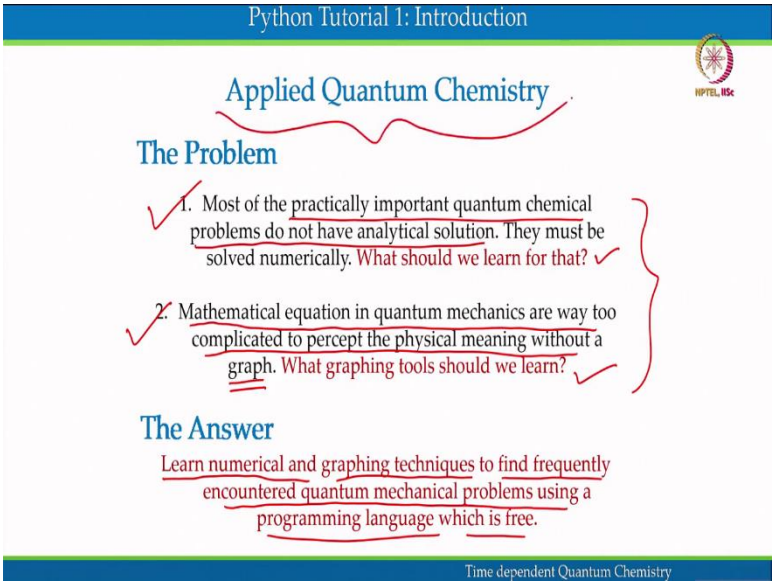
Time-Dependent Quantum Chemistry
Professor. Atanu Bhattacharya
Department of Inorganic and Physical Chemistry
Indian Institute of Science, Bengaluru
Lecture No. 05
Introduction to Python Programming

Welcome to Python Tutorial 1 of the course, Time-Dependent Quantum Chemistry.

(Refer Slide Time: 00:28)



The subject Quantum Chemistry is very often taught at both the undergraduate and graduate levels with almost entire focus on systems having known analytical solutions. Particle in one dimensional box, simple harmonic oscillator and hydrogen atom are very popular examples. On the other hand, most of the practical problems in quantum chemistry and quantum mechanics, even at the very preliminary research level, do not have any analytical solutions, they must be solved numerically.

So, this mismatch poses a vexing problem particularly, when it comes to encouraging students both in experimental and theoretical physical chemistry lab for free thinking of any quantum mechanical problem. So, this is one problem with which we have is most of the practically important quantum mechanical problems do not have analytical solution they must be solved numerically.

Question is: what should I learn for that? Furthermore, even for the problems having known analytical solutions, often inability to build physically perceptible mental picture in the end of a rigorous mathematical derivation also leads to a state of being not well equipped for free thinking of quantum mechanical problems. As the proverb goes, a picture is worth 1000

words, graphs are the best way of portraying the physical concept behind a mathematical expression.

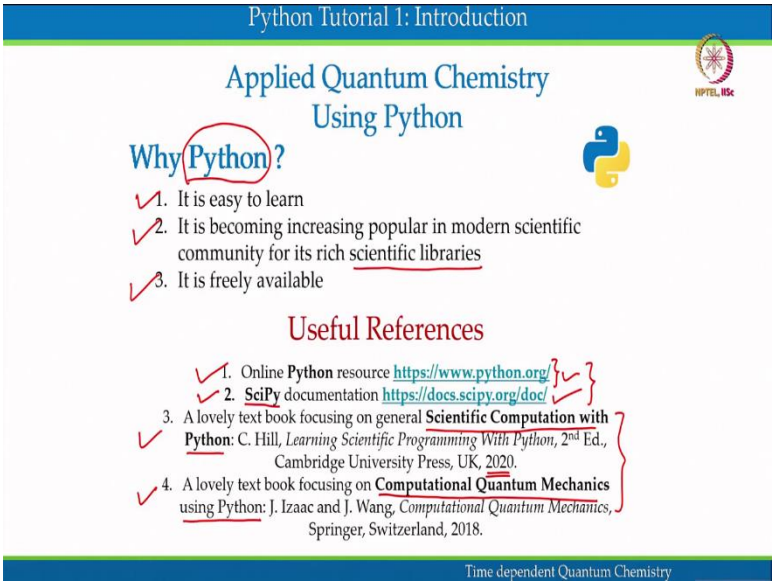
Therefore, to make the study of quantum mechanical systems, quantum chemistry, practically useful and to make the learning process comfortable one in my opinion, blackboard or PowerPoint presentation of quantum mechanics must be supplemented by computer programming based graphing tools. This is true both for time independent and time dependent version of quantum mechanics.

So, another problem is that even I can have analytical solutions in quantum mechanics, these analytical solutions are way too complicated to percept the physical meaning without a graph and question is, how do I use different graphing tools to portray the physical concept behind the quantum mechanical system? And the answer to both of these questions is, we have to learn numerical and graphing techniques to find frequently encountered quantum mechanical problems using programming language which is free, free of cost.

So, the aim of the Python tutorials given in this course, there are multiple tutorials will be given in this course, connected to each chapter of each module of this course, is to introduce numerical techniques, which are commonly required to find solutions to very frequently encountered quantum mechanical problems, such as solution to the Schrodinger equation for arbitrary potential, simulating quantum dynamics under the influence of time dependent or time independent external potential etc.

In addition to that, all the Python tutorials are prepared to provide a set of graphing tools to promote easy construction of mental picture for the analytical solution. So, we will, we will move forward with this introduction which will enable us to perform Applied Quantum Chemistry or Practical Quantum Chemistry going beyond the textbook.

(Refer Slide Time: 04:37)



So, question is what kind of programming language you should use to make this quantum mechanical, the study of quantum mechanical system practically useful? We have selected Python programming language for this course and there are a number of reasons why we have selected Python programming.

It is easy to learn, it is becoming increasingly popular in modern scientific community for its rich scientific libraries. It has been developed for a long time recently, and to support scientific community, and the most importantly, it is freely available. So, in this course, we are not expecting any prior knowledge of programming; we will expect that students does not have any programming knowledge.

So, we will start from the beginning of Python programming and will slowly nudge to the practical problems, which we will be encountering in quantum mechanics, in time dependent quantum mechanics in this course. Any enthusiastic students who will be attending this course would be able to go through the tutorials, without any experts help, that is expected and that is my belief.

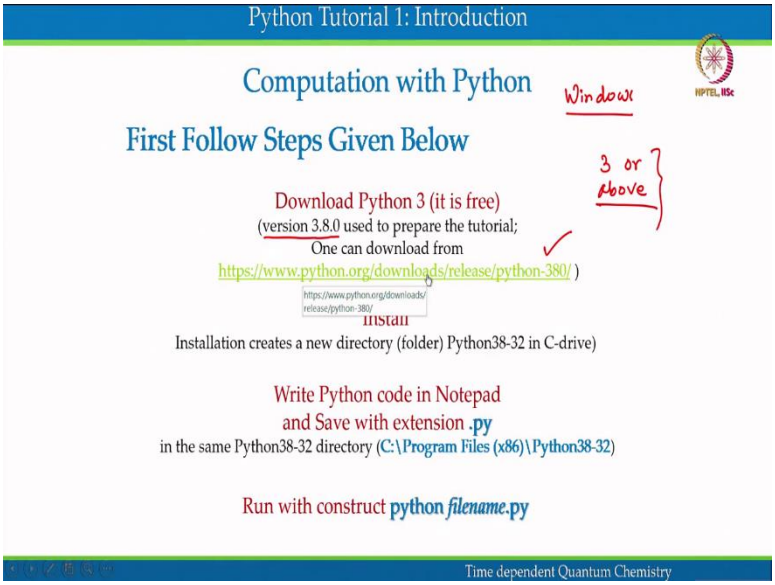
In the Python tutorials, first core concepts of programming for scientific computing are systematically introduced using a modern programming language Python, and, and these references would be very useful. The first thing is, one can navigate this website for Python programming and there are a lot of syntax arguments, methods, functionalities discussed in this website.

And there is a SciPy documentation, all the features are not available, in built in in Python, there are multiple useful features which we will be using for scientific computing, they would be available in SciPy Module or library of Python that can also be navigated in this website. So, both websites can be very useful and I strongly suggest that students, enthusiastic students who would like to learn Python programming and then subsequently would like to use Python programming to explore quantum mechanical problems must they must go through these two websites.

If you love to read textbooks, then then I would strongly suggest these two textbooks, they are lovely textbooks. The first one, this one Scientific Computation with Python, that is very new textbook, one can learn scientific, different scientific programming with Python, not necessarily useful for quantum mechanical problems, but for in general scientific problems.

On the other hand, the second textbook Computational Quantum Mechanics using Python, this textbook has entirely focused the solutions or problems associated with quantum mechanics using Python. So, I would encourage enthusiastic students to go through these two textbooks.

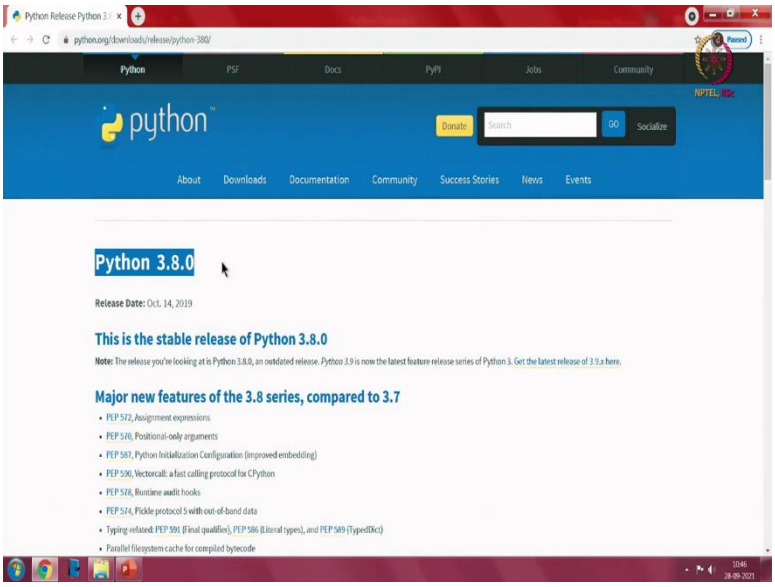
(Refer Slide Time: 08:47)



Before we jump into Python programming, there are a few steps we need to follow. Students can comfortably go through the instructions given in the Python tutorial of this course, if they have Windows base computer, so we are assuming that we are using Windows operating system and one can first download Python from this website.

