PYTHON

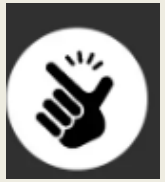
AGENDA

- What is Python?
- Features of Python
- Installation of Python
- Basics of Python
  - Data Types
  - Operators
  - Flow Control
  - Functions
  - File Handling

What is Python?

- Created by **Guido Van Rossum** in 1989
- Inspired by his favourite show's (Flying Circus) creator '**Monty Python**'
- **High level**, **Interpreted language** with **easy syntax** and **dynamic semantics**.

Features of Python



Simplicity

Open Source

Portability

Embeddable and Extensible

Interpreted

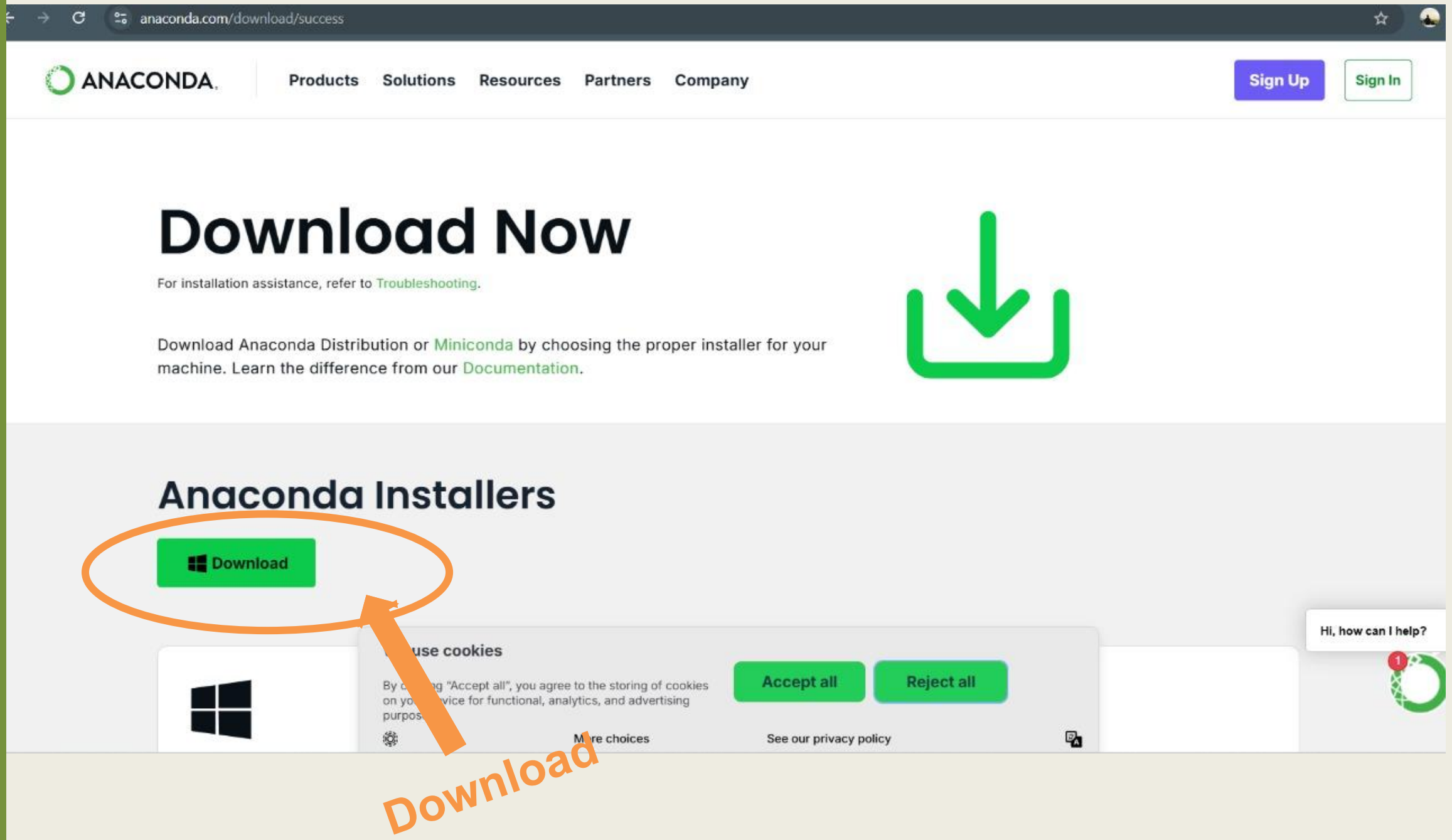
Huge Libraries

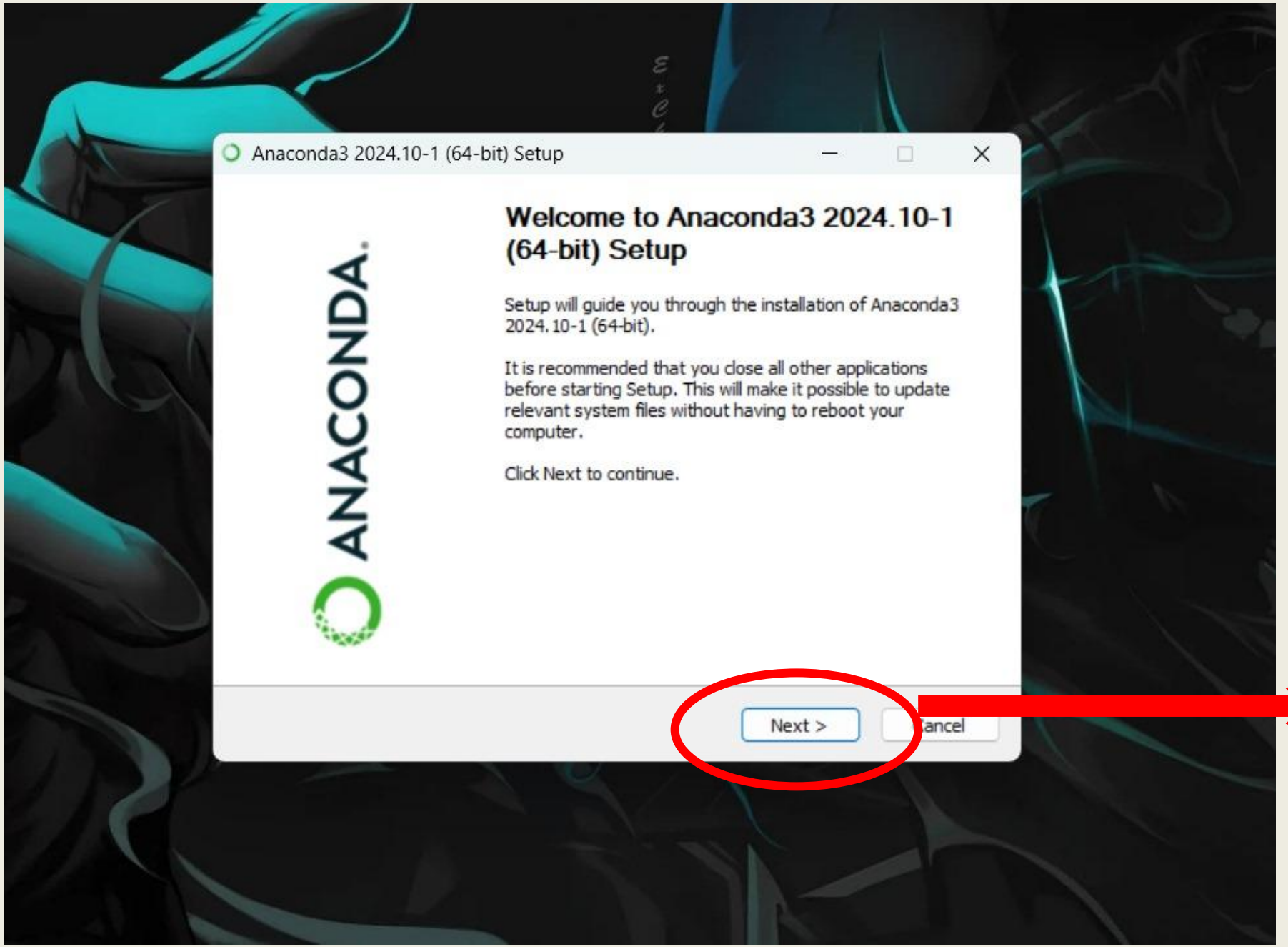
Object Orientation

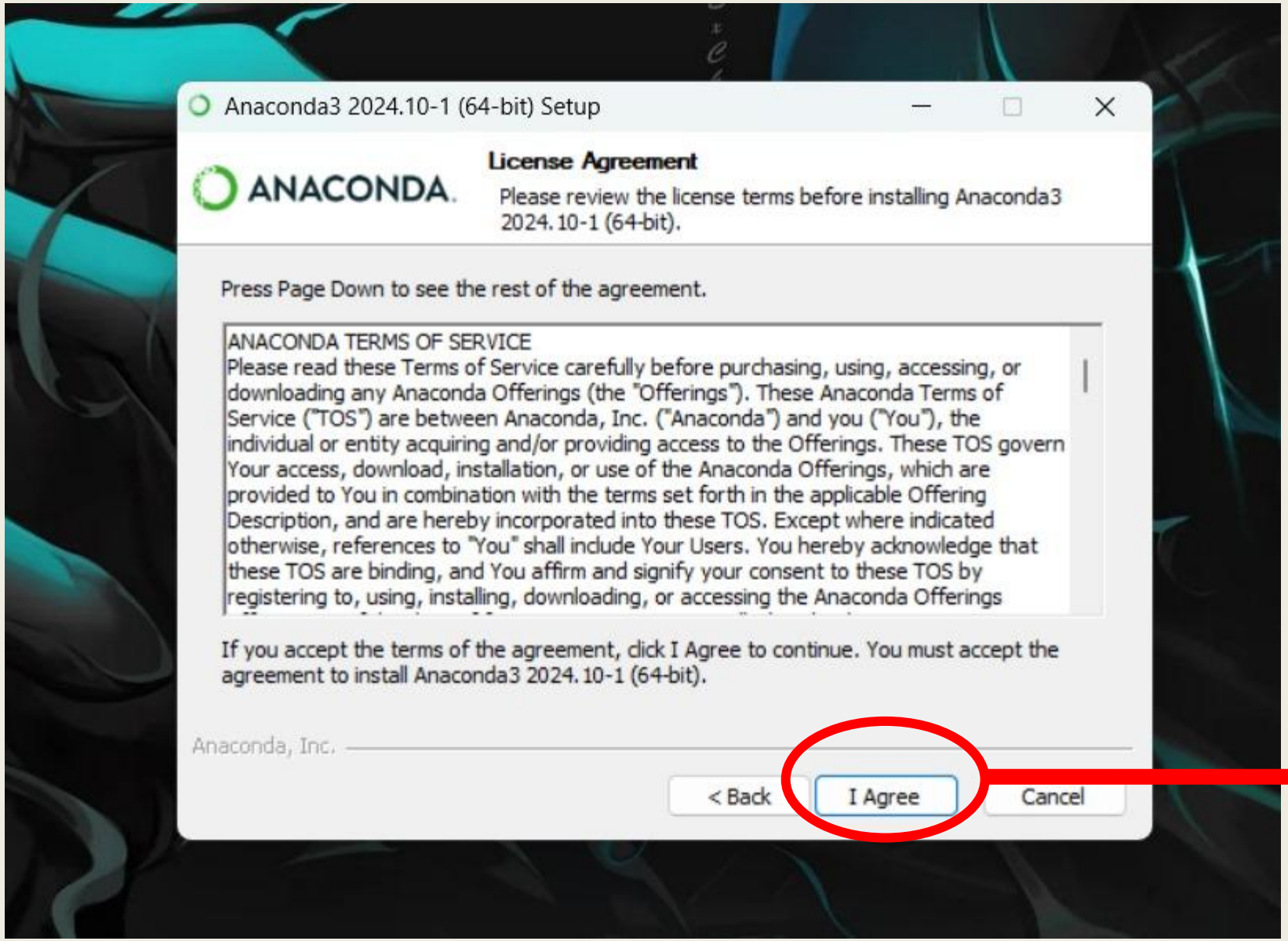
Installing Python

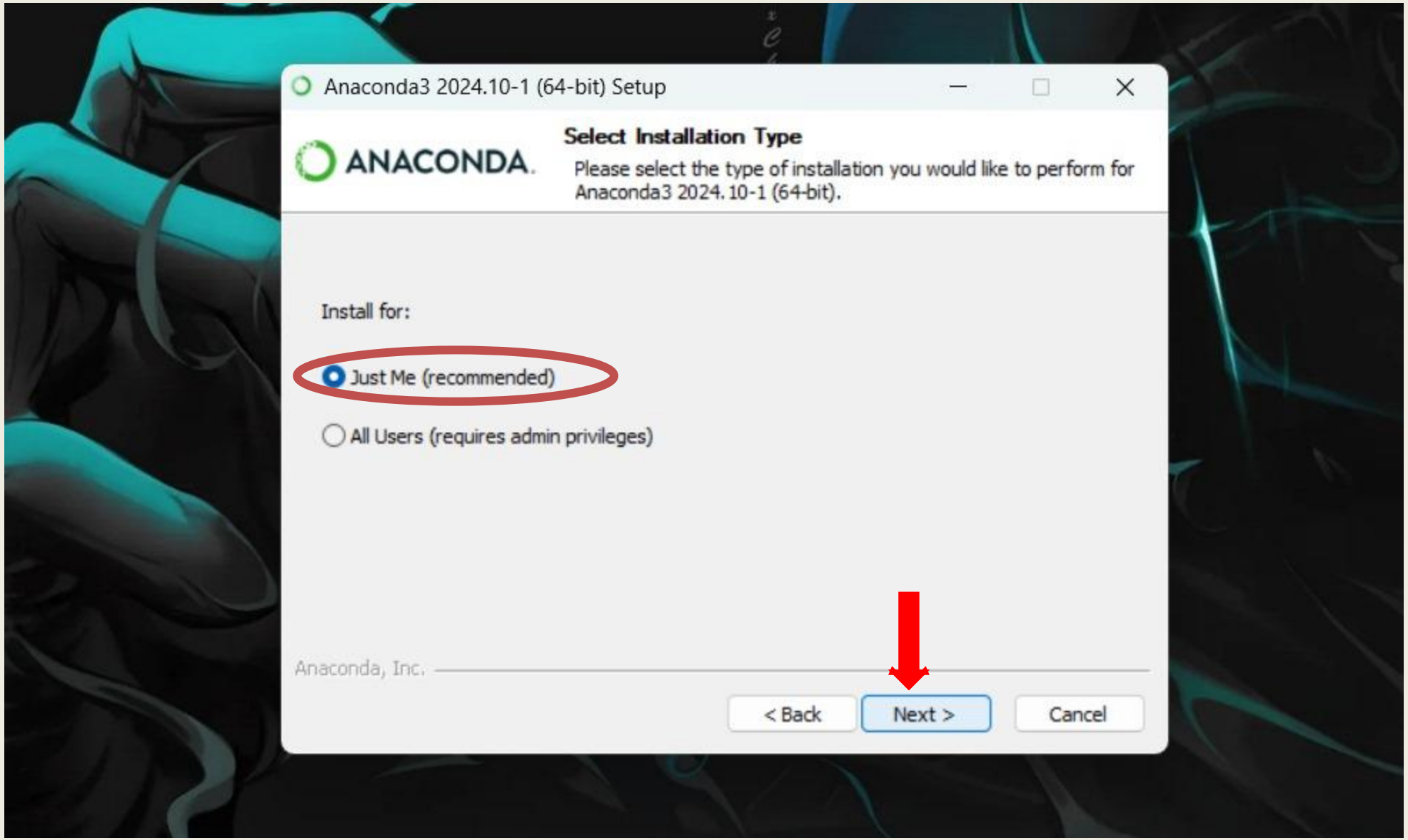
Anaconda official site:

