

18-09-2024

Training Day -3

Install Python

1. Download Python:

- Go to the [official Python website](https://www.python.org/downloads/).
- Click the **Download Python** button to get the latest version for your operating system (Windows, macOS, or Linux).

2. Install Python:

- **Windows:**
 1. Run the downloaded .exe file.
 2. Check the box **"Add Python to PATH"** (very important).
 3. Select **Customize Installation** if needed, or choose **Install Now**.
- **macOS:**
 1. Open the .pkg file and follow the instructions.
 2. Python is pre-installed on macOS, but it's recommended to use the latest version.
- **Linux:**
 1. Use your package manager:

```
bash
Copy code
sudo apt update
sudo apt install python3
```

3. Verify Installation:

- Open a terminal or command prompt.
- Run:

```
bash
Copy code
python --version
```

Or for some systems:

```
bash
Copy code
python3 --version
```

Install PyCharm

1. Download PyCharm:

- Go to the [official PyCharm website](https://www.jetbrains.com/pycharm/).
- Choose the **Community Edition** (free) or **Professional Edition** (paid).

2. Install PyCharm:

- **Windows:**
 1. Run the downloaded `.exe` file.
 2. Follow the installation wizard.
 3. Check the box to **add PyCharm to PATH** or create a desktop shortcut.
- **macOS:**
 1. Open the `.dmg` file and drag the PyCharm icon to the Applications folder.
- **Linux:**
 1. Extract the downloaded `.tar.gz` file.
 2. Navigate to the `bin` folder in the extracted directory.
 3. Run:

```
bash
Copy code
./pycharm.sh
```

3. Configure PyCharm:

- Open PyCharm.
- Configure your Python interpreter:
 1. Go to **File > Settings > Project > Python Interpreter**.
 2. Click **Add Interpreter** and choose the installed Python version.

Types of Application:

1. Console Applications

Console applications are programs that run in a command-line interface (CLI) or terminal window. They primarily use text-based input and output and do not have a graphical user interface (GUI).

Use Cases:

- Automation scripts.
- System utilities (e.g., disk cleanup tools).

Examples:

- `ping` (networking tool).
- `gcc` (C compiler).
- Python command-line scripts.

2. Windows Applications

Windows applications are programs with a graphical user interface (GUI) designed to run on the Microsoft Windows operating system. They are user-friendly and visually interactive.

Use Cases:

- Desktop productivity tools (e.g., Microsoft Word, Excel).
- Multimedia applications (e.g., VLC Media Player).
- Gaming applications.
- Enterprise software (e.g., accounting or ERP systems).

Examples:

- Microsoft Paint.
- Notepad++.
- Adobe Photoshop.

User Interaction:

1 Input:

Users interact through a graphical interface using a combination of mouse clicks, keyboard inputs, or touch (on touchscreen devices).

Interactive elements include text boxes, buttons, sliders, drop-down menus, and more.

2 Output:

Feedback is displayed visually via pop-ups, labels, images, charts, or other GUI components.

Allows for richer data presentation, like tables, graphs, or multimedia elements

Data Types: To represent different types of values

Single element:

int: Whole Numbers 22 -563

float: Decimal Point: 2.0 ,3.62

complex: $2+3j$, Real+Imaginary

Bool: Only 2 values, True, False

None Type: None

Multi element:

Iterators, str, list, tuple, dict, set, frozenset

#New Program

a=b #NameError: name 'b' is not defined

```
# print(a)
```

```
# #New Program
```

```
# a=2+3j
```

```
# print(type(a))
```

STRING

str: Syntax: characters in a sequence included within single quotes, double quotes, triple quotes

str is a collection of homogeneous data types ie collection of characters.

Single Line String: Single, Double or Triple Quotes

Multi Line String: Only triple quotes

```
"""
```

```
# s='Welcome to CETPA'
```

```
# print(s)
```

```
# s="Welcome to CETPA"
```

```
# print(s)
```

```
# s="Welcome to CETPA"
```

```
# print(s)
```

```
# s="""Welcome to CETPA"""
```

```
# print(s)
```

```
# #New Program
```

```
# a="a"
```

```
# print(type(a))
```

```
# #New Program
```

```
# s="Welcome to CETPA.
```

```
# CETPA is an award winning training company.
```

```
# CETPA is awarded by Chetan Bhagat, Shasho Tharoor and ..."
```

```
# print(s)
```

```
# s="""Welcome to CETPA.
```

```
# CETPA is an award winning training company.
```

```
# CETPA is awarded by Chetan Bhagat, Shasho Tharoor and ..."""
```

```
# print(s)
```

"""

len function is created to find the length of iterators

len function won't work on single elements data type

INPUT FUNCTION:

Syntax:

```
input("Message for user")
```

```
var_name=input("Message for user").
```

Input function will hold the program till the time we press enter or pass the value and press enter.

