



Apprendre
la programmation

LES 7 ERREURS À ÉVITER POUR BIEN DÉBUTER EN PROGRAMMATION

CHAPITRE 1

SOMMAIRE

1. Chapitre 1 : Sommaire
2. Chapitre 2 : Qui suis-je ?
3. Chapitre 3 : Ne pas savoir pourquoi vous programmez
4. Chapitre 4 : Croire que la programmation est compliquée
5. Chapitre 5 : Ne pas connaître les bases de l'algorithmie
6. Chapitre 6 : Ne pas être passionné
7. Chapitre 7 : Commencer grand au lieu de commencer petit
8. Chapitre 8 : Débuter sur le mauvais langage de programmation
9. Chapitre 9 : Programmer seul dans son coin
10. Chapitre 10 : Conclusion
11. Chapitre 11 : Copyright



2

QUI SUIS-JE ?

Avant de rentrer dans le vif du sujet, je tiens à me présenter.

Je suis Gaëtan Cottrez et ça fait maintenant plus de 10 ans que je code dont 7 ans dans le monde professionnel. J'ai toujours été passionné d'informatique dès le plus jeune âge mais c'est le développement, et plus précisément le web, qui m'a toujours attiré.

Analyser un besoin réel pour le **développer** par moi-même dans le but **d'automatiser** des tâches afin de me faire **gagner du temps** est quelque chose d'excitant et de passionnant.

Et pour y parvenir, on doit **parcourir un chemin semé d'embûches**. Heureusement, l'immense **communauté** des développeurs est là pour **nous permettre de les surmonter** grâce aux **conseils qu'il prodigue** et aux **différentes ressources** qu'elle peut nous offrir. Ce chemin m'a mené jusqu'à présent au rang de «**Chef de Projets**» mais **je garde toujours les mains dans le cambouis** comme on dit. J'ai donc toujours la double casquette : «Développeur» et «Chef de Projets». Je continue toujours de goûter de très près à la beauté de la programmation.

Pourquoi ai-je écrit ce guide ? La **première raison** est que **j'aime le partage** et que j'ai beaucoup profité de ce que la communauté a pu m'apporter qu'il me semble normal de contribuer à mon tour. La deuxième raison est que **j'ai moi aussi été un débutant** et que je sais ce que ça fait quand on commence à programmer. On ne sait pas **par où commencer**, ce qu'on doit faire lorsque l'on se retrouve **bloqué pendant des heures** sur un problème ou tout simplement **effrayé par la taille et le travail à accomplir** pour un projet. Je suis passé par là et je sais ce que c'est, rien de très confortable.

Ce guide sera ton **point de départ** dans ce merveilleux domaine qu'est le développement. Tu commenceras par **un avantage non négligeable sur les autres** puisque tu connaîtras **les erreurs à éviter pour bien débuter** ce qui te permettra de **ne pas abandonner au 1er obstacle** et de **gagner du temps** dans ton apprentissage.

Je te souhaite une très bonne lecture et on commence tout de suite dans le vif du sujet,

Gaëtan Cottrez

Apprendre la programmation, le blog

CHAPITRE 3

NE PAS SAVOIR POURQUOI VOUS PROGRAMMEZ

“Pour moi la programmation est plus qu'une pratique importante. C'est aussi un formidable moyen d'atteindre les bases de la connaissance.”

~ Grace Hopper

J'ai une question simple à te poser : **pourquoi veux-tu programmer ?**

Et je suis très sérieux quand je te pose cette question. Je rencontre **beaucoup de gens qui ne savent pas** réellement pourquoi ils veulent faire ça. Je pense avoir déjà entendu pas mal de raisons des plus farfelues au plus insensées d'entre elles. Il est impératif que **tu saches pourquoi tu veux programmer et surtout le but** que tu t'es fixé. Pourquoi est-ce que je te dis ça ? Tout simplement parce **si tu sais clairement** pourquoi tu veux programmer et dans quel but tu le fais, **tu pourras clairement définir tes objectifs d'apprentissage** afin de **n'apprendre que le strict nécessaire** pour arriver à tes fins.

