UPRIGHT PUBLICATIONS
https://upright.pub/index.php/ijrstp/

Original Contribution

Python Programming Language for Data-Driven Web Applications

Sreekanth Dekkati[1]

Keywords: Python, Programming Language, High-Level Language, Object-Oriented, Big Data, Data Sciences

International Journal of Reciprocal Symmetry and Theoretical Physics

Vol. 8, Issue 1, 2021 [Pages 1-10]

Selecting an appropriate programming language to carry out any necessary job associated with programming or any other pertinent field is the most essential duty for anyone just starting in the field of computers. Selecting a programming language approachable for novices is of the utmost significance, as this decision is a critical stepping stone to becoming a programmer. If we choose a complex programming language, it may cause us to lose interest in computer programming. This article focuses on the characteristics of Python, as well as its applications, benefits, drawbacks, and other aspects. It also briefly discusses object-oriented principles that may be easily implemented using Python. This may assist the reader in understanding why they should choose Python as their first programming language if they are starting.

INTRODUCTION

Python is a computer language that may be used for various purposes and supports several programming paradigms. A high-level programming language that, compared to other programming languages, requires a significantly smaller number of lines to carry out a specific task (Deming et al., 2018). As a result of its highly user-friendly writing style, Python is widely regarded as one of the simplest programming languages to get started in today's world. The standard library that comes along with the language includes a significant number of pre-defined methods (Gutlapalli, 2016a).

Python is a high-level programming language that is simple and easy to learn, free to use and open source, platform-independent, portable, dynamically typed, procedure-oriented and object-oriented, interpreted, extendable, embedded, and has an extensive library (Lal et al., 2017). These are the primary benefits of using Python.

In the late 1980s, Guido van Rossum started working on the Python programming language, and the first version of Python, version 0.9.0, was made available to the public in 1991. Because it is a dynamically typed programming language, the user is not required to specify the data type for the values that will be stored in the program for those values to be saved. Python 2.0 was made available to the public in 2000, whereas Python 3.0 was made available in 2008. Python 3.0 is flexible with prior language versions (Bogdanchikov et al., 2013).

To be able to work in big data analytics, we need to be one of the data scientists who specialize in this area, and we need to select the appropriate tools that will enable us to complete all of our analyzing duties appropriately (Thaduri, 2018). There are a lot of powerful analytical tools for data, and they can be broken down into three categories: programming languages, statistical solutions, and visualization tools (Ballamudi, 2016). There are also a lot of different analytical tools for data. The selection of one or more of them is contingent on our prior experience in programming and our familiarity with statistics.

For instance, if we wish to use the R programming language, we must have a strong background in scientific programming and statistics. On the other hand, when we use visualization tools, we don't need that knowledge to experiment with the data.

[1]Assistant Vice President (System Administrator), MUFG Bank, New York, USA [sreekanthd041987@gmail.com]

FEATURES OF PYTHON

Easy to Code: In contrast to C, C++, C#, and Java, the syntax of the Python programming language is extremely simple and user-friendly for developers. This means it is possible to quickly learn and write basic code in just a few hours or days. One of the simplest languages to pick up is Python (Pedersen et al., 2015).

Open Source and Free: Python is an open-source technology with an extensive community of programmers, data scientists, and other professionals who contribute to the development process of Python. Python also includes a large set of libraries that can be used for various applications (Bodepudi et al., 2019). There are a lot of online forums where people talk about Python, exchange their expertise about it with other programmers, and engage in conversations that cover a variety of subjects connected to the programming language. Reddit, StackOverflow, Source Python, DigiForum, and other similar websites are examples of well-known Python discussion forums. Windows, Mac OS X, or Linux users can get a free download of Python's most recent version from the official website.

GUI Support: Python provides access to a significant selection of graphical user interface (GUI) frameworks. Most software developed these days has an intuitive and user-friendly graphical user interface (GUI) that makes the findings more visible. TkInter, PyQt, and PySimpleGUI are just a few examples of GUI frameworks that can be implemented in Python (Ballamudi, 2019a).

Highly Portable: Thanks to its platform independence, Python can run on diverse computer systems. Code written in Python can run on various operating systems, including Linux, Mac OS X, and Microsoft Windows. The programmer can run the same code seamlessly across all supported platforms until any system-dependent code is added (Mandapuram et al., 2020).

High-Level Language: Because Python is a high-level programming language, the person writing the code can give little thought to the underlying hardware architecture when they are doing so (Meyer & Obermayer, 2016).

Dynamically Typed: Python does not need the user to define the data types for variables before writing code; instead, Python automatically determines the type and allocates memory accordingly, protecting the user from data type mismatch errors (Chen et al., 2019).

