Cours: Science des données avec Python

Programmation spécialisée en Python scientifique et numérique

Apprendre à utiliser les modules scientifiques et numériques du langage Python. La spécialisation Python scientifique et numérique traite autant les données numériques, textuelles que vectorielles (géographiques) ; d'où l'appellation science des données.

Le Python scientifique et numérique est différent du développement web, d'applications, de GUI ou de l'administration de systèmes. Toute personne avec une base minimale en Python peut entreprendre le cours puisqu'il s'agit d'une spécialité à part entière (consulter le site pour des vidéos de démonstration).

Le contenu du cours s'apparente au traitement, à la modélisation et à la visualisation de données tels qu'il est fait avec des tableurs, des bases de données SQL et NoSQL, des logiciels comme Tableau ou d'autres outils de l'informatique décisionnelle. Chaque section du cours donne une idée d'ensemble des possibilités.

À l'utilisateur de se spécialiser par la suite. Le Python scientifique et numérique vient bonifier le savoir-faire analytique de la personne et les autres compétences de son domaine d'expertise. Le contenu cours s'adresse donc autant à des métiers techniques (cégep) qu'universitaires.

Sommaire

Objectif du cours

Apprendre la programmation en Python ; très spécifiquement, apprendre le Python scientifique et numérique.

Le Python scientifique et numérique

Python offre de nombreuses possibilités (Figure 1). Une d'entre elles est le calcul et l'analyse des données. Nous nommons cette spécialité : science des données.

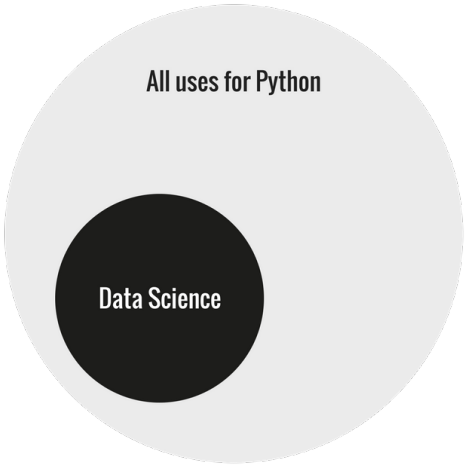

Figure 1 – *Usages du Python*

Pour ce faire, nous utilisons des objets, des fonctions et des méthodes (mathématiques, statistiques, algébriques, etc.) d'une série de modules scientifiques, ou *Scipy Stack* pour *Scientific Python Stack*. Cet ensemble inclut aussi la visualisation avec le module Matplotlib (et d'autres). Nous traitons les calculs avec le module Numpy, pour *Numerical Python*.

La science des données avec Python ou le Python scientifique et numérique correspondent à la même chose (Figure 2).

Figure 2 – *Python scientifique et numérique*

Ses applications

Le Python scientifique et numérique est un complément (ou un substitut) aux tableurs (limités en taille et vitesse de calcul), aux bases de données SQL et NoSQL (peu flexibles et incapables de faire de la visualisation) et à certains logiciels de calcul (en géomatique, par exemple, le logiciel QGIS possède une console Python intégrée pour l'automatisation de tâches).

Le Python scientifique et numérique est libre. Les licences sont ouvertes. Nous pourrions ainsi remplacer une série d'outils informatiques payants et faire des économies sur les licences. Par exemple, nous pourrions remplacer un tableur qui fait des tableaux croisés dynamiques et des graphiques, un logiciel de présentation, des outils pour faire des requêtes sur une base de données relationnelle et gérer des cubes OLAP, une série de logiciels pour le reporting, la visualisation et la conception de tableaux de bord.

Le Python scientifique et numérique peut aussi traiter les mégadonnées moyennant quelques ajouts technologiques (calculs distribués, services infonuagiques ou logiciels comme Spark).