Laisse-moi te raconter mon histoire. J'ai toujours été fasciné par l'informatique depuis que je suis en âge de le comprendre et c'est aujourd'hui ce domaine qui nous domine. Petit à petit, j'ai commencé à m'intéresser à la programmation. L'idée de concevoir des programmes informatiques pour répondre à des problématiques du

quotidien en automatisant des tâches pour faire gagner du temps m’a fasciné au plus au point. Je me suis sentis pousser des ailes et je sentais que grâce à la programmation, je pouvais contribuer au monde d’aujourd’hui et réaliser quelque chose d’utile mais surtout innovant. Oui, **c’est comme cela que je percevais la programmation à 14-15 ans**. Et c’est cette motivation avec l’énergie qui en découlait qui m’a convaincu que je voulais devenir un programmeur et en faire mon métier. Bien entendu, je me posais beaucoup de questions, comme toi certainement, qui avec le recul étaient des barrières que je tentais de m’imposer : **Vais-je y arriver ? Ai-je le niveau pour pouvoir faire cela ? Est-ce vraiment fais pour moi ?** Car pour être honnête en plus d’avoir à peine la moyenne à l’école et des professeurs qui ne croient en personne qu’en eux-mêmes, **je n’avais spécialement une grande confiance en moi** et en mes possibilités mais je savais 2 choses essentielles : **pourquoi je voulais programmer et dans quel but**. Aujourd’hui, je suis capable de répondre à ces questions et dans l’ordre les réponses sont : **non, non et non**. Je me devais donc d’**apprendre tous les aspects de ce domaine pour réussir** à en faire mon métier d’aujourd’hui et c’était loin d’être facile. Mais pourtant je suis en train d’écrire et de partager mon histoire avec vous dans ce guide.

Alors toi qui me lis, **pourquoi veux-tu programmer** ? Quel est **ton but** ? Quels sont **tes objectifs** ? Veux-tu en **faire ton métier** ? Est-ce que **la programmation te fascine et/ou te passionne** ? **Veux-tu développer un outil** dont tu as besoin à tout prix et tu veux avoir la satisfaction de **dire «c’est moi qu’il l’ait conçu»** ? Ou peut-être que tu es juste curieux et que tu souhaiterais explorer ce domaine pour voir si cela est fait pour toi ? Parce qu’au final vouloir **faire de la programmation pour en faire son métier** et **faire de la programmation par passion**, ce n’est pas la même chose ni le même objectif. Dans un cas, il sera nécessaire **d’approfondir la matière** et d’exercer régulièrement. Dans l’autre, l’apprentissage **ne sera pas aussi intense** et soutenu, tu pourras donc avancer à votre rythme.

Je te conseille d’arrêter de lire ce guide maintenant et de prendre quelques minutes pour réfléchir aux questions de ce chapitre. Je t’assure que c’est très important car cela va t’aider à tracer ton chemin pour savoir jusqu’où tu veux aller dans la programmation. Alors ça y est ? Tu sais maintenant pourquoi tu veux programmer ? Très bien, alors on passe à la suite.

CHAPITRE 4

CROIRE QUE LA PROGRAMMATION EST COMPLIQUÉE

“La plupart des gens trouvent le concept de la programmation évident, mais la réalisation impossible.”

~ Alan Jay Perlis

Beaucoup de personnes pensent que **pour programmer il faut être un petit génie** quand d'autres pensent qu'il faut **avoir un très bon niveau en math**. Et dans l'extrême certains croient que les programmeurs écrivent une suite de caractères incompréhensible aussi vite que l'éclair comme dans les films.

En clair, **ils pensent que la programmation est très compliquée** et est un monde à part éloigné de ce qu'on a l'habitude de voir. En fais-tu partie ? Il faut absolument te sortir cette idée de la tête. Franchement **si je te montrais mes résultats en maths à l'école** (et mes résultats de manière générale), je pense que **tu n'aurais rien à m'en-
vier**. J'étais très loin d'être le 1er de la classe et sans exagérer je pense que j'étais plutôt plus proche du 1er en partant de la fin. Si mes profs m'avaient dit qu'il fallait être bon en maths pour faire de la programmation, tu ne serais pas entrain de lire ce guide.

