

Quel langage de programmation apprendre lorsque l'on est soignant ?

L. BENICHOU(1)*, B. BENICHOU(2)

(1) Service de chirurgie maxillo-faciale et stomatologique
Groupe Hospitalier Paris St-Joseph
165 rue Raymond Losserand 75014 PARIS

(2) Wagram Dentaire
14 rue Brémontier 75017 PARIS

Introduction :

Notre métier est en pleine mutation.

Notre monde entièrement connecté offre l'opportunité de créer de nouveaux outils exploitant les données relevées en temps réel sur chacun des individus.

Alors qu'aujourd'hui elle servent essentiellement à vendre des brosses à dent "révolutionnaires" en drop shipping !

Imaginez la différence de prise en charge d'un signe fonctionnel aussi banal que la fièvre si une alerte nous informait que le GPS du smartphone du patient a relevé un séjour en pays endémique de paludisme.

La première génération de soignants "millennials" vient d'arriver sur le marché et les futur soignants n'auront pas connu le monde avant la révolution numérique, l'omniprésence des smartphone et l'hyper connexion.

Dans ce paysage numérique, le soignant devra avoir un rôle dans la conception et le développement des outils de demain.

En 2012 on comptait pas moins de 4000 applications médicales ou concernant la santé (1), malheureusement assez peu de soignants ne sont impliqués dans leur développement.

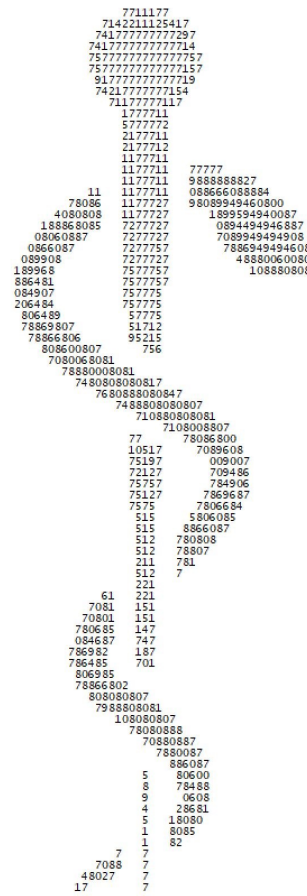
L'idéal est que le médecin soit impliqué dans toutes les phases du développement et de conception de ces outils. De la même façon qu'il l'est dans la recherche expérimentale, ou dans la recherche clinique.

On n'imagine pas un médecin participi à de la recherche clinique sans notion de statistique (d'ailleurs cette discipline est enseigné dans toutes les facultés de médecine), de la même façon il est logique que les soignants impliqués dans le développement des applications numériques connaissent les bases du codage, et des outils informatiques mis en oeuvre.

Ils auront également un rôle dans l'évaluation de ces outils à travers d'essais et d'études scientifiques ce qui manque cruellement aujourd'hui dans le domaine des applications médicales.

Enfin, ayant prêts serment ils sauront également les gardiens du secret médical mis en danger par les échanges de données de santé inhérents au fonctionnement de ces outils.

Afin de reprendre la main sur ces sujets, il est important que les soignants connaissent les bases de la programmation. Mais le paysage du codage présente de nombreux langages, outils et notions souvent obscures pour le soignant.



Spécificité du langage recherché

Simplicité

Il faut que le langage utilisé par les soignants soit simple. Simple à apprendre, simple à utiliser, simple à mettre en œuvre et simple à déployer.

Il existe 2 philosophies dans les langages de programmation : Les langages compilés et les langages interprétés.

Les langage compilés

Ils doivent être compilés (traduit) en langage machine par un compilateur pour pouvoir être compris par l'ordinateur.

Ils ont l'avantage d'être plus performant et rapide.

Le plus connu et utilisé est aujourd'hui le C++, mais d'autres langages tel que Rust ont de plus en plus de succès.

Les langage interprété

Contrairement au langage compilé, le langage interprété est traduit en temps réel par la machine.

Ceci a pour désavantage de nuire légèrement à la performance du programme. Mais en contrepartie, ils sont infiniment plus simples à mettre en œuvre, à configurer et à utiliser.

Même si les puristes préféreront un langage compilé, les performances des outils actuels font que les langages interprétés sont devenus extrêmement performants.

La balance performance / difficulté penche nettement vers les langage interprété

En tant que soignant, il faudra s'orienter vers un langage **interprété**

Bibliothèques et gestionnaire de paquet

Comme je l'ai précisé précédemment, ce n'est pas tant langage de programmation qui importe que les différents modules ou bibliothèques qui y sont associés.

Le langage de programmation choisi devra être associé à des bibliothèques compatibles avec la pratique de la médecine.

Ces bibliothèques traiteront en particulier de la cryptographie, de la gestion de base de données, du traitement d'image, ou même de la gestion de fichier type dicom consacré à l'imagerie médicale.

Plus important encore, ces bibliothèques devront être facilement accessibles et facilement importées.

Nous privilégierons donc un langage de programmation possédant ce que nous appelons des gestionnaires de paquets.

Il s'agit d'une commande permettant d'importer n'importe quelle bibliothèque grâce à une base de données dédiée.