Le Python scientifique et numérique permet le traitement statistique avancé et l'apprentissage statistique ou *machine learning*. Il ouvre la porte à des techniques plus sophistiquées comme les réseaux de neurones et l'apprentissage profond.

Le Python scientifique et numérique donne une base de programmation pour progresser en développement Web, de logiciels, d'applications, de GUIs, etc.

La clientèle cible

Nous définissons la clientèle cible à la fois par son background et ses attentes.

Les backgrounds typiques

Il faut avoir utilisé ou être familier avec **quelques uns** des concepts qui vont suivre. En d'autres mots, il existe des cheminements professionnels qui justifient et facilitent l'apprentissage du Python scientifique et numérique.

- Avoir une formation technique (cégep) ou universitaire dans un secteur à nature scientifique : sciences pures sciences de la santé, sciences appliquées et génie.
- Avoir une formation technique (cégep) ou universitaire dans un secteur quantitatif : sciences sociales et administratives avec une concentration en méthodes quantitatives (économie, sciences comptables, markéting quantitatif, géographie, etc.) ;
- Connaitre d'autres langages de programmation et/ou la programmation à même certains logiciels (logique, algorithmie), utiliser des logiciels à base de calculs, d'analyse ;
- Être familier avec un terminal et ses commandes ou une interface par lignes de commandes (UNIX, terminal de Mac OS X, bash de Linux, console ou PowerShell de Windows) ;
- Utiliser des logiciels de calcul avec souris ou pointer-cliquer (logiciels mathématiques, statistiques, d'optimisation ou de recherche opérationnelle) ;
- Utiliser des méthodes quantitatives telles les mathématiques, la statistique, l'informatique ou des méthodes computationnelles dans des contextes de données non numériques ou non structurées (l'analyse de texte et du langage naturel, le travail dans les médias sociaux, le web, l'expérience utilisateur, etc.) ;
- Utiliser des bases de données non structurées ou NoSQL, des fichiers vectoriels et JSON.

Secteurs et métiers d'application

Formations techniques (cégep) et universitaires.

- Administration de bases de données, architecture de données, analyse d'entrepôts de données ;
- Analyse de médias sociaux, de médias numériques, de plateforme de services numériques ;
- Biologie (biométrie), biostatistique, écologie, sciences environnementales ;
- Chaine logistique, données de production, de qualité, de distribution ;
- Commerce en ligne ;
- Conception de plateformes de service, mesure d'audiences, de base d'abonnés ;
- Conception, développements Web, d'applications, de GUI, de logiciels ;
- Extraction de données du Web ou d'autres entrepôts de données (wikis, bibliothèques, intranets) ;
- Finance, assurance, économie, sciences comptables ;
- Géomatique, géographie, cartographie ;
- Gestion analytique du Web, ciblage publicitaire sur le Web ;
- Gestion de données scientifiques, grappes de calculs ;
- Gestion de données sur des architectures virtuelles, infonuagiques ou distribuées ;
- Histoire, humanités numériques ;
- Informatique décisionnelle (BI) ;
- Internet des objets, microcontrôleurs ;
- Journalisme numérique ;
- Markéting quantitatif, relation client, analyse quantitative de la clientèle et prospection (CRM), ventes ;
- Mesure et évaluation de politiques publiques, comparaisons de groupes, de régimes ;
- R&D ;

- Recherche d'informations avec API, automatisation avec moteurs de recherche Web ;
- Sciences médicales et pharmaceutiques, tests ;
- Trading à haute fréquence ;
- Traitement du langage naturel (textes, base de connaissances, courriels, messagerie, chat, commentaires) ;
- Transport, recherche opérationnelle ;
- Utilisation d'outils intelligents, connectés ;
- Visualisation, graphisme et cartographie.

À quoi s'attendre

La clientèle cible doit aussi comprendre la portée du cours, à quoi il mène.

