Share on your Social Media

Python Tutorial for Beginners

Published On: October 1, 2024

Python Tutorial for Beginners: Installation to Operations

Building dependable software for nearly any purpose is possible with Python's ease of use, intuitiveness, and robustness. In this Python tutorial, you will learn the installation and operations of Python programming to get started on your learning journey.

Download Python Tutorial PDF

Introduction to Python

Python is a popular computer programming language used for creating software and websites, task automation, and data analysis. In this Python tutorial, we cover the following:

- Overview of Python
- Various Python Versions
- Proper Installation of Python
- How Do I Install Python Correctly?
- Operations on File in Python
- Advantages of Python

Overview of Python

Python is a general-purpose language that may be

Featured Articles

Related Courses at SLA

- Python Training in OMR
- Python Training in Chennai
- Python Online Training

Related Posts

used to develop a wide range of programs; it is not tailored to solve any particular issues.

- On a server, web applications can be made with Python.
- Workflows can be made with Python in addition to applications.
- Database systems are connectable from Python.
- Complex mathematics and large data handling are possible with Python.
- Python is useful for both software development and is suitable for production and quick prototyping.

Key Features of Python

- Easy to Code.
- Open Source & Free Software.
- Support for GUI.
- Object-Oriented Methodology.
- High-Level Language.
- Highly portable language.
- Integrated by nature.
- Extremely Dynamic.

Python Interview Questions and Answers

Various Python Versions

When choosing a Python version to ensure compatibility with third-party modules, it is always safer to go with one major point revision following the current one.

The most recent version of Python is 3.12 as of this writing. Therefore, it is safest to utilize Python 3.11's most recent upgrade. Going back to one version ensures the best compatibility with popular third-party Python packages.

Python Project Ideas

Published On: October 26, 2024

Introduction A Python professional uses the Python programming language for software development, automation, and data...

MERN Stack Tutorial for Web Development Aspirants

Published On: October 14, 2024

MERN Stack Tutorial for Web Development Aspirants There is a growing need for competent MERN...

Tableau Developer Salary in Chennai

Published On: October 12, 2024

Introduction A Tableau Developer designs, develops, and maintains dashboards and visualizations using Tableau software. Key...

VMware Tutorial for Cloud Computing

Still, you may always test out the latest version of Python in a controlled manner, such as on a virtual machine or test computer.

Similar to Linux, Python is available in a multitude of distributions. However, unlike Linux, Python has a single, reliable, “official” edition that you can always rely on, CPython, made available at python.org by the Python Software Foundation. Gain expertise with faster app development with our **DevOps training in Chennai**.

Choosing between the 32-bit and 64-bit versions of Python is a crucial decision you’ll need to make, particularly on Windows. 64-bit is the most likely response for the following reasons:

- The bulk of modern operating systems use 64-bit Python by default. Users of Windows can run 32-bit versions of Python on 64-bit Windows but with little performance impact.
- 32-bit applications and 32-bit Python programs can access a maximum of 4 GB of RAM at once.
- Many Python data analysis and machine learning tools work better in 64-bit versions since this limit does not apply to 64-bit programs. Some are exclusive to 64-bit versions.
- If a third-party module you’re using is limited to 32-bit editions, or if you’re compelled to use a 32-bit version of Windows, then you should only choose the 32-bit version of Python.

Python Course Syllabus PDF

Proper Installation of Python

Python uses an installer that walks you through the setup process and installs on Windows just like any other application.

By default, the Python installer for Windows stores its executables under the user's AppData directory, preventing the need for administrator access. Install Python in a higher-level directory if you are the only user of the machine to make it easier to discover. The target directory can be specified using the Windows installer. Kickstart your testing career with our **software testing training courses** available with us.

Tips to choose the right Python installer for Windows

Python is available for Windows in several versions, according to Python.org. You have the option of using the embeddable zip file, the executable installer, or the web-based installer in addition to the 32-bit ("x86") and 64-bit ("x86-64") versions that were previously mentioned. This is the main idea behind those:

- The Python setup is carried out via the executable installer, which is simply a '.exe' file. The most popular and simple default option is this one.
- Similar to the executable installer, the web-based installer downloads the necessary files individually to complete the installation. This needs a network connection, but it significantly minimizes the size of the installer itself.
- With no dependencies and the ability to execute in a single folder, the embeddable zip file is a simple and self-contained version of the Python program.
    - When you need a fast, one-time Python install to test something on the spot or when you want to manually distribute a Python application, a bundle can be helpful.
    - Yet, pip and all the other helpful tools that come with a full install are not included in the embeddable zip, so it should only be used by experts.