Data Structures: Because Python includes numerous rich, dynamic data structures as part of its standard installation, such as lists, sets, tuples, dictionaries, and so on, it can be put to use as soon as it has been installed (Reddy et al., 2020).

Interpreted: Unlike other compiled languages, Python's ability to execute code line by line without first requiring it to be collected makes it significantly more accessible and more effective for beginners to debug their programs (Desamsetti & Mandapuram, 2017).

Language Interoperability: Python's support for the execution of code written in a variety of other programming languages, such as C and C++, as well as Java, is one of its many admirable qualities. Users of Python can use F2pyCtypes, Cython, and similar tools to call libraries and functions written in other programming languages such as C and C++ (Desamsetti, 2018).

Object Oriented: Python is sometimes referred to as a hybrid programming language since it can support both completely object-oriented programming notions and a structured programming style. Python treats everything as if it were an object (Gutlapalli, 2016b). Python's object-oriented programming (OOP) not only assists in defining real-world entities in programming but also specifies the relationships between those entities (Desamsetti, 2020).

- **Class:** A user-defined data structure follows a blueprint to construct an object.
- **Object:** An instance of a class that has been given memory to carry out its functions is an object. It includes the properties and operations that are specified by the class specification. Python has an easy-to-understand syntax for creating objects.
- **Abstraction**: The process of concealing the internal workings of an implementation and exposing only the required components.
- **Encapsulation**: Bringing together data and methods into a single unit, which is then represented by a class and given the form of an object.
- **Inheritance:** The process of transferring properties from a parent class to a child class so that the child class can make use of its parent class's reusability feature. This is done so the child class's code can quickly improve.
- **Polymorphism:** The capacity of an object to assume various shapes; alternatively, the capability of carrying out a single task in several distinct ways.

DATA ANALYZING IN PYTHON

Python is one of the most well-known data analysis languages, and data scientists frequently direct their attention to it when doing research. Because of its high-level interactive character and scientific ecosystem libraries, this programming language is widely regarded as the superior option for developing analytical algorithms and discovering previously unknown aspects of data (Meurer et al., 2017).

When one focuses on the scientific computing community, it is simple to see how Python has become increasingly popular in this field (beginning in the early 2000s) in business solutions (applications) and academic research.

Python has its scientific ecosystem in addition to a large number of helpful libraries, some of which are as follows:

Numpy: The name "Numerical Python" refers to the core package in the Python programming language and the foundation data structure. Knowing that all Python input data is stored in a numpy array makes it simple to deduce that all Python libraries are built on top of the numpy package. Numpy has the following capabilities:

- Ndarray is an object that is both effective and quick when working with multidimensional arrays.
- A comprehensive list of functions that may be used to alter arrays by doing element-wise computations on them or by providing mathematical operators that can be used between arrays.
- Instrumentation for reading and writing array-based data sets to and from a storage medium.
- The Fourier transform, operations in linear algebra, and the production of random numbers are some topics covered.
- Instruments for incorporating the source code of other languages, such as C, C++, and FORTRAN, into Python.

Pandas: This package helps scientists speed up their work on structured data. It does this by providing ready-made and pre-designed functions and rich data structures, all of which combine to make the scientists' work more straightforward and meaningful (Gouillart et al., 2016).

Matplotlib: This is a program that is widely used for the visualization of data as well as the drawing of expressive charts. This tool is quite effective for these tasks, particularly for 2D plots.

Ipython: is an environment for interactive computing and development. We can use it to increase our productivity in interactive computing and software development. IPython's primary focus is on the former. In addition, it features a comprehensive graphical user interface console equipped with inline plotting, a web-based interactive notebook format, and a lightweight and swift parallel computing engine. It is a condensed version of the Python shell, and its primary purpose is to speed up the processes of writing, testing, and debugging Python programs.

SciPy: is a collection of packages that contain effective methods for things like linear algebra, sparse matrix representation, special functions, and fundamental statistical functions [2]. These deals have the following items:

- scipy.integrate
- scipy.linalg
- scipy.optimize
- scipy.signal
- scipy.sparse
- scipy.special
- scipy.stats
- scipy.weave

Additionally, bindings for several Fortran-based standard numerical packages, such as LAPACK, are available in the scipy programming language.

Cython: A package that combines C and Python, allowing data scientists to use Python syntax and high-level operations while also increasing compiling performance to reach the level of performance of compiled languages. This is made possible by the combination of the two languages (Perez et al. 2012).