Les plus connus sont NPM pour le javascript, PIP pour le python, ou même COMPOSER pour le PHP.

Nous privilégierons donc pour le soignant à langage riche en bibliothèque et possédant un **gestionnaire de paquets**

Communauté

Il faudra également choisir un langage possédant une grande communauté d'utilisateurs.

En effet, il n'est pas rare lors de développement de application ou logiciel que l'on soit confronté à des problèmes auxquels il est difficile de trouver une solution.

Une communauté nombreuse permet d'augmenter les probabilités de trouver un autre développeur ayant rencontré exactement le même problème et pour lequel il a publié la solution.

Dans la communauté des développeurs il existe tout un écosystème de forum, et de site dédié aux problèmes de langage où se réunit une **communauté** très importante.

Mieux encore, c'est la communauté qui implémente de nouvelle bibliothèque au langage. Lorsqu'un développeur trouve une solution à un problème en créant des modules pour ses propres projets, il n'est pas rare qu'il le partage dans les bases de données des bibliothèques.

Une communauté nombreuse permet l'implémentation de nombreuses bibliothèques pour de nombreux cas de figure.

Typiquement les langages les plus populaires en 2021 sont le python et le javascript qui réunissent une communauté incroyablement nombreuses

Cross-plateforme

Vous n'êtes pas sans savoir qu'il existe de nombreuses plateformes pour accueillir les solutions que nous créerons dans l'avenir.

Le développement des logiciels pouvant être lu par les différents systèmes d'exploitation ne se fait en général pas avec les mêmes langages.

Pour être synthétiques Windows, Mac, Linux, Android, chacun ont des langages de prédilection.

L'idéal est de partir sur un langage compatible avec le plus de plateformes possible.

Ceci à deux avantages tout d'abord maîtriser le mieux possible le langage sans se disperser vers une multitude de langages à étudier.

Par ailleurs le temps humain utilisé pour le développement des différentes applications n'en sera que réduit et donc le coût de la solution sera également réduit d'autant.

Il faut bien être conscient qu'en matière de développement ce qui coûte cher c'est le temps passer à coder plus que le matériel.

Le langage idéal pour le soignant devrait donc être **cross plateforme**.

polyvalence

De la même façon, il faudra employer le langage le plus polyvalent possible afin de pouvoir couvrir plus de cas de figures adapté à notre pratique.

Il n'y a pas si longtemps il était indispensable de connaître une multitude de langages pour couvrir la totalité d'un projet.

Les choses ont changé depuis l'apparition de ce qu'on appelle le développement full stack.

Typiquement un langage comme JavaScript permet aujourd'hui de couvrir la programmation du serveur, du navigateur ou même de créer des applications natives pour Android et iOS.

Le langage choisi dépend du projet

Quel langage pour quel projet ?

Data science

Un autre type de travail de recherche qui effectuée en science de la santé est l'exploitation des données médicales, et notamment l'utilisation d'algorithmes d'intelligence artificielle.

La communauté des développeurs est assez unanime sur le fait que lorsqu'il s'agit de data science, le langage de prédilection sera le python pour sa simplicité, et sa richesse de bibliothèque de Deep learning.

D'autres langages se développent, en particulier R.

La bibliothèque de Deep learning tensor Flow est également en cours d'adaptation pour JavaScript.

Echange de données

Il s'agit en général de plateformes de messagerie, de prise de rendez-vous, de stockage de données des dossiers médicaux, de partage de données pour avis.

Bref tout ce qui permet d'échanger des informations d'un individu à un autre.

Pour ce genre de projet, il faudra absolument se pencher vers un langage axé sur le web, notamment permettant de créer des serveurs, des bases de données et permettant de manipuler le navigateur.

Ces langages sont nombreux, du côté navigateur il sera indispensable de connaître JavaScript (en association au HTML5 et CSS3), du côté serveur beaucoup plus d'options existent, les plus populaires étant le PHP, le python ou le javascript avec NodeJS.

Outils annexes

Le langage de programmation comme nous l'avons vu plus haut n'est pas seulement la syntaxe mais tout l'écosystème qu'il y a autour, notamment la communauté.

Il existe aujourd'hui pléthore d'outils de communication afin de pouvoir mener des projets à plusieurs.

