

ANNEXURE

PYTHON

Python is a high-level, interpreted programming language that was first released in 1991. It was designed to be easy to read, write, and understand, with a clear and concise syntax that emphasizes code readability. Python is an object-oriented language that supports many programming paradigms, including procedural, functional, and imperative programming. It has a large and active community of developers who contribute to its development and maintenance, and a vast library of pre-built modules and frameworks that make it easier to build complex applications. Python is widely used in many industries and domains, including web development, data science, machine learning, scientific computing, artificial intelligence, and more. It is also the language of choice for many introductory programming courses due to its simplicity and ease of learning.

HISTORY OF PYTHON

Python was first created by Guido van Rossum in the late 1980s and released in 1991. Guido van Rossum named the language after the British comedy group Monty Python. Python was developed as a successor to the ABC language and aimed to be a simple and easy-to-use language for general-purpose programming. Python's syntax and semantics were designed to be easy to read and understand, and its modular architecture made it easy to use and extend. Python gained popularity in the mid-1990s, particularly in academic and scientific communities. The release of Python 2.0 in 2000 introduced several new features, including list comprehensions, garbage collection, and support for Unicode. In 2008, Python 3.0 was released, which introduced several major changes to the language, including

improved Unicode support, the removal of some features that were considered redundant, and a more consistent syntax.

PYTHON FEATURES

- **Easy to learn and read:** Python has a simple and straightforward syntax that is easy to read and write, making it an excellent language for beginners to learn.
- **Object-oriented programming:** Python supports object-oriented programming (OOP) principles, allowing developers to create reusable and modular code.
- **Large standard library:** Python comes with a large standard library of pre-built modules and functions that can be used to perform a range of tasks, from file I/O to network programming.
- **Cross-platform compatibility:** Python code can be run on a variety of platforms, including Windows, macOS, and Linux, making it highly versatile and adaptable.
- **Interpreted language:** Python is an interpreted language, meaning that code can be executed without the need for compilation, making it easier to test and debug.
- **Dynamic typing:** Python is a dynamically typed language, meaning that variable types are determined at runtime, making it more flexible and easier to use.
- **Community support:** Python has a large and active community of developers who contribute to its development and maintenance, as well as creating a wealth of resources, tutorials, and libraries that make it easier to develop Python applications.

ADVANTAGES OF PYTHON

Easy to learn and read: Python has a simple and straightforward syntax that is easy to read and write, making it an excellent language for beginners to learn.

Large standard library: Python comes with a large standard library of pre-built modules and functions that can be used to perform a range of tasks, from file I/O to network programming, saving time and effort in development.

Cross-platform compatibility: Python code can be run on a variety of platforms, including Windows, macOS, and Linux, making it highly versatile and adaptable.

Interpreted language: Python is an interpreted language, meaning that code can be executed without the need for compilation, making it easier to test and debug.

Object-oriented programming support: Python supports object-oriented programming (OOP) principles, allowing developers to create reusable and modular code.

High-level language: Python is a high-level language, meaning that it abstracts many low-level programming details, making it more productive and easier to use.

Dynamic typing: Python is a dynamically typed language, meaning that variable types are determined at runtime, making it more flexible and easier to use.

Community support: Python has a large and active community of developers who contribute to its development and maintenance, as well as creating a wealth of resources, tutorials, and libraries that make it easier to develop Python applications.

Versatility: Python is used in a wide range of applications, from web development to scientific computing and artificial intelligence.

Scalability: Python's performance can be improved with the use of third-party libraries and tools, making it scalable for large-scale applications.

Flexibility: Python is a versatile language that can be used for a variety of applications, from small scripts to complex applications.

DISADVANTAGES OF PYTHON

Speed: Python is an interpreted language and can be slower than compiled languages like C and C++. While Python's performance can be improved with the use of third-party libraries and tools, it may not be the best choice for high-performance applications that require low latency.

Memory consumption: Python's dynamic typing and garbage collection can result in higher memory consumption compared to languages like C++.

Design restrictions: Python's dynamic typing and high-level abstractions may make it less suitable for some applications that require low-level control over hardware resources.

Global interpreter lock: Python's global interpreter lock (GIL) limits the ability to use multiple CPU cores for certain types of applications, such as CPU-bound or multithreaded applications.