Installing Python using Package

Manager

Using one of the Windows package management solutions is still an additional choice.

NuGet is a '.NET package management tool' that has Python available in its repository.

But the primary purpose of having Python available there is to use it as an element of a '.NET program', not to set up a stand-alone instance of Python for everyday usage.

Installing the standard Python method should make managing your Python instance easier.

Python is also available through the more all-encompassing Windows package manager, Chocolatey.

- Chocolatey is a more practical option than NuGet for installing the Python language runtime and monitoring its presence on your system.
- On the same machine, it is advisable to refrain from combining conventional Python installations with Chocolatey installs.

Installing Python on Linux Correctly

Python is usually installed on Linux via the package manager of the particular distribution, as Linux distributions vary greatly from one another. For example, the installation process for Python varies greatly between Fedora and Ubuntu. Usually, the Python version number determines the destination location for the installation on Linux (and MacOS).

Example: */usr/bin/python3.X on Linux, or /usr/local/opt/python/* on the Mac.

Using a Python runtime that is containerized is one method to get around the complexities of Linux package management.

You can stop worrying about multiple Python runtimes stomping on each other's toes because containers operate in isolation from the rest of the system.

Note: If containers aren't already a part of your process, you'll have to invest time and effort in learning Docker.

A useful tool in this context is also 'asdf-vm'. Numerous Python runtimes, as well as numerous runtimes for Node.js, Ruby, Elixir, and many more languages, may be managed on Unix-like platforms (Linux and MacOS) with the help of asdf-vm.

You should therefore explore asdf-vm if you frequently find yourself juggling versions of other programs in addition to Python.

Installing Python on MacOS Correctly

An older version of Python—Python 2.7—has never been included in the MacOS installation bundle. The two versions frequently clashed, which caused issues when Python 3 was released.

The official Python manual offers a few statements to that effect, but the only more detailed advice is to make sure you use the correct path for the Python instance you want.

Using the Homebrew package manager on Mac OS is a popular method of managing Python runtimes. For obtaining, installing, managing, and uninstalling Python and other third-party command-line applications, Homebrew offers a standardized user interface. Check out our **full-stack developer course** to have a better career in web development.

Installing Python Packages Correctly

Even if you only want to use Python for one project, once you have a base install of a Python version set up, you should not begin installing packages straight into it using pip.

After configuring your project directories, establish virtual environments for Python and install packages within them. In this manner, the base installation is kept tidy.

An overview of managing several projects with virtual environments and dependencies may be seen in the Poetry Project. Poetry provides a high-level command-line tool for controlling dependencies and virtual environments.

Install different versions of Python simultaneously

Managing several Python versions installed side by side is the single most difficult problem to solve when it comes to Python installations. Here, two general guidelines come into play:

Install every version in a distinct directory at all times.

- Make sure that all system paths are set up to automatically start at the version you wish to run.
- Per-project virtual environments are strongly supported by running various versions of Python. All Python activity within the project's context is automatically routed toward the appropriate version of Python upon activation of the virtual environment.
- The 'py' launcher software gives Windows users another way to choose which Python version to use when multiples are installed.
  - Installing the 'py' launcher, a tiny application that allows you to choose the version of Python to use for a particular script using command-line options is an option that appears during the Python setup process.
  - For example, py – 3.7 -m pip would be entered to execute pip for Python 3.7.

Upgrade Python Correctly

Python revision upgrades of minor versions, from 3.7.2 to 3.10, are usually not too difficult. When an installer finds the previous version on Windows, it updates it. The installer or package manager usually performs the same function on Linux and MacOS.

The virtual environments you have built, however, do not automatically upgrade; you will also need to upgrade them. Simply navigate to the virtual environment directory and type 'venv –upgrade' to upgrade Python in a virtual environment. Remember that this is only effective for small point revision upgrades, such as from Python 3.7.2 to Python 3.7.3.