D'abord, le cours est purement technique. Par exemple, il montre comment agréger un jeu de données selon des catégories et à calculer une moyenne sur chaque catégorie. Cependant, il ne forme pas la personne du point de vue analytique : comment approcher un problème, pourquoi agréger ces catégories de telle manière, pourquoi telles fonctions statistiques, comment interpréter les résultats, etc. Cet aspect du travail relève du background analytique de la personne.

Ensuite, le cours montre la programmation procédurale. Plutôt que de travailler dans un logiciel avec un souris, de refaire les mêmes étapes cas après cas, nous montrons comment analyser des données avec des lignes de code, avec un script de calcul qui automatise une bonne partie de l'analyse.

Le cours dote la personne d'un nouvel outil de travail pour l'aider dans ses tâche. Le cours ne forme pas la personne à une nouvelle profession. Par analogie, ce n'est pas un cours d'électricien, mais un cours sur l'utilisation d'outils électriques pour être plus productifs dans son métier d'électricien.

Par exemple, une biologiste environnementale maitrise quelques logiciels (techniques), des méthodologies de travail (analytique) et possède des connaissance en sa matière (savoir-faire). Elle veut s'investir dans la science des données avec Python pour adopter des outils de calcul plus puissants pour remplacer certains de ses logiciels et renforcer sa gestion de bases de données alimentées par une myriade de stations d'échantillonnage.

De même que pour un technicien en génie électrique qui veut analyser les données de capteurs sur une chaine de montage, une historienne qui veut cartographier une quantité d'évènements qu'elle a colligé, un marketer qui veut analyser ses cohortes de clients pour chaque initiative markéting, une analyste du service à la clientèle qui veut systématiser l'évaluation du sentiment général des commentaires reçus, etc.

Préparation au cours

Il est possible d'acquérir une base minimale en Python pour passer au Python scientifique et numérique. Il est possible d'éviter des cours plus complets qui mènent au développement Web ou de logiciels. Il sera toujours possible d'y retourner par la suite avec un nouveau bagage de connaissances en Python scientifique et numérique.

La base à maitriser se compose de cinq thèmes.

1. Variables et types de données (conversions), opérateurs ;
2. Structures de données et notation ;
3. Fonctions et méthodes de base ;
4. Contrôles de flux (`if elif else`) ;
5. Boucles (`for, while`) ;

Grosso modo, elles se résument à ceci (Figure 3), tout en évitant la programmation orientée objet (`classe`) :

La plupart des cours Python poursuivraient sur la programmation orientée objet. Pour poursuivre en Python scientifique et numérique, nous pouvons stopper avant, pour y retourner plus tard.

Par contre, nous ajoutons quelques savoirs essentiels à la programmation avec Python.

6. Combiner des boucles et des contrôles de flux ;
7. Importer des modules ;
8. Maitriser la syntaxe Python et les meilleurs pratiques du code.

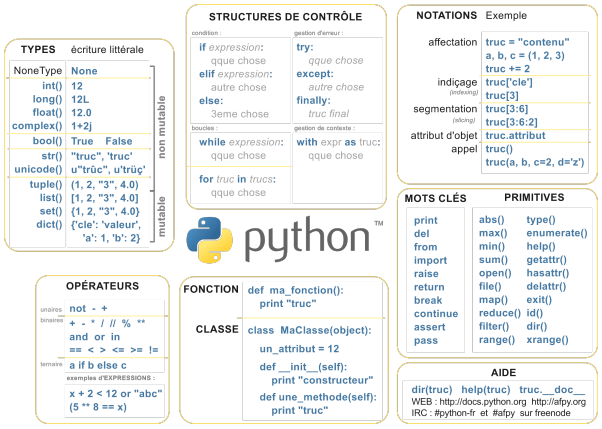

Figure 3 – *Cinq thèmes*

Pour en connaitre davantage sur cette préparation, consulter la page du site.

Éviter les confusions

Comme nous l'avons défini au tout début, la science des données avec Python est synonyme de Python scientifique et numérique.

