# Python Orientation

About Python PF0

## Learning Objectives

1 Understand the foundations and goals of Python as a coding language

## Introduction

All programming languages are created to meet a specific need or use case. As such, not one language is better than another, but possibly better in specific use cases. Here we'll highlight some of reasons for Python in the biosciences an some of its history.

## Why program in Python?

Python has become one of the most widely used programming, regarded for its simplicity, stringency, readability and versatility. From web development and automation, all the way through to scientific research and artificial intelligence Python is a standard for many programmers. On GitHub, the world's leading code-sharing platform, Python consistentl ranks as one of the most widely used and actively developed languages. It is also considered a **general-purpose** language, meaning that it can be used for building everything from mobile apps to robotic control systems.

But where Python truly excels is in data handling, automation and scientific computing: making it especially valuable i fields like biosciences and medicine, where researchers work with increasingly large and complex data. Part of what makes Python powerful is its community-driven ecosystem, with thousands of freely available libraries -- such as **Pandas** for data analysis, **NumPy** for numerical computing and **Matplotlib** for visualisation -- Python allows researchers to build powerful tools, making spartan use of economical, clean code. Since these libraries are open-source, they are constantly being refined and extended by a global community of users.

Advanced programming paradigms such as **Object Oriented Programming (OOP)** -- where code is structured by organising data and behaviours into reusable units called **objects** -- are well-supported in Python. This makes it a great first language to learn when getting started with OOP, with its characteristically clean syntax mirroring everyday English. It is easy to write clean, scalable and modular code using Python, which is of particular importance in researcl software and simulations. For biologists, clinicians and data scientists, Python allows for reproducible, efficient and customisable research workflows.

## Who invented Python?

Python was created by **Guido van Rossum** in the late 1980s, with its official release being first published in 1991. Var Rossum developed Python as a hobby project, where he was aiming to create a language that was easy to read, allowin developers to express ideas with both clarity and the fewest possible lines of code; particularly relative to other languages, such as C and Perl. As a fun fact, van Rossum named the language after the British comedy group *Monty Python*, rather than the snake.

## Why Python is important in biosciences and medicine

In biosciences and medicine, Python has now become largely indispensable. Both researchers and clinicians are routinely faced with complex datasets: from nucleic and amino acid sequence data, through to protein structures, electronic health records and imaging data. Python's many freely-available libraries for scientific computing (such as **Pandas**, **NumPy** and **scikit-learn**, all of which we will employ extensively throughout the course), offer powerful tool for analysing data -- both in isolation, and in synergy.

With the recent rise of machine learning and artificial intelligence, Python's popularity continues to cement. Among many different applications, libraries such as **scikit-learn**, **TensorFlow** and **PyTorch** allow bioscientists to build predictive models for diagnosis, drug discovery, patient stratification and much more. The ability to automate tasks, prototype rapidly and collaborate using open-source tools makes Python an ideal fit for biomedical research and translational medicine.

## Reproducibility

Beyond analysis and modelling, Python plays a central role in ensuring **reproducibility**. Publishing research should always hold the ability to reproduce it as a cornerstone of good scientific practice. It is therefore essential that workflows or programs can be repeated, with output and results verified and expanded upon, by others. In order to do this, Python code should be easily legible by nascent users. Its stringency -- with singular ways to do certain tasks -- means that it is easily legible by other users.

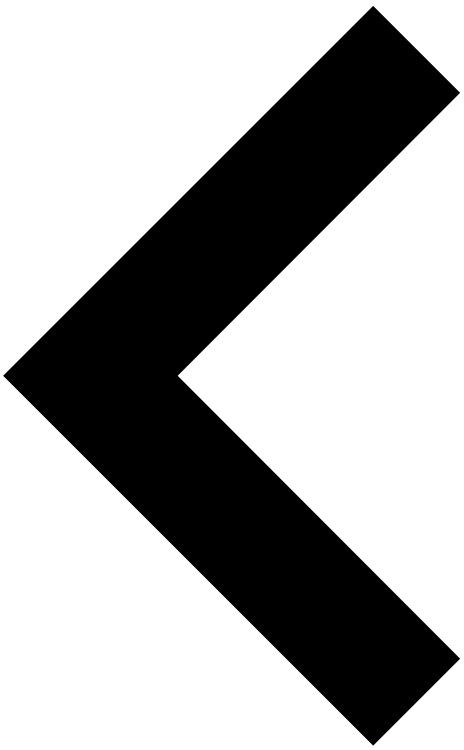
Python allows you to script every step of a data-processing or analysis pipeline: from importing raw datasets and performing statistical operations on these, through to generating figures and exporting results. And it is *via* platforms like **GitHub**, that these scripts can be fully version-controlled, shared and reused, forming a transparent and traceable record of your analysis. As part of this course, you'll learn how to write clean, well-documented code that supports reproducible research and scientific practice; whether this is strictly for your own use, or sharing your code with others

Python's interaction with other languages and programs is also one of its more attractive features. It is often seen as a 'glue' language, integrating with other popular tools and platforms used in bioscience research. Whether you're interfacing with **R** for statistical methods, calling established command-line tools for sequence alignment, or processin imaging data, Python makes it possible to build seamless and efficient workflows across these. Its compatibility with web-based environments such as Jupyter Notebooks and cloud platforms like GitHub and Azure means your work can scale -- from local experiments to large, collaborative projects. Throughout the course, we'll explore these tools in practice, helping you build research workflows that are not only powerful but also accessible and future-proof.

## Community

Python's success is strongly linked to its global community. There are millions of Python users around the world, contributing to a rich ecosystem of libraries, tutorials, forums and collaborative projects. Platforms like **GitHub** and **Stack Overflow** provide access to an enormous body of shared knowledge, allowing newcomers to find help quickly and build on others' work.

This community-driven model ensures that Python remains at the cutting edge of research and development. For those in biosciences and medicine, this means being part of a digital ecosystem where new methods are constantly being innovated, shared, updated and improved by peers across the globe.



Previous Next