- Using venv to establish a new, distinct virtual environment subdirectory in the project directory, reinstalling any dependencies into it, and switching to using the new virtual environment is your best option if you're undertaking a big point revision update, like upgrading Python 3.10 to 3.11.
- You can transition between several virtual environments in a project by using the majority of IDEs that support Python, including Microsoft Visual Studio Code.

If you are a serious job seeker, our **Python full-stack job seeker training program** provides you with a promising career in the app development field.

Python Developer Salary for Freshers

How Do I Install Python Correctly?

Python can be installed on Windows in a few simple steps.

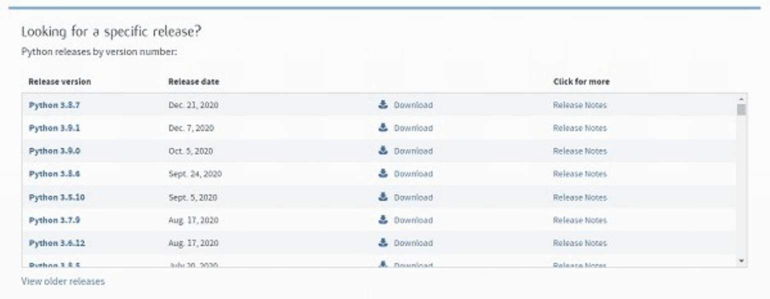
Step 1: Choose the Python version to install

There are multiple versions of Python available, and each version differs in terms of syntax and functionality. Selecting the version that we require or wish to use is necessary. For Python 2 and Python 3, there are various versions available.

Step 2: Download the Python Executable Installer

Navigate to the Download for Windows section on the Python official website (www.python.org) using your web browser.

There will be a list of every Python version that is available. After choosing the version you need, click "Download." Assume for the moment that we selected Python 3.9.1.



Looking for a specific release?

Python releases by version number:

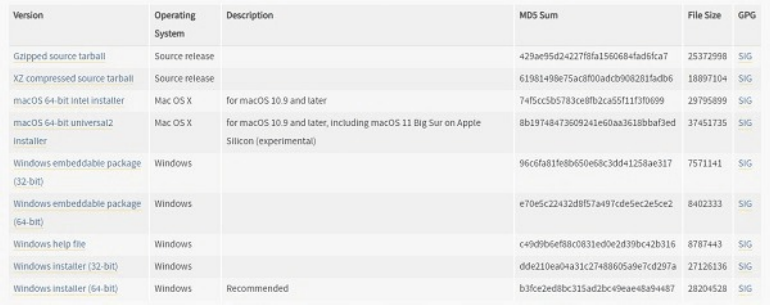
Release version	Release date		Click for more
Python 3.8.7	Dec. 21, 2020	Download	Release Notes
Python 3.9.1	Dec. 7, 2020	Download	Release Notes
Python 3.9.0	Oct. 5, 2020	Download	Release Notes
Python 3.8.6	Sept. 24, 2020	Download	Release Notes
Python 3.5.10	Sept. 5, 2020	Download	Release Notes
Python 3.7.9	Aug. 17, 2020	Download	Release Notes
Python 3.6.12	Aug. 17, 2020	Download	Release Notes
Python 3.8.5	July 20, 2020	Download	Release Notes

View older releases

Python Tutorial

A list of possible executable installers with varying operating system requirements will appear when you click "download." Select the installer that corresponds with your operating system and start the installation process.

Assuming we choose the Windows 64-bit installation, the download size is less than 30MB.