Steep learning curve for some libraries: While Python has a large and diverse range of libraries and modules, some of them have a steep learning curve, which may require significant time investment to use effectively.

Security: Python's dynamic nature and flexibility can make it vulnerable to certain types of security risks, such as code injection and cross-site scripting attacks.

APPLICATIONS

Web development: Python can be used to build web applications using popular web frameworks like Django and Flask.

Data science and analysis: Python's libraries like NumPy, Pandas, and SciPy are commonly used for data analysis, visualization, and machine learning.

Artificial intelligence and machine learning: Python's libraries like TensorFlow, Keras, and PyTorch are used for developing artificial intelligence and machine learning applications.

Automation and scripting: Python is used for automating tasks and creating scripts for various applications like system administration, network automation, and test automation.

Gaming: Python can be used for developing games and simulations using libraries like Pygame.

Education: Python is commonly used for teaching programming concepts and is the language of choice for many introductory computer science courses.

Scientific computing: Python is used in scientific computing applications, including computational physics, biology, and chemistry.

Financial applications: Python is used for financial modeling, quantitative analysis, and algorithmic trading in the finance industry.

Desktop applications: Python can be used to develop desktop applications using GUI toolkits like Tkinter, PyGTK, and PyQt.

VISUAL STUDIO CODE

VS Code is a free and open-source source-code editor developed by Microsoft for Windows, macOS, and Linux operating systems. It is a lightweight editor that offers a range of features, including syntax highlighting, code completion, debugging, Git integration, and customizable user interface. VS Code is designed to be highly customizable and extensible through a vast library of plugins and extensions that can be installed directly from the integrated Marketplace. This makes it a popular choice among developers across different programming languages, including JavaScript, Python, Java, and C++.

VSCODE FEATURES

Support for multiple programming languages: Supports multiple programming languages. So earlier, programmers needed Web-Support: a different editor for different languages, but it has built-in multi-language support. This also means it easily detects if there's any fault or cross-language reference, it'll be able to detect it easily.

Intelli-Sense: It can detect if any snippet of code is left incomplete. Also, common variable syntaxes and variable declarations are made automatically. Ex: If a certain variable is being used in the program and the user has forgotten to declare, intelli-sense will declare it for the user.

Cross-Platform Support: Traditionally, editors used to support either Windows or Linux or Mac Systems. But Visual Studio Code is cross-platform. So it can work on all three platforms. Also, the code works on all three platforms; else, the open-source and proprietary software codes used to be different.

Extensions and Support: Usually supports all the programming languages but, if the user/programmer wants to use the programming language which is not supported then, he can download the extension and use it. And performance-wise, the extension doesn't slow down the editor as it runs as a different process.

Repository: With the ever-increasing demand for the code, secure and timely storage is equally important. It is connected with Git or can be connected with any other repository for pulling or saving the instances.

Web-Support: Comes with built-in support for Web applications. So web applications can be built and supported in VSC.

Hierarchy Structure: The code files are located in files and folders. The required code files also have some files, which may be required for other complex projects. These files can be deleted as per convenience.

Improving Code: Some code snippets can be declared a bit differently, which might help the user in the code. This function prompts the user, wherever necessary, to change it to the suggested option.

Terminal Support: Many of the times, the user needs to start from the root of the directory to start with a particular action, in-built terminal or console provides user support to not to switch in-between two screens for the same.

Multi-Projects: Multiple projects containing multiple files/folders can be opened simultaneously. These projects/folders might or might not be related to each other.

Git Support: Resources can be pulled from Git Hub Repo online and vice-versa; saving can be done too. Resource pulling also means cloning the code which is made available on the internet. This code can later be changed and saved.

Commenting: A common feature, but some of the languages do not support it. Commenting on the code helps the user to recall or track according to the sequence he wants.

ADVANTAGES OF VSCODE

Cross-platform support: VS Code is available for Windows, macOS, and Linux, making it a versatile option for developers using different operating systems.

Lightweight and fast: VS Code is a lightweight editor that uses minimal system resources, making it faster than many other code editors.

Built-in debugging: VS Code has a built-in debugger that allows developers to debug their code without the need for an external debugger.

Extensible: VS Code can be extended with various plugins and extensions that provide additional functionality, such as support for different programming languages and frameworks.

Git integration: VS Code has built-in Git integration, making it easy for developers to manage their code repositories without having to switch to a separate Git client.