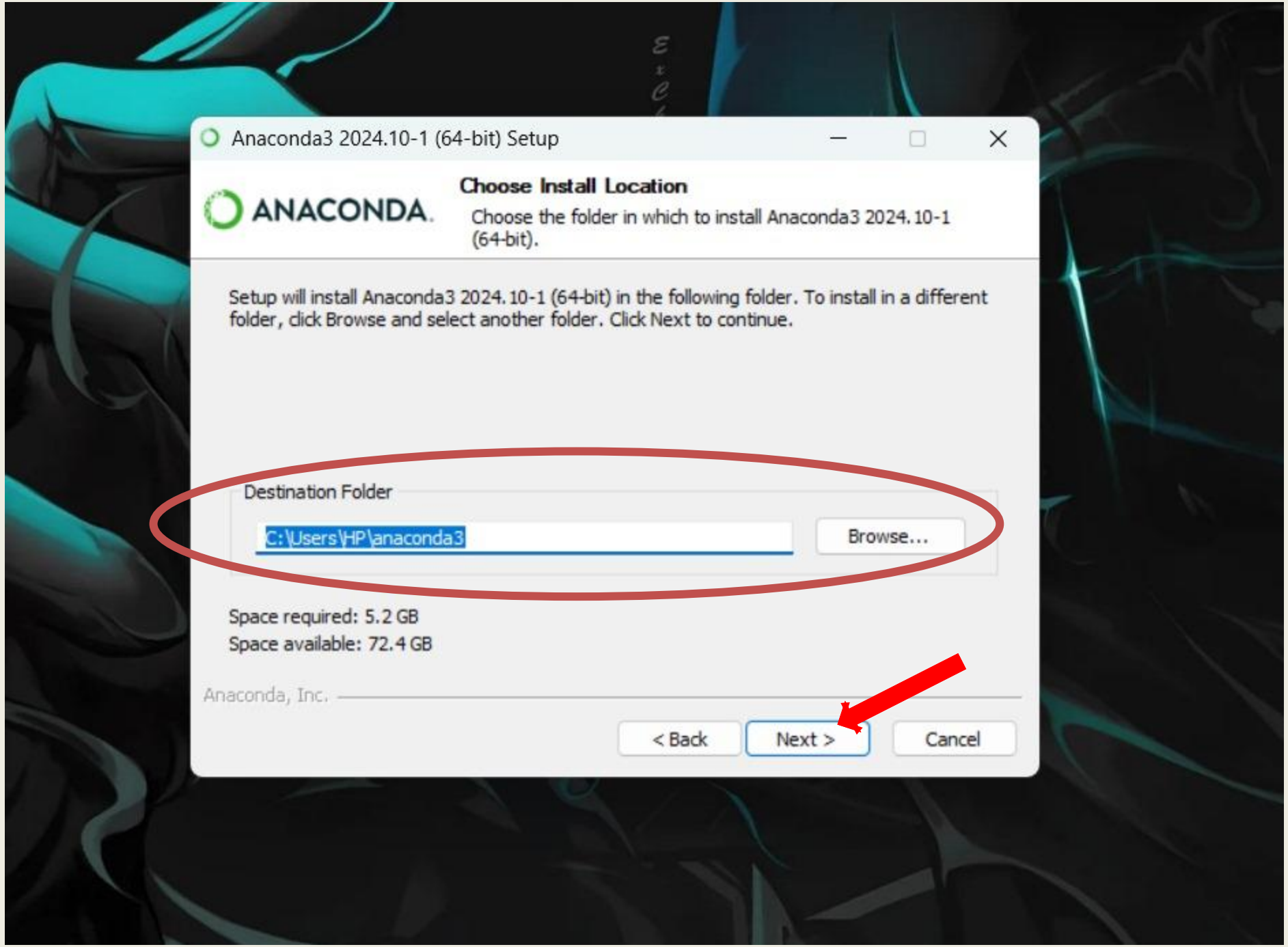
https://www.anaconda.com/download/success

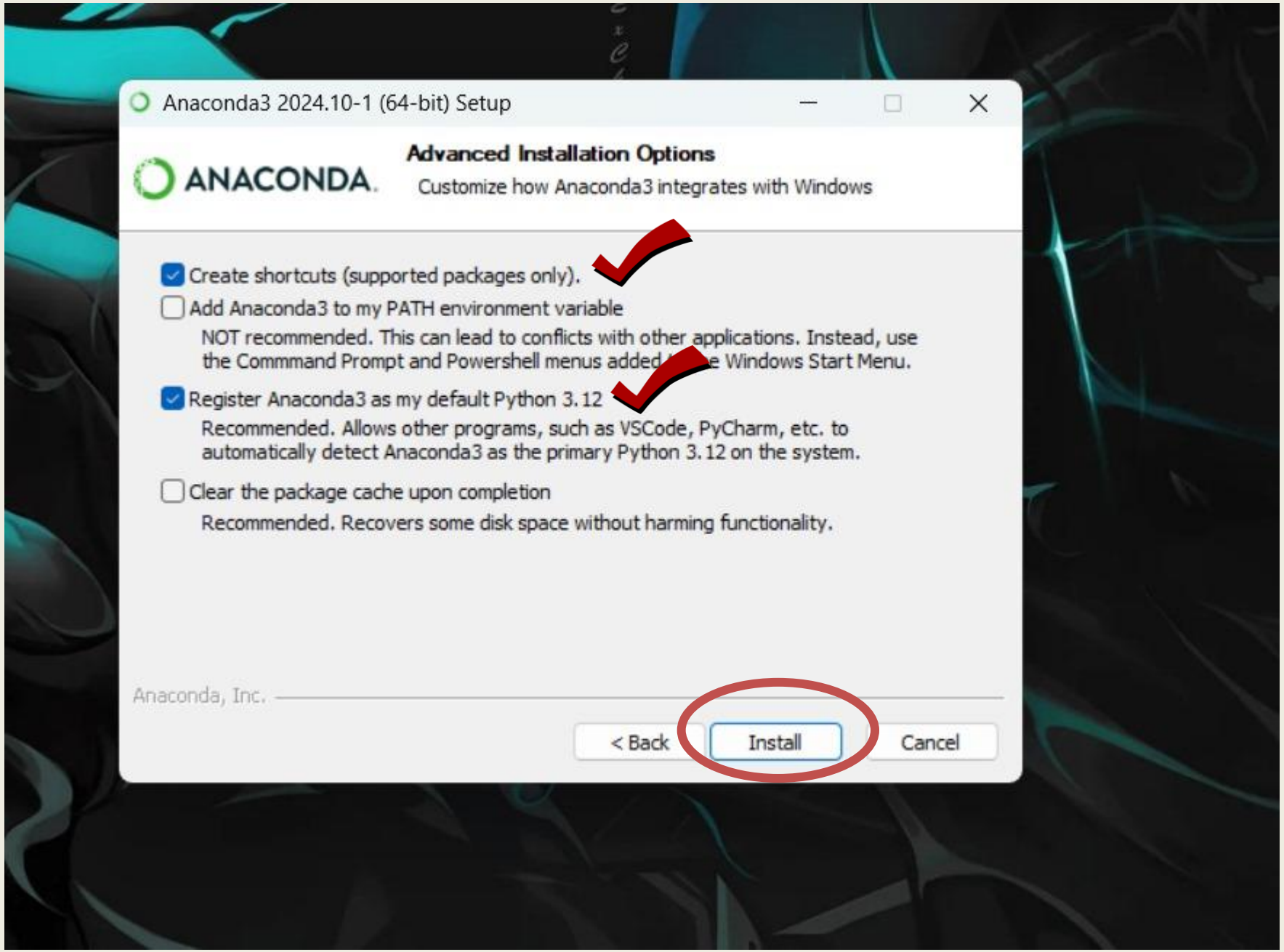

anaconda.com/download/success
ANACONDA.
Products
Solutions
Resources
Partners
Company
Sign Up
Sign In
Download Now
For installation assistance, refer to Troubleshooting.
Download Anaconda Distribution or Miniconda by choosing the proper installer for your machine. Learn the difference from our Documentation.
Anaconda Installers
Download
use cookies
Accept all
Reject all
See our privacy policy
Hi, how can I help?
Download