Science vs scientifique

La science des données est un vaste domaine. En plus du langage Python, la science des données fait usage de bien d'autres langages tels R, SQL, Julia, Scala, etc.

Le cours ne porte que sur la science des données avec Python.

Il ne s'agit pas non plus de former un scientifique de données. Cette profession comporte bien d'autres compétences techniques et analytiques.

Par contre, un cours en science des données avec Python s'intègre parfaitement dans un plan de formation plus large pour devenir scientifique de données.

Profil : scientifique de données

Le scientifique de données est un spécialiste de l'économie numérique, au croisement de l'informatique et de l'analyse statistique. Il possède une expertise scientifique de niveau universitaire et un savoir-faire dans un domaine d'activité économique qui lui permet d'appuyer la prise de décisions administratives.

Quelques images (Figures 4 et 5) valent plus qu'une description complète.

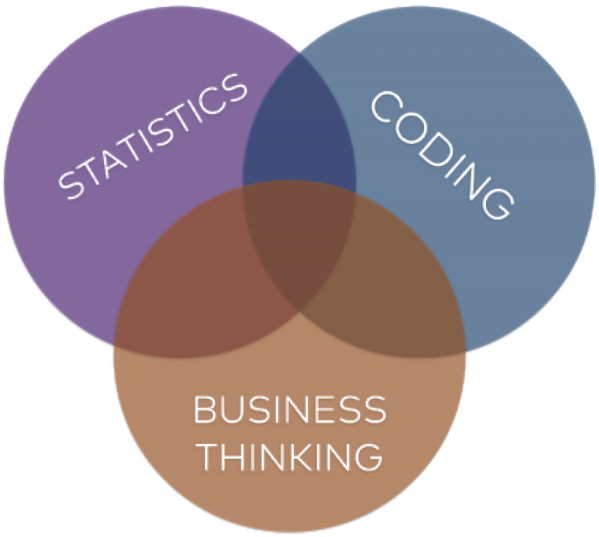

Figure 4 – *Compétences*

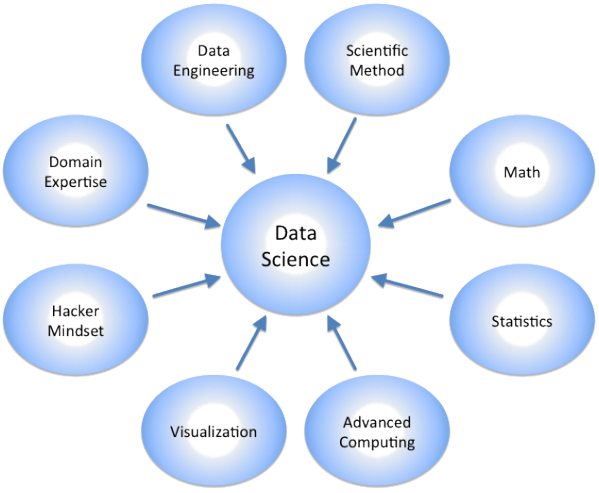

Figure 5 – *Compétences*

Pour en connaitre davantage, nous proposons cette définition.

Le langage Python

Le Python scientifique et numérique n'est qu'une des possibilités du langage Python. La science des données avec Python peut constituer un tremplin vers d'autres spécialités du langage comme cela peut être un complément pour un développeur Python.

Poursuivre en Python

Avec une base minimale en Python, un cheminement en Python scientifique et numérique mène naturellement vers la programmation orientée objet, d'autres domaines de spécialisation en Python et le développement de sites Web, de logiciels, de GUIs et d'applications.

Les développeurs qui maitrisent les autres spécialisations du langage Python ont un avantage, car ils connaissent déjà la syntaxe de Python sur laquelle est calqué le Python scientifique et numérique. En d'autres mots, il est plus facile d'apprendre un langue latine si nous parlons déjà une langue latine que si notre langue principale est d'origine asiatique.