Language R: R is a flexible open-source programming language used chiefly for statistical and data-science purposes. We can perform any statistical computation using R's functional-based grammar or program-based code. R's debugging facilities are highly sophisticated, and the language provides a lot of interfaces to other programming languages. After that, the collected statistics can be shown using the advanced graphical tool R delivers.

Even though they might work in sectors as diverse as medicine or academia, the vast majority of data scientists use the R environment and its associated packages when conducting research in big data-related

fields such as big data business, industry, or government (a discussion of the differences between R and other languages will follow). R possesses the following characteristics:

- A concise and streamlined syntax that will speed up the actions We do on our data.
- It is capable of loading and storing data in various formats and can do so locally and via the Internet.
- Capability to carry out one's duties using memory and a syntax that is consistent throughout.
- A comprehensive listing of tools (functions, packages) for performing data analysis activities; some tools are already built-in, while the remaining tools are open source.
- It possesses a variety of straightforward methods to depict the statistical findings in graphical formats, as well as the capacity to save these graphs on the disk.
- Capability to automate analyses, develop new functions (R is a programming language), and extend the features of the language that are already there.
- Users do not need to constantly reload their data because the system stores the data between sessions and saves the history of their executed instructions.
- If we prefer GUI, several free GUI options are available for R, including the following: RStudio, R Command, StatET, ESS, and JGR Java GUI for R.
- It is downloadable for all operating systems, including Windows, Macintosh, and Linux.

Data scientists have found that the R programming language provides a wide variety of tools for all statistical, machine learning (linear and nonlinear modeling, classic statistical tests, time-series analysis, classification, and clustering), and graphical approaches, and that it is massively expandable. This is because R is an open-source programming language. R offers built-in and extended application functions, including statistical analysis, machine learning, and data visualization (Campbell et al., 2015).

SAS

The SAS software (with its language) is a well-known and widely used solution for gaining access to data, transforming it, and reporting on it by utilizing its adaptable, extendable, and web-based interface (Mandapuram & Hosen, 2018).

The SAS Analytics Platform comprises various analytical applications that form an application framework. This framework makes the SAS Analytics Platform a beneficial tool for data scientists to employ in their work. The following is a list of the most essential helpful applications of analysis:

SAS Text Miner: is an option for the SAS Enterprise Miner environment that can be installed as a plug-in. This is because it helps simplify the most critical component of text mining, the prediction aspect, and supports it with a comprehensive collection of tools. The SAS Text Miner can manipulate a variety of different textual data sources.

SAS Forecast Server: Because of its automation and scalability, businesses can make decisions that are better for their future in terms of both efficiency and effectiveness, and they are also able to generate vast amounts of high-quality forecasts rapidly and automatically. This tool enables forecasters to identify the essential forecasts and concentrate their efforts on them, boosting the efficiency of the forecasts provided by businesses for a broad range of routine problems (planning difficulties).

SAS Model Manager: It arranges and organizes the phases of generating analytical model collections, beginning with creating, continuing through maintaining and monitoring, and arriving at administering as the final step. When it comes to choosing and identifying, during the development process, the priorities of the models to deployment, or even when managers want to ensure that all current new conditions are reflected in the models after updating, SAS Model Manager provides decision makers with a web-based environment that makes their jobs easier with perfectly managed tools and supports lifecycle management and governance of models.

APPLICATIONS OF PYTHON

Web Applications: Python provides access to a sizable selection of frameworks, which facilitates the development of web applications. Pyramid or Django may be utilized for tasks requiring great force, while a flask or bottle may need moderate force. Python's libraries are designed to make it simple to work with various internet technologies, including HTML, JSON, and others. The development of secure applications may also use the implementation of cryptographic functions. Python allows for implementing connection-based and connectionless protocols (Miles, 2016).

Scientific and Numeric: Artificial intelligence and machine learning are increasingly getting integrated into

everyday life (Dekkati & Thaduri, 2017). As a result, more and more apps are being developed with intricate structures and algorithms that carry out a significant amount of mathematical and scientific computation to complete the work at hand (Lal, 2015). Because Python provides a high level of data abstraction, its built-in libraries, such as Numpy, Pandas, Scipy, and Scikit-learn, can be utilized to complete the work. Python also includes various libraries that can be used as a last resort.

Software Development: SCons builds control and management applications with Python as the primary programming language. Buildbot and Apache Gump can carry out automatic continuous compilation and testing. Roundup and Trac are two options that can be used for bug tracking and managing projects, respectively (Thaduri, 2019).

Image, Audio, and Video Processing: Image processing is one of the areas where Python is making significant strides in development. Some examples of libraries that can process and edit images are OpenCV, SimpleITK, and Pillow. Pyglet, QT Phonon, and similar libraries allow Python to process audio and video data.