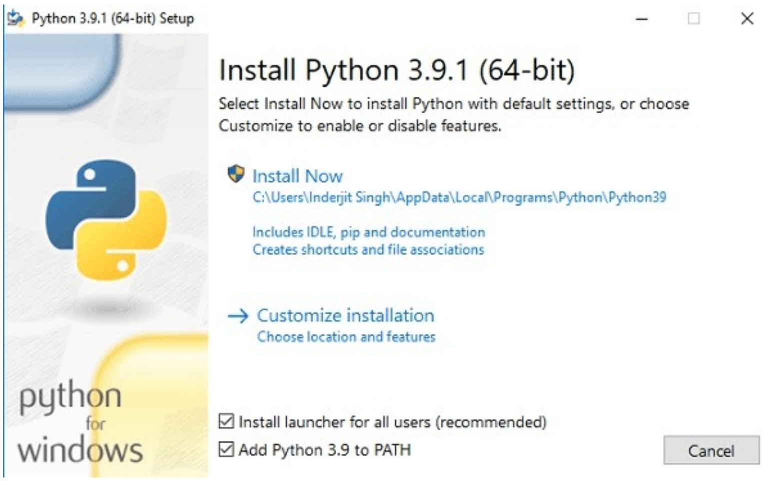
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		429ae95d24227f8fa1560684fad6fca7	25372998	SIG
XZ compressed source tarball	Source release		61981498e75ac8f00adcb908281fadb6	18897104	SIG
macOS 64-bit Intel installer	Mac OS X	for macOS 10.9 and later	74f5cc5b5783ce8fb2ca55f11f3f0699	29795899	SIG
macOS 64-bit universal2 installer	Mac OS X	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	8b19748473609241e60aa3618bbaf3ed	37451735	SIG
Windows embeddable package (32-bit)	Windows		96c6fa81fe8b650e68c3dd41258ae317	7571141	SIG
Windows embeddable package (64-bit)	Windows		e70e5c22432d8f57a497cde5ec2e5ce2	8402333	SIG
Windows help file	Windows		c49d9b6ef88c0831ed0e2d39bc42b316	8787443	SIG
Windows installer (32-bit)	Windows		dde210ea04a31c27488605a9e7cd297a	27126136	SIG
Windows installer (64-bit)	Windows	Recommended	b3fce2ed8bc315ad2bc49eae48a94487	28204528	SIG

Python Tutorial 2

Step 3: Launch the Executable Installer

The Python 3.9.1 Windows 64-bit installation was downloaded.

Launch the installation. After confirming that the two checkboxes at the bottom are selected, click Install New.



Python 3.9.1 (64-bit) Setup

Install Python 3.9.1 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Inderjit Singh\AppData\Local\Programs\Python\Python39

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.9 to PATH

Python Tutorial 3

Clicking the Install Now button starts the installation procedure.

Python Tutorial 4

It will take a few minutes to finish the installation process, and when it does, the following screen appears.

Step 4: Confirm that Python is installed correctly on Windows.

To verify whether Python has been installed on your system successfully, take the prescribed actions:

- Get the command prompt open.
- Enter "python" after typing it.
- If Python has been installed on your Windows computer successfully, the version of Python that you have installed will be shown.

Step 5: Confirm that Pip was set up.

Pip is a robust mechanism for managing Python software packages. Therefore, confirm that it is installed.

To confirm if the pip was installed, adhere to the instructions provided.

- Get the command prompt open.
- To find out if pip was installed, enter pip –V.
- If the pip is installed successfully, the output is shown in the command line.

Python and pip have been successfully installed on our Windows computer.

Operations on File in Python

Operations on files in Python are vital for manipulating, analyzing, and storing data, as well as for reading and writing data to and from files.

The basics of file operations in Python, including how to open, read, write, and manipulate files, as well as more advanced techniques for efficient file management, will be covered here:

Why are file operations in Python important?

Working with files is a basic requirement when working with huge datasets for machine learning tasks. Python is one of the most used languages for data science; therefore, you should be familiar with all its file operations. Now, let's explore a few Python file operations. Upgrade your skills with our **Data Science with Python Course**.

Opening a File using the open() function

To work with files in Python, you must first learn how to open files. To open files, use the open() function. Python's open() method takes two parameters. The file name and entire path are the first, while the file open mode is the second.

The following are popular reading modes for files:

Reading Modes	**Descriptions**
r	This mode indicates that the file is intended to be read-only.
w	This option indicates that writing to the file will only be permitted. It will create a new file if the one with that name doesn't exist.
a	This mode indicates that the program's output will be appended to the file's earlier output.
	This option indicates

r+	that writing and reading of the file will be permitted.
w+	To read and write data. It will replace the current data.
a+	To obtain and append data from the file. It is not going to replace the current data.