Avec un bon background, une motivation liée à ses attentes et du travail, tout s'apprend.

Python : un langage générique

Nous présentons quelques spécialités du langage Python (Figures 6). Consulter l'Annexe.

- Administration de réseaux et communications : Twisted, PySerial), pybluez ;
- Administration de systèmes : Ansible, Salt, OpenStack ;
- Bioinformatique : Biopython ;
- Développement d'applications multi-touch : kivy ;
- Développement de GUI : tkInter, PyGObject, PyQt, PySide, Kivy, wxPython ;
- Développement de jeux vidéos en 2D : Pygame ;
- Développement de logiciels : Buildbot, Trac, Roundup ;

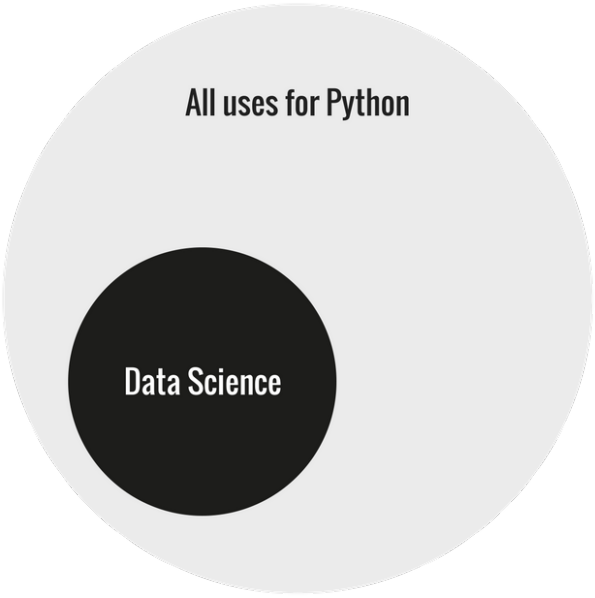

Figure 6 – *Usages du Python*

- Développement Web : Django, Pyramid, Bottle, Tornado, Flask, web2py ;
- Gestion de base de données : MySQLdb ;
- Python scientifique et numérique : SciPy, Numpy, Pandas, IPython, matplotlib ;
- Traitement d'images, de la vision artificielle par caméra : PIL, SimpleCV ;
- Traitement du son, de la synthèse vocale : eSpeak ;
- Et bien plus.

Les combinaisons de spécialités ouvrent beaucoup de possibilités !

Pédagogie

Le cours est échelonné sur plusieurs leçons de 3h pour un total de 36h. L'espacement est nécessaire pour permettre à l'étudiant de réviser les concepts, de travailler les exercices et les corrigés entre les cours. En fait, selon le niveau de l'étudiant, le travail hors classe en presque aussi important que le travail en classe : exercices, révisions, etc.

L'enseignement normal se déroulerait devant des groupes de 6 à 12 apprenants.

L'étudiant travaille sur des logiciels en nuage (moyennant un accès Internet). La plateforme Microsoft Azure Notebooks est un espace virtuel de stockage de données et d'hébergement de logiciels en ligne pour programmer en Python. Il est possible d'installer un environnement Python et les mêmes logiciels qu'en ligne sur un ordinateur personnel pour travailler sur son ordinateur en classe, mais le support technique n'est pas prévu dans le cours. La procédure d'installation sur ordinateur est expliquée sur le site.

Le premier cours est différent des autres, car il couvre le démarrage de l'environnement de travail en ligne. L'installation de cet environnement doit être complété par l'étudiant avant le premier cours. Ce n'est pas long : une inscription en ligne, puis une exploration de l'environnement suivant une documentation transmise par courriel. Le formateur prend donc contact avec son groupe avant le premier cours pour la préparation et l'envoi de fichiers préliminaires. Le premier cours permet de finaliser l'installation, de démontrer comment bien utiliser la plateforme de travail en ligne avant d'enchainer sur la matière.