Education: Python's excellent readability and user-friendly syntax make it the ideal language for someone just starting in computer programming (Ballamudi, 2020). Python is a perfect choice for individuals who may or may not have prior experience with coding, as it is appropriate for beginners up to advanced users (Lal, 2016). A person can quickly learn Python and begin experimenting with domain-specific libraries within a few days because of the extensive community support and availability of learning resources (Thaduri, 2020). This can be accomplished within a short amount of time.

SET UP DEVELOPMENT ENVIRONMENT (WINDOWS)

Visit the website for Python, download the most recent stable release of the Python programming language (which, as of November 2017, was Version 3.8), and then install it by following the instructions provided on the Python website. It is recommended to have a dedicated directory (folder) on our computer where we will keep our Python programs (for example, programming-historian), and we can save it anywhere we like on our hard drive. To stay organized, have a dedicated directory (folder) (Turkel & Crymble, 2012).

Komodo Edit is a free and open-source code editor, but if we wish to use another tool, we have a wide variety of other options for editing text (Ballamudi, 2019c). We can get a copy by going to the Komodo Edit website and downloading it. If the Toolbox pane is not visible on the right-hand side, select View -> Tabs -> Toolbox from the menu bar. It makes no difference whether the Project pane is open or closed. Spend some time getting accustomed to the navigational structure of the Komodo editor. The Help file is of very high quality. At this point, we need to configure the editor to execute Python programs.

Select Browse after selecting Edit > Preferences > Languages > Python 3, where Python 3 is listed as the active language. Now, navigate to the following directory: C: Users ourUserNameAppDataLocalProgramsPythonPython38-32). If it appears like this, then click the OK button:

> Next, in the section labeled Preferences, select the Internationalization option. Choose Python from the drop-down menu labeled Language-specific Default Encoding, and then check to see that UTF-8 is set as the default encoding method.

Next, select Toolbox > Add > New Command from the menu bar. This will open up a new window for the dialog. Modify our command so that it reads "Run Python." In the 'Command' section, type:

> Python version 3
> If we neglect this command, Python will inexplicably halt since it isn't receiving a program as input. This is because Python expects to receive a program.
> In the 'Start in' field, type: %D.

If it appears like this, then click the OK button:

> The Toolbox window should now reflect the addition of our new command. After we have finished this step, our computer may require a restart before Python will work with Komodo Edit.

HELLO WORLD

It is common practice to begin learning a new programming language by attempting to write a program that displays the phrase "hello world" and then exits the environment (Dekkati et al., 2019). We will walk through the process of accomplishing this goal using Python and HTML.

Because Python is a relatively high-level programming language, it is an excellent choice for novices (Ballamudi & Desamsetti, 2017). In other words, building short programs capable of achieving a great

deal is feasible. When a program is more straightforward, it is more possible that the entirety of it can be displayed on a single screen, and when this occurs, it is much simpler to keep all of the information straight in our head (Thaduri & Lal, 2020).

The programming language known as Python is called an "interpreted" language. This indicates that there is specialized computer software for computers, an interpreter, that can comprehend and carry out instructions in the respective language (Ballamudi, 2019b). The interpreter can be used in several different ways, one of which is to save all of our instructions to a file, which the interpreter interprets (Gutlapalli, 2017a). A "program" is the name given to the file that stores the instructions for a programming language. The interpreter will carry out each of the instructions we provided in the program, and once it has finished, it will terminate (Thaduri et al., 2016). Let's give it a shot.

Using our preferred text editor, make a new file, type the two-line program below, and save the file in the programming-historian directory. hello-world.py (in Python)

```
# Hello-World.py
print('hello world')
```

Our preferred text editor ought to provide us with a "Run" button that we may click to initiate the execution of our program. If everything goes according to plan, it should look like this (an example of which may be found in Komodo Edit. We may view a larger version of the picture by clicking on it here:

INTERACTING WITH A PYTHON SHELL

Using something known as a shell is still another option for interacting with an interpreter. When we write a statement into the shell and then press the Enter key, the shell will carry out the command we gave (Gutlapalli, 2017b; Desamsetti, 2016b). It is highly recommended that we use a shell when testing statements to ensure that they do the actions we expect them to take.

Simply double-clicking the python.exe file will launch a Python Shell for us. If we have installed version 3.8, which is the most recent version available as of November 2017, then this file is most likely stored in the directory that can be found at C:\Users\OurUserName\AppData\Local\Programs\Python\Python38-32 directory. When the shell window appears on our screen, type in the following:

```
print('Hello world')
```

Then, hit the enter key. The computer will give us a response of

```
Hello world
```

As seen in the following example, when describing an interaction with the shell, we will use -> to signify the shell's answer to our command.

```
print('hello world')

-> hello world
```

It should appear something like this when it comes up on our screen:

Python's Command Prompt in Windows (View Image): Increase the size of this picture.
Python's command prompt for Windows.