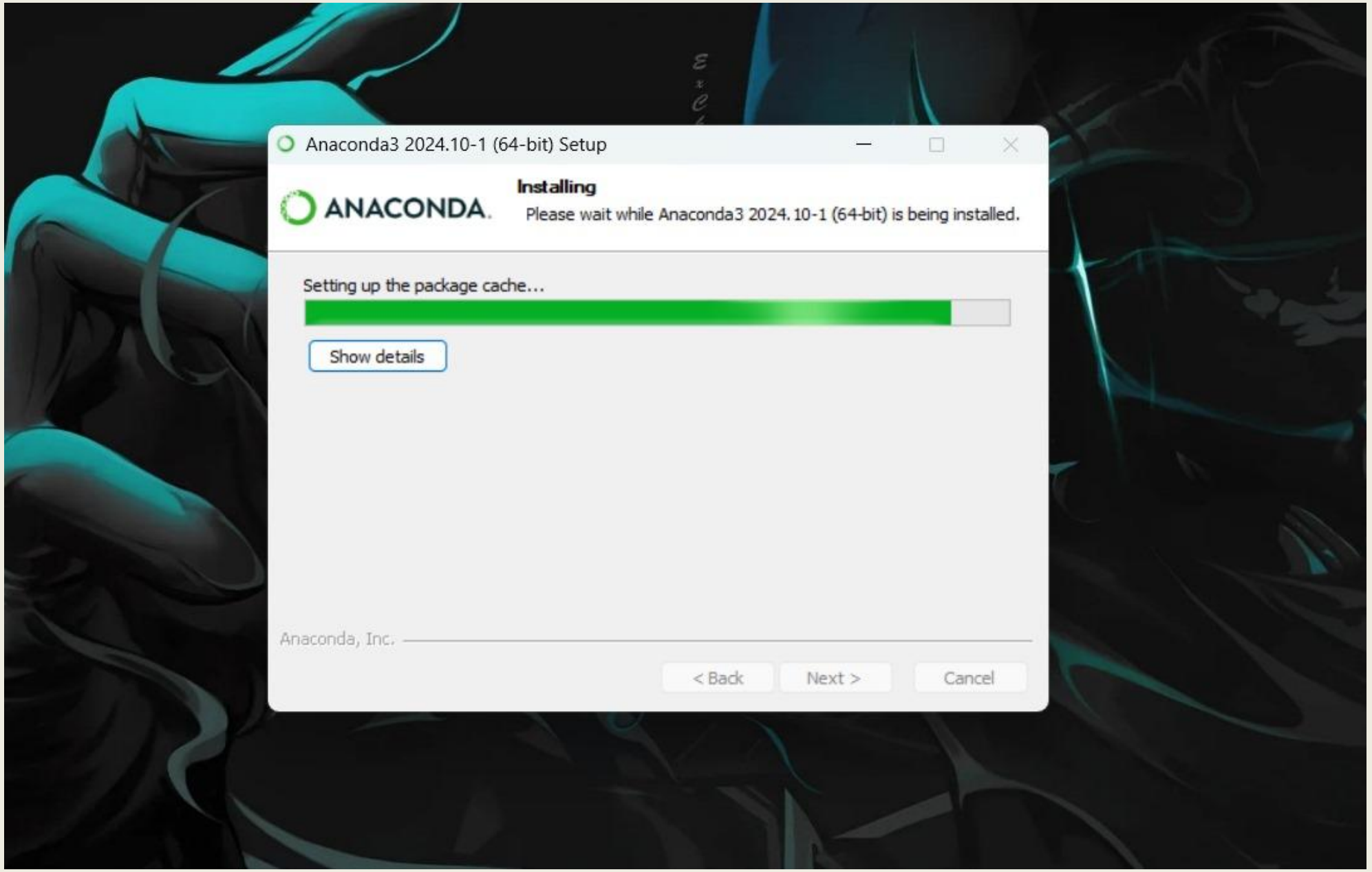

Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Welcome to Anaconda3 2024.10-1 (64-bit) Setup
Setup will guide you through the installation of Anaconda3 2024.10-1 (64-bit).
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel
Next

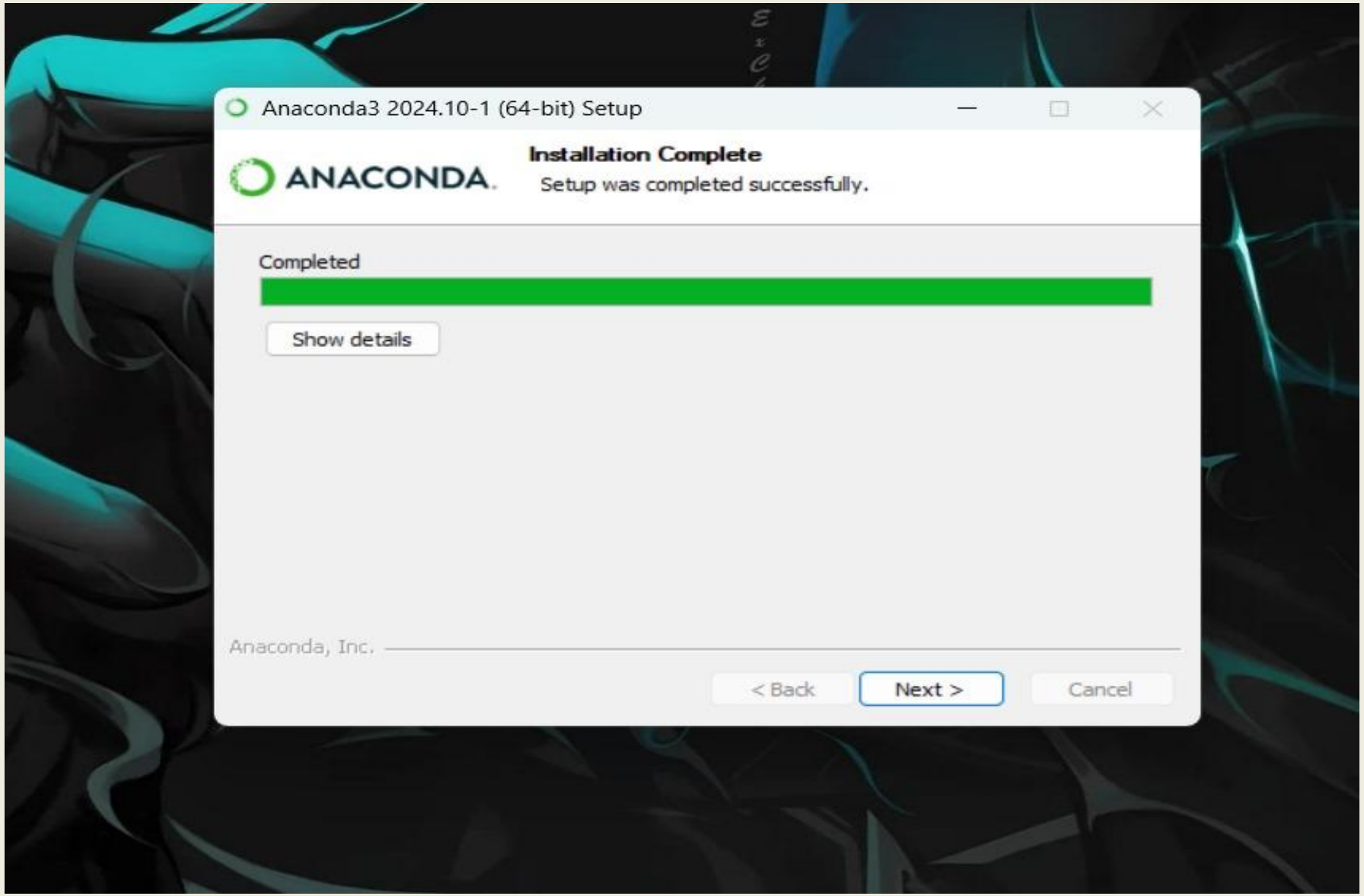

Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
License Agreement
Please review the license terms before installing Anaconda3 2024.10-1 (64-bit).
Press Page Down to see the rest of the agreement.
ANACONDA TERMS OF SERVICE
Please read these Terms of Service carefully before purchasing, using, accessing, or downloading any Anaconda Offerings (the "Offerings"). These Anaconda Terms of Service ("TOS") are between Anaconda, Inc. ("Anaconda") and you ("You"), the individual or entity acquiring and/or providing access to the Offerings. These TOS govern Your access, download, installation, or use of the Anaconda Offerings, which are provided to You in combination with the terms set forth in the applicable Offering Description, and are hereby incorporated into these TOS. Except where indicated otherwise, references to "You" shall include Your Users. You hereby acknowledge that these TOS are binding, and You affirm and signify your consent to these TOS by registering to, using, installing, downloading, or accessing the Anaconda Offerings
If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2024.10-1 (64-bit).
Anaconda, Inc.
< Back
I Agree
Cancel
I Agree


Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2024.10-1 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