Dès le premier cours, la feuille de route pédagogique entrecoupe la démonstration de quelques concepts et le travail dans le logiciel sur des morceaux de code et des exercices.

La programmation s'apprend par la pratique et les blocages, les déblocages. À tout moment, l'interaction étudiant-formateur est basée sur les questions-réponses. Cet aspect du cours est important : il corrige la faiblesse des cours de programmation en ligne. D'ailleurs, dès le deuxième cours, nous commençons la séance avec une période de questions-réponses, un retour sur les exercices (corrigés) et un déblocage des bogues. Le support informatique est crucial pour que l'étudiant progresse sans perdre son temps.

L'étudiant se voit remettre (par dépôt de données en ligne : Google Docs, Bitbucket, GitLab ou autres moyens de diffusion en ligne) le contenu de la matière tout au long du cours : les notes, les données, les exercices et les corrigés. Ce contenu est entièrement numérique. Le dépot est sécurisé.

Sous la supervision du formateur, l'étudiant suit le contenu du cours et exécute des calculs dans ses outils de programmation : l'interpréteur interactif Jupyter Notebook. Par la suite, l'étudiant développe ses propres notes sur la base de ses expérimentations et de ses exercices.

Licences

Le cours utilise le langage Python et un interpréteur interactif nommé Jupyter Notebook sur une plateforme en ligne : Microsoft Azure Notebooks. Sur ordinateur, on peut utiliser les mêmes outils en plus d'un Environnement de Développement Intégré (EDI) nommé Pyzo pour encadrer Jupyter Notebook local.

- Python est un langage libre : Python licence 2.0e. Cette licence est vulgarisée ici, L'interpréteur (par ligne de code) fait partie de l'environnement Python ;
- Il existe des modules Python tiers payants ou restreints, mais ils ne font pas partie du cours.
- L'interpréteur interactif Jupyter Notebook est un standard ouvert pour la recherche scientifique et l'éducation que les entreprises peuvent aussi utiliser sans frais ;
- La plateforme Microsoft Azure Notebooks fait partie d'Office Online. Un individu peut ouvrir un compte personnel dont il est responsable pour travailler où bon lui semble (comme avec Google Docs). L'étudiant travaille à partir de son compte en ligne sur Jupyter Notebook et utilise le langage Python en ligne.
- L'EDI Pyzo est doté d'une licence ouverte.

Autres cours et ateliers

Suivant le cours, il existe d'autres cours et ateliers de 4h à 12h.

- Visualisation en science des données avec Python ;
- Reporting en langage Python ;
- Mégadonnées et traitement de données massives ou *Big Data*).

Annexe A - Croissance de Python

La 4ème révolution industrielle et la montée des secteurs STEM (Figure 7) sont à l'origine de changements sur le marché de l'emploi.

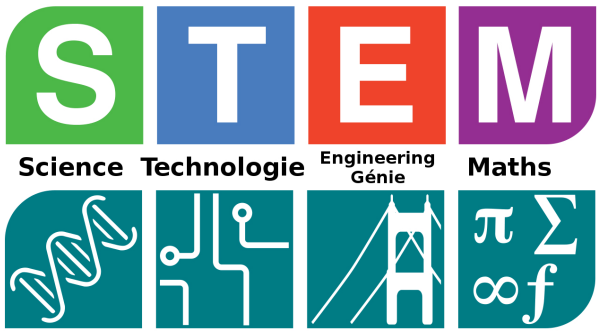

Figure 7 – *STEM*

Les besoins du marché de l'emploi au Québec (ou dans les pays de l'OCDE) suivent la tendance de certains écosystèmes avancés. Des marchés de l'emploi précurseurs comme Silicon Valley-San Francisco-Oakland-Fremont, Austin-Houston-Sugarland-Baytown au Texas et Seattle illustrent ce qui attend d'autres écosystèmes technologiques.

