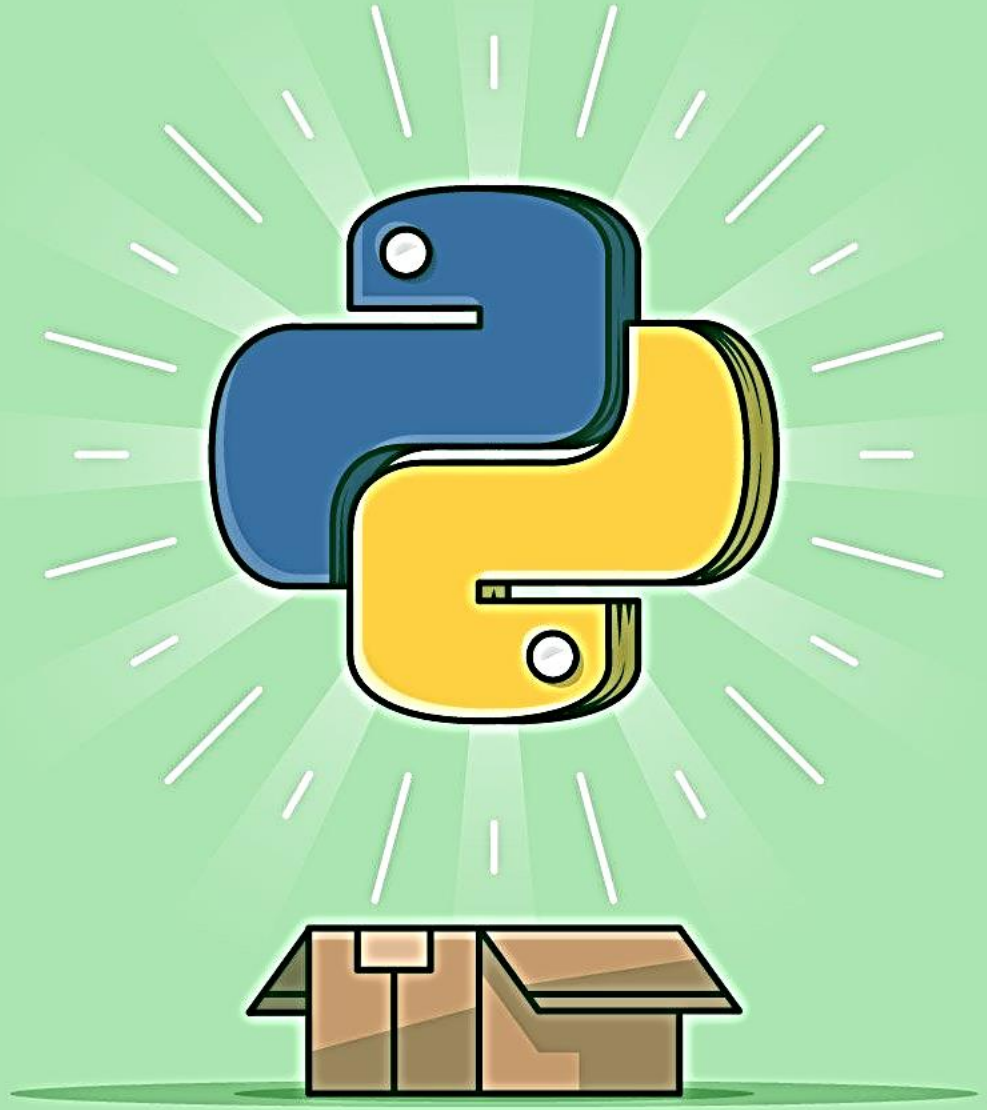
ISLab Python Course

Session 1: Introduction to Python



Presenters:

Shahrzad Shashaani

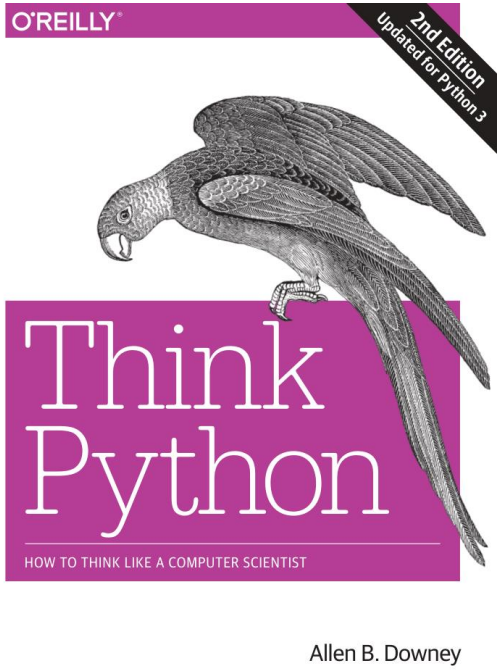
Hamed Homaei Rad

Saeed Samimi

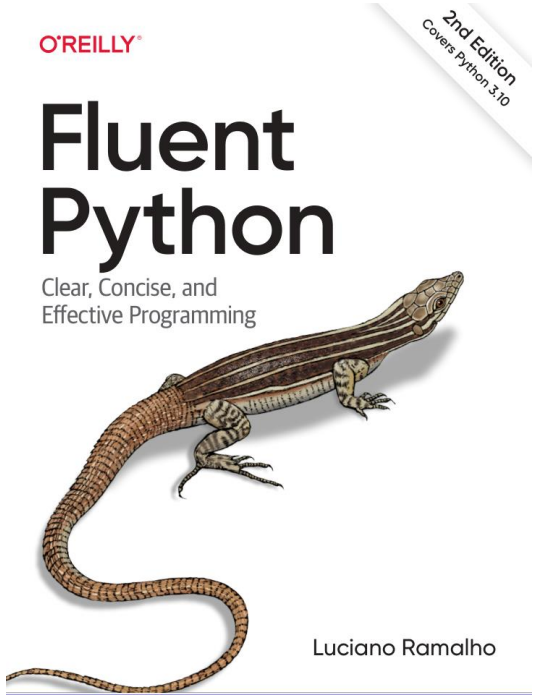
Summer 2023

K.N.Toosi University of Technology

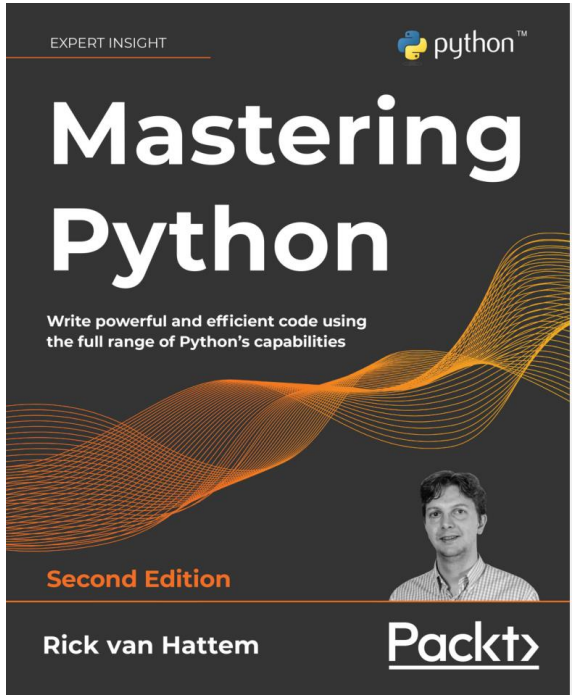
Reference Books



[O'Reilly] Think Python (2015)



[O'Reilly] Fluent Python (2nd Edition, 2022)



[Packt] Mastering Python (2nd Edition, 2022)

Reference Books

[Alexander Shvets] Dive Into Design Patterns (2019)

Course Material

All the material for this course including slides, jupyter notebooks for codes, exercises, etc., are available via the following GitHub repo:

https://github.com/hamedonline/islab2023-python-course

There's also a dedicated private Telegram group in which you'll be added, giving you the option to ask questions and seek guidance from mentors regarding any problems you may encounter during the course, and also discuss and share your progress with your classmates.

Getting Python

- https://www.python.org/
- The most up-to-date and current source code, binaries, documentation, news, etc..
- https://www.python.org/doc/
- Download Python documentation

Python Availability

- Windows
- Linux
- Mac OS

Do You Already Have Python?

CMD →

```
>> C:\Users\User_name>python
```

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021,
11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license"
for more information.
```

Installing Python

- Unix and Linux Installation
  - Open a Web browser and go to https://www.python.org/downloads/.
  - Follow the link to download the *zipped source code* available for *Unix/Linux*.
  - Download and extract files.
  - Editing the Modules/Setup file if you want to customize some options.