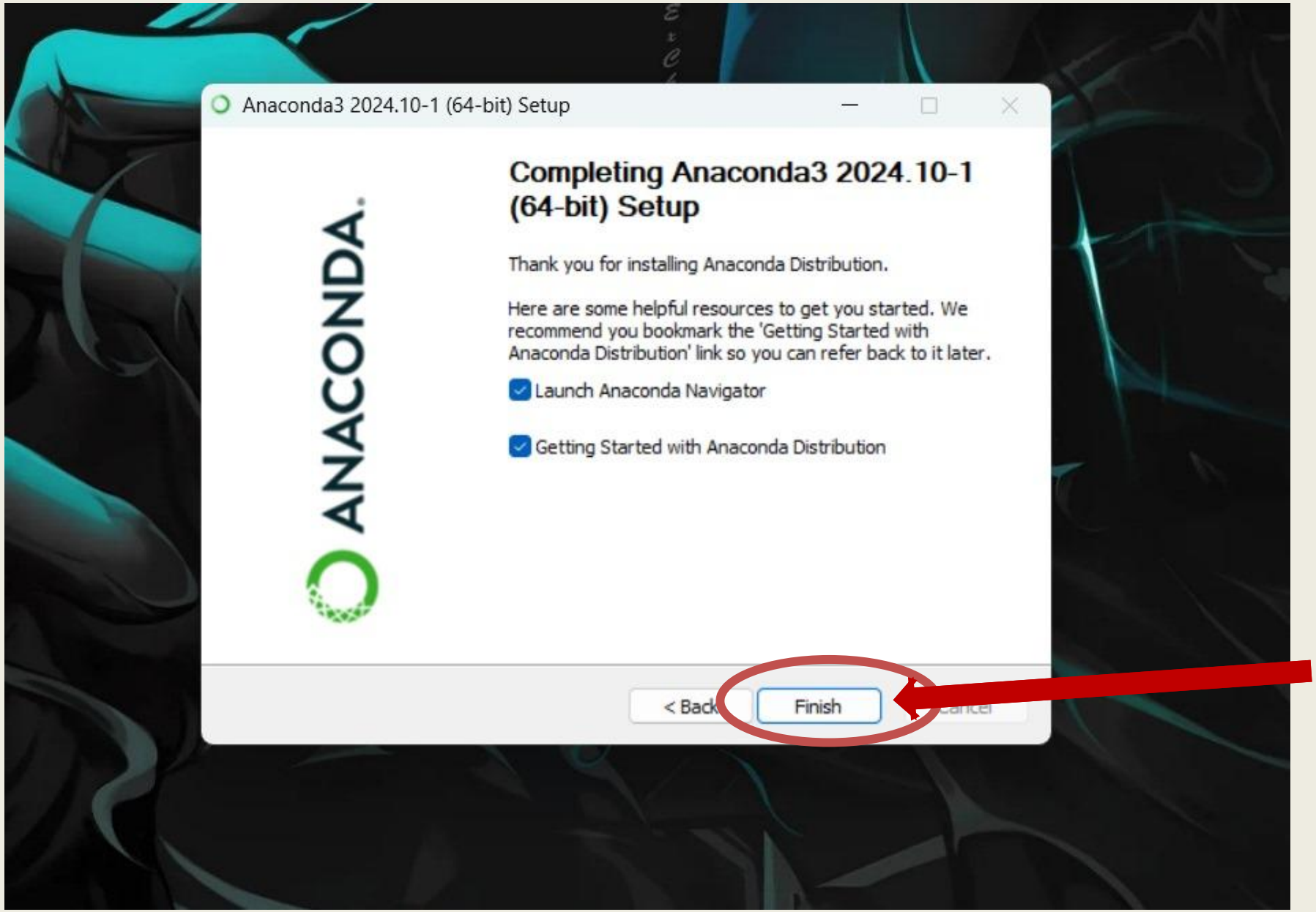

Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Choose Install Location
Choose the folder in which to install Anaconda3 2024.10-1 (64-bit).
Setup will install Anaconda3 2024.10-1 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Users\HP\anaconda3
Browse...
Space required: 5.2 GB
Space available: 72.4 GB
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda3 integrates with Windows
Create shortcuts (supported packages only).
Add Anaconda3 to my PATH environment variable
NOT recommended. This can lead to conflicts with other applications. Instead, use the Commmand Prompt and Powershell menus added
Windows Start Menu.
Register Anaconda3 as my default Python 3.12
Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Anaconda3 as the primary Python 3.12 on the system.
Clear the package cache upon completion
Recommended. Recovers some disk space without harming functionality.
Anaconda, Inc.
< Back
Install
Cancel


Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Installing
Please wait while Anaconda3 2024.10-1 (64-bit) is being installed.
Setting up the package cache...
Show details
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Installation Complete
Setup was completed successfully.
Completed
Show details
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2024.10-1 (64-bit) Setup
ANACONDA.
Anaconda3 2024.10-1 (64-bit)
Code with Anaconda in the Cloud
Everything you love about Anaconda now available from the cloud with Anaconda's fully-loaded Jupyter notebook.
No setup required. Get started for free.
Code in the cloud with Anaconda Notebooks
ANACONDA.
jupyter
Anaconda, Inc.
< Back
Next >
Cancel

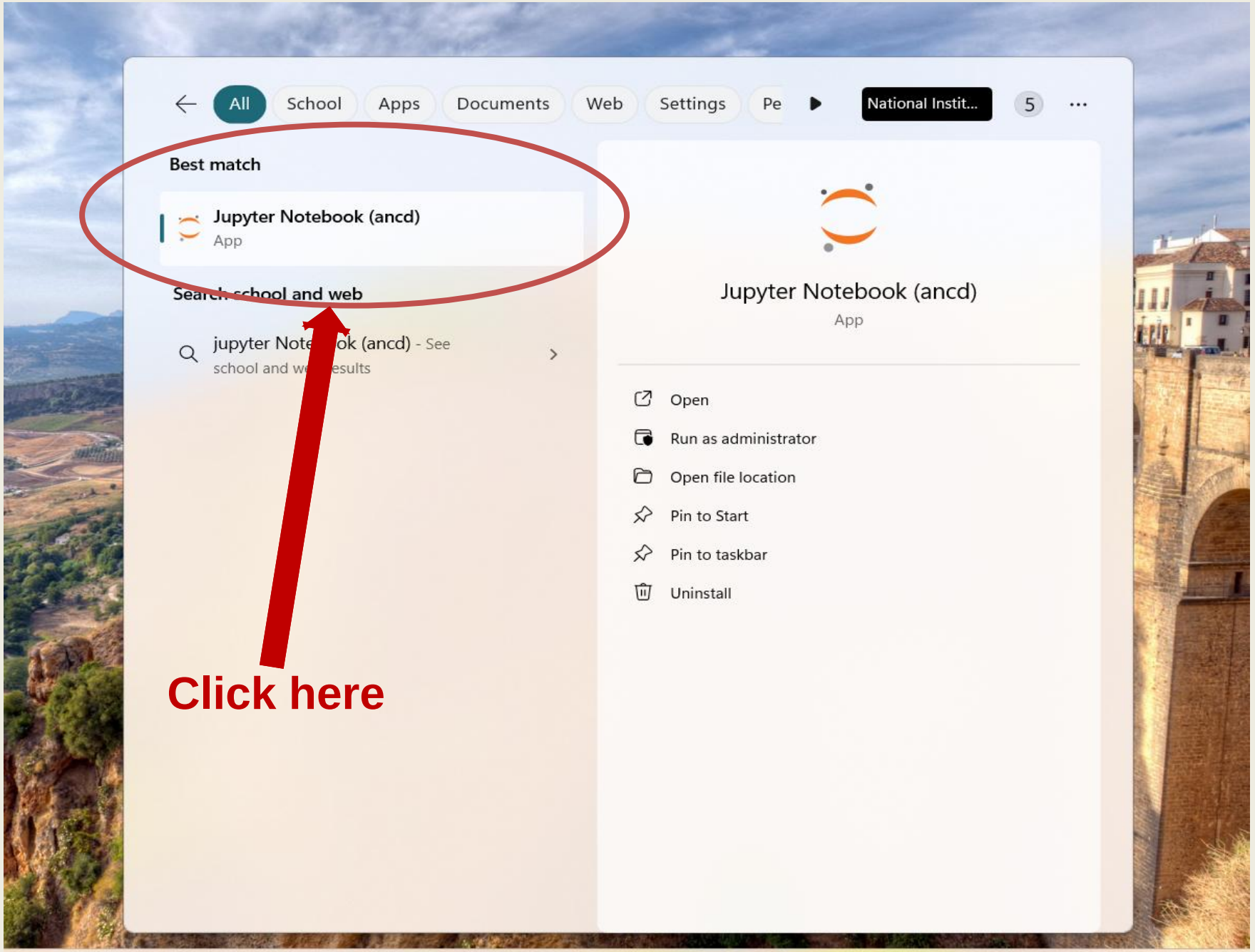

Anaconda3 2024.10-1 (64-bit) Setup
Completing Anaconda3 2024.10-1 (64-bit) Setup
Thank you for installing Anaconda Distribution.
Here are some helpful resources to get you started. We recommend you bookmark the 'Getting Started with Anaconda Distribution' link so you can refer back to it later.
Launch Anaconda Navigator
Getting Started with Anaconda Distribution
ANACONDA.
< Back
Finish
Finish