.....**New Program**.....

```
# s=input("Enter Your Name:")
```

```
# print(s)
```

.....**New Program**.....

Addition of two input numbers

```
# a=input("Enter First No:")    #a=5
```

```
# b=input("Enter Second No:")   #b=7
```

```
# r=a+b                        #
```

```
# print(r)                     #
```

Whenever we interact with the user in python then data is always travelled in string format.

input function always returns strings type of data in our program.

print function firstly call a **__str__** method on the arguments of the print function, and this str method always returns string data type and this string value returned from **__str__** method is printed on the screen.

```
# # print(25)
```

```
# L=[10,20,30]
```

```
# s=str(L)
```

```
# print(s)
```

```
# print(L)
```

```
# r=L.__str__()
# print(r)+ operator if applied on string then it concatenate the
strings ie join the strings.
+ operator works as a concatenation operator in case of
strings.
"""
```

.....**New Program**.....

```
# a=5
# b=7
# s=a+b
# print(s)
# a="Ram"
# b="Shyam"
# s=a+b
# print(s)
```

.....**New Program**.....

```
# a="5"
# b="7"
# r=a+b
# print(r)
```

.....**New Program**.....

Addition of Two numbers: Incorrect Approach

```
# no1=input("Enter First No:") #no1="5"
# no2=input("Enter Second No:") #no2="7"
# res=no1+no2 #res="57"
# print(res)
```

Whenever we take input from user then first and foremost thing is plan to consider the data type of input.

Type Casting

To convert one data type to another data type in programming then this concept is called type casting. also called type conversion, is the process of converting a variable from one data type to another. It is commonly used in programming to make operations compatible with variables of different types. There are two main types of type casting:.

1. Implicit Type Casting (Type Coercion):

- **Performed automatically by the compiler or interpreter.**
- Happens when a conversion is safe, such as from a smaller data type to a larger one (e.g., int to float).

Example in Python:

```
python
Copy code
x = 5      # Integer
y = 2.5    # Float
z = x + y  # x is implicitly converted to a float
print(z)   # Output: 7.5
.
```

2. Explicit Type Casting (Type Conversion):

- **Performed manually by the programmer.**
- Requires using casting functions or methods to convert one type to another.
- May result in loss of precision or data if not done carefully

Type Casting Syntax:

dest_var=dest_type(src_var)

.....New Program.....

```
# x=5
# y=float(x)
# print(y,type(y))
```

.....New Program.....

```
# x="5"
# y=float(x)
# print(y,type(y))
```

.....New Program.....

```
# x="5"
# y=int(x)
# print(y,type(y))
```

.....New Program.....

```
# x=2.35
# y=int(x)
# print(y,type(y))
```

STRING DATA TYPE:

"5" vs 5 in memory

str and int

integers are directly converted to binary and saved

5: 101

ASCII Standard: American Standard Code for Information Interchange

ASCII code were initially developer of 7 bit length and later

it was designed with 8 bit length. ASCII is a standard to represent different standard keys of English Keyboard.

A: 65 a: 97

B: 66 b: 98

C: 67 c: 99

Z: 90 z: 122

0: 48

1: 49

2: 50

3: 51

4: 52

5: 53

9: 57

"5":

Through ASCII numbers we can represent only English Alphabets, Numbers or some special symbols because ASCII code is of only 8 bits.

To consider alphabets or special symbols of other languages of the world, a new standard was created ie Unicode Standard.

16 Bits Combinations: 2^{16} : 65536

Unicode support 16 bits, 32 bits, 64 bits...

Now Python 3 support Unicode

Unicode is a universal character encoding standard designed to represent text in most of the world's writing systems. It assigns a unique code point to every character, regardless of the platform, program, or language, ensuring consistency in text representation.

A: ASCII Code 65, and Unicode 6

ord function returns the Unicode of the character passed

chr function returns the Character of the unicode passed

```
# #New Program
```

```
# print(chr(2346))
```

```
# print(ord("ᵀ"))
```

```
" "
```