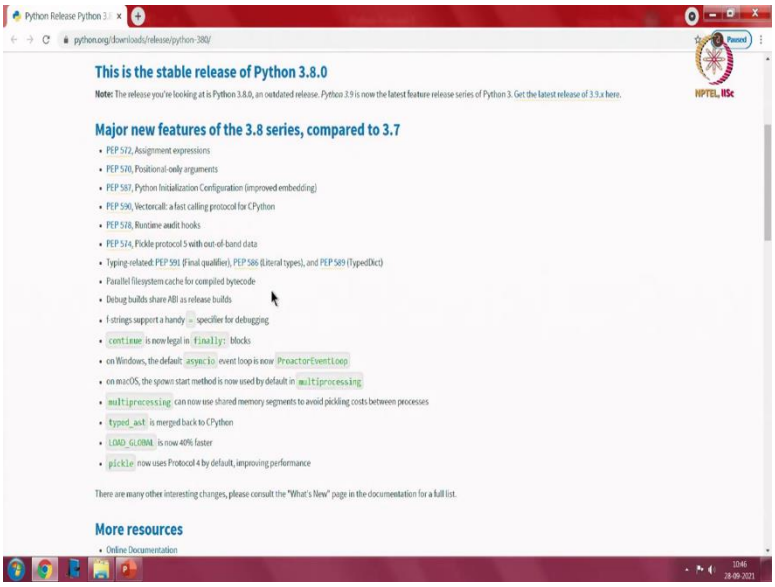
The entire tutorial has been prepared using the Python version 3.8.0 and I suggest that definitely Install Python, version 3 or above. Do not use Python 2, because all the libraries which we will be using, they may not be available with Python 2, but Python 3 and above they are available. And if you go to this site, this is Python site.

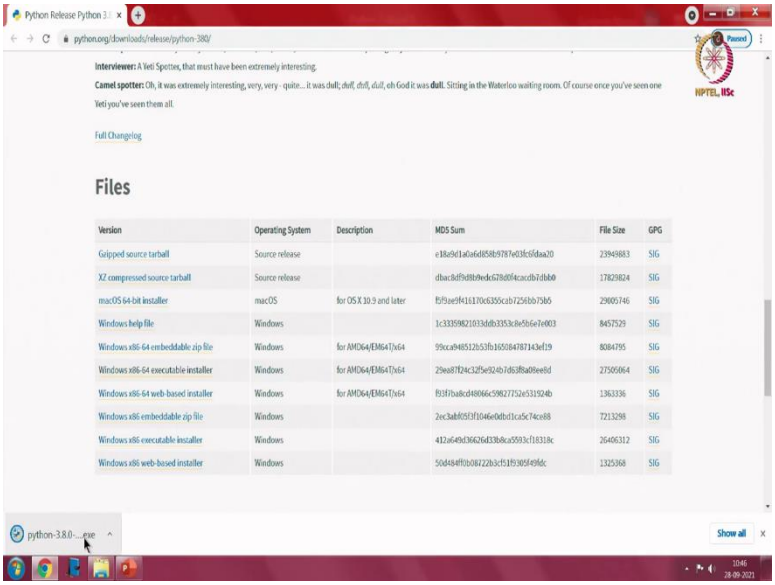
(Refer Slide Time: 10:02)

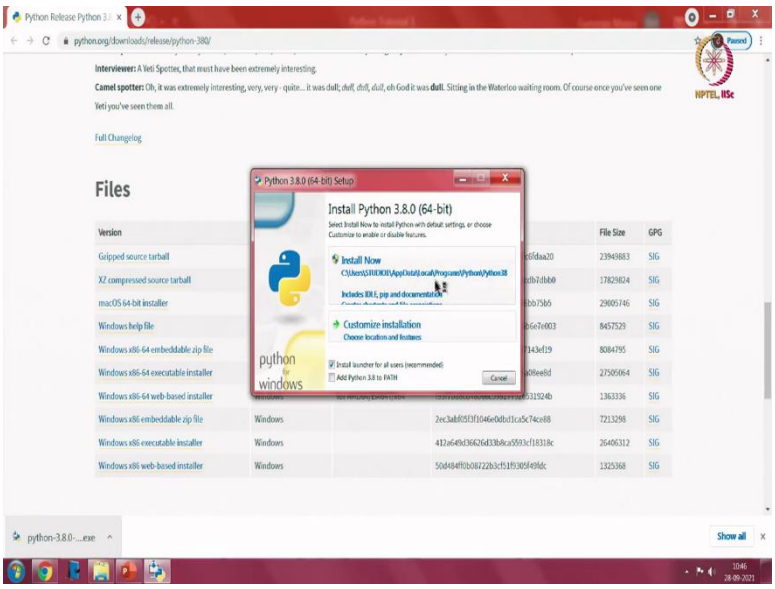


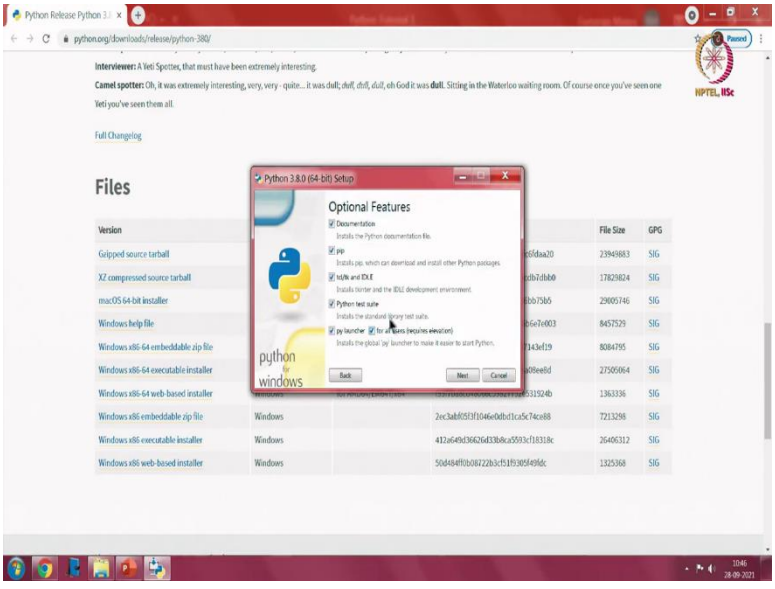
Then one can find out this Python 3.8.0 it was released in 2019.

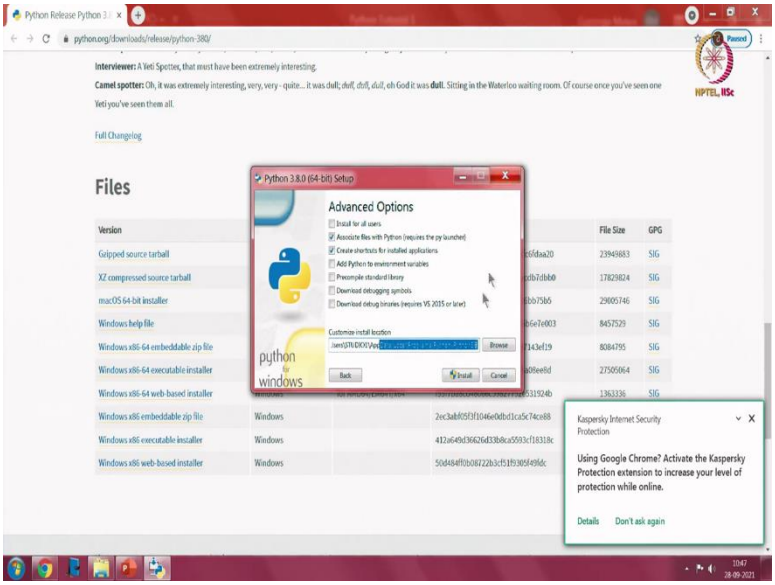
(Refer Slide Time: 10:10)