La programmation n'est pas compliquée. Ce qui est compliqué en revanche c'est de s'accrocher et de ne pas abandonner au premier obstacle. Si tu as de **bonnes bases en informatique** de manière générale et que tu sais **comment fonctionne de façon générale un ordinateur** alors tu pourras apprendre facilement la programmation. Ce qui est certain en revanche c'est que tu devras acquérir certaines compétences comme **avoir un esprit logique**, être patient et être quelqu'un qui n'abandonne pas à la 1^{er} difficulté rencontrée.

Tu n'as aucune de ces compétences ? C'est très bien car j'ai une bonne nouvelle pour toi ! Tu vas les acquérir en pratiquant la programmation car comme la passion ces compétences vont s'acquérir au fur et à mesure que tu vas progresser dans ton apprentissage.

Maintenant j'ai une nouvelle question : **penses-tu qu'il est compliqué d'apprendre une nouvelle langue** ? Oui tu as bien lu, une nouvelle langue car la programmation est une façon de communiquer et d'échanger des informations. Une langue bizarre bien entendu au premier abord d'autant que l'on s'adresse à un ordinateur mais une langue quand même. D'ailleurs on pourrait identifier **le programmeur comme un traducteur**. Le programmeur doit en réalité traduire un besoin exprimé en français dans un autre langage, le langage informatique, qu'un ordinateur pourra comprendre. Comme toute nouvelle langue, cela peut sembler compliqué et fastidieux au départ surtout si on n'est pas bilingue mais je te rassure : c'est beaucoup plus simple d'apprendre la programmation qu'une nouvelle langue et j'en sais quelque chose tu peux me croire et tu t'en rendras très vite compte par toi-même.

Ce qui peut ajouter une complexité à la programmation c'est que tu as certainement entendu qu'il **fallait être bon en anglais**. C'est un vieux préjugé mais si cela peut être un bon atout mais **il n'est pas nécessaire d'être un bilingue** pour faire de la programmation. Tu dois apprendre essentiellement du vocabulaire technique que tu pourras acquérir très aisément au cours de ton apprentissage lorsque tu en auras besoin. Et puis si tu ne comprends pas un mot en particulier Google est ton ami n'est-ce pas ?

Ce que tu dois assimiler c'est qu'avec un peu d'investissement, de rigueur, de patience et de la bonne volonté, tu apprendras très vite à programmer et tu te diras qu'en fin de compte ce n'est pas si difficile que ça.

CHAPITRE 5

NE PAS CONNAÎTRE LES BASES DE L'ALGORITHMIE

“Enseigner la programmation va à l’encontre de l’éducation moderne : Quel est le plaisir à planifier, se discipliner à organiser ses pensées, faire attention aux détails et apprendre à être autocritique ?”

~ Alan J. Perlis

Tout le monde peut s’improviser programmeur mais on atteint vite les limites. Des problèmes de performances dans l’exécution du code se font vite sentir ou bien notre programme ne fait finalement pas ce que l’on souhaite. On rédige énormément de lignes jusqu’à se rendre compte que l’on peut faire exactement la même chose en moins de lignes. Mais c’est trop tard le code est déjà fait et nul besoin de repasser dessus. C’est **L’ERREUR la plus fréquente** du débutant : ne pas connaître les bases de l’algorithmie. Beaucoup diront qu’il n’est pas nécessaire de connaître l’algorithmie mais je ne suis vraiment pas de cet avis. Il est nécessaire de connaître cette compétence pour plusieurs raisons. L’une d’entre elles et pas des moindres, c’est que cela va vous permettre de programmer plus vite, de mieux hiérarchiser son code et ainsi éviter des problèmes de conception logique dans sa programmation.

Tu te poses certainement cette question : **qu’est-ce qu’un algorithme ou l’algorithmie** ? Je répondrais simplement : tout ce que tu fais chaque jour dans un but précis pour obtenir un résultat. Tu ne le savais peut-être pas mais tu en utilises en

réalité tous les jours dans ton quotidien. En fait, je devrais dire « exécute » car un algorithme s'exécute. Lorsque tu prends une douche, que tu manges, que tu marches ou que tu utilises ta voiture par exemple, tu exécutes en réalité un algorithme. Ce que tu dois savoir c'est qu'**un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit à un résultat donné.**

L'avantage de l'algorithmie s'est que cela s'écrit dans un langage très compréhensible pour tout le monde, **un langage humain**. Et tu pourras le réécrire par la suite dans un langage informatique : un langage de programmation. La trame et la logique seront globalement conservées d'un langage à une autre ce qui te permettra de gagner beaucoup de temps et d'obtenir une clarté homogène dans ce que tu réalises.