Creation of Virtual Environment

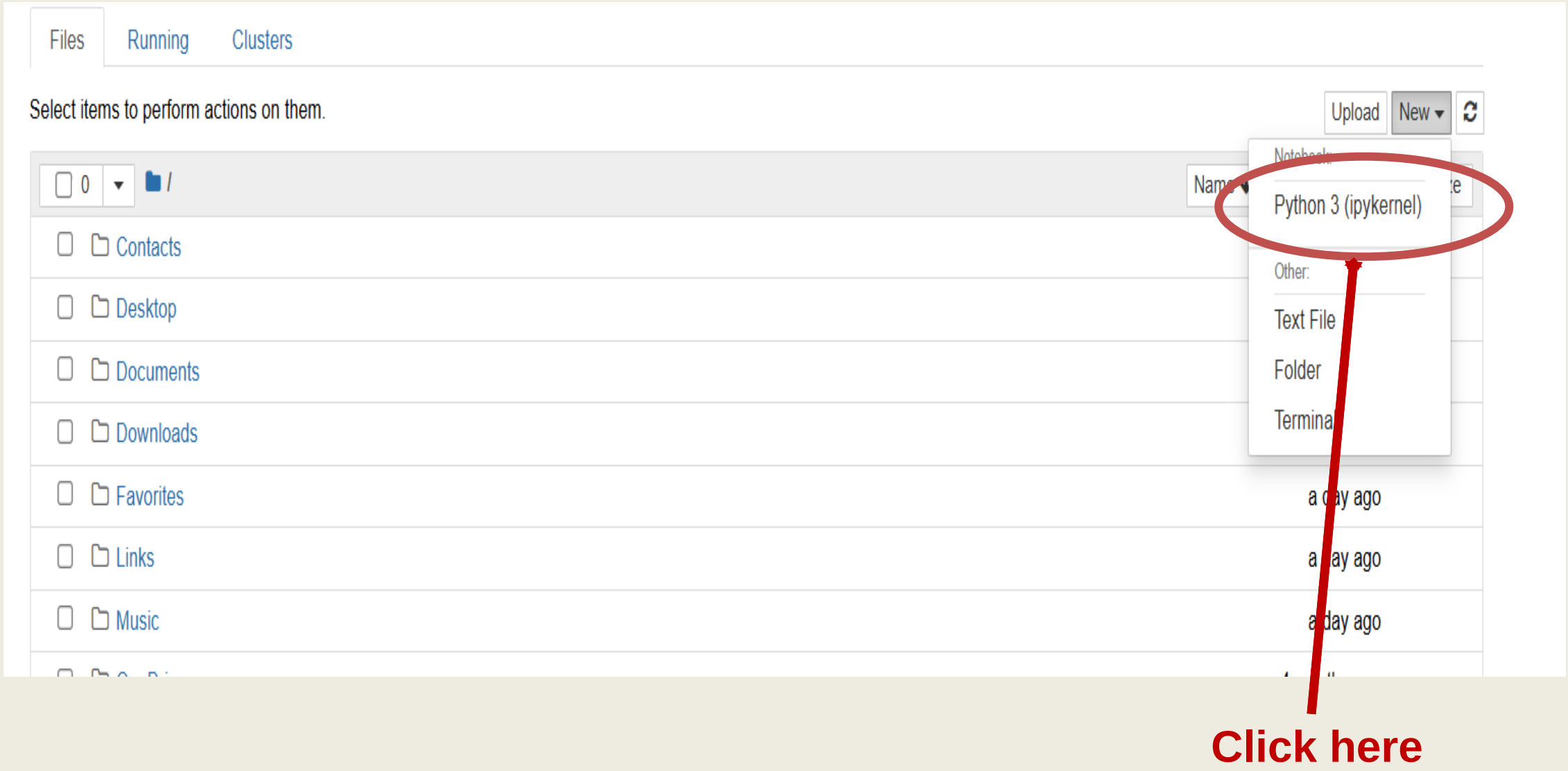
https://www.geeksforgeeks.org/set-up-virtual-environment-for-python-using-anaconda/

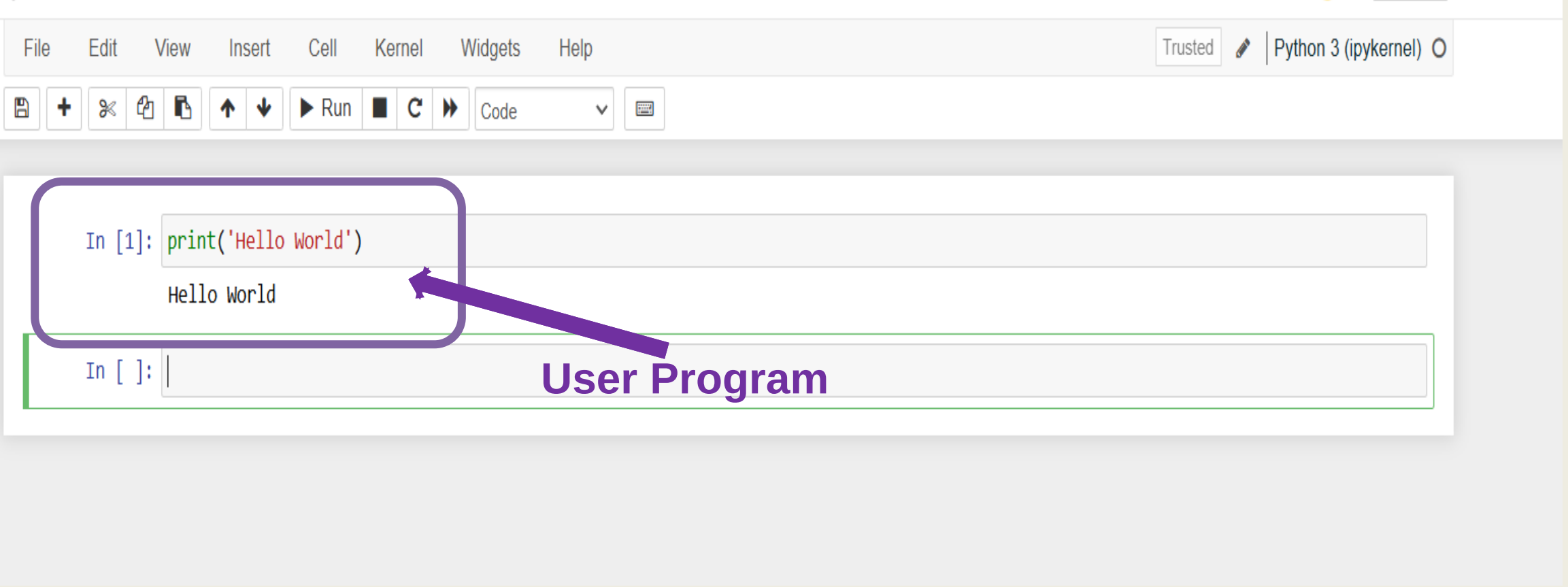
Creation of Virtual Environment

```
# Create a new environment called "myproject" with Python 3.8
conda create -n myproject python=3.8

# Activate the "myproject" environment
conda activate myproject

# Navigate to your project directory
cd path/to/your/project

# Start coding using your Python script
python your_script.py
```


All
School
Apps
Documents
Web
Settings
Pe
National Instit...
5
Best match
Jupyter Notebook (ancd)
App
Search school and web
jupyter Note ok (ancd) - See
school and w esults
Jupyter Notebook (ancd)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall
Click here


Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
Contacts
a day ago
Desktop
an hour ago
Documents
a day ago
Downloads
an hour ago
Favorites
Click here


Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Python 3 (ipykernel)
Other:
Text File
Folder
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
Click here



- Python program communicates its results to user using print
- Python 3 uses input to read user input as a string (str)

In [2]:
```
a = 10
b = 20
c = a+b
print('The addition of a and b is: %d' %c)
```

```
The addition of a and b is: 30
```