```
$ run ./configure script
$ make
$ make install
```

This installs Python at standard location /usr/local/bin and its libraries at /usr/local/lib/pythonXX where XX is the version of Python.

- Red Hat Enterprise Linux (RHEL 8)

```
$ sudo yum install python3
```

Installing Python

- Windows Installation
  - Open a web browser tab and go to https://www.python.org/downloads/.
  - Follow the link for the *Windows installer python-XYZ.exe* file where XYZ is the version you need to install.
  - To use this installer python-XYZ.exe, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
  - Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the installation is finished, and you are done.

Setting up Path

- Unix and Linux
  - **In the csh shell** − type setenv PATH "$PATH:/usr/local/bin/python" and press Enter.
  - **In the bash shell (Linux)** − type export PATH="$PATH:/usr/local/bin/python" and press Enter.
  - **In the sh or ksh shell** − type PATH="$PATH:/usr/local/bin/python" and press Enter.

/usr/local/bin/python is the path of the Python directory.

Setting up Path

- Windows
  - **In the command prompt** − type path %path%;C:\Python and press Enter.

```
C:\Python is the path of the Python directory.
```

Package Management

- pip

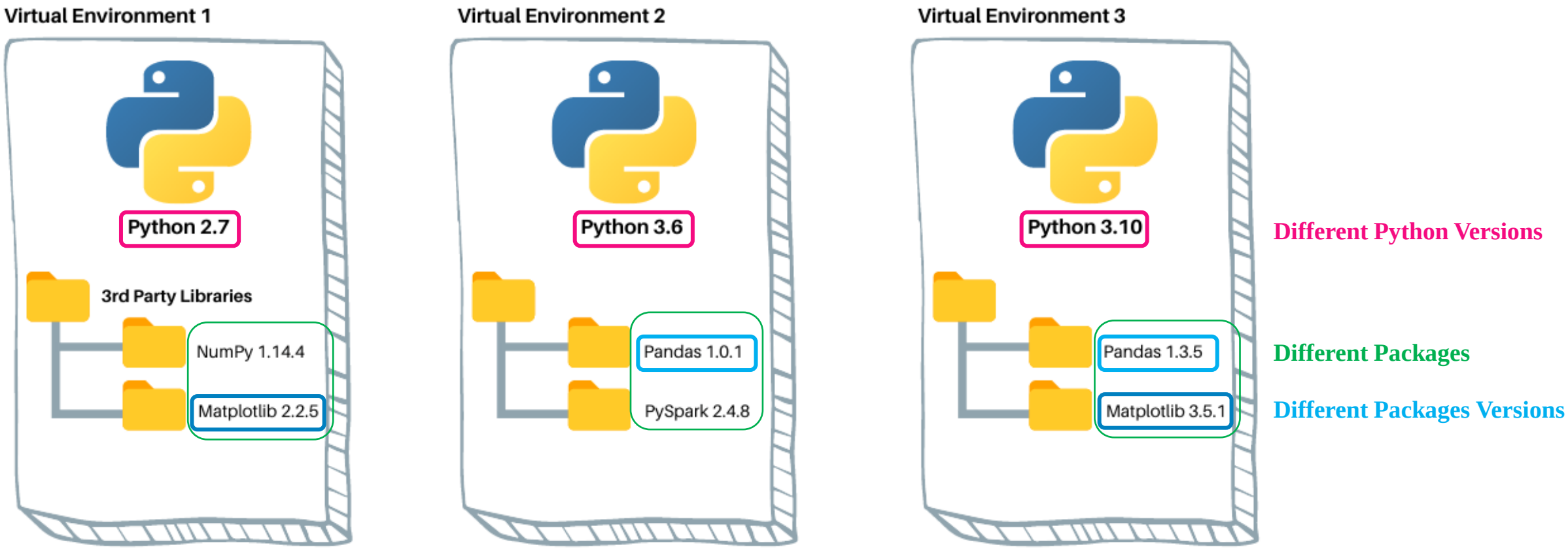
Installing packages	
Linux	Windows
`$ python3 -m pip install package_name`	`>python -m pip install package_name`
Installing packages from a link	
Linux	Windows
`$ python3 -m pip install complete_URL`	`>python -m pip install complete_URL`
Installing packages	
Linux	Windows
`$ python3 -m pip install file_name`	`>python -m pip install file_name`

Package Management

- pip

Install multiple packages using a requirements file	
Linux	Windows
`$ python3 -m pip install -r requirements.txt`	`>python -m pip install -r requirements.txt`
Upgrade a package	
Linux	Windows
`$ python3 -m pip install --upgrade package_name`	`>python -m pip install --upgrade package_name`
Uninstall a package	
Linux	Windows
`$ python3 -m pip uninstall package_name`	`>python -m pip uninstall package_name`

Environment Management



Environment Management

- Installing virtualenv
  - Linux

