Introduction

* Open source, high-level programming language developed by Guido van Rossum in the late 1980s.

|  |  |
| --- | --- |
| Topic | Description |
| Less learning time | Python is relatively easy to learn. Many find Python a good first language for learning programming because it uses simple syntax and shorter codes |
| high-level language | Reading and writing codes in Python is much like reading and writing regular English statements |
| interpreted language | This means that every time a program is run, its interpreter runs through the code and translates it into machine-readable byte code. |
| object-oriented | that allows users to manage and control data structures or objects to create and run programs |
| First class | Everything in Python is, in fact, first class. All objects, data types, functions, methods, and classes take equal position in Python |
| Readability | Python programs use clear, simple, and concise instructions that are easy to read even by those who have no substantial programming background. Programs written in Python are, therefore, easier to maintain, debug, or enhance |
| Higher productivity | Codes used in Python are considerably shorter, simpler, and less verbose than other high level programming languages such as Java and C++. In addition, it has well-designed built-in features and standard library as well as access to third party modules and source libraries. These features make programming in Python more efficient |
| Platform independent | Runs across different platforms .Python works on Windows, Linux/UNIX, Mac OS X, other operating systems and small form devices. It also runs on microcontrollers used in appliances, toys, remote controls, embedded devices, and other similar devices and use to create games, write GUIs, and develop web applications. |

Syntax – Keywords

|  |  |  |
| --- | --- | --- |
| import | while | global |
| def | class | in |
| print | and | is |
| if | assert | lambda |
| else | del | not |
| elif | except | or |
| for | exec | pass |
| break | try | raise |
| Continue | finally | with |
| return | from | yield |

Syntax – Identifiers rules

|  |
| --- |
| An identifier can be a combination of uppercase letters, lowercase letters, underscores, and digits (0-9). Hence, the following are valid identifiers: myClass, my\_variable, var\_1, and print\_hello\_world. |
| Special characters such as %, @, and $ are not allowed within identifiers. |
| An identifier should not begin with a number. Hence, 2variable is not valid, but variable2 is acceptable. |
| Python is a case-sensitive language and this behavior extends to identifiers. Thus, Labor and labor are two distinct identifiers in Python. |
| You cannot use Python keywords as identifiers. |
| Class identifiers begin with an uppercase letter, but the rest of the identifiers begin in lowercase. |
| You can use underscores to separate multiple words in your identifier |