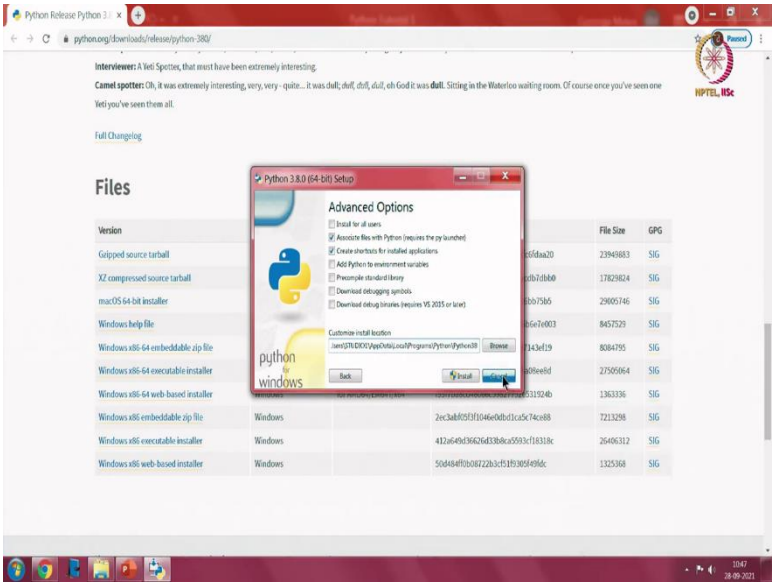


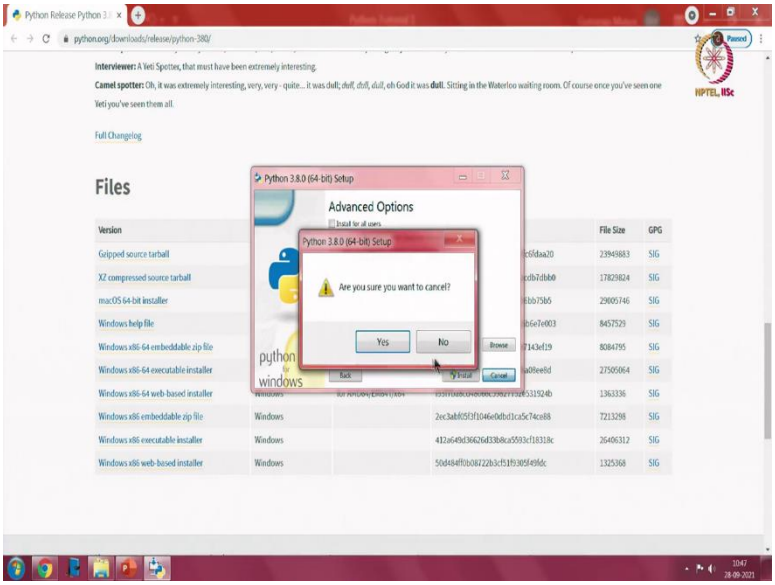

Python Release Python 3.8
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Version
Operating System
Description
MD5 Sum
File Size
GPG
Gzipped source tarball
Source release
e18a9d1a0a6d858b9787e03fc6fdaa20
23949883
SIG
XZ compressed source tarball
Source release
dbac8df9d8b9edc678d0f4cacdb7dbb0
17829824
SIG
macOS 64-bit installer
macOS
for OS X 10.9 and later
f3f3ae9f416170c6355cab7256bb75b6
29005746
SIG
Windows help file
Windows
1c333598210330db3353c8e5b6e7e003
8457529
SIG
Windows x86-64 embeddable zip file
Windows
for AMD64/EM64T/x64
99cca948512b53fb16508478714ef19
8084795
SIG
Windows x86-64 executable installer
Windows
for AMD64/EM64T/x64
29ea87f24c32f5e924b7d63f8a08ee8d
27505064
SIG
Windows x86-64 web-based installer
Windows
for AMD64/EM64T/x64
f83f7ba8cd480866c59827752e531924b
1363336
SIG
Windows x86 embeddable zip file
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
7213298
SIG
Windows x86 executable installer
Windows
412a649d36626d338c8ca5593cf18318c
26406312
SIG
Windows x86 web-based installer
Windows
50d484ff0b08722b3cf51f9305f49fdc
1325368
SIG
python-3.8.0-....exe
Show all
NPTEL IISc


Python Release Python 3.8
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Python 3.8.0 (64-bit) Setup
Install Python 3.8.0 (64-bit)
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
Install Now
C:\Users\SHUKRIT\AppData\Local\Programs\Python\Python38
Includes IDLE, pip and documentation
Customize installation
Choose location and features
python
for
windows
Install launcher for all users (recommended)
Add Python 3.8 to PATH
Cancel
Version
File Size
GPG
Gzipped source tarball
23949883
SIG
XZ compressed source tarball
17829824
SIG
macOS 64-bit installer
29005746
SIG
Windows help file
8457529
SIG
Windows x86-64 embeddable zip file
8084795
SIG
Windows x86-64 executable installer
27505064
SIG
Windows x86-64 web-based installer
1363336
SIG
Windows x86 embeddable zip file
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
7213298
SIG
Windows x86 executable installer
Windows
412a649d36626d338c8ca5593cf18318c
26406312
SIG
Windows x86 web-based installer
Windows
50d484ff0b08722b3cf51f9305f49fdc
1325368
SIG
python-3.8.0-....exe
Show all
NPTEL IISc


Python Release Python 3.8
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Python 3.8.0 (64-bit) Setup
Optional Features
Documentation
Installs the Python documentation file.
pip
Installs pip, which can download and install other Python packages.
tcl/tk and IDLE
Installs tkinter and the IDLE development environment.
Python test suite
Installs the standard library test suite.
py launcher
for all users (requires elevation)
Installs the global 'py' launcher to make it easier to start Python.
python
for
windows
Back
Next
Cancel
Version
File Size
GPG
Gzipped source tarball
23949883
SIG
XZ compressed source tarball
17829824
SIG
macOS 64-bit installer
29005746
SIG
Windows help file
8457529
SIG
Windows x86-64 embeddable zip file
8084795
SIG
Windows x86-64 executable installer
27505064
SIG
Windows x86-64 web-based installer
1363336
SIG
Windows x86 embeddable zip file
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
7213298
SIG
Windows x86 executable installer
Windows
412a649d36626d338c8ca5593cf18318c
26406312
SIG
Windows x86 web-based installer
Windows
50d484ff0b08722b3cf51f9305f49fdc
1325368
SIG
NPTEL IISc

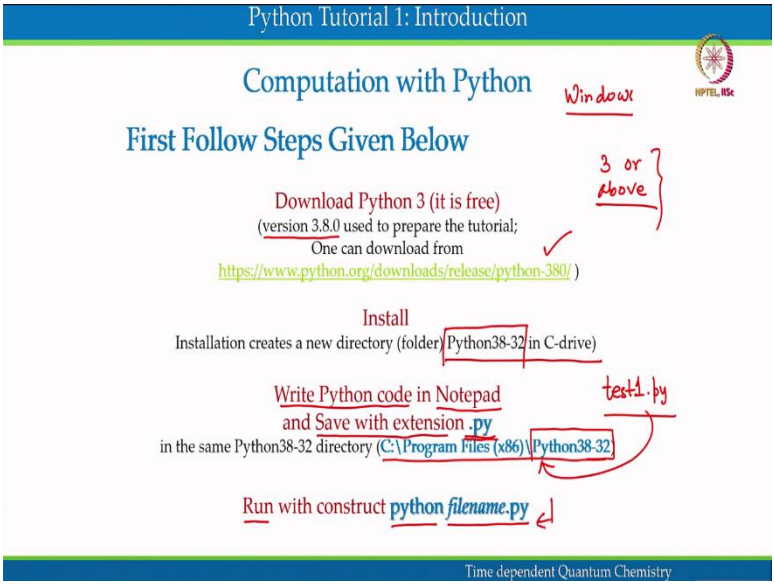

Python Release Python 3.8.0
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Python 3.8.0 (64-bit) Setup
Advanced Options
Install for all users
Associate files with Python (requires the py launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2015 or later)
Customize install location
Browse
Back
Install
Cancel
python for windows
Version
File Size
GPG
Gzipped source tarball
XZ compressed source tarball
macOS 64-bit installer
Windows help file
Windows x86-64 embeddable zip file
Windows x86-64 executable installer
Windows x86-64 web-based installer
Windows x86 embeddable zip file
Windows x86 executable installer
Windows x86 web-based installer
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
412a649d36626d338c8ca5593cf18318c
50d484ff0b08722b3cf51f9305f49fdc
SIG
Kaspersky Internet Security
Protection
Using Google Chrome? Activate the Kaspersky Protection extension to increase your level of protection while online.
Details
Don't ask again
Passed
NPTEL IISc
1047
28-09-2021


Python Release Python 3.8.0
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Python 3.8.0 (64-bit) Setup
Advanced Options
Install for all users
Associate files with Python (requires the py launcher)
Create shortcuts for installed applications
Add Python to environment variables
Precompile standard library
Download debugging symbols
Download debug binaries (requires VS 2015 or later)
Customize install location
Browse
Back
Install
Cancel
python for windows
Version
File Size
GPG
Gzipped source tarball
XZ compressed source tarball
macOS 64-bit installer
Windows help file
Windows x86-64 embeddable zip file
Windows x86-64 executable installer
Windows x86-64 web-based installer
Windows x86 embeddable zip file
Windows x86 executable installer
Windows x86 web-based installer
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
412a649d36626d338c8ca5593cf18318c
50d484ff0b08722b3cf51f9305f49fdc
SIG
Passed
NPTEL IISc
1047
28-09-2021


Python Release Python 3.8.0
python.org/downloads/release/python-380/
Interviewer: A Yeti Spotter, that must have been extremely interesting.
Camel spotter: Oh, it was extremely interesting, very, very - quite... it was dull; dull, dull, dull, oh God it was dull. Sitting in the Waterloo waiting room. Of course once you've seen one Yeti you've seen them all.
Full Changelog
Files
Python 3.8.0 (64-bit) Setup
Advanced Options
Python 3.8.0 (64-bit) Setup
Are you sure you want to cancel?
Yes
No
Browse
Back
Install
Cancel
python for windows
Version
File Size
GPG
Gzipped source tarball
XZ compressed source tarball
macOS 64-bit installer
Windows help file
Windows x86-64 embeddable zip file
Windows x86-64 executable installer
Windows x86-64 web-based installer
Windows x86 embeddable zip file
Windows x86 executable installer
Windows x86 web-based installer
Windows
2ec3abf05f3f1046e0dbd1ca5c74ce88
412a649d36626d338c8ca5593cf18318c
50d484ff0b08722b3cf51f9305f49fdc
SIG
Passed
NPTEL IISc
1047
28-09-2021

And I have prepared all the tutorials notes using this version, but one can go for a higher version as well if you scroll down, then there is a file, executable file, you can install this file in your computer, when you try to install it will ask you where to install you can customise the installation and you can change the directory where you would like to install. So, these are the installation procedure one can follow.

(Refer Slide Time: 10:53)

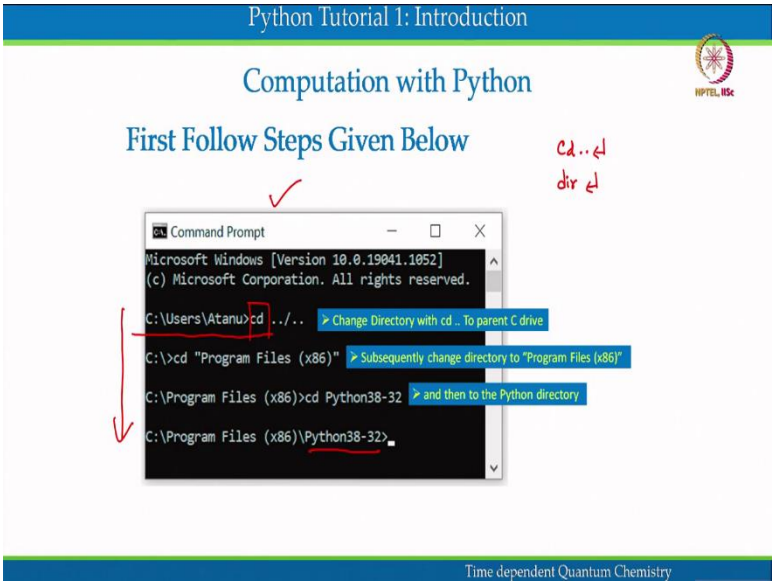


Once you have installed it, there will be Python folder which will be created in your desired drive. In my computer, I have in my laptop, I have installed it in C drive, that is why this Python file will be showing up in C drive in my computer. But based on your convenience, you can change the installation directory. Once you have installed it one can write Python code in different ways.

There are multiple ways one can use right now. But we will use very conventional and very old-fashioned way we will use a notepad to write down the Python code and we will save the file with dot py extension that is the Python programming extension .py so, if I name the file as test 1, then I will place .py as a file name So, this is going to be the file name after writing the code in Notepad and then we have to save the file in the same directory where the Python you have installed.