```
$ python3 -m pip install --user virtualenv
```

  - Windows

```
>python -m pip install --user virtualenv
```

Environment Management

- Creating a Virtual Environment
  - Linux

```
$ python3 -m venv venv_name
```

  - Windows

```
>python -m venv venv_name
```

Environment Management

- Activating a Virtual Environment
  - Linux

```
$ source venv_name/bin/activate
```

…/env/bin/python

  - Windows

```
>.\venv_name\Scripts\activate
```

…\env\Scripts\python.exe

Environment Management

- Installing Packages
  - Linux

```
$ python3 -m pip install package_name
```

  - Windows

```
>python -m pip install package_name
```

Environment Management

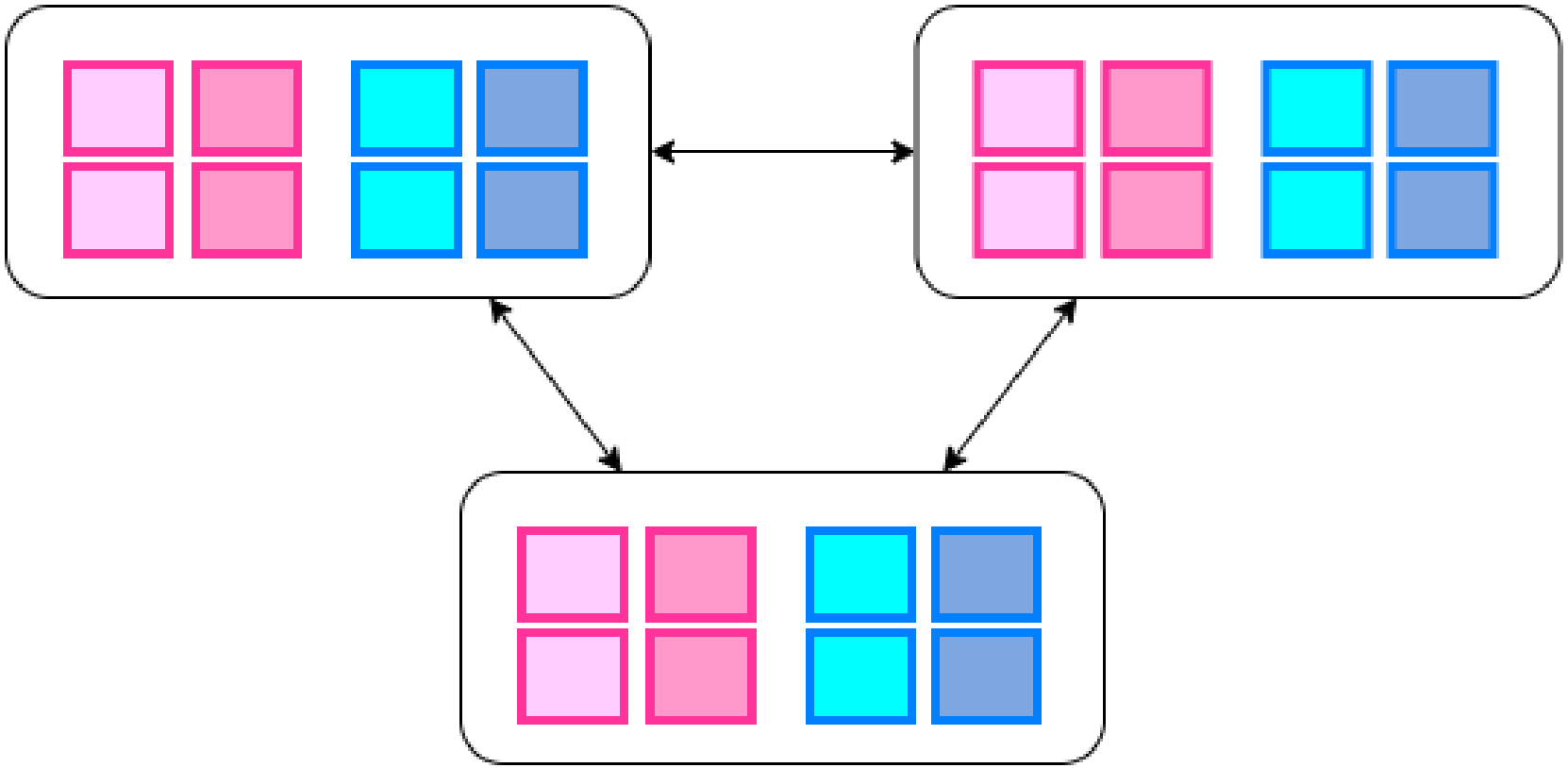
- Leaving the Virtual Environment
  - Linux