Since we and our computer are operational, we can move on to more engaging activities now that they are both ready. If we follow the Python lectures in the recommended sequence, the next tutorial we should attempt is "Understanding Web Pages and HTML" (Desamsetti & Lal, 2019).

SET UP DEVELOPMENT ENVIRONMENT (MAC)

Python 2 may already be set up and running on our PC. We will need to install Python 3 as this version of Python will no longer be supported after the year 2015 comes to a close. Installing Python programming requires downloading the most recent stable release (version 3.8 as of November 2017) and following the installation instructions provided on the Python website (Turkel & Crymble, 2012).

We recommend choosing a specific directory (folder) on our computer to store all our Python applications (for example, programming-historian) to maintain order. We can save it in any location on our chosen hard drive.

We can write Python commands in various text editors, store them, and run them using any of them. In this particular tutorial, we will be using Komodo Edit. It is a code editor that is both open-source and free to use (Dekkati et al., 2016). A wide variety of other text editors are available if we would instead use one of those (Gutlapalli, 2017c). The editing software known as BBEdit is the one that is favored by some of our testers (Thodupunori & Gutlapalli, 2018). It is entirely up to us which one we use; however, to maintain coherence throughout these lectures, we will use Komodo Edit. We

can obtain a copy of Komodo Edit by going to the website for Komodo Edit and downloading it. It should be installed via the .DMG file. At this point, we need to configure the editor to execute Python programs (Mandapuram, 2017b).

Select View > Tabs & Sidebars > Toolbox if we do not see the Toolbox window on the right-hand side of the screen. To create a new command, open the Toolbox window, click the gear icon, and pick "New Command..." from the drop-down menu. This will open up a new window for the dialog. Change the name of our command to "Run Python," and we are free to select a different icon if we prefer. Type "command" into the box labeled "command."

%(python3) %f
Moreover, on the Advanced Options tab, in the section labeled "Start in," type in
%D
Select the OK button. The Toolbox window should now display our newly created command to run Python.

HELLO WORLD

It is common practice to begin learning a new programming language by attempting to write a program that displays the phrase "hello world" and then exits the environment (Mandapuram et al., 2018). We will walk through the process of accomplishing this goal using Python and HTML.

Because Python is a relatively high-level programming language, it is an excellent choice for novices. In other words, building short programs capable of achieving a great deal is feasible. When a program is more straightforward, it is more possible that the entirety of it can be displayed on a single screen, and when this occurs, it is much simpler to keep all of the information straight in our head (Desamsetti, 2016a).

The programming language known as Python is called an "interpreted" language (Dekkati, 2020). This indicates that there is specialized computer software for computers, an interpreter, that can comprehend and carry out instructions in the respective language (Lal et al., 2018). The interpreter can be used in several different ways, one of which is to save all of our instructions to a file, which the interpreter interprets. A "program" is the name given to the file that stores the instructions for a programming language (Mandapuram, 2017a). The interpreter will carry out each of the instructions we provided in the program, and once it has finished, it will terminate. Let's give it a shot.

Using our preferred text editor, make a new file, type the two-line program below, and save the file in the programming-historian directory. hello-world.py (in Python)

```
# hello-world.py

print('hello world')
```

INTERACTING WITH A PYTHON SHELL

Our preferred text editor ought to provide us with a "Run" button that we may click to initiate the execution of our program. If we use BBEdit, select the "#!" button and then select the "Run" option. If everything goes according to plan, it ought to look somewhat like this:

Using something known as a shell is still another option for interacting with an interpreter (Koehler et al., 2020). When we write a statement into the shell and then press the Enter key, the shell will carry out the command we gave. It is highly recommended that we use a shell when testing statements to ensure that they do the actions we expect them to take (Gutlapalli et al., 2019). On Mac, Linux, and Windows, this is accomplished marginally differently.

We may run a Python shell by running the 'terminal.' Once we have opened the Finder on our Mac and double-clicked on the Applications > Utilities > Terminal menu item, we must type "python3" into the new window on our display. At the prompt given by the Python shell, type

```
print('hello world')
```

Then, hit the enter key. The computer will give us a response of

```
hello world!
```

As seen in the following example, when describing an interaction with the shell, we will use -> to signify the shell's answer to our command.

```
print('hello world')

-> hello world
```

Since we and our computers are both operational, we can move on to more engaging activities now that they are both ready (Thaduri, 2017). If we progress through the Python tutorials in the recommended sequence, the next course we should attempt is "Understanding Web Pages and HTML.