IntelliSense: VS Code provides intelligent code completion and suggestions based on the context, making coding faster and more accurate.

Integrated terminal: VS Code has an integrated terminal that allows developers to run commands and scripts without leaving the editor.

DISADVANTAGES OF VSCODE

Steep learning curve: Despite being a lightweight editor, It has many features and can be overwhelming for new users, which may take some time to learn.

Resource-intensive extensions: Although VS Code's extensibility is a significant advantage, some extensions may use a lot of system resources, which can slow down the editor.

Limited visual design capabilities: VS Code is primarily focused on code editing, so it has limited visual design capabilities. Developers who need to create graphical user interfaces or design layouts may need to use a different tool.

No built-in support for some programming languages: Although VS Code supports many programming languages out of the box, some languages may require additional plugins or extensions.

Limited debugging capabilities: While VS Code has a built-in debugger, it may not be as advanced as some external debugging tools, which could limit the ability to debug complex applications.

APPLICATIONS

Web development: VS Code is well-suited for web development tasks such as HTML, CSS, and JavaScript development, as well as building web applications using frameworks like Angular, React, and Vue.

Mobile app development: VS Code supports mobile app development for both iOS and Android platforms using tools like Xamarin, React Native, and Flutter.

Cloud computing: VS Code has several extensions that allow developers to work with cloud computing platforms like Microsoft Azure, AWS, and Google Cloud.

Machine learning and data science: VS Code supports the development of machine learning and data science projects with tools like Jupyter Notebooks and extensions for Python, R, and other languages.

CHAPTER 5

REFERENCES

- [1] Ashanti Widiana, Mohammad Iqbal Jerusalem, Budi Yumechas “The Application of Text-to-Speech Technology in Language Learning” ICOLLITE 2022.
- [2] Chooi Shir Ley, Aimi Syammi Binti Ab Ghafar “Handwritten Character Recognition Using Convolutional Neural Network” Progress in Engineering Application and Technology, Volume 2, Issue 1, 2021.
- [3] Dr.S.A. Ubale, Girish Bhosale, Ganesh Nehe, Avinash Hubale, Avdhoot Walunjkar “A Review on Text-to-Speech Converter” International Journal of Innovative Research in Technology, Volume 9, Issue 1, June 2022.
- [4] Karun Somasunder M, Amal J.S, Gopal Gopakumar, Suraj V Thomas, Keerthi Krishnan “Text to Speech Conversion Using Google Vision API” International Journal of Innovative Research in Technology, Volume 8, Issue7, 2021.
- [5] Nisha P, Dr.J. Vijayakumar “Survey paper: Text to Speech conversion for visually impaired peoples” International Research Journal of Engineering and Technology, Volume 8, Issue 6, June 2021.
- [6] Paras Doshi, Yash Shirke, Tejas Hegde, Pranav Dhanvij “Text Reader for Visually Impaired Using Google Cloud Vision API” International Journal of Innovative Research in Technology, Volume 4, Issue 12, May 2018.
- [7] Prof. Mrunalinee Patole, Akhilesh Pandey, Kaustubh Bhagwat, Mukesh Vaishnav, Salikram Chadar Survey on “Text-to-Speech Systems for Real-Time

Audio Synthesis” International Journal of Advanced Research in Science, Communication and Technology, Volume 6, Issue 1, June 2021.

[8] Sharvari S, Usha A, Karthik P, Mohan Babu C “Text to Speech Conversion using Optical Character Recognition” International Research Journal of Engineering and Technology, Volume 7, Issue 7, July 2020.

[9] Shruti Mankar, Nikita Khairnar, Mrunali Pandav, Hitesh Kotecha, Manjiri Ranjanikar “A Recent Survey Paper on Text-To-Speech Systems” International Journal of Advanced Research in Science, Communication and Technology, Volume 3, Issue 2, January 2023.

[10] Vaishnavi R.Ambaskar, Vinod M. Lokhande, Dr. Avinash S. Kapse A Review on “Text-to-Speech System” International Journal of Advance Research and Innovative Ideas in Education, Volume 8, Issue 3, March 2022.

[11] [BUILD OPTICAL CHARACTER RECOGNITION](#)

[12] [Translation an advanced resource](#)

[13] [Speech and Language Processing \(3rd ed. draft\)](#)