So, in my laptop where I will be performing all those programming this is where I have installed it and that is why after saving this file, we will save this file in this folder only. And finally, we will be able to run the programme with the construct Python filename.py so, that is the construct we will be using followed by pressing the Enter.

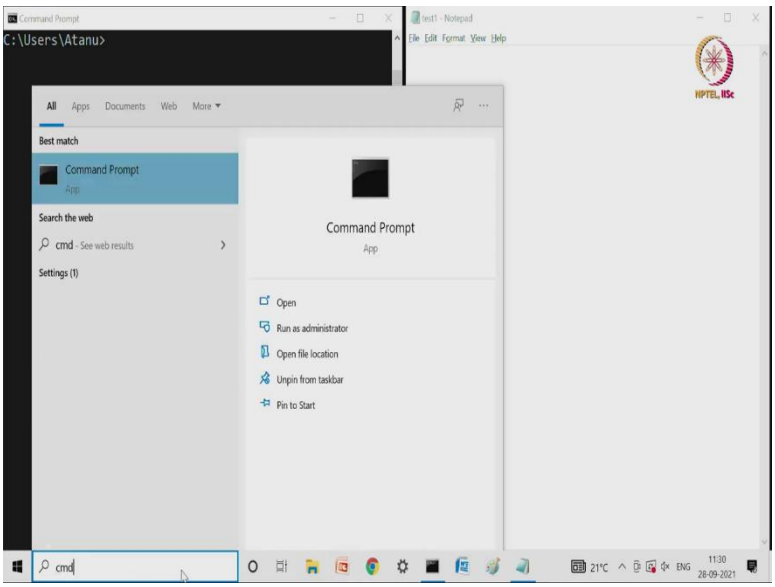
(Refer Slide Time: 12:57)

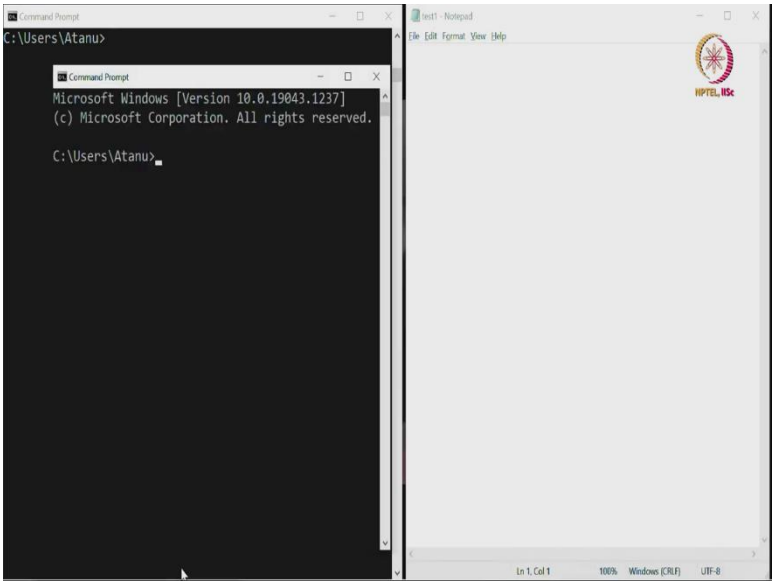


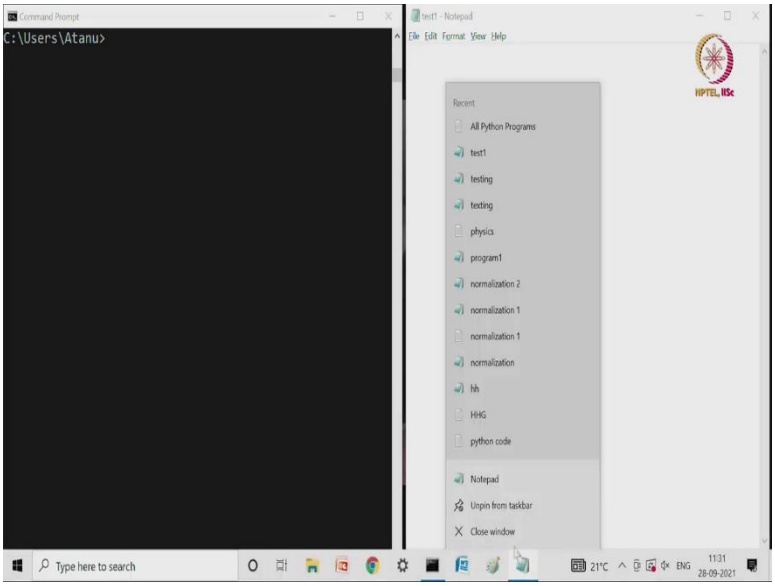
We will be running Python programming using command prompt as I told you before that there are multiple ways you can save file, execute Python files, Python programming, but we will be using very conventional way and later once we understand everything, we will introduce different techniques to make it easier.

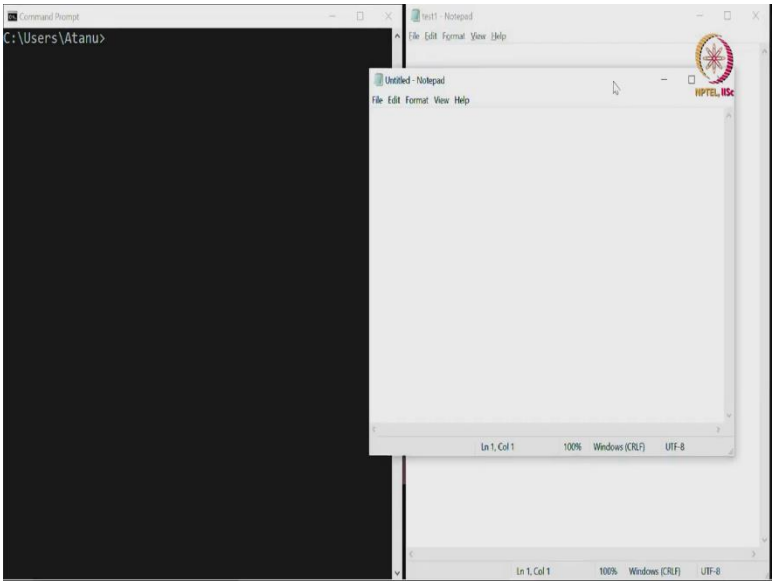
But right now, we will follow the conventional technique, notepad to write the programme and to run the programme to execute the programme we will use command prompt. Command prompt can be launched using your computer I will move to laptop right now.

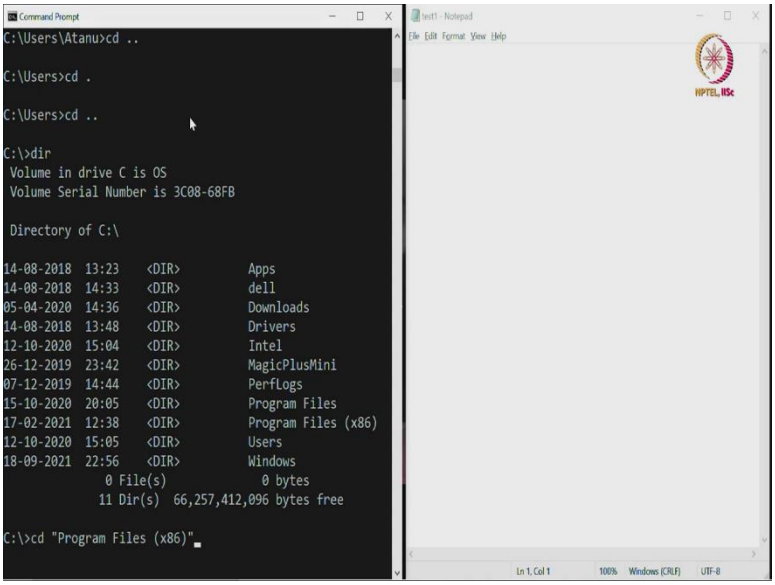
(Refer Slide Time: 13:56)

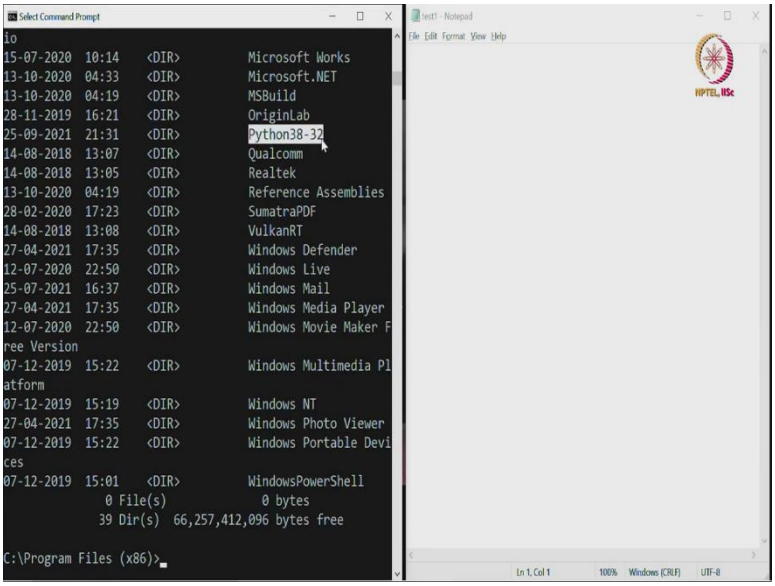


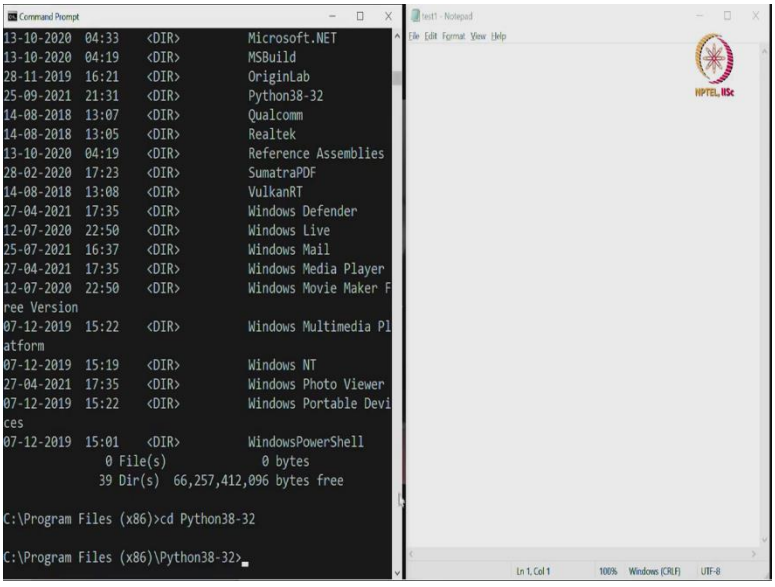

Command Prompt
C:\Users\Atanu>
Command Prompt
Microsoft Windows [Version 10.0.19043.1237]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Atanu>
test1 - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8