CONCLUSION

After reading all the information presented in the paper, one can conclude that Python is an excellent choice for novices as a first language to learn to model real-world things utilizing object-oriented concepts. This is because Python is very user-friendly. Python is a sophisticated programming language that can run on various platforms, is portable, has a low learning curve, is open-source, and does not cost anything to download or use. The availability of support for several programming paradigms encourages programmers with varying programming styles and techniques to collaborate on a single platform. The paper clearly shows that the demand for Python is expanding in various work domains. The study also mentions several recent applications which project a clear view of the need. Python's high level of abstraction and the availability of a large number of both standard and third-party libraries make it an excellent choice for programmers of all experience levels, from novices to advanced programmers, not only for its effectiveness in resolving complex problems but also for its speed and ease with which it completes tasks without requiring excessive attention to the details of their implementation.

REFERENCES

Ballamudi, V. K. R. (2016). Utilization of Machine Learning in a Responsible Manner in the Healthcare Sector. *Malaysian Journal of Medical and Biological Research*, *3*(2), 117-122. https://mjmbr.my/index.php/mjmbr/article/view/677

Ballamudi, V. K. R. (2019a). Artificial Intelligence: Implication on Management. *Global Disclosure of Economics and Business*, *8*(2), 105-118. https://doi.org/10.18034/gdeb.v8i2.540

Ballamudi, V. K. R. (2019b). Road Accident Analysis and Prediction using Machine Learning Algorithmic Approaches. *Asian Journal of Humanity, Art and Literature*, *6*(2), 185-192. https://doi.org/10.18034/ajhal.v6i2.529

Ballamudi, V. K. R. (2019c). Hybrid Automata: An Algorithmic Approach Behavioral Hybrid Systems. *Asia Pacific Journal of Energy and Environment*, *6*(2), 83-90. https://doi.org/10.18034/apjee.v6i2.541

Ballamudi, V. K. R. (2020). Militarization of Space. *Asian Journal of Applied Science and Engineering*, *9*(1), 169–178. https://doi.org/10.18034/ajase.v9i1.38

Ballamudi, V. K. R., & Desamsetti, H. (2017). Security and Privacy in Cloud Computing: Challenges and Opportunities. *American Journal of Trade and Policy*, *4*(3), 129–136. https://doi.org/10.18034/ajtp.v4i3.667

Bodepudi, A., Reddy, M., Gutlapalli, S. S., & Mandapuram, M. (2019). Voice Recognition Systems in the Cloud Networks: Has It Reached Its Full Potential?. *Asian Journal of Applied Science and Engineering*, *8*(1), 51–60. https://doi.org/10.18034/ajase.v8i1.12

Bogdanchikov, A., Zhaparov, M., Suliyev, R. (2013). Python to Learn Programming. *Journal of Physics: Conference Series*, *423*(1). https://doi.org/10.1088/1742-6596/423/1/012027

Campbell, J. C., Hindle, A., Amaral, J. N. (2015). Error Location in Python: Where the Mutants Hide. *PeerJ PrePrints*. https://doi.org/10.7287/peerj.preprints.1132v1

Chen, S., Thaduri, U. R., & Ballamudi, V. K. R. (2019). Front-End Development in React: An Overview. *Engineering International*, *7*(2), 117–126. https://doi.org/10.18034/ei.v7i2.662

Dekkati, S. (2020). Blender and Unreal Engine Character Design and Behavior Programming for 3D Games. *ABC Journal of Advanced Research*, *9*(2), 115-126. https://doi.org/10.18034/abcjar.v9i2.704

Dekkati, S., & Thaduri, U. R. (2017). Innovative Method for the Prediction of Software Defects Based on Class Imbalance Datasets. *Technology & Management Review*, *2*, 1–5. https://upright.pub/index.php/tmr/article/view/78

Dekkati, S., Lal, K., & Desamsetti, H. (2019). React Native for Android: Cross-Platform Mobile Application Development. *Global Disclosure of Economics and Business*, *8*(2), 153-164. https://doi.org/10.18034/gdeb.v8i2.696

Dekkati, S., Thaduri, U. R., & Lal, K. (2016). Business Value of Digitization: Curse or Blessing?. *Global Disclosure of Economics and Business*, *5*(2), 133-138. https://doi.org/10.18034/gdeb.v5i2.702

Deming, C., Dekkati, S., & Desamsetti, H. (2018). Exploratory Data Analysis and Visualization for Business Analytics. *Asian Journal of Applied Science and Engineering*, *7*(1), 93–100. https://doi.org/10.18034/ajase.v7i1.53

Desamsetti, H. (2016a). A Fused Homomorphic Encryption Technique to Increase Secure Data

Storage in Cloud Based Systems. *The International Journal of Science & Technoledge*, *4*(10), 151-155.