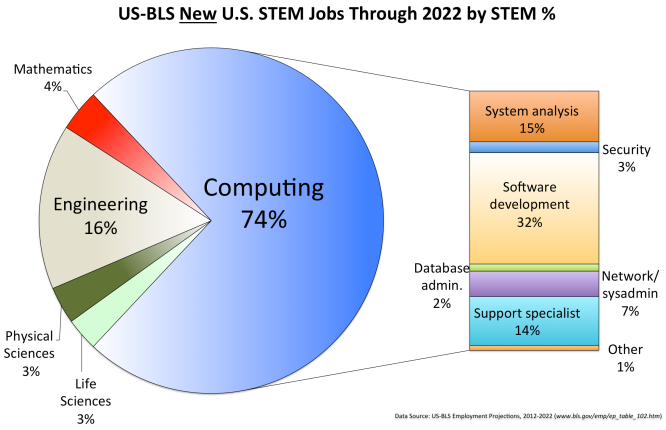

Figure 8 – *Marché*

La création d'entreprise et d'emplois provient et proviendra principalement des technologies de l'information (TI) (Figure 8 et Figure 9).

Donc, non seulement le scientifique, le technologue, l'ingénieur ou le mathématicien (STEM), mais d'autres profils doivent et devront maitriser les TI. Une bonne partie du savoir-faire en TI repose sur la programmation.

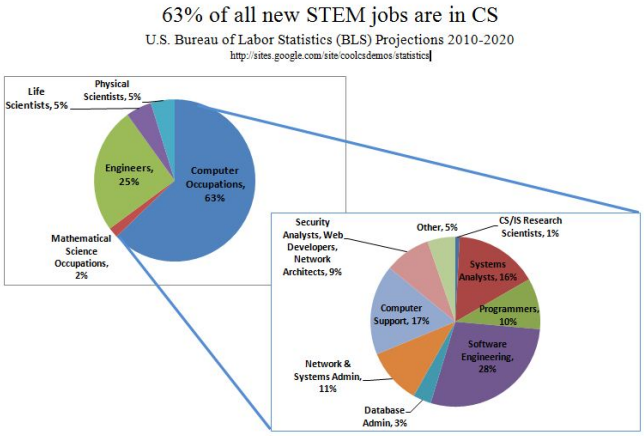

Figure 9 – *Marché*

Parmi tous les langages de programmation, le langage Python est celui qui enregistre la plus grande croissance, selon des estimations telles des requêtes sur des forums de discussion comme Stack Overflow (Figure 10) ou des firmes d'analyse tels TIOBE et RedMonk.

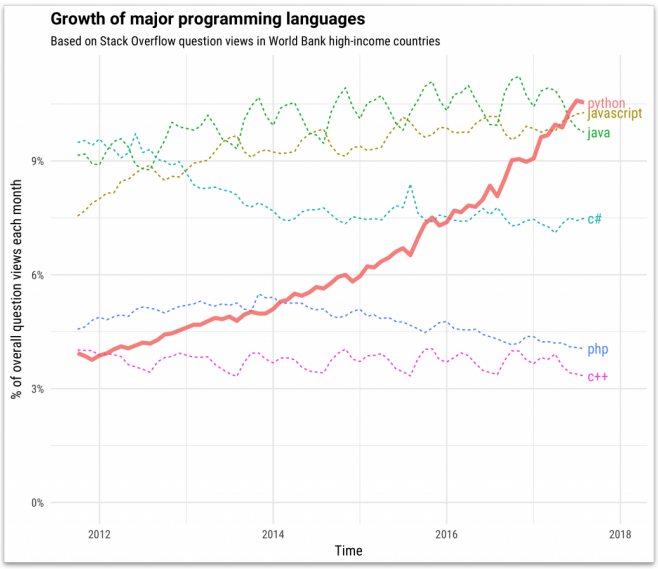

Figure 10 – *Python*

La percée de Python s'explique par sa simplicité (apprentissage et utilisation) et son universalité. Nous avons vu dans une section précédente comment Python est un langage générique.