Command Prompt
C:\Users\Atanu>
test1 - Notepad
File Edit Format View Help
Recent
All Python Programs
test1
testing
texting
physics
program1
normalization 2
normalization 1
normalization 1
normalization
hh
HHG
python code
Notepad
Unpin from taskbar
Close window
Type here to search
21°C
ENG
11:31
28-09-2021


Command Prompt
C:\Users\Atanu>
test1 - Notepad
File Edit Format View Help
Untitled - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8


Command Prompt
C:\Users\Atanu>cd ..
C:\Users>cd .
C:\Users>cd ..
C:\>dir
Volume in drive C is OS
Volume Serial Number is 3C08-68FB
Directory of C:\
14-08-2018 13:23 <DIR> Apps
14-08-2018 14:33 <DIR> dell
05-04-2020 14:36 <DIR> Downloads
14-08-2018 13:48 <DIR> Drivers
12-10-2020 15:04 <DIR> Intel
26-12-2019 23:42 <DIR> MagicPlusMini
07-12-2019 14:44 <DIR> PerfLogs
15-10-2020 20:05 <DIR> Program Files
17-02-2021 12:38 <DIR> Program Files (x86)
12-10-2020 15:05 <DIR> Users
18-09-2021 22:56 <DIR> Windows
0 File(s) 0 bytes
11 Dir(s) 66,257,412,096 bytes free
C:\>cd "Program Files (x86)"
test1 - Notepad
File Edit Format View Help
NPTEL IISc
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8


Select Command Prompt
io
15-07-2020 10:14 <DIR> Microsoft Works
13-10-2020 04:33 <DIR> Microsoft.NET
13-10-2020 04:19 <DIR> MSBuild
28-11-2019 16:21 <DIR> OriginLab
25-09-2021 21:31 <DIR> Python38-32
14-08-2018 13:07 <DIR> Qualcomm
14-08-2018 13:05 <DIR> Realtek
13-10-2020 04:19 <DIR> Reference Assemblies
28-02-2020 17:23 <DIR> SumatraPDF
14-08-2018 13:08 <DIR> VulkanRT
27-04-2021 17:35 <DIR> Windows Defender
12-07-2020 22:50 <DIR> Windows Live
25-07-2021 16:37 <DIR> Windows Mail
27-04-2021 17:35 <DIR> Windows Media Player
12-07-2020 22:50 <DIR> Windows Movie Maker F
ree Version
07-12-2019 15:22 <DIR> Windows Multimedia Pl
atform
07-12-2019 15:19 <DIR> Windows NT
27-04-2021 17:35 <DIR> Windows Photo Viewer
07-12-2019 15:22 <DIR> Windows Portable Devi
ces
07-12-2019 15:01 <DIR> WindowsPowerShell
0 File(s) 0 bytes
39 Dir(s) 66,257,412,096 bytes free
C:\Program Files (x86)>
test1 - Notepad
File Edit Format View Help
NPTEL IISc
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8


Command Prompt
13-10-2020 04:33 <DIR> Microsoft.NET
13-10-2020 04:19 <DIR> MSBuild
28-11-2019 16:21 <DIR> OriginLab
25-09-2021 21:31 <DIR> Python38-32
14-08-2018 13:07 <DIR> Qualcomm
14-08-2018 13:05 <DIR> Realtek
13-10-2020 04:19 <DIR> Reference Assemblies
28-02-2020 17:23 <DIR> SumatraPDF
14-08-2018 13:08 <DIR> VulkanRT
27-04-2021 17:35 <DIR> Windows Defender
12-07-2020 22:50 <DIR> Windows Live
25-07-2021 16:37 <DIR> Windows Mail
27-04-2021 17:35 <DIR> Windows Media Player
12-07-2020 22:50 <DIR> Windows Movie Maker F
ree Version
07-12-2019 15:22 <DIR> Windows Multimedia Pl
atform
07-12-2019 15:19 <DIR> Windows NT
27-04-2021 17:35 <DIR> Windows Photo Viewer
07-12-2019 15:22 <DIR> Windows Portable Devi
ces
07-12-2019 15:01 <DIR> WindowsPowerShell
0 File(s) 0 bytes
39 Dir(s) 66,257,412,096 bytes free
C:\Program Files (x86)>cd Python38-32
C:\Program Files (x86)\Python38-32>
test1 - Notepad
File Edit Format View Help
NPTEL IISc
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

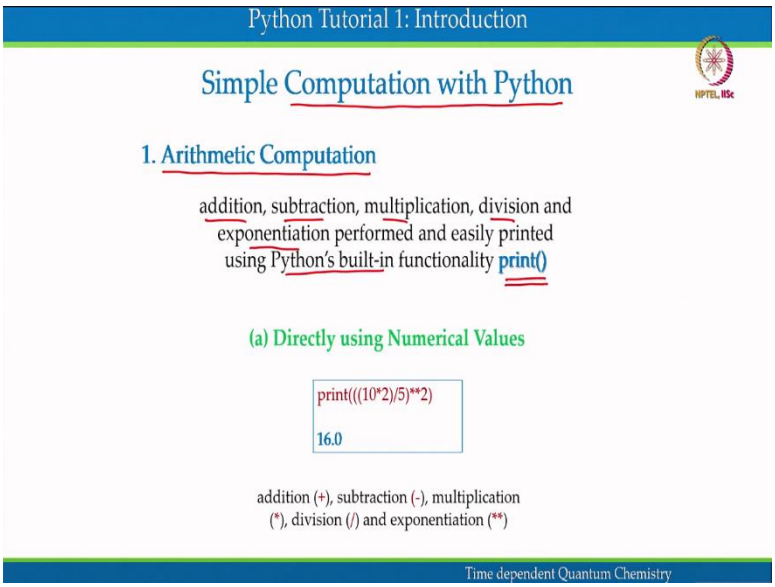
So, I have already created, command, launched the command prompt and Notepad and I kept it side by side for my convenience. But one can always type CMD enter to launch the command prompt and from notepad also one can launch the new notepad for writing the programme. So, it is already there. Once you launch the command prompt as I have moved to this slide right now, let us look at this slide.

Once you launch the command prompt, we would be in the home directory right now and that is the user directory. We have to move to the directory where the Python is installed. So, these are the procedures which has been given cd.. will enable you to go to the previous directory. So, this is what we are going to use to move the directory. And if I want to check the content of a directory, we can always use direct, dir return.

So, these are the two dos comments we need to move, I will show in my laptop right now. If I move cd.. I am moving now, I will check the directory I have to move to this programme files cd. Instead of writing everything I can press, I can write down few letters first, and then press Tab computer will automatically try to take the remaining letters.

So, that is exactly what I am doing and once I reach there, I will check the directory again, there is a Python directory given here and that is where I need to go. So, this is the directory where I will be saving the file as well as execute the file.

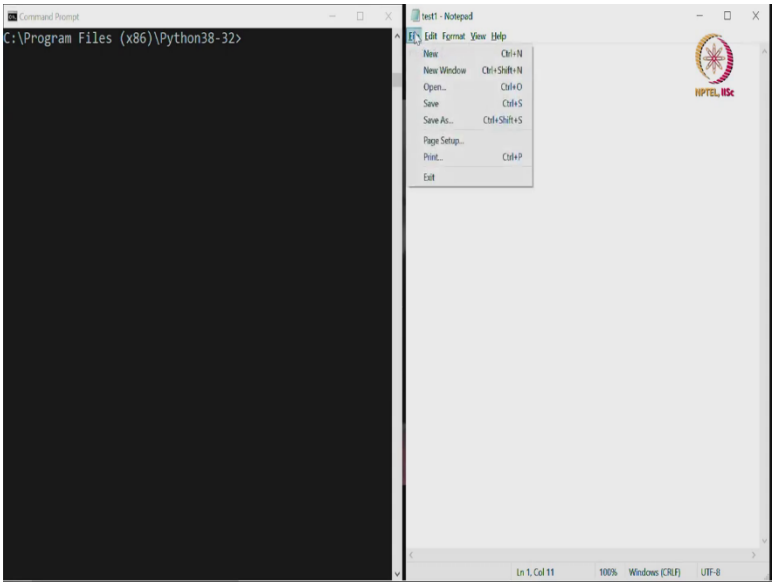
(Refer Slide Time: 16:43)

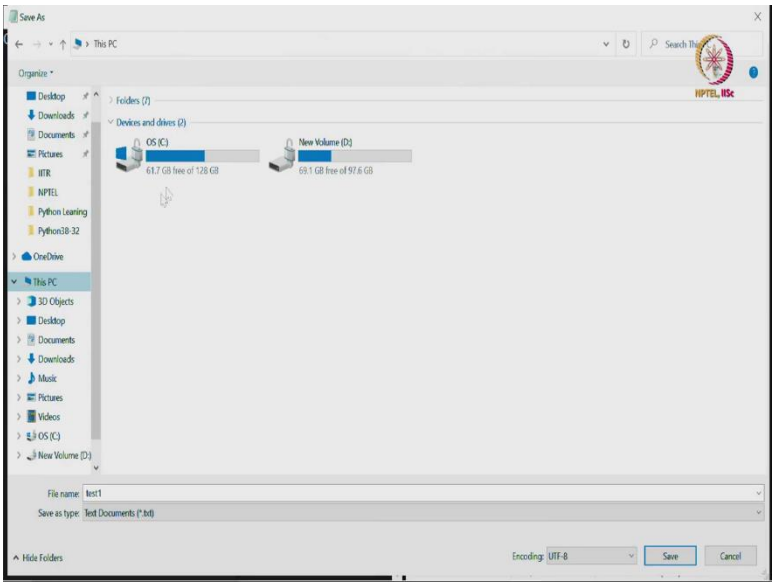


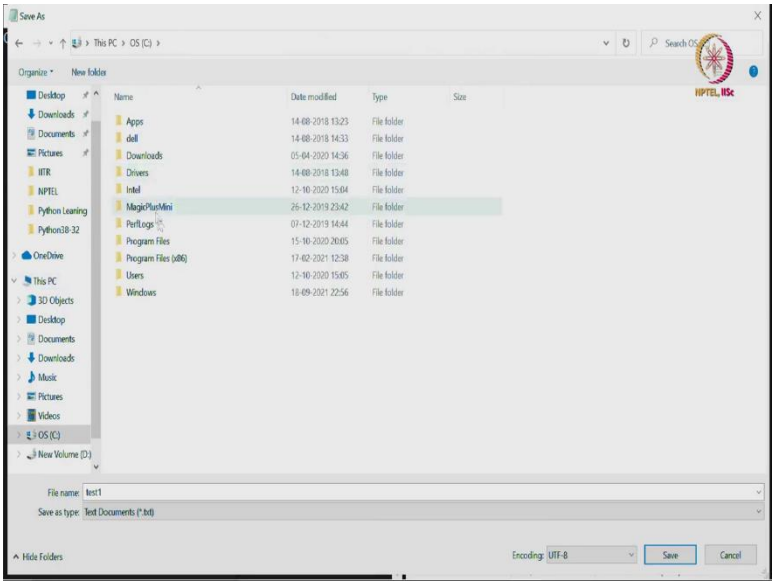
We will move to these slides right now and we will first begin the computation with Python demonstrating simple Arithmetic Computation. Arithmetic Computations such as addition,

