l'emploi pour cela, tel que [Foundation](#) ou [Bootstrap](#), qui propose des modèles de page, que l'on peut adapter, et qui s'afficheront correctement quelque soit la taille de l'écran du visiteur.

Mise en place

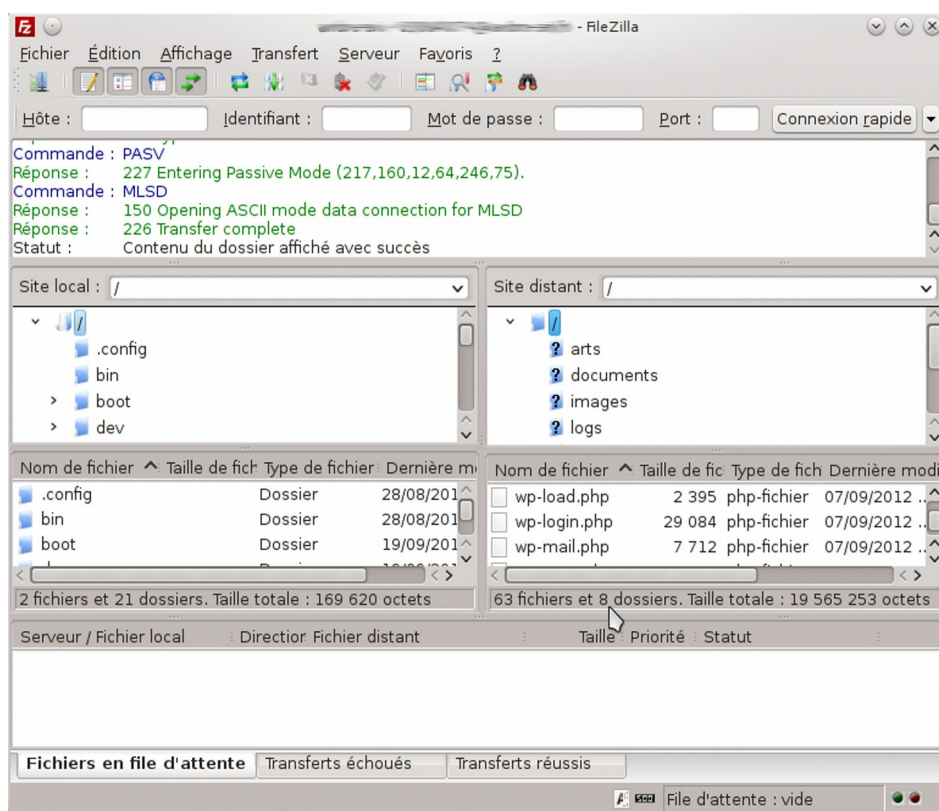
Publication

Une fois les pages du site créées sur votre ordinateur, il faut, afin que le site soit accessible sur internet, les mettre en ligne sur un hébergeur. On se sert pour cela d'un client FTP pour envoyer les fichiers composant votre site de votre ordinateur vers votre espace réservé sur le serveur de votre hébergeur.

On peut se servir de *FileZilla* pour cela, c'est un logiciel libre, complet et performant. Pour se connecter sur son hébergeur, il faut l'adresse du serveur FTP de son hébergeur, ainsi que les informations du compte tels que login et mot de passe. Puis, on transfère ses fichiers de son ordinateur vers son espace web.

Maintenant le site est accessible depuis un navigateur internet en tapant son nom de domaine dans la barre d'adresse du navigateur.

Présentation de FileZilla : En haut, il faut renseigner les informations de connexion. Sur la partie gauche, ce sont les fichiers locaux, le site web que vous souhaitez mettre en ligne, situés sur l'ordinateur. Dans la fenêtre de droite, c'est le serveur FTP où vous devez placer votre site web.



Référencement

Référencer un site web est un travail qui prend du temps. Et ce travail commence dès la conception du site. Dans les balises *title* et *description* mettez des mots clés, au début si possible. Pareil dans vos pages, il faut des mots clé en début de texte, et dans les titres h1, h2 etc. C'est également une bonne chose de mettre les mots clés sur des liens, en gras, en italique, pour les faire ressortir.

Pour les liens, il faut un bon maillage interne, c'est à dire des liens entre toutes les pages de votre site. Pour les liens sortants, inutile d'en mettre trop, préférer la qualité à la quantité, avec des sites en rapport avec votre thématique. De même pour les liens pointant vers votre site, il vaut mieux des sites de qualité en rapport avec votre activité.

Un fichier *sitemap* est utile également, afin que le site soit bien visité par les moteurs de recherche. Voici un [outils](#) gratuit en ligne.

Vérifier également les [liens](#) afin de ne pas avoir de liens morts et d'erreurs 404.

