|  |  |  |  |
| --- | --- | --- | --- |
| Date: | **10-07-2020** | Name: | **Varun G Shetty** |
| Course: | **Introduction to Internet of Things** | USN: | **4AL17EC093** |
| Topic: |  | Semester & Section: | **6th & ‘B’** |
| GitHub Repository: | **Varunshetty4** |  |  |

**Report-**

Python Interpreter: Python is an interpreted language; therefore, an interpreter is required to parse and execute Python code. The Python interpreter understands and executes Python code. Python code can be created in any text editor and Python interpreters are available for many operating systems.

Python developers can create and deploy Python programs in practically any operating system. Third party tools such as Py2exe And Pyinstaller can also be used to package the Python source code into an executable file, eliminating the need for the Python interpreter when running Python code. In Linux machines, the Python interpreter is usually installed in /usr/bin/python or /usr/bin/python3 (depending on the available Python versions on the system).

With the new Windows Python installer, Python is installed by default into the user’s home directory. In older Windows machines, Python is often placed in C:\PythonXX (where XX is the version of Python). After the Python interpreter has been installed, it operates somewhat like the Linux shell. This means that when called with no arguments, it reads and executes commands interactiveely.

When called with a file name argument or with a file as standard input, it reads and executes a script from that file. To start the interpreter, simply type Python or python3 at the shell prompt.Some legacy systems are still running on an older versi on of Python 2, but many new systems are moving to use the new Python version 3. Python’s version is printed on the first line when the interpreter is launched .

This course is built on Python 3 code. When the Python interpreter is called with no arguments, and commands are entered via the keyboard, the interpreter is said to be in interactive mode. In this mode, the interpreter waits for commands. The primary prompt is represented by three greater-than signs (>>>). Continuation lines are represent ted by three dots (...). Continuation is the default secondary prompt. The>>>prompt indicates the interpreter is ready and waiting commands. Continuation lines are needed when entering multi-line code.

Another way of using the interpreter is python -c command [arg] ... which executes the statement(s) in the command. Because Python statements often contain spaces or other characters that are particular to the shell, it is suggested to enclose the entire command between single quotes.

Functions and Data Types in Python: Python supports many useful functions and datatypes. Some of the more important ones are as

follows:

The range () function generates a list of numbers usually used to iterate with FOR loops.

● Range (stop) -This is the number of integers (whole numbers) to generate, starting from zero.

● Range ([start], stop [, step] –This is the starting number of the sequence, the ending number in the sequence, and the difference between each number in the sequence.

Tuples: A tuple is a sequence of unchangeable Python objects. Tuples are sequences, separated by parentheses. Figure 2 shows examples of tuples.

Lists: Lists are a sequence of changeable Python objects. Lists can be created by putting different comma-separated values between square brackets. Figure 3 shows examples of lists and how they can be updated.

Sets: Sets are unordered collections of unique elements. Common uses include membership testing, removing duplicates from a sequence, and computing standard math operations on sets such as intersection, union, difference, and symmetric difference. Figure 4 shows examples of sets.

Dictionary: A dictionary is a list of elements that are separated by commas. Each element is a combination of a value and a unique key. Each key is separated from its value by a colon. The entire dictionary is written within braces. Dictionary elements can be accessed, updated, and deleted. There are also many built-in dictionary functions such as a function that compares elements within different dictionaries and another that provides a count of the total number of elements within a dictionary.

What is Big Data?

Data is information that comes from a variety of sources, such as people, pictures, text, sensors, and web sites. Data also comes from technology devices like cell phones, computers, kiosks, tablets, and cash registers. Most recently, there has been a spike in the volume of data generated by sensors.

Sensors are now installed in an ever growing number of locations and objects. These include security cameras, traffic lights, intelligent cars, thermometers, and even grape vines! Big Data is a lot of data, but what is a lot? No one has an exact number that says when data from an organization is considered “Big Data.” Here are three characteristics that indicate an organization may be dealing with Big Data:

● They have a large amount of data that increasingly requires more storage space (volume).

● They have an amount of data that is growing exponentially fast (velocity).

● They have data that is generated in different formats (variety). How much data do sensors collect? Here are some estimated examples:

● Sensors in one autonomous car can generate 4,000 giga bits (Gb) of data per day.

● An Airbus A380 Engine generates 1 petabyte (PB) of data on a flight from London to Singapore.

●Safety sensors in mining operations can generate up to 2,4 terabits (TB) of data every minute.

● Sensors in one smart connected home can produce as much as 1 gigabyte (GB) of information a week. While Big Data does create challenges for organizations in terms of storage and analytics, it can also provide invaluable information to fine-tune operations and improve customer satisfaction.