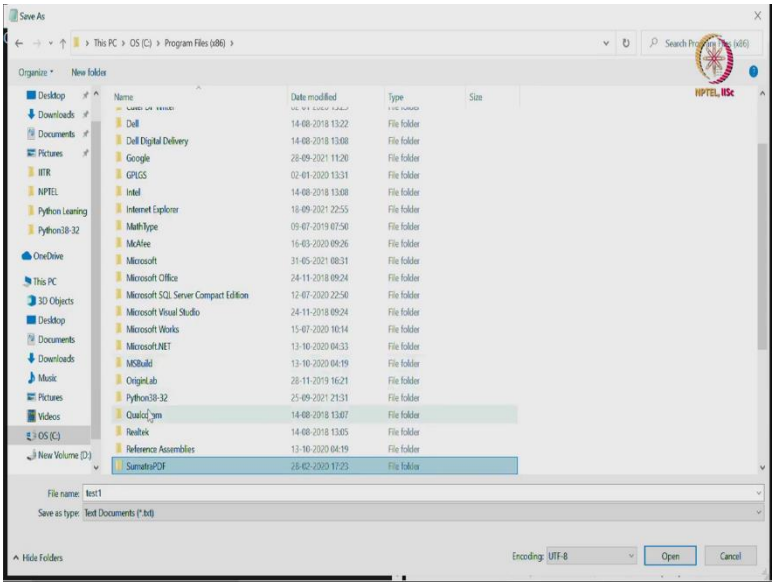
subtraction, multiplication, division, exponentiation, all these can be performed with Python's built-in functionality print bracket and I will show right now in my laptop.

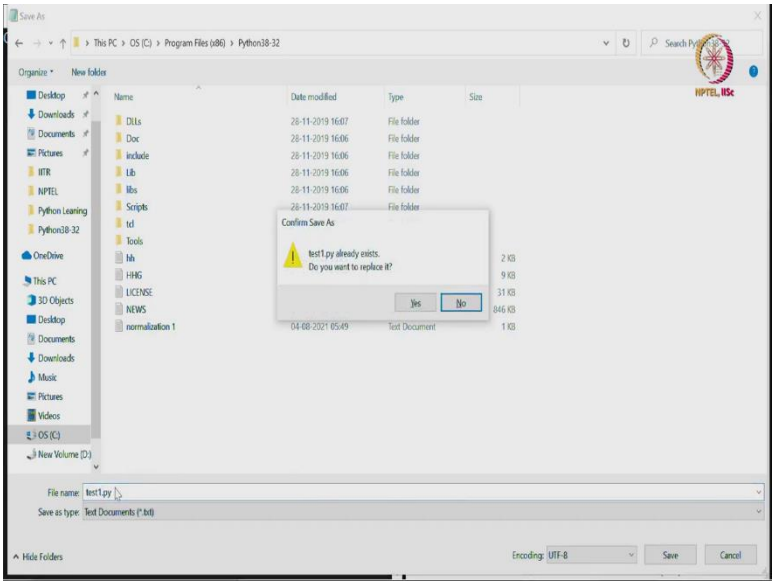
(Refer Slide Time: 17:18)

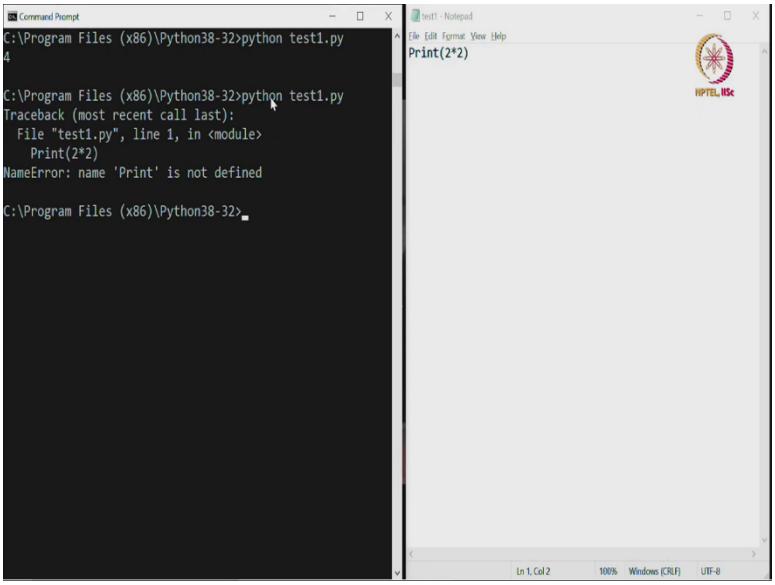


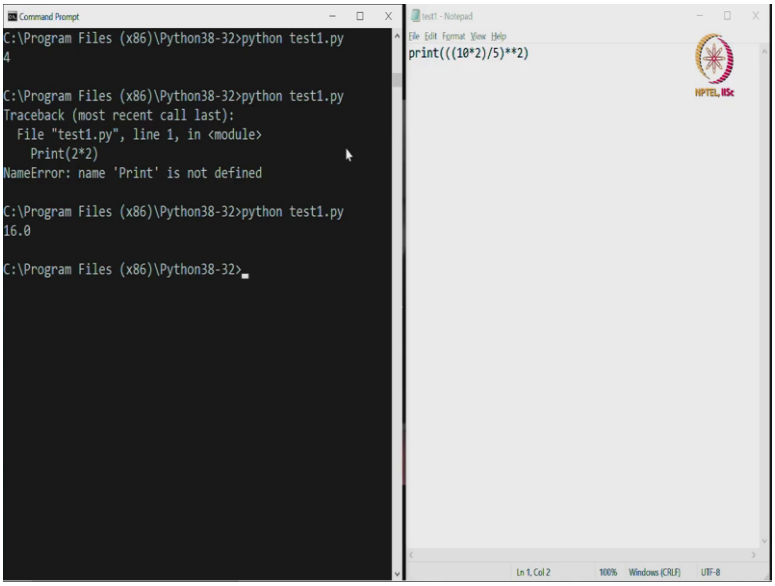


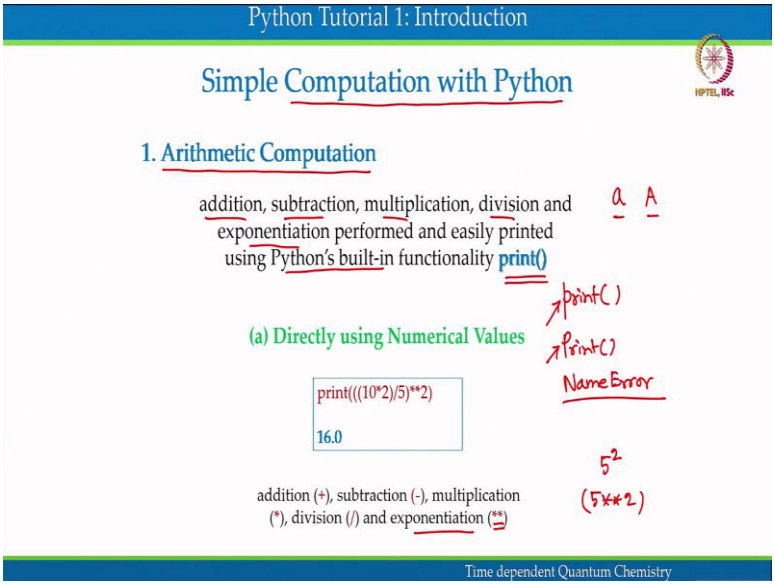

Save As
This PC › OS (C:) ›
Organize
New folder
Name
Date modified
Type
Size
Apps
dell
Downloads
Drivers
Intel
MagicPlusMini
PerfLogs
Program Files
Program Files (x86)
Users
Windows
File name: test1
Save as type: Text Documents (*.txt)
Hide Folders
Encoding: UTF-8
Save
Cancel


Save As
This PC › OS (C:) › Program Files (x86) ›
Organize
New folder
Name
Date modified
Type
Size
Dell
Dell Digital Delivery
Google
GPLGS
Intel
Internet Explorer
MathType
McAfee
Microsoft
Microsoft Office
Microsoft SQL Server Compact Edition
Microsoft Visual Studio
Microsoft Works
Microsoft.NET
MSBuild
OriginLab
Python38-32
Realtek
Reference Assemblies
SumatraPDF
File name: test1
Save as type: Text Documents (*.txt)
Hide Folders
Encoding: UTF-8
Open
Cancel


Save As
This PC › OS (C:) › Program Files (x86) › Python38-32
Organize
New folder
Name
Date modified
Type
Size
DLLs
Doc
include
Lib
libs
Scripts
tcl
Tools
hh
HHG
LICENSE
NEWS
normalization 1
Confirm Save As
test1.py already exists.
Do you want to replace it?
Yes
No
File name: test1.py
Save as type: Text Documents (*.txt)
Hide Folders
Encoding: UTF-8
Save
Cancel


Command Prompt
C:\Program Files (x86)\Python38-32>python test1.py
4
C:\Program Files (x86)\Python38-32>python test1.py
Traceback (most recent call last):
File "test1.py", line 1, in <module>
Print(2*2)
NameError: name 'Print' is not defined
C:\Program Files (x86)\Python38-32>
test1 - Notepad
File Edit Format View Help
Print(2*2)
NPTEL IISc
Ln 1, Col 2
100%
Windows (CRLF)
UTF-8