Une fois que tout cela est vérifié, il faut dire à internet que votre site est là.

C'est donc l'étape de référencement, c'est à dire, prévenir les moteurs de recherche et annuaires que votre site est disponible à telle adresse et qu'il traite de tel ou tel sujet.

Il faut se rendre sur les moteurs de recherche et les annuaires pour inscrire son site en donnant l'adresse web (URL), le nom du site, et son thème.

Tout cela pour permettre aux internautes de trouver facilement le site qu'ils recherchent d'après leurs mots clés sur un moteur de recherche (comme [Qwant](#) par exemple, qui ne surveille pas ce que vous faite sur internet).

Vous pouvez aussi vous inscrire sur des réseaux sociaux, créer une page sur votre activité pour vous présenter, rejoindre des groupes et des forums autour de votre thématique, et participer à des discussions, pour faire connaître votre site et améliorer son référencement.

Aller plus loin

Nous avons vu les bases du développement web, pour de petits sites. Nous allons aborder ici des outils permettant la création de sites plus complexes et adaptatifs.

Framework



Illustration 5: Logo du framework AngularJS

Un framework est un ensemble de composants permettant de construire et gérer l'architecture d'une application ou d'un site web. Leur utilisation doit donc simplifier le travail du développeur en fournissant un cadre ainsi que des fonctionnalités prêtes à l'emploi.

Il en existe un grand nombre, pour divers usages et basés sur certains langages. Une fois que l'on a appris un langage on peut s'initier à l'utilisation d'un framework, c'est une nécessité si l'on veut pouvoir faire des choses évoluées.

Les principaux frameworks [PHP](#) sont notamment ***Symfony***, ***Laravel***, ***Zend*** ou encore ***CodeIgniter***.

En [Javascript](#), il existe aussi un grand nombre de framework comme ***Angular***, ***ReactJS***, ***Vue***.

Django est un framework web très populaire développé en Python.

Ruby on Rails est quand à lui conçu en Ruby.

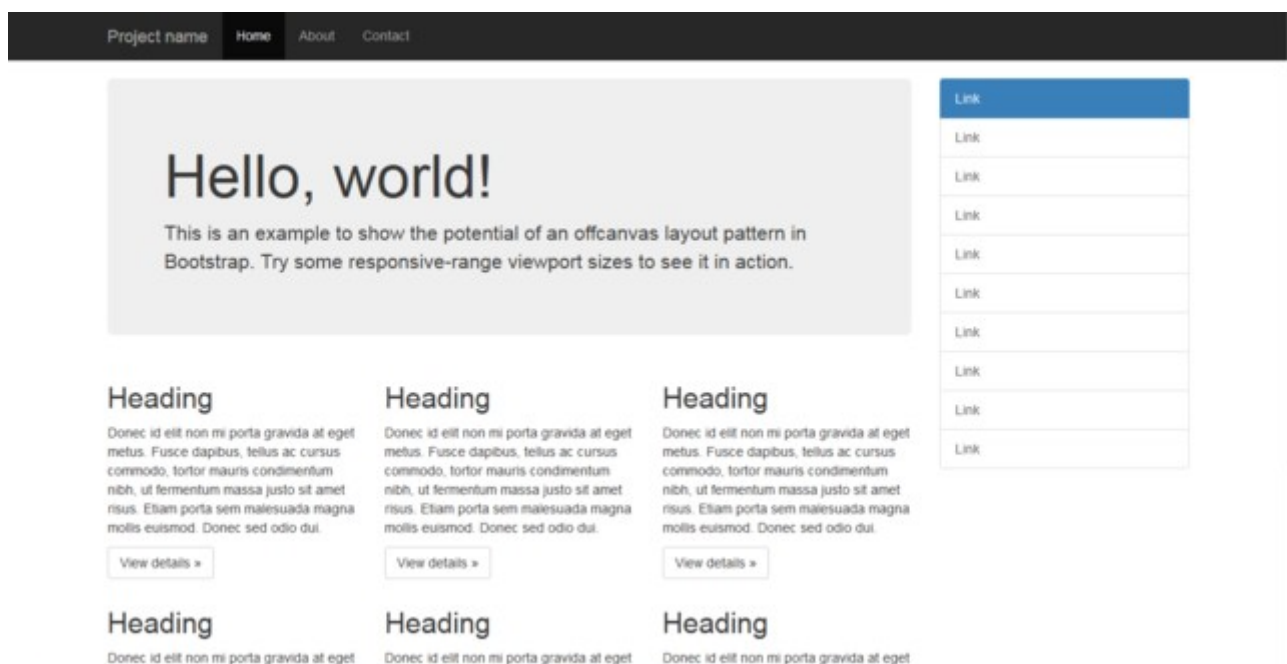


Illustration 6: Le framework Bootstrap de Twitter

[Bootstrap](#) est un framework développé par Twitter et publié sous licence libre. Mais contrairement aux frameworks présentés plus haut, celui-ci ne s'occupe que du design, de l'apparence de l'application. Il permet de gérer le *front-end*, c'est à dire ce que l'utilisateur voit, contrairement au *back-end*, qui est la partie d'administration d'un site ou d'une application.