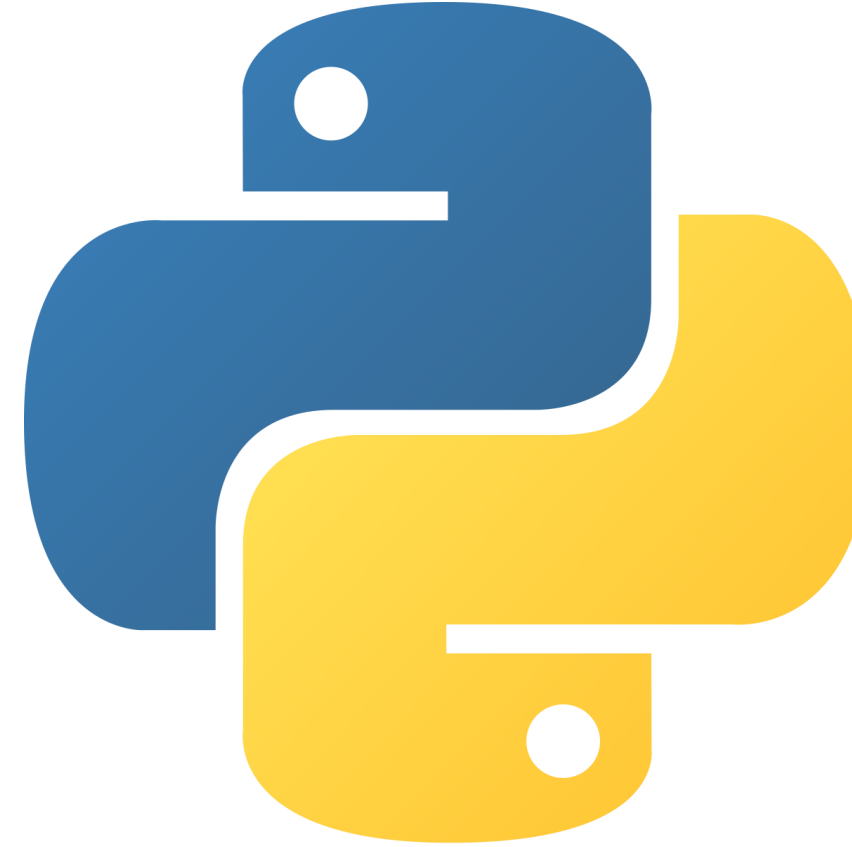
Je pense particulièrement aux outils de versionning comme GIT qui devront être absolument étudiés lors de l'apprentissage de la programmation moderne.

Il faudra également s'intéresser à tout ce qui est administration système si vos projets concernent le développement du web.

Aujourd'hui la barrière entre développeur et administrateur système est en train de s'effondrer grâce à une nouvelle approche que je ne développerai pas ici nommée le DevOps basé sur l'utilisation de conteneur.

Résultat

Cette réflexion n'engage que moi et bien sûr j'apporte simplement ma conviction sur le sujet afin de permettre d'ouvrir une discussion. Mais je pense qu'en tant que soignant il faut s'intéresser particulièrement à deux langages selon le projet que l'on souhaite développer.



Projet d'application web ou mobile

Il s'agit là de réaliser des applications basées sur l'échange d'information, ou le stockage d'informations sur le Cloud. Dans ces cas-là, mon langage de prédilection sera un langage qui permettra de travailler sur navigateur (en frontend) et également au niveau du serveur (en backend).

L'idéal étant d'utiliser le même langage pour ces deux aspects.

JavaScript me semble alors le langage le plus adapté, en effet il permet de programmer tous les navigateurs et permet depuis peu de travailler au niveau du serveur grâce à NodeJS.

Par ailleurs, JavaScript permet également de travailler, grâce à des bibliothèques très puissantes, sur les bases de données notamment les bases de données non relationnelles type mongoDB parfaitement adaptées pour le stockage et récupération de grande quantité de données.

JavaScript présente aussi l'avantage de permettre de travailler sur les applications mobiles, en effet aujourd'hui certaines bibliothèques type React natif permettent de convertir JavaScript en langage adapté au système d'exploitation mobile Android ou iOS.

Les Data Science

En revanche, si nous voulons travailler sur intelligence artificielle et exploitation de grandes quantités de données (Data science), le langage le plus adapté sera le python (2).

Les bibliothèques intelligence artificielle et de manipulation d'image de Python sont particulièrement puissantes et bénéficient d'une communauté et d'un soutien des industriels énorme.

Par ailleurs, le python est un langage simple et très polyvalent. Il permet également de réaliser des applications de bureau sur les différentes plateformes (Apple, Windows et Linux).

Il peut à peu près être utilisé pour toutes les indications, en dehors de la programmation au niveau des navigateurs web et des applications mobiles.

Conclusion

Je pense qu'en tant que soignant il y a deux langages qui peuvent nous intéresser : JavaScript et Python.

Il s'agit pour tous les deux de langage interprété extrêmement simple, avec une richesse de bibliothèque très importante et gérer de manière très simple grâce à des gestionnaires de paquets. Ces langages bénéficient également d'une communauté absolument gigantesque.

Je crois beaucoup en l'avenir des applications Web dans notre métier.

Les navigateurs permettent aujourd'hui de réaliser des choses tout à fait intéressantes au niveau de l'ergonomie, de la simplicité des échanges de données.

Ils sont devenus également très puissants et permettent maintenant d'afficher des vidéos, des images en 3D de manière très performante, permettant même et les échanges temps réel avec le webRTC.

Elles sont cross plate-forme et peuvent être utilisées d'un ordinateur à un autre grâce à un simple système d'authentification.

Bibliographie

- Rathbone AL, Prescott J. The use of mobile apps and SMS messaging as physical and mental health interventions: systematic review. J Med Internet Res. 2017 Aug 24;19(8):e295.
- Pittard WS, Li S. The essential toolbox of data science: python, R, git, and docker. Methods Mol Biol. 2020;2104:265-311.