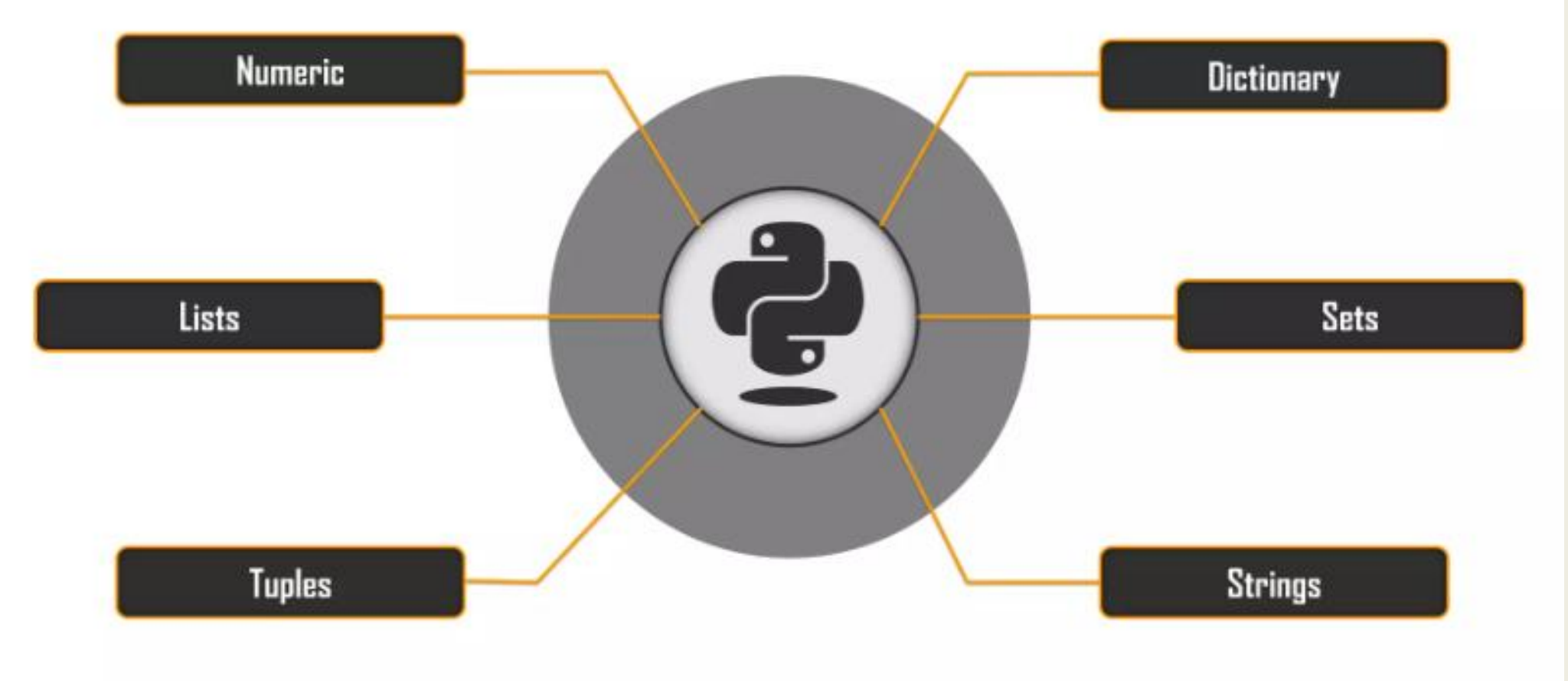
In [3]:
```
a = 10
b = 20
print('The addition of a and b is: %d' %(a+b))
```

```
The addition of a and b is: 30
```

Elements of Python

- A Python program is a sequence of **definitions** and **commands (statements)**
- Commands manipulate **objects**
- Each object is associated with a **Type**
- **Type:**
  - A set of values
  - A set of operations on these values
- **Expressions**: An operation (combination of objects and **operators**)

DATATYPES



1. Numeric

- Numeric Datatypes are used to store numerical values in the variables.
- These are **not mutable**.

Types	Examples
Integer	1, 5, 356
Float	3.142
Complex	10+3j

```
In [4]: int(2)
Out[4]: 2

In [5]: float(2.3)
Out[5]: 2.3

In [6]: int(asd)
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_23324\1830309894.py in <module>
----> 1 int(asd)

NameError: name 'asd' is not defined
```

Type Conversion

- Conversion of value of one type to other
- We are used to int ↔ float conversion in Math
  - ✓Integer 2 is treated as float 2.0 when a real number is expected
  - ✓Float 2.5 is truncated as 2 for integer contexts
- Type names are used as type converter functions

```
In [8]: int(2.5)
Out[8]: 2

In [9]: float(2)
Out[9]: 2.0

In [10]: str(2333)
Out[10]: '2333'

In [ ]:
```

```
In [11]: age = input('How old are you ? ')

         How old are you ? 24

In [13]: print('In 5 years, your age will be', int(age) + 5)

         In 5 years, your age will be 29

In [ ]:
```

2. Lists

- Lists can have heterogeneous datatypes in them.
- These are **mutable**.
- Ordered sequence of values.
- Written as a sequence of comma-separated values between square brackets.

```
In [14]: lst = [1,2,3,4,5]

In [15]: lst
Out[15]: [1, 2, 3, 4, 5]

In [17]: type(lst)
Out[17]: list
```

More Operations on Lists

- L.append(x)
- L.extend(seq)
- L.insert(i, x)
- L.remove(x)
- L.pop(i)
- L.pop()
- L.index(x)
- L.count(x)
- L.sort()
- L.reverse()

x is any value, seq is a sequence value (list, string, tuple, …),
i is an integer value

3. Tuples

- A tuple consists of a number of values separated by commas.

```
In [18]: t = 'Intro to python' , 'Shreeharsha', 101

In [19]: t
Out[19]: ('Intro to python', 'Shreeharsha', 101)
```

- Empty and Singleton tuples
- Tuples can be nested
- Tuples can be concatenated, repeated, indexed and sliced.

```
In [20]: course = 'Python', 'Program', 101

In [21]: student = 'MachineLearning', 34,course

In [22]: empty = ()

In [27]: len(empty)
Out[27]: 0

In [30]: len(student)
Out[30]: 3

In [31]: len(course)
Out[31]: 3
```

4. Dictionaries

- Dictionaries are used to hold key:value pairs.
- These are mutable.

```
In [32]: mydict = {'Name' : 'Shreeharsha' , 'Surname' : 'Dash'}

In [33]: mydict
Out[33]: {'Name': 'Shreeharsha', 'Surname': 'Dash'}
```

Operation	Meaning
len(d)	Number of key:value pairs in d
d.keys()	List containing the keys in d
d.values()	List containing the values in d
k in d	True if key k is in d
d[k]	Value associated with key k in d
d.get(k, v)	If k is present in d, then d[k] else v
d[k] = v	Map the value v to key k in d (replace d[k] if present)
del d[k]	Remove key k (and associated value) from d
for k in d	Iterate over the keys in d

5. Sets

- An unordered collection with no duplicate elements
- Supports
  - membership testing
  - eliminating duplicate entries
  - Set operations: union, intersection, difference, and symmetric difference.

```
In [35]: A = set('acads')
         B = set('institute')

In [36]: A
Out[36]: {'a', 'c', 'd', 's'}

In [37]: B
Out[37]: {'e', 'i', 'n', 's', 't', 'u'}
```

6. Strings

- Strings in Python have type str
- They represent sequence of characters
  - ✓Python does not have a type corresponding to character.
- Strings are enclosed in single quotes(') or double quotes(“)
  - ✓Both are equivalent
- Backslash (\) is used to escape quotes and special characters

Indexing

- Strings can be indexed.
- Negative indices start counting from the right.
- Negatives indices start from -1.

```
In [38]: name = 'Acads'

In [39]: name[-1]
Out[39]: 's'

In [40]: name[-5]
Out[40]: 'A'

In [41]: name[-2]
Out[41]: 'd'
```

Slicing

- To obtain a substring
- s[start:end] means substring of s starting at index start and ending at index end-1
- s[0:len(s)] is same as s
- Both start and end are optional
  - If start is omitted, it defaults to 0
  - If end is omitted, it defaults to the length of string
- s[:] is same as s[0:len(s)], that is same as s