Construit à l'aide de HTML et CSS, mais contenant également des outils en Javascript, il contient tous le nécessaire à la mise en forme d'un site ou d'une application web (en responsive design, qui s'adapte à la taille de l'écran), formulaire, bouton et autres outils de navigation. Il est composé d'un système de grille permettant de gérer chaque éléments de l'interface. Il est très populaire et est utilisé, notamment, dans de nombreux thèmes pour *WordPress*.

Voilà un exemple de design généré par Bootstrap :



Outils

Voici les principaux outils libres à connaître pour le développement web.

NetBeans et *Eclipse* ne sont pas des IDE (environnement de développement) spécifiques au développement web, contrairement à *BlueGriffon* ou *Kompozer*, mais, à l'aide de plugins, peuvent devenir des outils très puissants. Il existe également Visual Code Studio ou encore Brackets qui sont bien faits et disposent de nombreux plugins afin d'augmenter leurs fonctionnalités.

L'installation d'une architecture *LAMP* (Linux Apache MySQL PHP) permet de travailler en local sur son propre serveur web. Ce système est également disponible pour Windows. Bien *qu'Apache* soit le serveur web le plus utilisé sur internet, *Nginx* est lui aussi très populaire.

Pour gérer ses bases de données, *PhpMyAdmin* est un incontournable.

Et le client FTP *FileZilla* pour transférer ses fichiers sur le serveur de son hébergeur.



Illustration 7: Logo de Git

[Git](#) est un système de gestion de version décentralisé très puissant permettant de gérer finement les différentes versions d'un projet. C'est un outil incontournable pour tous les développeurs, notamment si l'on travaille en groupe. Il permet de surveiller l'évolution du code suivant les évolutions et ajouts et d'en garder tout l'historique.

CMS



Illustration 8: Logo du CMS Wordpress

Les CMS, systèmes de gestion de contenu, sont des logiciels qui permettent de mettre en place rapidement un site internet disposant de nombreuses fonctionnalités et adaptables à divers objectifs. Les CMS les plus connus sont *Wordpress* évidemment, *Drupal*, *Joomla*, *SPIP*, mais aussi *Prestashop* pour les boutiques en ligne ou *MediaWiki* pour les wikis.

Pour finir

Vous avez découvert avec ce tutoriel les bases du développement web, les langages et outils nécessaires. Vous avez les connaissances de base pour vous initier et vous lancer dans la programmation. De nombreux sites internet permettent de commencer à programmer en ligne. De nombreuses ressources en ligne sont disponibles, il faut avoir la curiosité de chercher, et de poser des questions sur les forums également.

Maintenant que vous avez cet aperçut du métier, peut-être voudrez-vous aller plus loin et vous lancer dans l'apprentissage d'un langage ? HTML et CSS sont les bases pour commencer. Ensuite, Javascript et PHP pour faire des choses plus dynamiques et évoluées. Et pour finir, la maîtrise d'un framework sera la dernière étape pour devenir un professionnel.

Le développement web est un métier vraiment intéressant, aux diverses facettes. Il permet de créer toutes sortes de sites ou d'applications web. Mais, et c'est comme tout, il requiert beaucoup de travail, il faut être précis et consciencieux. Tout le monde n'est pas fait pour cela. A vous de voir. Prêt à vous lancer ?

Bibliographie

Apprendre le web sur [MDN](#)

Conception de site sur [Wikipedia](#)

Les bases de la programmation sur [Zeste de Savoir](#)

Développement web sur [MDN](#)

Créer un site web sur [Commentcamarche](#)

Hébergeur web sur [Wikipedia](#)

Référencement sur [Commentcamarche](#)

Standards du web sur [Wikipedia](#)

Standards sur le [W3C](#) (en anglais)

Accessibilité web sur le [W3C](#)

Langage de programmation sur [Wikipedia](#)

Programmation web sur [Wikipedia](#)

CMS sur [Wikipedia](#)

Framework sur [Wikipedia](#)

Nombreuses ressources sur le développement web sur [Developpez](#)

Ce tutoriel est publié sous licence libre GNU Free Documentation License :



Texte de la licence :

<https://www.gnu.org/licenses/fdl.html>