Desamsetti, H. (2016b). Issues with the Cloud Computing Technology. *International Research Journal of Engineering and Technology (IRJET)*, *3*(5), 321-323.

Desamsetti, H. (2018). Internet of Things (IoT) Technology for Use as Part of the Development of Smart Home Systems. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *5*, 14–21. https://upright.pub/index.php/ijrstp/article/view/89

Desamsetti, H. (2020). Relational Database Management Systems in Business and Organization Strategies. *Global Disclosure of Economics and Business*, *9*(2), 151-162. https://doi.org/10.18034/gdeb.v9i2.700

Desamsetti, H., & Lal, K. (2019). Being a Realistic Master: Creating Props and Environments Design for AAA Games. *Asian Journal of Humanity, Art and Literature*, *6*(2), 193-202. https://doi.org/10.18034/ajhal.v6i2.701

Desamsetti, H., & Mandapuram, M. (2017). A Review of Meta-Model Designed for the Model-Based Testing Technique. *Engineering International*, *5*(2), 107–110. https://doi.org/10.18034/ei.v5i2.661

Gouillart, E., Nunez-Iglesias, J., Stéfan, V. D. W. (2016). Analyzing Microtomography Data With Python and the Scikit-Image Library. *Advanced Structural and Chemical Imaging*, *2*(1), 1-11. https://doi.org/10.1186/s40679-016-0031-0

Gutlapalli, S. S. (2016a). An Examination of Nanotechnology's Role as an Integral Part of Electronics. *ABC Research Alert*, *4*(3), 21–27. https://doi.org/10.18034/ra.v4i3.651

Gutlapalli, S. S. (2016b). Commercial Applications of Blockchain and Distributed Ledger Technology. *Engineering International*, *4*(2), 89–94. https://doi.org/10.18034/ei.v4i2.653

Gutlapalli, S. S. (2017a). Analysis of Multimodal Data Using Deep Learning and Machine Learning. *Asian Journal of Humanity, Art and Literature*, *4*(2), 171–176. https://doi.org/10.18034/ajhal.v4i2.658

Gutlapalli, S. S. (2017b). The Role of Deep Learning in the Fourth Industrial Revolution: A Digital Transformation Approach. *Asian Accounting and Auditing Advancement*, *8*(1), 52–56. Retrieved from https://4ajournal.com/article/view/77

Gutlapalli, S. S. (2017c). An Early Cautionary Scan of the Security Risks of the Internet of Things. *Asian Journal of Applied Science and Engineering*, *6*, 163–168. Retrieved from https://ajase.net/article/view/14

Gutlapalli, S. S., Mandapuram, M., Reddy, M., & Bodepudi, A. (2019). Evaluation of Hospital Information Systems (HIS) in terms of their Suitability for Tasks. *Malaysian Journal of Medical and Biological Research*, *6*(2), 143–150. https://mjmbr.my/index.php/mjmbr/article/view/661

Koehler, S., Desamsetti, H., Ballamudi, V. K. R., & Dekkati, S. (2020). Real World Applications of Cloud Computing: Architecture, Reasons for Using, and Challenges. *Asia Pacific Journal of Energy and Environment*, *7*(2), 93-102. https://doi.org/10.18034/apjee.v7i2.698

Lal, K. (2015). How Does Cloud Infrastructure Work?. *Asia Pacific Journal of Energy and Environment*, *2*(2), 61-64. https://doi.org/10.18034/apjee.v2i2.697

Lal, K. (2016). Impact of Multi-Cloud Infrastructure on Business Organizations to Use Cloud Platforms to Fulfill Their Cloud Needs. *American Journal of Trade and Policy*, *3*(3), 121–126. https://doi.org/10.18034/ajtp.v3i3.663

Lal, K., & Ballamudi, V. K. R. (2017). Unlock Data's Full Potential with Segment: A Cloud Data Integration Approach. *Technology & Management Review*, *2*(1), 6–12. https://upright.pub/index.php/tmr/article/view/80

Lal, K., Ballamudi, V. K. R., & Thaduri, U. R. (2018). Exploiting the Potential of Artificial Intelligence in Decision Support Systems. *ABC Journal of Advanced Research*, *7*(2), 131-138. https://doi.org/10.18034/abcjar.v7i2.695