Command Prompt
C:\Program Files (x86)\Python38-32>python test1.py
4
C:\Program Files (x86)\Python38-32>python test1.py
Traceback (most recent call last):
File "test1.py", line 1, in <module>
Print(2*2)
NameError: name 'Print' is not defined
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>
test1 - Notepad
File Edit Format View Help
print(((10*2)/5)**2)
NPTEL IISc
Ln 1, Col 2
100%
Windows (CRLF)
UTF-8


Python Tutorial 1: Introduction
Simple Computation with Python
NPTEL IISc
1. Arithmetic Computation
addition, subtraction, multiplication, division and exponentiation performed and easily printed using Python's built-in functionality print()
a A
print( )
Print( )
NameError
(a) Directly using Numerical Values
print(((10*2)/5)**2)
16.0
5^2
(5**2)
addition (+), subtraction (-), multiplication (*), division (/) and exponentiation (**)
Time dependent Quantum Chemistry

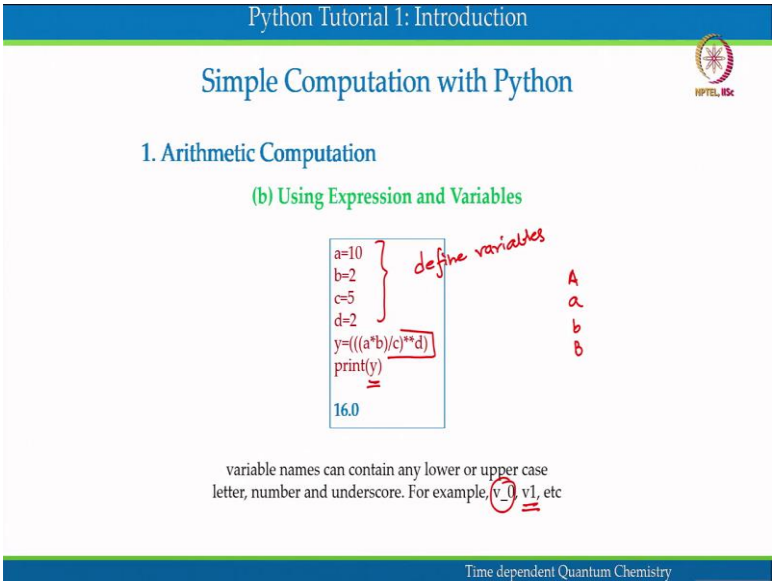
So, if I write print 2* 2, it is 2 multiplied by 2 and if I save the file in the same directory in the Python directory, I will give the name test1.py. So, this is saved in the same directory and I can now run this programme Python then test1.py I get back the value 2 multiplied by 2 equals 4. If I make p to be capital and re-run the programme, I can use Page Up and Page Down arrows to get my previous comment.

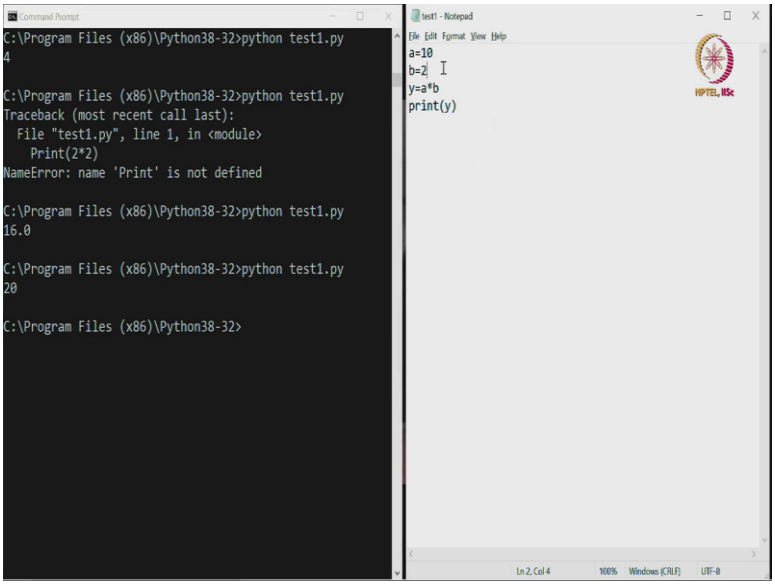
I see that if I use capital P for the print command, the print functionality I move to this slide. Print functionality instead of I am using print then I get an name error in the programming and the reason why I am getting the name error is because Python will distinguish small letter and capital letter and this is an important instruction to everybody. If you are copying Python programming, make sure that you are not changing any syntax.

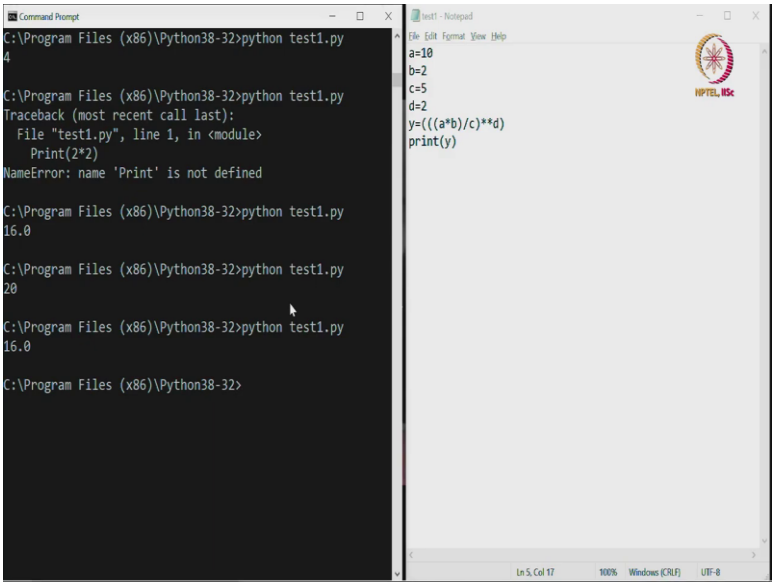
So, in Python programming a and A they are two different variables. They are two different syntax. We will move on and print what I have written here. 10 multiplied by 2. I am using number of parentheses to preserve the order of the operation. So, that the one can read easily and one can understand what kind of operation we have. So, if we perform this with a small letter P, then we get 16.

That is the value we have. So, one thing to note here, I have moved this to this slide, one thing to remember is this sign for each arithmetic computation double star is used for exponentiation it is so, 5 square is written as 5**2 in Python.

(Refer Slide Time: 21:14)





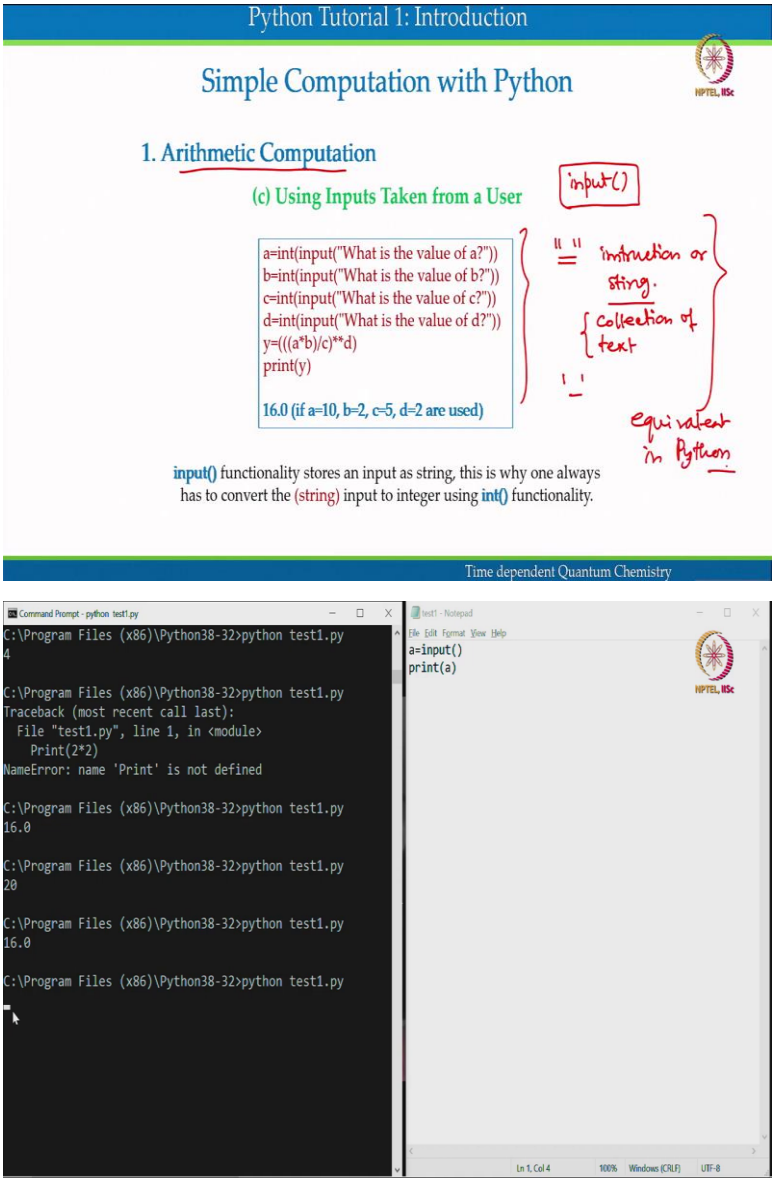


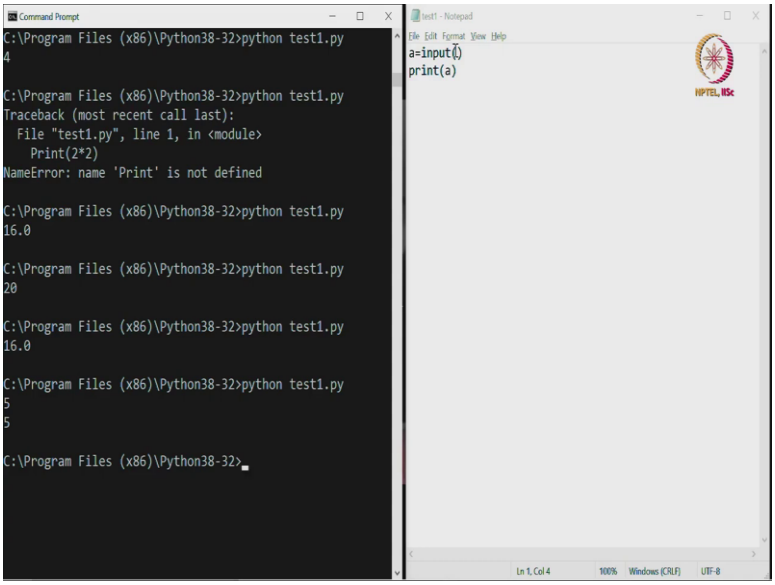
We will move on to the next kind of arithmetic computation in this we will first define variables and then we will use a mathematical expression and then we will print the output that is exactly what we are going to do. So, we will move to this laptop, a= 10, b = 2; I can write down print I can define another function is multiplied by b and then I can print the value of y.