OPERATORS

- Arithmetic

+	-	*	//	/	%	**

- Comparison

==	!=	>	<	>=	<=

- Assignment

=	+=	-=	*=	//=	/=	%=	**=

- Logical

and	or	not

- Bitwise

&	|	^	~	>>	<<

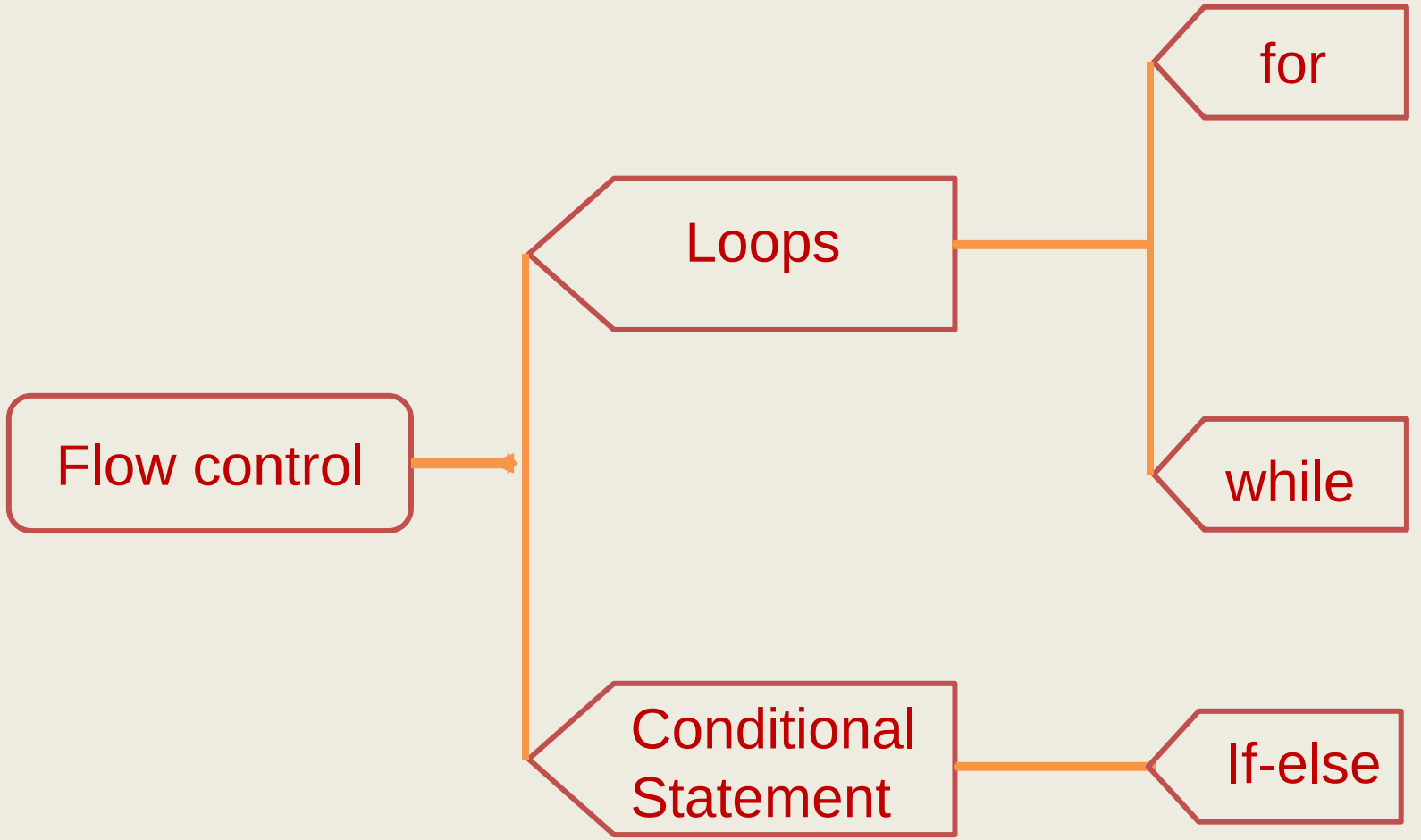
- Membership

in	not in

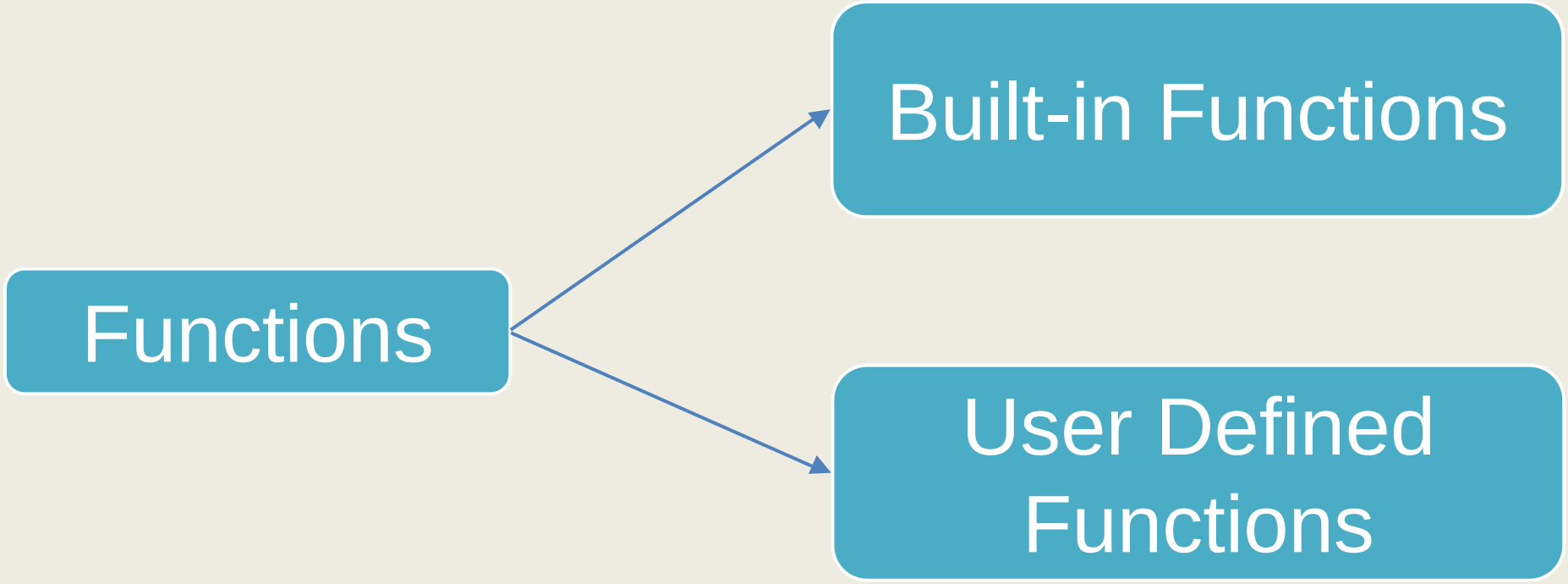
- Identity

is	is not

FLOW CONTROL



FUNCTIONS



FILE HANDLING

- Files are persistent storage
- Allow data to be stored beyond program lifetime
- The basic operations on files are
  - ✓ open, close, read, write
- Python treat files as sequence of lines
  - ✓sequence operations work for the data read from files

Python Libraries

Many popular Python toolboxes/libraries:

- NumPy
- SciPy
- Pandas
- SciKit-Learn

Visualization Libraries

- matplotlib
- Seaborn

Numpy

- Introduces objects for multidimensional arrays and matrices, as well as functions that allow to easily perform advanced mathematical and statistical operations on those objects

- Provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance

- Many other python libraries are built on NumPy

Link: http://www.numpy.org/

SciPy

- collection of algorithms for linear algebra, differential equations, numerical integration, optimization, statistics and more

  - ✓ part of SciPy Stack
  - ✓ built on NumPy

Link: https://www.scipy.org/scipylib/

Pandas

- adds data structures and tools designed to work with table-like data (similar to Series and Data Frames in R)
- provides tools for data manipulation: reshaping, merging, sorting, slicing, aggregation etc.
- allows handling missing data

Link: http://pandas.pydata.org/

SciKit-Learn

- provides machine learning algorithms: classification, regression, clustering, model validation etc.

- built on NumPy, SciPy and matplotlib

Link: http://scikit-learn.org/

Matplotlib

- python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- a set of functionalities similar to those of MATLAB
- line plots, scatter plots, barcharts, histograms, pie charts etc.
- relatively low-level; some effort needed to create advanced visualization

Link: https://matplotlib.org/

Seaborn

- based on matplotlib
- provides high level interface for drawing attractive statistical graphics

Link: https://seaborn.pydata.org/

THANK YOU