Prenons un exemple pour illustrer : le trajet en voiture d'un point «A» vers un point «B». Quelle serait la suite d'instructions pour effectuer cette action ? Voici une proposition minimaliste qui peut être plus complète :

1. Je me déplace vers ma voiture
2. Je déverrouille ma voiture
3. J'ouvre la porte et je monte dans ma voiture
4. Je mets la clé dans le contact et je tourne la clé pour démarrer ma voiture
5. Je paramètre mon GPS avec ma destination
6. Je conduis en respectant les instructions de mon GPS

Cette liste d'instructions peut permettre à n'importe quel être humain de comprendre et de pouvoir effectuer un trajet d'un point «A» vers un point «B». Pourquoi ? Parce que l'être humain est capable de comprendre ce langage et cette liste d'instructions. De plus, il connaît le résultat attendu de cette série d'instructions : la destination jusqu'au point B. Pour un algorithme informatique c'est exactement pareil, l'ordinateur va comprendre la série d'instructions qu'il va exécuter les unes à la suite des autres dans un langage qu'il comprend pour vous faire obtenir le résultat attendu.

Maintenant que tu visualises et que tu sais ce qu'est l'algorithmie, tu te demandes certainement **pourquoi il faut connaître les bases avant de pouvoir programmer** ? Ma réponse est simple : l'algorithmie va t'aider à mieux concevoir ton pro-

gramme en visualisant et en découpant chacune de tes instructions dans le but d'obtenir un résultat par rapport à la logique que tu vas mettre en place. Le bonus, qui est tout aussi important, c'est que tu vas comprendre la logique de l'ordinateur pour développer ensuite ton programme informatique quelque soit le langage de programmation que tu emploieras. Bien entendu, il est possible de programmer sans connaître l'algorithmie mais je ne te le recommande pas. Ta programmation ne sera pas qualitative. Elle sera très compliquée à retoucher et le programme sera au final moins performant. Pas top ! Si le résultat attendu n'est pas celui souhaité, il y a de fortes chances que la rectification de ton programme soit fastidieuse et très compliquée car tu vas perdre ton temps pour trouver la source à ton anomalie.

Saches que cela ne prend pas énormément de temps d'apprendre l'algorithmie et que le retour sur investissement de ce temps consacré dans ta programmation sera très vite regagné. Je te l'assure !

CHAPITRE 6

NE PAS ÊTRE PASSIONNÉ

“Ce qui ne me passionne pas m’ennuie”

~ Vincent Lindon

Comme lorsque tu découvres et débutes un nouveau domaine il faut être un minimum passionné. **Avoir cette petite flamme** qui t’anime et qui te motive pour ne pas lâcher prise dès le premier obstacle rencontré en plus d’apporter un côté très ludique à la pratique. C’est d’autant plus important pour l’apprentissage de la programmation puisque cela va t’aider à surmonter plus facilement les difficultés que tu vas rencontrer.

Je ne veux pas te faire peur mais je veux être très honnête et très transparent avec toi : tu vas en avoir **des problèmes et des difficultés**, et par certains moments tu ne comprendras pas tout ce tu vas tenter d’apprendre, et tu vas perdre du temps à en rester bloqué sur des problèmes de bas niveau qui pourront te frustrer au plus au point que tu auras l’impression d’être si mauvais, que **tu feras beaucoup, encore et toujours des erreurs...**

Tu sais quoi, je vais te rassurer : **C’est tout à fait normal !**

Ce sont des étapes par lesquelles de nombreuses personnes sont passées bien avant toi (j'en fais partie) mais beaucoup d'entre elles ont **malheureusement abandonné**. Je ne veux pas que tu en fasses partie car ces difficultés peuvent te faire renoncer plus vite que prévu et ce serait vraiment dommage. Pour éviter cela et te motiver à les surmonter en les affrontant, **la passion sera ton plus grand allié**. Bien sûr il n'est pas nécessaire que tu aies besoin d'avoir une passion débordante et intense dès le départ, mais une infime passion peut conduire à une véritable révélation pour ce domaine. J'en parle en connaissance de cause car avant d'avoir un minimum de passion pour la programmation, j'étais en premier curieux et fasciné.