Moreover, you can append “b” to access the file in binary for the Windows operating system. This is a result of Windows’ ability to distinguish between ordinary text files and binary text files.

Let’s say we put a text file called “file.txt” in the same directory as our code. It’s time to open the file. Nevertheless, the open(filename, mode) function returns a file object. You can continue with your task using that file object.

> ### *Example*
>
> ```
> text_file = open(‘file.txt’,’r’)
>
> text_file2 = open(‘/home/imtiaz/file.txt’,’r’)
>
> print(‘First Method’)
>
> print(text_file)
>
> print(‘Second Method’)
>
> print(text_file2)
> ```

Output

```
First Method

Second Method
```

Reading and Writing Files in Python

Python provides many ways to read from and write to files, with each function exhibiting unique behavior. The file operations mode is one item to be aware of. You must open a file in read or write mode to read it. Python requires that the file be open in write mode to write to it. The following are a few Python functions that let you read and write to files:

Python Functions	**Descriptions**
read()	This function retrieves a string after reading the full file.
readline()	This function retrieves a string representation of the lines it reads from the file. If it is invoked the nth time, it retrieves line n.
readlines()	This function yields a list with a single line from that file for each element.
write()	This function creates a file with a predetermined character sequence.
writelines()	This function outputs a string list.
append()	Rather than overwriting the file, this function adds a string to it.

Example

```
text_file = open(‘/Users/SLA/abc.txt’,’r’)
```

```
line_list = text_file.readlines();

for line in line_list:

    print(line)

text_file.close() #don’t forget to close the file
```

Output

```
>>

========= RESTART: /Users/SLA/Desktop/read-file.py ====

Hi SLA

I am here

>>
```

Now that we understand how to read a file in Python, let’s use the writelines() function to execute a write operation.

Example

```
text_file = open(‘/Users/SLA/file.txt’,’w’)

word_list= []

for i in range (1, 5):

    print(“Please enter data: “)

    line = input() #take input

    word_list.append(line)

text_file.writelines(word_list) #write 4 words to the file

text_file.close() #don’t forget to close the file
```

Output

```
>>>

========= RESTART: /Users/SLA/Desktop/write-
```

```
file.py ======

Please enter data:

1

Please enter data:

2

Please enter data:

3

Please enter data:

4

>>>
```

Copying Files in Python using the shutil() method

Python’s shutil method can be used to copy files. With the help of this tool, we can move and copy files between different Python environments.

For example

```
import shutil

shutil.copy2(‘/Users/SLA/abc.txt’,
‘/Users/SLA/abc_copy2.txt’)

shutil.copyfile(‘/Users/SLA/abc.txt’,
‘/Users/SLA/abc_copyfile.txt’) #alternative way

print(“File Copy Done”)
```

Delete Files in Python with shutil.os.remove() method

Use the remove() function from the shutil module in Python to remove files from the file system. Let’s see how to execute a delete operation in Python.

```
import shutil

import os
```

```
shutil.os.remove(‘/Users/SLA/abc_copy2.txt’)
```

```
os.remove(‘/Users/SLA/abc_copy2.txt’)
```

Closing Files in Python using close() method

You must close any Python files you open after making any modifications to them. This eliminates the file from memory, saves whatever changes you’ve already made, and stops the application from reading or writing to the file again.

Syntax

```
fileobject.close()
```

Example

```
text_file = open(‘/Users/SLA/abc.txt’,’r’)
```

```
text_file.close()
```

If you use the with block, you can also avoid having to manually close files. The files are closed and become unreadable and unwritable as soon as the with block is run.

Python FileNotFoundError

When interacting with files in Python, it’s typical to encounter a ***FileNotFoundError***. By giving whole file paths while creating the file object, it may be easily prevented.

```
File “/Users/SLA/Desktop/string1.py”, line 2, in <module>
```

```
text_file = open(‘/Users/SLA/Desktop/abc.txt’,’r’)
```

```
FileNotFoundError: [Errno 2] No such file or directory: ‘/Users/SLA/Desktop/abc.txt’
```

You only need to confirm that the path you provided for the file open method is accurate to resolve the *FileNotFoundError*.