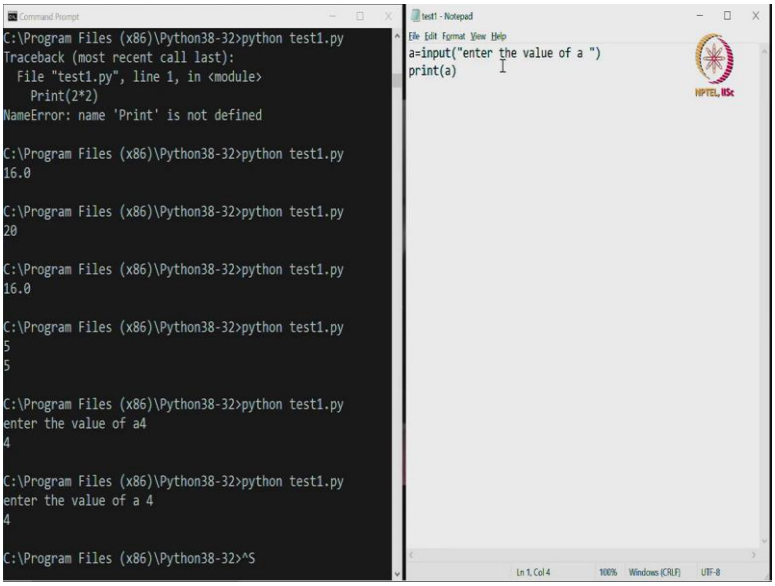
We get 20, 10 multiplied by 2 is 20. I will write down what we had c equals 5 and d equals 2 then we will define this function a * b/c, I am again using a number of parentheses to preserve the order anybody can read and check what kind of ordered we have for the operation we get 16.

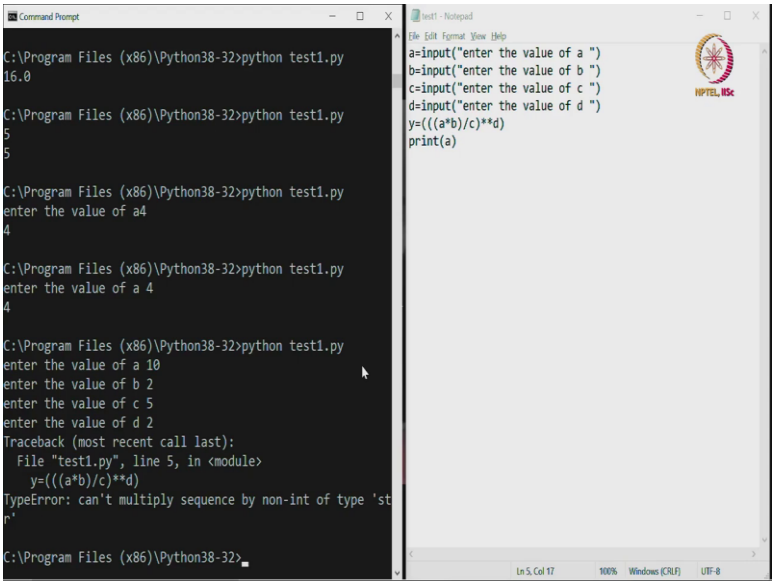
So, let us move to this slide right now, one can in general variable names can contain any lower or uppercase letter. So, I can give name a, b like this way. Anything is possible and also one can use underscore number to define the variables.

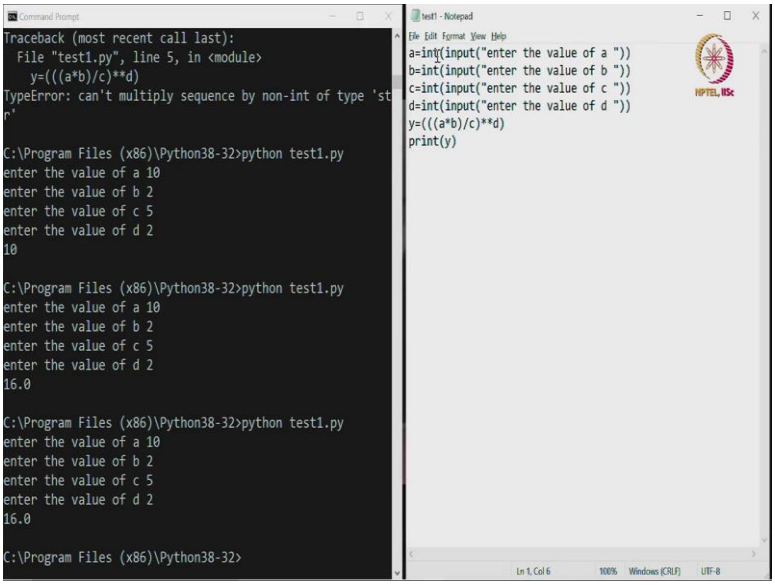
(Refer Slide Time: 23:48)




Command Prompt
C:\Program Files (x86)\Python38-32>python test1.py
4
C:\Program Files (x86)\Python38-32>python test1.py
Traceback (most recent call last):
File "test1.py", line 1, in <module>
Print(2*2)
NameError: name 'Print' is not defined
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>python test1.py
20
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>python test1.py
5
5
C:\Program Files (x86)\Python38-32>
test1 - Notepad
File Edit Format View Help
a=input()
print(a)
NPTEL, IISc
Ln 1, Col 4
100%
Windows (CRLF)
UTF-8


Command Prompt
C:\Program Files (x86)\Python38-32>python test1.py
Traceback (most recent call last):
File "test1.py", line 1, in <module>
Print(2*2)
NameError: name 'Print' is not defined
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>python test1.py
20
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>python test1.py
5
5
C:\Program Files (x86)\Python38-32>python test1.py
enter the value of a4
4
C:\Program Files (x86)\Python38-32>python test1.py
enter the value of a 4
4
C:\Program Files (x86)\Python38-32>^S
test1 - Notepad
File Edit Format View Help
a=input("enter the value of a ")
print(a)
NPTEL, IISc
Ln 1, Col 4
100%
Windows (CRLF)
UTF-8


Command Prompt
C:\Program Files (x86)\Python38-32>python test1.py
16.0
C:\Program Files (x86)\Python38-32>python test1.py
5
5
C:\Program Files (x86)\Python38-32>python test1.py
enter the value of a4
4
C:\Program Files (x86)\Python38-32>python test1.py
enter the value of a 4
4
C:\Program Files (x86)\Python38-32>python test1.py
enter the value of a 10
enter the value of b 2
enter the value of c 5
enter the value of d 2
Traceback (most recent call last):
File "test1.py", line 5, in <module>
y=(((a*b)/c)**d)
TypeError: can't multiply sequence by non-int of type 'st
r'
C:\Program Files (x86)\Python38-32>
test1 - Notepad
File Edit Format View Help
a=input("enter the value of a ")
b=input("enter the value of b ")
c=input("enter the value of c ")
d=input("enter the value of d ")
y=(((a*b)/c)**d)
print(a)
NPTEL, IISc
Ln 5, Col 17
100%
Windows (CRLF)
UTF-8



We will still continue arithmetic computation, but this time we will take the input from our user. So, let us check how to do that. And for that what we need the Python's inbuilt functionality. That is called input, parentheses. This is the inbuilt functionality. So, we will move to laptop. If we write a equals input, then print a. It suggests that the a variable will be given by user that will be taken as input and then it will print a.

So, let us see if we run the programme. It is, the cursor is waiting for an input. There is no instruction showing up. I will tell you how to include an instruction in this. But there is an instruction right now I have to give the value if I give the value, it will print the value. And that is exactly what is going on right now. So, I will show you how to give instruction. Instruction will be given with double quote or single quote; both are equivalent in Python.

So, I will move to this slide. So, double quote, I can use double code to write a string, that is the instruction or some string, that is the collection of text we can use double quote, or single quote, both are equivalent in Python. So, we will move to slide I will use double quote right now and I write enter the value of a then it is asking me enter the value of a, I enter it return, it will show up the value.

That is exactly what I am asking to do. I can give a space here and that space will show up. So, that and its printing. So, we have learned how to take input from user. So, now we will go up, go ahead and take the input all the inputs for the arithmetic computation, which I am doing right now, b equals I will just copy and keep the string. This is for b, enter the value of b, c equals this is for c and d equals this is for d.

All these inputs will be taken from user and then I am going to define that function a, multiplied by b divided by c exponentiation d. That is my string. So, I will be now running the programme, is asking to enter the value of a I will use the same value 10, which I used before 10, then 2, then 5, and then 2, I get an error. And this is something which we should remember. I will move to this slide.

Whenever I am using input, any value given through input it returns a string and I cannot use multiplication, division or arithmetic computation with the help with string I need an numerical value So, this string has to be converted to a numerical value before we perform any arithmetic computation. So, numerical value numerical value can be integer, for which I will use this int. I can use float, floating point to convert it.

So, these are the ways I can convert the numerical values. So, what I will do right now, I will move to slide and I will change. I will make int within bracket everything. I am specifying that this is going to be integer. If I do not want I if I want to convert it to string to float, I have to use float. So, this int, parenthesis built in functionality of Python will convert the string to integer now a, b, c, d are integer although the values will be taken by taken from an from a user, but it will be converted to integer so that I can perform this arithmetic computation.

So, I will give the same value 10, 2, 5, 2 and I get back. Okay, So, if print y, then what I get is if I run rerun the programme, I get the I use the same value c 5 and d 2, I get back 16. So, now it is doing the arithmetic computation without any problem. So, this is something which we have to remember that whenever you are taking an input from user by a keyboard, that input is given in the string format.

We have to convert it to integer or floating-point numerical value and then only we will be able to perform the arithmetic computation. We will stop here and we will continue this tutorial in the next session.