

## « Je n'imaginai pas que Python connaîtrait un tel succès »

Entretien avec Guido van Rossum, le créateur du langage informatique le plus utilisé dans les domaines de pointe.

LE MONDE | 25.07.2018 à 17h32 | Propos recueillis par Damien Leloup ([journaliste/damien-leloup/](#))

Guido van Rossum est le créateur du langage informatique Python – nommé en hommage aux Monty Python –, l'un des plus populaires au monde. Après près de trente ans comme responsable de l'évolution du langage, cet informaticien néerlandais, passé par Google et qui travaille aujourd'hui pour DropBox, a annoncé début juillet son intention de prendre sa retraite, et de laisser le développement du langage à la communauté des développeurs. Il revient pour Pixels sur l'étonnant parcours de Python, qui a connu un succès exponentiel depuis le début des années 2010.

**Pixels : Une étude récente du site d'aide aux développeurs Stack Overflow montre que Python, qui a 30 ans, est devenu le langage le plus recherché sur la plate-forme. Les tendances Google soulignent que depuis 2013, Python a quasiment éclipsé tous les autres langages. Comment expliquez-vous ce succès ?**

**Guido van Rossum :** Quand j'ai commencé à travailler sur Python, je n'imaginai certainement pas qu'il connaîtrait un tel succès ! Dans les années 1990, Python était un « petit » acteur, même au sein de la catégorie des langages dynamiques. Le grand acteur, c'était Perl. Je connaissais Larry [Wall, le créateur de Perl], et j'étais en désaccord avec certaines de ses idées. Le temps a montré que d'autres personnes partageaient ma vision, et j'ai été très content quand, au milieu des années 2000, Python a commencé à prendre le dessus sur Perl.

La croissance récente a plusieurs explications, mais l'un des éléments-clefs, c'est que le secteur du big data a adopté Python comme l'un de ses principaux outils. Le big data a très fortement progressé ces derniers temps, et c'est clairement l'une des raisons qui ont poussé le succès de Python.



Guido van Rossum en 2014, au siège de DropBox. Daniel Stroud / CC Creative Commons Attribution-Share Alike 4.0

**Python est aussi très utilisé par les chercheurs en intelligence artificielle, un autre secteur qui a le vent en poupe. Pourtant, ces deux disciplines n'existaient pas, en tout cas sous cette forme, à la création de Python...**

C'est vrai. Mais à la fin des années 1990, il y avait un certain nombre de personnes qui voulaient utiliser des bases de code écrites dans les langages de programmation les plus rapides, comme C ++, mais qui avaient aussi besoin d'un langage leur permettant de lier ces briques logicielles ensemble. C'est alors qu'est né Numeric Python, et plus tard NumPy. Cela a lancé tout un mouvement : des gens ont commencé à ajouter des outils de visualisation de données, [*la librairie graphique*] matplotlib...

Une fois qu'il y a un certain nombre d'outils disponibles, dans une discipline donnée, les chercheurs ont tendance à en ajouter d'autres ! Par exemple IPython, qui leur a permis de jouer avec leurs données, de manière dynamique et expérimentale. Que se passe-t-il si je projette telle ou telle donnée sur tel ou tel axe ? Cette interactivité était incroyablement importante. IPython a grossi, puis est arrivé IPython Notebook, un autre outil très important pour les communautés de chercheurs, et plus généralement pour les gens qui manipulent de grandes quantités de données.

Toute la communauté Python est incroyablement impliquée dans l'open source [*le partage libre et ouvert des logiciels créés*]. Les développeurs copient ce qu'ils connaissent et ce qu'ils aiment, l'améliorent ou l'étendent, et rendent le résultat open source. C'est une philosophie commune.

**D'autres langages, comme Perl, étaient aussi open source, mais ont eu moins de succès à long terme. Qu'est-ce qui faisait la différence pour Python ?**

Il y a deux raisons, je pense. D'abord, c'est très facile de brancher du code C ++ à Python, et pour quelqu'un qui n'a pas une expérience d'administrateur système Linux, Python est beaucoup plus facile à apprendre que Perl.

En administration système, on utilise beaucoup de scripts – Linux a tous ces outils que l'on peut combiner pour faire des choses utiles. Perl utilise de manière très astucieuse beaucoup de choses issues de ce système – le dollar pour les variables, les expressions régulières... – et est donc devenu une meilleure version des scripts Unix. Mais quand vous enseignez à des gens qui n'ont aucune expérience en programmation, Perl n'est pas un bon langage. Pour l'apprendre, il faut maîtriser beaucoup de choses, qui ne sont pas du niveau d'un lycéen ou d'une lycéenne par exemple.

La philosophie de Python est que vous avez uniquement besoin de connaître un peu les mathématiques, niveau lycée, et quelques clefs du langage pour commencer à faire des choses. C'est un langage très facile à apprendre pour des chercheurs, qui connaissent les mathématiques et ont quelques bases en informatique. C'est notamment pour cela que la communauté scientifique s'en est rapidement emparée.

**Python est effectivement très utilisé dans l'enseignement de l'informatique, y compris en lycée en France. Était-ce l'un de vos buts quand vous l'avez créé ?**

Ce n'était pas sur ma liste ! Mais Python a emprunté beaucoup de choses à ABC [*un langage jamais finalisé sur lequel M. van Rossum avait précédemment travaillé*], qui était conçu pour être facile à enseigner. Dans les années 1990, j'ai rencontré un développeur qui m'a raconté avoir donné Python à son fils, 12 ans à l'époque, et qu'il avait adoré. Son fils a même découvert un bug qui faisait planter l'interpréteur [*le logiciel qui exécute le programme*] ! Python est assez intuitif, et les enfants n'aiment pas devoir lire des pages et des pages de documentation. Ils apprennent en tapant sur le clavier ! Mais à un certain point, il faut au minimum savoir comment chercher dans la documentation – arrivé à un certain niveau, chercher la réponse sur Google ne sera plus suffisant.