Quoi qu'il en soit, la passion va te guider durant tout ton parcours et ton apprentissage. Elle t'aidera à tenir dans les mauvais moments et te fera apprécier la réussite de faire un programme. **Sans passion, tu n'iras pas très loin dans ce domaine**. Je préfère te le dire.

Tu dois aimer pratiquer la programmation car il n'y a rien de plus frustrant et navrant que de faire quelque chose qu'on n'aime pas faire. Si tu n'aimes pas, alors ton chemin sera très court. **Coder avec plaisir et par passion et non pas par nécessité et obligation!**

CHAPITRE 7

COMMENCER GRAND AU LIEU DE COMMENCER PETIT

“C'est en forgeant qu'on devient forgeron.”

Cela peut paraître évident mais de nombreuses personnes voient trop grand lorsqu'elles commencent à programmer. Rendons-nous à l'évidence c'est une grosse erreur. Comment peut-on prétendre mener à bien un grand projet sans avoir commencé à exercer la programmation ? Je comprends parfaitement ces personnes. Elles veulent débiter rapidement et entreprendre de grandes choses parce que ce qui les motive c'est de voir les choses énormes que l'on peut réaliser. Mais malheureusement ce n'est pas la bonne stratégie.

Prenons un exemple : imaginons que tu souhaites construire par toi-même ta maison. Pour cela, tu auras besoin de plusieurs compétences non négligeables pour y arriver comme savoir monter une charpente, savoir maçonner, savoir faire les fondations, savoir carreler, savoir peindre et j'en passe.

Nous sommes d'accord que tu ne vas jamais construire ta maison sans t'avoir exercé dans chacune de ses compétences. Tu vas d'abord apprendre, essayer et

t'entraîner à carreler, à maçonner, à peindre, etc... Bref, tu vas commencer par t'entraîner sur des tâches de plus petite envergure histoire de tâter le terrain en vue de mettre à profit ce que tu as appris pour bâtir ta maison. Par la même occasion, tu pourras constater par toi-même si cela te plaît en effectuant des petites tâches.

Pour la programmation ? C'est la même chose. Tu vas devoir dans un premier temps acquérir des compétences où tu vas t'exercer et t'entraîner pour les valider sur des petits projets en prévision de réaliser ton «Saint Graal». Ne perds pas de vue ton objectif pour la programmation pour mieux orienter ton apprentissage. Apprends ce dont tu as besoin au moment où tu en as besoin. C'est le plus important et le plus efficace car cela ne sert à rien d'apprendre des choses que tu n'utiliseras pas dans l'immédiat. En plus de perdre du temps inutilement, tu devras réapprendre car tu n'auras pas eu l'occasion de mettre en application ce que tu as appris.

En parlant de perte de temps, voici une astuce pour en perdre le moins possible sur des petits projets pour t'exercer : fais en qui te feront monter rapidement en compétences et qui touchent de près ton but et tes objectifs. Si par exemple, tu sais que ton grand projet devra être muni d'un système d'authentification sécurisée. Pourquoi ne pas le réaliser en petit projet en vue de t'exercer ? Cela te permettra de récupérer ton code et de l'améliorer dans le cadre de ton projet de plus grande envergure.

Pour résumer, privilégie plutôt les petits projets utiles pour commencer afin de mieux te préparer sur un projet de plus grande envergure.

CHAPITRE 8

DÉBUTER SUR LE MAUVAIS LANGAGE DE PROGRAMMATION

“Trop de connaissance ne facilite pas les plus simples décisions”

~ Frank Herbert

C'est un choix à ne pas négliger. Beaucoup commencent sur un langage de programmation qui ne leur est pas approprié car ils n'ont pas réfléchi au pourquoi ils voulaient programmer et dans quel but. Tu ne dois pas faire la même erreur de choisir un langage de programmation soit parce qu'il est à la mode soit parce que c'est un nouveau langage de programmation. Tu dois le choisir en fonction de tes convictions et du but que tu t'es fixé dans le [chapitre 3](#).