```
$ deactivate
```

  - Windows

```
>deactivate
```

Computer Programs



Variables

- Definition: a reserved memory location to store values
  - For a limited while as needed
  - How much space? It depends on data type

- Naming Rules
  - ✓ Use descriptive names
  - ✓ Case-sensitive (my_name, My_Name, MY_NAME)
  - ✓ Variable names must start with a letter or "_" character
  - ✗ We can't use numbers in the beginning of a variable name
  - ✓ Only use alpha-numeric or "_"
  - ✗ Don't use pre-defined keywords (for, if, …)

PEP 8 – Style Guide for Python Code:
https://peps.python.org/pep-0008/

Data Types

- Basic data types
  - Numbers (integer, float, complex)
  - Strings
  - Booleans
- Composite data types
  - Lists
  - Tuples
  - Dictionaries
  - Sets

Assignment Operators in Python

Operator	Description	Syntax
=	Assign value of right side of expression to left side operand	x = y + z
+=	Add and Assign: Add right side operand with left side operand and then assign to left operand	a += b
-=	Subtract AND: Subtract right operand from left operand and then assign to left operand: True if both operands are equal	a -= b
*=	Multiply AND: Multiply right operand with left operand and then assign to left operand	a *= b
/=	Divide AND: Divide left operand with right operand and then assign to left operand	a /= b
%=	Modulus AND: Takes modulus using left and right operands and assign result to left operand	a %= b
//=	Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand	a //= b
**=	Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand	a **= b
&=	Performs Bitwise AND on operands and assign value to left operand	a &= b
|=	Performs Bitwise OR on operands and assign value to left operand	a |= b
^=	Performs Bitwise xOR on operands and assign value to left operand	a ^= b
>>=	Performs Bitwise right shift on operands and assign value to left operand	a >>= b
<<=	Performs Bitwise left shift on operands and assign value to left operand	a <<= b

Arithmetic Operators in Python

Operator	Description	Syntax
+	Addition: adds two operands	x + y
–	Subtraction: subtracts two operands	x – y
*	Multiplication: multiplies two operands	x * y
/	Division (float): divides the first operand by the second	x / y
//	Division (floor): divides the first operand by the second	x // y
%	Modulus: returns the remainder when the first operand is divided by the second	x % y
**	Power (Exponent): Returns first raised to power second	x ** y

Arithmetic Operators in Python

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	**(a == b) is not true.**
!=	If values of two operands are not equal, then condition becomes true.	**(a != b) is true.**
<>	If values of two operands are not equal, then condition becomes true.	**(a <> b) is true. This is similar to != operator.**
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	**(a > b) is not true.**
<	If the value of left operand is less than the value of right operand, then condition becomes true.	**(a < b) is true.**
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	**(a >= b) is not true.**
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	**(a <= b) is true.**

Logical Operators in Python

Operator	Description	Syntax
and	Logical AND: True if both the operands are true	**x and y**
or	Logical OR: True if either of the operands is true	**x or y**
not	Logical NOT: True if operand is false	**not x**

X	Y	X and Y	X or Y	not(X)	not(Y)
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

Identity Operators in Python

- Identity operators are used to compare the objects if both the objects are actually of the same data type and share the same memory location.
  - **‘is’ operator** – Evaluates to True if the variables on either side of the operator point to the same object, and false otherwise.