

Apprendre par la pratique

Aujourd'hui, nous allons parler de « Learn By Doing ». Ce sujet est extrêmement important. L'apprentissage est crucial, et l'importance d'apprendre à apprendre va de soi. J'ai découvert que pour créer des choses, la meilleure méthode est d'apprendre en faisant et de faire en apprenant. Par exemple, si je veux maîtriser le développement Android, je dois vraiment me lancer dans la pratique du développement Android, en réalisant un projet après l'autre, chacun présentant des défis.

Paul Graham explique comment les étudiants en informatique devraient passer leur vie universitaire. Résoudre un grand nombre de problèmes difficiles. D'une part, en résolvant des problèmes difficiles du début à la fin, on peut apprendre énormément de choses. Par exemple, en réalisant un projet très ambitieux, comme écrire un client WeChat. Si on fait tout soi-même, on devrait, à la fin, être capable de traiter des images avec Photoshop et maîtriser tous les aspects d'Android. Si on n'utilise pas de service de notification tiers, écrire soi-même un serveur de communication en temps réel permet d'apprendre encore plus.

Après avoir lu un livre, nous avons souvent l'impression d'avoir globalement compris son contenu. Mais cela est loin d'être suffisant. Lorsque j'ai terminé la lecture de *"Crazy Android Lecture Notes"*, je pensais avoir acquis une bonne compréhension de la programmation Android et que j'avais maîtrisé l'essentiel du livre. J'ai passé environ 10 jours à lire les centaines de pages de ce livre. Ensuite, j'ai consacré 4 mois à maîtriser les connaissances qu'il contenait. Ce n'est qu'après avoir réalisé plusieurs petits projets et un projet plus important que j'ai commencé à me familiariser un peu plus avec le développement Android.

En d'autres termes, les connaissances acquises en lisant des livres sont loin d'être suffisantes. La lecture est superficielle. Bien sûr, comme je n'ai passé que 10 jours à lire, ce n'est pas non plus très chronophage.

Je me suis dit que peut-être la majorité du temps dans l'enseignement secondaire, collégial et primaire était gaspillée. Tout cela repose sur une éducation basée sur les livres.

La dernière fois, j'ai discuté avec mon oncle de la manière dont la chaux est fixée sur les murs. En fait, je ne suis toujours pas sûr. Est-ce que CaO et O_2 s'oxydent pour former CaCO_3 , et ensuite CaCO_3 est un solide, donc assez dur ? Dans la vie quotidienne, je ne fais pas bien la distinction entre CaO et CaCO_3 , même si je peux équilibrer leurs réactions chimiques. Grâce à la pratique et aux exercices, je sais bien ce qu'ils peuvent devenir et quelles sont leurs propriétés. Mais rapidement, j'oublie. Cela fait presque un an et demi que j'ai quitté le lycée, et les connaissances que je peux me rappeler activement ne sont plus très nombreuses. J'ai

obtenu 590 points au Gaokao, mais si je devais le repasser maintenant, je pense que je n'aurais que 300 points.

Heureusement, pendant mes trois années de lycée, j'ai consacré beaucoup de temps à apprendre les algorithmes et la programmation. Je n'ai pas passé tout mon temps à jouer à des jeux ennuyeux sur papier, comme les livres et les examens. Mais malheureusement, je n'ai pas appris plus de programmation à l'époque.

Bien que le processus d'apprentissage des matières scientifiques comme la physique, la chimie, la biologie, ainsi que des langues et des mathématiques, m'ait forgé et ait renforcé ma patience et ma persévérance pour atteindre la perfection à maintes reprises, ses inconvénients sont également évidents. Il vous fait percevoir l'apprentissage comme une souffrance, les connaissances comme ennuyeuses, et tout cela uniquement pour les classements et les notes élevées. Cela affecte ensuite vos intérêts naturels, votre curiosité innée, et vous fait, même sur le chemin de la programmation, vous préoccuper des classements et de savoir si vous êtes meilleur que untel ou untel, sans profiter du plaisir intrinsèque de la connaissance elle-même.

TJ (vision media) a dit qu'il ne va jamais à l'école et ne lit jamais de livres. Il apprend la programmation en regardant le code des autres, puis il est curieux et veut toujours comprendre pourquoi.

Paul Graham dit que les adultes ambitieux sont ceux qui concentrent leur curiosité sur un petit domaine.

Plus les connaissances sont complexes, moins elles peuvent être acquises par la motivation des gains matériels ou par la simple volonté, mais seulement par une curiosité naturelle et spontanée.

Depuis mon plus jeune âge, j'ai acquis des connaissances principalement pour la renommée, les classements et pour être meilleur que les autres. J'ai un grand besoin de reconnaissance extérieure. Bien sûr, j'aime aussi assez la programmation en soi.

Ce qui me motive principalement, c'est que, comparé à certains de mes amis, je suis encore très médiocre. Ils sont vraiment impressionnants, du même âge que moi, en deuxième année d'université, 20 ans. L'un d'eux écrit des compilateurs en dehors de ses loisirs et travaille très bien dans une entreprise de sous-traitance. Un autre ami possède deux entreprises et occupe un emploi à temps plein où il s'occupe du développement back-end. Et puis il y a les camarades de Tsinghua que je connais, qui remportent toutes sortes de prix internationaux.

Il est évident que je ressens un écart considérable. Ce n'est pas qu'ils ont commencé à apprendre avant moi ou quoi que ce soit, mais plutôt qu'ils sont plus efficaces et passent chaque jour plus de temps que moi.

Il faut dire que la pression des pairs et la motivation de vouloir surpasser les gens de son âge sont énormes. Il y a aussi des collègues très compétents. Mais inconsciemment, je pense qu'ils sont plus âgés que moi, et peut-être qu'à leur âge, je serai à peu près au même niveau. En y réfléchissant ainsi, cela ne me dérange plus vraiment.

Par conséquent, nous devons essayer de connaître des personnes du même âge que nous, mais qui sont plus compétentes. Bien que cela puisse être difficile, car cela nous oblige à constamment reconnaître nos propres limites, le résultat en vaut la peine. Tant que vous faites face à tout avec courage, sans vous décourager, que vous voyez les écarts et que vous travaillez discrètement, les résultats seront excellents.

Il y a peut-être encore plus d'avantages. Parce qu'ils sont doués, plus talentueux, plus compétents dans une certaine compétence, ils sont plus susceptibles d'être ceux qui aiment vraiment cette chose. Ils sont plus susceptibles d'être des passionnés de programmation. Par conséquent, cela apporte une influence positive sur eux-mêmes et leur permet de ressentir le plaisir de la programmation.

La programmation a ceci de merveilleux qu'une fois que vous vous y plongez, vous ne pouvez vous empêcher de l'aimer, oubliant ainsi les classements, vos propres compétences, ainsi que la renommée et la richesse.

J'avoue que je consacre du temps à la programmation, d'abord parce que c'est mon travail et que je dois rendre des comptes à mes supérieurs, ensuite parce que je me sens encore très novice et que si je ne m'améliore pas rapidement, il sera trop tard, et enfin parce que j'ai un rêve de devenir un expert en programmation et de lancer une startup full-stack. Mais parce que je m'investis, je ressens rapidement le plaisir de créer quelque chose de mes propres mains. La satisfaction de résoudre un problème après l'autre est incomparable.

Se mettre en action permet de mieux assimiler les connaissances et rend l'apprentissage plus intéressant.