Par exemple si ton objectif est de programmer une application Windows, tu vas plutôt t'orienter vers un langage du type .NET comme le «C#» mais certainement pas vers un langage web comme PHP. L'exemple est assez extrême mais il résume vraiment ce qui se passe dans la tête des débutants. Ils vont vouloir naturellement se tourner vers un langage à la mode plutôt qu'un langage correspondant plus à leurs attentes.

Tu as dû aussi voir sur la toile que **beaucoup de personnes proposent** de commencer sur des langages de programmation simples pour bien débiter comme **le «C» ou bien le «Python»** car l'un va t'apprendre à comprendre comment fonctionne votre machine et l'autre est plus abordable et respecte pas mal de bonnes pratiques concernant la programmation. **Ces personnes ont tout à fait raison** car beaucoup sont passées par là et il est vrai que cela permet d'assimiler et de comprendre très rapidement les aspects de la programmation. Mais **TOI**, est-ce que tu veux vraiment débiter sur ces 2 langages ? Est-ce que cela remplit ton but et tes objectifs dans l'apprentissage de la programmation ? Je pense que l'on connaît tous les 2 ta réponse.

Alors finalement **comment faire pour choisir son langage de programmation** ? Il y a principalement 2 choses à déterminer pour choisir ton langage de programmation sur lequel tu vas débiter.

Tout d'abord tu dois **définir la plateforme sur laquelle tu veux programmer** suivant ton but et tes objectifs. Par exemple si tes objectifs sont d'en apprendre plus sur la conception d'un objet connecté de A à Z et que ton but final est d'en réaliser un, il est alors utile de commencer à programmer sur une carte Arduino avec le langage Arduino plutôt que d'apprendre le Java par exemple.

Enfin, tu dois réfléchir à la **finalité de ta future application : où elle sera installée et comment elle sera accessible**. Si tu souhaites réaliser une application mobile, il sera alors judicieux de choisir un langage de programmation orientée mobile plutôt qu'un langage orienté web ou «desktop» par exemple.

Tu dois prendre le temps de bien réfléchir et de bien choisir car cela va fortement influencer ta motivation et ton apprentissage de la programmation.

CHAPITRE 9

PROGRAMMER SEUL DANS SON COIN

“Plus on est de fous, plus on rit”

Je terminerais ce guide avec ce chapitre tout aussi important que les autres. Tu dois savoir que tu n'es pas seul et qu'il y a une grande communauté de développeurs qui existe dans le monde entier et prêt à t'aider. J'en fais partie et je te rends ce service avec ce guide.

Tout d'abord, tu seras amené à programmer seul dans ton coin sur un projet personnel ou tout simplement parce que tu en as envie. Mais ce qui est important c'est que tu dois garder en tête que tôt ou tard tu auras besoin d'aide parce que tu seras ralenti par un problème que tu auras du mal à résoudre. Vouloir résoudre son problème tout seul dans son coin est très honorable d'autant plus que tu peux monter en compétence assez rapidement malgré le temps que tu risques d'y passer. Car se débrouiller seul et être un autodidacte sont de très bon avantages en programmation mais si tu as la possibilité d'être aidé par quelqu'un pourquoi s'en priver ?

Lorsque j'ai débuté la programmation dans le monde professionnel, j'étais seul et je ne pouvais compter que sur moi-même. Pourquoi ça ? Tout simplement parce que j'étais le 1er développeur de l'entreprise et que le challenge à la sortie de mes études était de haut niveau : développer un ERP sur mesure. Je devais acquérir rapidement :

- des compétences dans certains langages de programmation web que je ne connaissais pas
- la philosophie de l'entreprise
- les méthodes de travail types de l'entreprise
- les flux métiers à intégrer dans l'ERP

Ça faisait beaucoup d'un coup car j'aurais aimé avoir un mentor sur qui compter et qui puisse me conseiller et m'aider dans les difficultés que j'ai pu rencontrer tout au long de ce périple. Bien entendu, j'ai réussi à les surmonter seul même si à plusieurs reprises l'abandon et l'échec m'avaient traversé l'esprit mais au final ai-je appliqué les bonnes pratiques ? Ai-je fait les choses correctement ? Pas du tout ! Aurais-je pu faire les choses autrement et plus efficacement ? Avec le recul sans aucun doute : oui !