Advantages of Python File Operations

The following are the advantages of Operations on Files in Python

Versatility: You can do a lot of different actions with files in Python, including creating, reading, writing, appending, renaming, and deleting them.

Flexibility: Working with many file types (text, binary, CSV, etc.) and performing various actions on files (read, write, append, etc.) makes file handling in Python extremely flexible.

User-friendly: It's easy to create, read, and edit files with Python's file management interface.

Cross-platform: Python routines for managing files are compatible and easily integrated with a variety of operating systems and platforms, including Windows, Mac, and Linux.

Disadvantages of Python File Operations

The limitations of operations on files in Python

Error-prone: Python file handling operations may be prone to mistakes, particularly if the code is poorly designed or the file system (such as file permissions, file locks, etc.) has problems.

Security risks: Managing files in Python may also give rise to security problems, particularly when the application allows user input that could be exploited to gain access to or alter confidential files on the system.

Complexity: Working with increasingly sophisticated file formats or operations in Python might make file handling more difficult. For files to be handled correctly and securely, careful attention to the coding is required.

Performance: Python's operations on files can be slower than those of other programming languages, particularly when processing huge files or carrying out intricate tasks.

Implementations of Operations on File in Python

We will go over each of the previously seen principles in this example. In addition to that, we'll also look at how to use the Python OS module's remove() function to remove a file.

> ***Example***

```
import os

def create_file(filename):

    try:

        with open(filename, ‘w’) as f:

            f.write(‘Hello, world!\n’)

        print(“File ” + filename + ” created successfully.”)

    except IOError:

        print(“Error: could not create file ” + filename)

def read_file(filename):

    try:

        with open(filename, ‘r’) as f:

            contents = f.read()

            print(contents)

    except IOError:

        print(“Error: could not read file ” +
filename)
def append_file(filename, text):
    try:
        with open(filename, ‘a’) as f:
            f.write(text)
        print(“Text appended to file ” +
filename + ” successfully.”)
    except IOError:
        print(“Error: could not append to file ”
+ filename)
def rename_file(filename, new_filename):
    try:
        os.rename(filename, new_filename)
        print(“File ” + filename + ” renamed to ”
+ new_filename + ” successfully.”)
    except IOError:
        print(“Error: could not rename file ” +
filename)
def delete_file(filename):
    try:
        os.remove(filename)
        print(“File ” + filename + ” deleted
successfully.”)
```

```
    except IOError:
        print(“Error: could not delete file ” + filename)

if __name__ == ‘__main__’:
    filename = “example.txt”
    new_filename = “new_example.txt”
    create_file(filename)
    read_file(filename)
    append_file(filename, “This is some additional text.\n”)
    read_file(filename)
    rename_file(filename, new_filename)
    read_file(new_filename)
    delete_file(new_filename)
```

Output

```
File example.txt created successfully.
Hello, world!
Text appended to file example.txt successfully.
Hello, world!
This is some additional text.
File example.txt renamed to new_example.txt successfully.
Hello, world!
```

```
This is some additional text.

File new_example.txt deleted successfully.
```

Python Online Training

Advantages of Python

The popular programming language Python has a lot of benefits, such as:

Easy to learn: Python is well-known for being readable and writeable even for beginners.

Flexible: Python is a high-level language that can be used for many tasks, such as developing scripts, data analysis, and web development.

Huge library: The modules and packages available in Python have a vast and rich library.

Scalable: Python is capable of supporting distributed computing and handling massive volumes of data.

Cross-platform: Python is cross-platform compatible.

Safe: The OWASP Python Security Project assists programmers in writing more secure code.

Open-source: Python is free and open-source software.

Active community: Python boasts a sizable and vibrant development community that offers tools and assistance.

Beginner-friendly: Python is a great language for prototyping because of its versatility and ease of use.

Conclusion

For beginners, Python is a simple and intuitive language to learn. This is due to its extremely understandable grammar, which is similar to that of the English language. Begin your learning journey with this Python tutorial. Learn comprehensively with hands-on exposure in our **Python training in Chennai**.

Share on your Social Media

Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training
- Corporate Training
- Hire With Us
- Job Seekers
- SLA’s Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python

Social Media Links

- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development
- DevOps

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed
- Software Suggest
- Sitejabber