En Python scientifique et numérique, les innovations amenées par des modules comme Pandas ont dopé l'adoption du langage Python

(Figure 11).

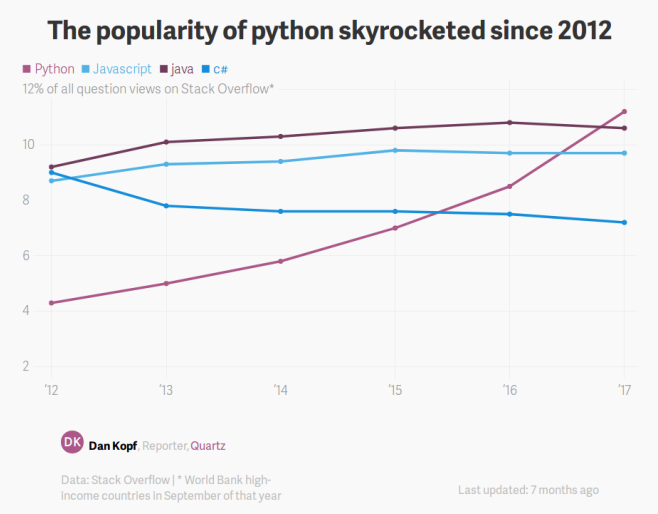

Figure 11 – *Python*

Règle générale (Figure 12), Python est plus simple que Java et Perl et plus rapide pour l'écriture de prototypes que C/C++/C#. Python peut remplacer PHP et Ruby pour concevoir des sites web. Python et ses modules de visualisation sont libres (gratuits) contrairement à MATLAB.

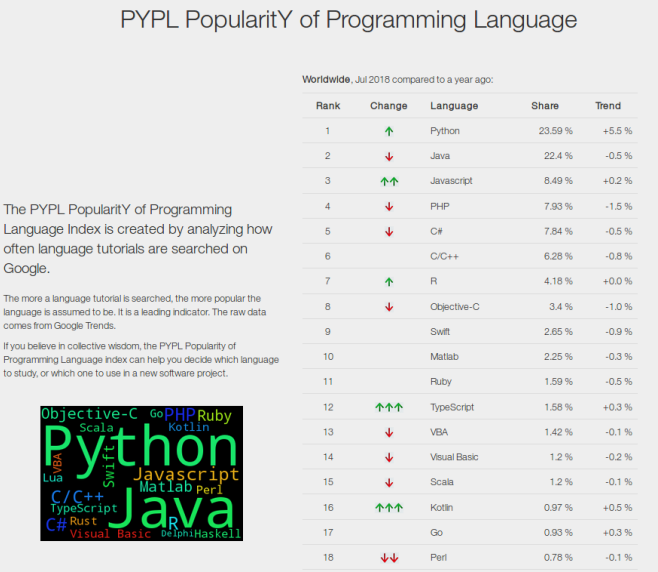

Figure 12 – *Python*

Python peut se substituer à d'autres langages spécialisés comme VBA et Visual Basic. Il peut remplacer SQL pour des requêtes sur des bases de données. Le module Pandas (encore) possède des méthodes pour modéliser des jeux de données. Python peut aussi réaliser de grands calculs sur des mégadonnées à la manière de Scala. Bien que ce dernier soit plus rapide, Python reste un langage générique qu'il est pratique d'apprendre pour sa polyvalence.

Annexe B - Plan de cours

- Jupyter Notebook
- Module Numpy
- Module Pandas
- Pandas – Objet 'Series'
- Pandas – Objet 'DataFrame'
- Pandas – Regrouuper et agréger
- API
- Pandas – Modéliser
- Statistiques et apprentissage statistique
- Pandas – Séries chronologiques
- Pandas – Miscellanées
- Modules de visualisation
- Visualisation graphique
- Visualisation géographique

Total : 24h. Le contenu se segmente en blocs.