C'est indéniable sur le fait que tu monteras plus vite et plus rapidement en compétence si tu as un mentor sur qui compter et qui pourra parcourir le code que tu fais. J'en sais quelques choses puisque je gère une équipe tous les jours et j'ai dû les faire monter rapidement en compétence pour qu'ils soient les plus autonomes possibles. Cela implique aussi qu'un mentor doit voir le code de ses développeurs.

Tu as peut-être peur que l'on voit ce que tu fais ? C'est tout à fait normal mais poses-toi cette question : comment puis-je savoir si ce que je fais est bien réalisé ? Si tu programmes seul dans ton coin sans soumettre ton code à la critique, comment être sûr que ce que tu fais est bien ? Et comment peux-tu t'améliorer ? Je n'ai pas eu cette chance à mes débuts et j'aurais voulu avoir cette chance, je te l'assure !

Alors qu'il n'est pas obligatoire d'avoir un mentor, il est vivement recommandé de ne pas hésiter à s'adresser à la communauté. Le plus simple est internet en parcourant des forums spécialisés ou tout simplement en consultant des blogs de développeur. Une autre façon de s'adresser à la communauté pour t'aider et monter rapidement en compétence, c'est d'en rencontrer une partie physiquement. Comment ?

C'est très simple : en participant à des séminaires et/ou des événements de développeur. Il n'y a rien de plus enrichissant que ça, non seulement tu rencontreras des développeurs de tous niveaux mais tu rencontreras des experts dans leur domaine qui te mettront à disposition leurs sciences infuses et leurs connaissances lors de leurs interventions. En plus d'apprendre énormément de choses, tu ressortiras de l'événement avec le plein de motivations et énormément d'idées et de créativité. C'est pour moi la meilleure façon et la plus rapide pour en apprendre plus sur la programmation plutôt que d'apprendre seul dans son coin.

CHAPITRE 10

CONCLUSION

Nous sommes à la fin de ce guide et j'espère avoir répondu à une partie de tes interrogations en t'apportant des réponses concrètes basées sur mon vécu et mes connaissances. **Peut-être as-tu d'autres frustrations concernant la programmation dont je n'ai pas mentionné et que tu voudrais que j'aborde avec toi ?**

Tu peux m'envoyer tes principales frustrations et questions sur mon adresse e-mail : **gaetan.cottrez@apprendre-la-programmation.net**. N'hésite pas à sauter le pas en me contactant, je serais très ravi de pouvoir t'aider.

Et maintenant ? Prends le temps de **répondre aux différentes questions que je t'ai posées** dans ce guide pour déterminer ton but et tes objectifs de programmation. C'est vraiment **très important pour toi** pour bien commencer. Tu sauras de façon plus précise **jusqu'où tu veux aller**, à quel **rythme approfondir la matière**, le **type de langage de programmation** par lequel commencer mais surtout savoir si cela te branche toujours autant d'apprendre la programmation.

Et ensuite ? Je te conseille de **débuter la programmation** en apprenant les bases et pratiquant. De t'amuser et prendre du plaisir à le faire. Je donne **beaucoup de méthodes et de bonnes pratiques** autour de la programmation. Je te conseille donc de me suivre sur les réseaux sociaux pour être au courant de mes nouveaux articles afin de t'accompagner dans ton apprentissage et tes débuts.

Je te dis à très bientôt sur mon blog et je te remercie de t'être inscrit à ma newsletter.

Et n'oublie pas ! **Début** la programmation avec envie et programme avec passion !



Gaëtan Cottrez

Apprendre la programmation, le blog

CHAPITRE 11

COPYRIGHT

Ce livre numérique est protégé par le droit d’auteur. Tous les droits sont exclusivement réservés à Gaëtan Cottrez et aucune partie de cet ouvrage ne peut être republiée, sous quelque forme que ce soit, sans le consentement écrit de l’auteur. Vous n’avez pas aucun des droits de revente, ni de diffusion, ni d’utilisation de cet ouvrage sans accord préalable de l’auteur. Vous ne disposez d’aucun droit de label privé. Toute violation de ces termes entraînerait des poursuites à votre égard.