Mandapuram, M. (2017a). Application of Artificial Intelligence in Contemporary Business: An Analysis for Content Management System Optimization. *Asian Business Review*, *7*(3), 117–122. https://doi.org/10.18034/abr.v7i3.650

Mandapuram, M. (2017b). Security Risk Analysis of the Internet of Things: An Early Cautionary Scan. *ABC Research Alert*, *5*(3), 49–55. https://doi.org/10.18034/ra.v5i3.650

Mandapuram, M., & Hosen, M. F. (2018). The Object-Oriented Database Management System versus the Relational Database Management System: A

Comparison. *Global Disclosure of Economics and Business*, *7*(2), 89–96. https://doi.org/10.18034/gdeb.v7i2.657

Mandapuram, M., Gutlapalli, S. S., Bodepudi, A., & Reddy, M. (2018). Investigating the Prospects of Generative Artificial Intelligence. *Asian Journal of Humanity, Art and Literature*, *5*(2), 167–174. https://doi.org/10.18034/ajhal.v5i2.659

Mandapuram, M., Gutlapalli, S. S., Reddy, M., Bodepudi, A. (2020). Application of Artificial Intelligence (AI) Technologies to Accelerate Market Segmentation. *Global Disclosure of Economics and Business 9*(2), 141–150. https://doi.org/10.18034/gdeb.v9i2.662

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., et al. (2017). SymPy: Symbolic Computing in Python. *PeerJ Computer Science*. https://doi.org/10.7717/peerj-cs.103

Meyer, R., Obermayer, K. (2016). pypet: A Python Toolkit for Data Management of Parameter Explorations. *Frontiers in Neuroinformatics*. https://doi.org/10.3389/fninf.2016.00038

Miles, M. (2016). Using web2py Python Framework for Creating Data-Driven Web Applications in the Academic Library. *Library Hi Tech*, *34*(1), 164-171. https://doi.org/10.1108/LHT-08-2015-0082

Pedersen, M., Phillips, A., Plotkin, G. D. (2015). A High-Level Language for Rule-Based Modelling. *PLoS One, 10*(6), e0114296. https://doi.org/10.1371/journal.pone.0114296

Perez, R. E., Jansen, P. W., Joaquim R. R. A. M. (2012). pyOpt: a Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization. *Structural and Multidisciplinary Optimization*, *45*(1), 101-118. https://doi.org/10.1007/s00158-011-0666-3

Reddy, M., Bodepudi, A., Mandapuram, M., & Gutlapalli, S. S. (2020). Face Detection and Recognition Techniques through the Cloud Network: An Exploratory Study. *ABC Journal of Advanced Research*, *9*(2), 103–114. https://doi.org/10.18034/abcjar.v9i2.660

Thaduri, U. R. (2017). Business Security Threat Overview Using IT and Business Intelligence. *Global Disclosure of Economics and Business*, *6*(2), 123-132. https://doi.org/10.18034/gdeb.v6i2.703

Thaduri, U. R. (2018). Business Insights of Artificial Intelligence and the Future of Humans. *American Journal of Trade and Policy*, *5*(3), 143–150. https://doi.org/10.18034/ajtp.v5i3.669

Thaduri, U. R. (2019). Android & iOS Health Apps for Track Physical Activity and Healthcare. *Malaysian Journal of Medical and Biological Research*, *6*(2), 151-156. https://mjmbr.my/index.php/mjmbr/article/view/678

Thaduri, U. R. (2020). Decision Intelligence in Business: A Tool for Quick and Accurate Marketing Analysis. *Asian Business Review*, *10*(3), 193–200. https://doi.org/10.18034/abr.v10i3.670

Thaduri, U. R., & Lal, K. (2020). Making a Dynamic Website: A Simple JavaScript Guide. *Technology & Management Review*, *5*, 15–27. https://upright.pub/index.php/tmr/article/view/81

Thaduri, U. R., Ballamudi, V. K. R., Dekkati, S., & Mandapuram, M. (2016). Making the Cloud Adoption Decisions: Gaining Advantages from Taking an Integrated Approach. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *3*, 11–16. https://upright.pub/index.php/ijrstp/article/view/77

Thodupunori, S. R., & Gutlapalli, S. S. (2018). Overview of LeOra Software: A Statistical Tool for Decision Makers. *Technology & Management Review*, *3*(1), 7–11.

Turkel, W. J., Crymble, A. (2012). Setting Up an Integrated Development Environment for Python (Mac). *The Programming Historian.* https://doi.org/10.46430/phen0012

Turkel, W. J., Crymble, A. (2012). Setting Up an Integrated Development Environment for Python (Windows). *The Programming Historian.* https://doi.org/10.46430/phen0019

--0--