# W3 Schools - Python Introduction

## What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* system scripting.

## What can Python do?

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

## Why Python?

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-oriented way or a functional way.

## Good to know

* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

## Python Syntax compared to other programming languages

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

# Keywords:

**Web Development** - The process of building and maintaining websites or web applications that run in web browsers. It includes front-end (user interface) and back-end (server-side) development.

**Software Development** - The systematic process of designing, coding, testing, and maintaining computer programs and applications to solve problems or perform specific tasks.

**Mathematics** - The abstract science of numbers, quantities, shapes, and patterns, used as a foundational tool in logic, computation, and problem-solving across various disciplines, including programming.

**System Scripting** - Writing short programs (scripts) to automate tasks and manage system operations, often using languages like Bash, PowerShell, or Python.

**Web Applications** - Software applications that run on web servers and are accessed through a web browser, often interactive and designed to perform specific tasks (e.g., email, shopping carts, dashboards).

**Workflows** - A sequence of tasks or processes, often automated, that are carried out in a specific order to complete a larger task or business process.

**Database Systems** - Software tools (like MySQL, PostgreSQL, or MongoDB) used to store, manage, and retrieve structured or unstructured data efficiently.

**Prototyping** - Creating a simplified version or model of a system or software to test concepts, design features, or gather user feedback before full-scale development.

**Production-ready software development** - The creation of software that is stable, secure, scalable, and optimized for real-world use in live environments, with proper testing and documentation.

**Syntax** - The set of rules that defines the combinations of symbols that are considered valid in a programming language, much like grammar in human languages.

**Object-Oriented** - A programming paradigm based on the concept of "objects" that contain both data and methods; emphasizes reusability, modularity, and encapsulation.

**Interpreter System** - A program that reads and executes code line by line, translating high-level programming languages into machine-understandable instructions on the fly (e.g., Python interpreter).

**Integrated Development Environment (IDE)** - A software suite that provides developers with tools like a code editor, debugger, compiler/interpreter, and other features to streamline the coding process (e.g., VS Code, IntelliJ, PyCharm).