



**HTML** - HyperText Markup Language. Langage par balise, site statique type minitel



C'est moche !!

Je dois répéter mon style partout !

**CSS** - Les feuilles de style en cascade



Ça manque d'interaction et de mouvement !!

**JavaScript** - Langage de programmation de scripts coté client browser



Comment je rends mon site dynamique ?

Comment je sauvegarde des données ?

Comment j'interagis avec mes utilisateurs ?

**PHP** – Langage Hypertext Preprocessor pour générer du HTML coté serveur et avoir une application unique pour chaque client, permet d'avoir du rendu sur mesure en fonction de la demande

**MySQL** - (*Structured Query Language*), langage de requête structurée



Je dois toujours réécrire le même code !

Quand ça devient compliqué c'est le bordel !

Reprendre le code d'un autre c'est chiant !

**Framework** - ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel et qui impose un cadre de travail respectant les bonnes pratiques de programmations (aussi bien back que front).

**Git / SVN** – Travailler à plusieurs, se partager les taches



Ça a déjà été fait pourquoi m'enmerder ?

Comment je peux réutiliser le code d'un autre ?

**Composer, npm** – Les Librairies d'applications, mettre à profit le savoir collectif.



Des problèmes complexes !

De la communication entre mes programmes !

Comment mieux coder ?

**Design pattern** – Développement Orienté objet – Interface - apprendre à gérer la complexité



C'est chiant faut recharger la page !

J'ai besoin de plus de rapidité !

**Les websockets** – connexion directe et gestion d'évènements avec le client

# Le chemin de croix d'un développeur Full Stack

## --- Les bases

1. Apprendre le HTML5, connaître les balises disponibles, utiliser les bonnes balises aux bons endroits. Respecter les règles du w3c.
2. Apprendre le CSS, maîtriser le responsive design en fonction de la taille de l'écran du client. Comprendre l'intérêt du SASS pour mettre des variables dans votre CSS et simplifier vos fiches de style. Réunir et compresser vos fichiers css pour la production.
3. Apprendre le javascript, maîtriser les interactions côté client et dynamiser vos applications, savoir interroger une API. Réunir et compresser vos fichiers js pour la production.
4. Apprendre un des langages serveurs PHP/Python/Java/.Net/C pour créer du contenu dynamique, communiquer avec l'utilisateur, sauvegarder des données, gérer les sessions et organiser la logique de votre application.
5. Apprendre à créer et à requêter des bases de données (SQL)

## --- La professionnalisation

6. Apprendre à utiliser des Frameworks pour gagner en productivité et utiliser les bonnes pratiques de la programmation. Au moins un Framework back (Laravel, Symfony, Django) et un Framework front (React, VueJs, Angular, JQuery)
7. Apprendre à installer des librairies et les adapter à notre cas d'usage en lisant les documentations. Réutiliser le travail d'autres développeurs et ne pas réinventer ce que d'autre ont déjà fait mieux que vous.
8. Apprendre la programmation orientée objet pour simplifier et rendre le code plus facile à réutiliser et à partager au sein d'un projet et pour vos futures applications.
9. Apprendre les designs patterns qui répondent au mieux aux problèmes les plus courants.
10. Apprendre à se servir d'un logiciel de versioning pour travailler en groupe et revenir facilement en arrière (Git/SVN)
11. Mettre une application en production et optimiser le SEO et le cache pour améliorer les performances
12. Créer des tests unitaires et fonctionnels sur les points sensibles de l'application

## --- La spécialisation

13. Identifier le ou les domaines de compétences où on est le plus à l'aise et entrer dans les détails des frameworks ou des librairies pour maîtriser le cœur du savoir et participer aux projets open source.
14. Faire de la veille active sur les nouveautés et continuer à se former pour être à jour dans ses domaines de compétences

## Conseils pratiques :

PRATIQUER, PRATIQUER, PRATIQUER. Créer des projets et recommencer en tentant d'améliorer à chaque fois. Demander conseil à des développeurs senior demandez-leur de relire votre code et de vous proposer des solutions plus performantes. N'hésitez pas à lire le code des autres sur les projets open source de github pour découvrir d'autre méthodologie de codage.

# Les bases et les docs officielles

## 1. HTML

Tout ce qui à savoir sur le HTML (les normes, les bonnes pratiques)

<https://www.w3schools.com/html/default.asp>

Toutes les références (balises)

<https://www.w3schools.com/tags/default.asp>

Les browsers support

[https://www.w3schools.com/tags/ref\\_html\\_browsersupport.asp](https://www.w3schools.com/tags/ref_html_browsersupport.asp)

## 2. CSS

Tout ce qui à savoir en CSS

<https://www.w3schools.com/css/default.asp>

Toutes les références

<https://www.w3schools.com/cssref/index.php>

Les browsers support (important)

[https://www.w3schools.com/cssref/css3\\_browsersupport.php](https://www.w3schools.com/cssref/css3_browsersupport.php)

## 3. Javascript

Tout ce qui à savoir en javascript

<https://www.w3schools.com/js/default.asp>

Toutes les références

<https://www.w3schools.com/jsref/default.asp>

## 4. SQL

Tout ce qui à savoir sur le SQL

<https://www.w3schools.com/sql/default.asp>

Toutes les références

[https://www.w3schools.com/sql/sql\\_ref\\_keywords.asp](https://www.w3schools.com/sql/sql_ref_keywords.asp)

## 5. PHP

Tout ce qui à savoir sur le PHP

<https://www.w3schools.com/php/default.asp>

Toutes les références

[https://www.w3schools.com/php/php\\_ref\\_overview.asp](https://www.w3schools.com/php/php_ref_overview.asp)

L'ensemble des références

<https://www.w3schools.com/references/index.php>

Des modèles de tous les principaux types de composant utile sur les app

<https://www.w3schools.com/howto/>

Vous pourrez également trouver des Quiz et des exercices corrigés sur chacun des langages sur le site w3c pour mettre à l'épreuve votre savoir et jauger vos compétences et votre progression.

<https://www.w3schools.com/exercises/index.php>

<https://www.w3schools.com/quiztest/default.asp>

Sur LinkedIn vous avez des tests sur les différents langages et les différents skills pour valider vos compétences sur votre profil. C'est intéressant pour valoriser vos compétences.

<https://www.linkedin.com/skill-assessments/hub/quizzes/>

En plus du portfolio, avoir un compte github avec quelques projets publics pour montrer votre façon de travailler et de coder peut-être une bonne chose. Attention à ne partager que du code de qualité dont vous êtes fier. Mieux vaut ne pas avoir de compte github qu'un compte avec du mauvais code.