**Quels conseils donneriez-vous à quelqu'un qui débute la programmation avec Python ?**

« POSER DES  
QUESTIONS EST  
UNE MANIÈRE

Je leur dirais de s'amuser ! Et de lire le code d'autres personnes, à partir du moment où ils ont le niveau suffisant pour comprendre comment il fonctionne. Un autre conseil important : ne pas s'y attaquer seul. Trouvez un ami, résolvez les problèmes ensemble ; si personne dans votre entourage n'est intéressé, trouvez un groupe amical sur Internet. Poser des questions est une manière incroyablement efficace d'apprendre. Il faut juste trouver le bon moment pour appeler à l'aide, une fois que vous avez

**INCROYABLEMENT EFFICACE D'APPRENDRE.** » vraiment essayé de trouver par vous-même. En général, c'est le moment où vous commencez à avoir envie de vous cogner la tête contre le mur !

**Dans un discours il y a deux ans** (<http://neopythonic.blogspot.com/2016/04/kings-day-speech.html>), vous disiez que « *les langages de programmation sont la manière dont les programmeurs expriment et communiquent leurs idées* ». Quelle idée vouliez-vous communiquer en créant Python ?

Je voulais que Python soit un peu différent des autres, et montrer qu'il n'y a rien de mal à être différent. A l'époque, la clarté n'était pas au cœur de la philosophie des langages. Ils se concentraient sur l'efficacité du code, c'était un héritage des débuts de l'informatique, dans les années 1940 : quand il faut une journée entière pour faire tourner un programme, vous avez intérêt à ce que votre code soit aussi efficace que possible. Mais cette efficacité se faisait au détriment de la lisibilité, et les langages étaient très complexes pour les utilisateurs.

Dans la philosophie de Python, le temps d'exécution n'est pas aussi important que le temps nécessaire à l'écriture du code. Prenons un exemple : pour accomplir une certaine tâche, il vous faut une journée pour écrire un programme, qui s'exécute en une seconde. Dans bien des cas, certains utilisateurs préféreraient, pour accomplir la même tâche, avoir un programme qui s'exécute en cinq minutes, mais qui s'écrit en une demi-heure. C'est une meilleure solution, sauf si vous devez faire tourner ce programme des centaines de fois par jour. L'idée était vraiment de dire : « *tant pis pour toutes ces astuces très malines qui rendent la vie des utilisateurs impossible* ». D'accord, le code s'exécutait un peu moins vite, mais quand la performance est devenue un problème, la communauté a amélioré le langage. Comme le disait le célèbre informaticien Donald Knuth, « *l'optimisation prématurée est la source de tous les maux* ».

**Pourquoi abandonnez-vous aujourd'hui la direction du projet ? Vous aviez pourtant le titre honorifique de « dictateur bienveillant à vie »...**

Bien sûr, cette idée de « dictature à vie » est une blague – originellement, le titre complet était même « dictateur bienveillant à vie par intérim ». C'était juste une manière amusante de désigner le leader d'un projet open source. Je l'ai fait pendant vingt-huit ans. J'en ai aujourd'hui 62, et cela fait plusieurs années que j'envisage de prendre ma retraite. C'est un travail stressant.

La goutte d'eau qui a fait déborder le vase a probablement été la manière dont les discussions en ligne se sont compliquées ces dernières années. Pendant tout ce temps, j'ai connu beaucoup de désaccords avec mes décisions, mais plus récemment, les discussions ont pris un ton désagréable, et parfois insultant. Cela a fini par me ronger. Un matin, je me suis levé – la veille, j'avais validé une nouvelle décision controversée –, et j'ai senti que c'était trop pour moi. Il est temps de laisser la communauté des développeurs trouver une autre manière de prendre les décisions, une manière qui ne me place pas en permanence en première ligne.

**Qu'est-ce qui a changé dans la manière dont les discussions se déroulent ?**

« ON NE PEUT PAS GAGNER UN DÉBAT SUR TWITTER. »

J'ai vécu avec les désaccords pendant quarante ans, et ce n'était pas un problème. Ce qui est nouveau, c'est la manière dont les gens expriment leur désaccord. Cela m'a donné l'impression qu'on ne me faisait plus confiance. Un message agressif publié sur un réseau social par un développeur important, ce n'est pas une bonne manière de faire un retour. Auparavant, quand quelqu'un me disait « *je ne suis pas d'accord* », je pouvais dire « *OK, je prends la responsabilité de cette décision* ». Sur les réseaux sociaux, je ne veux pas me noyer dans une discussion infinie avec 100 000 followers. On ne peut pas gagner un débat sur Twitter.

**Est-ce plus généralement le climat sur les réseaux sociaux qui a changé ? Après l'élection de Donald Trump, beaucoup d'Américains se sont plaint d'une hostilité grandissante...**

Je ne suis pas un expert, mais sur les réseaux sociaux, les débats politiques ont tendance à virer à la foire d'empoigne. Si c'est ce que le président des Etats-Unis ou l'Union européenne, veulent faire, c'est malheureux. Je ne suis pas fait pour ça – je n'aime pas dire des choses méchantes sur d'autres personnes.