

# Scientific Computing with Python

Ayesh Gunawardana

University of Kelaniya

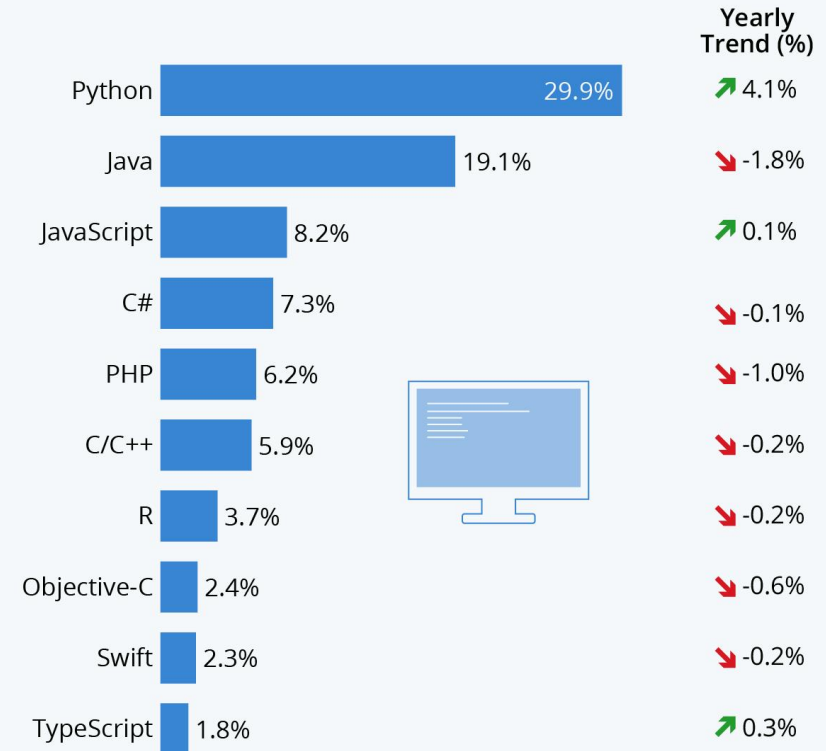
2021

# Why Python

- Easy to learn
  - Imitates the regular English
  - Less number of lines per code
- Large community
- Support for Data science
  - Sklearn
  - TensorFlow(Google)
  - Keras
  - PyTorch(Facebook)

## Python Remains Most Popular Programming Language

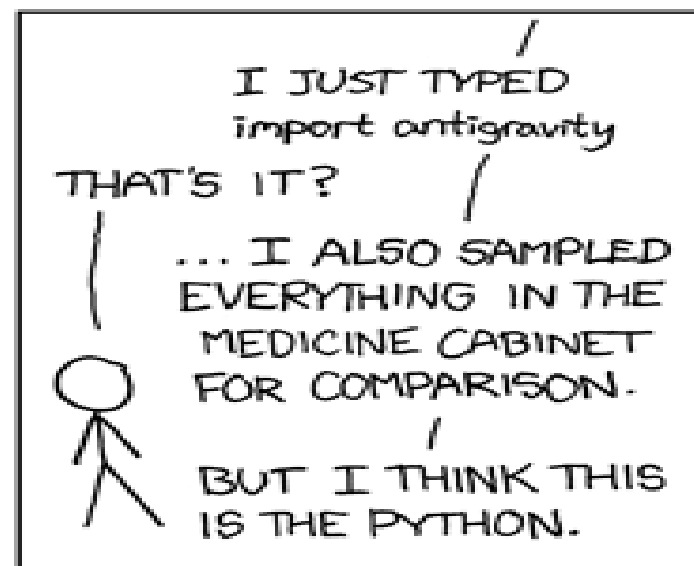
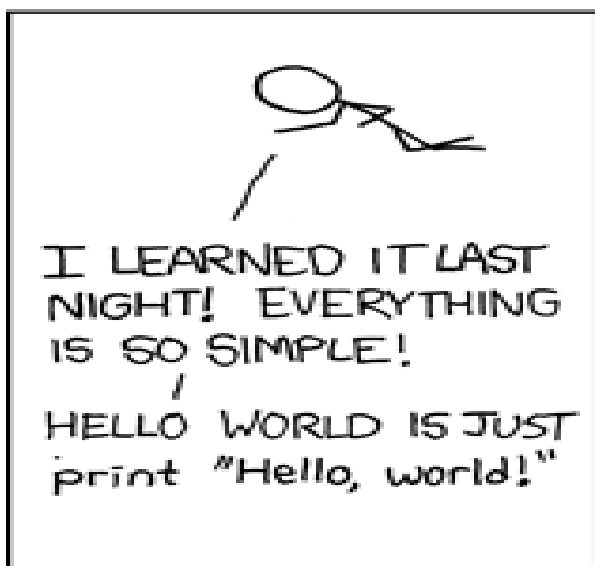
Popularity of each programming language based on share of tutorial searches in Google



Yearly trend compares percent change from Feb 2019 to Feb 2020  
Sources: GitHub, Google Trends

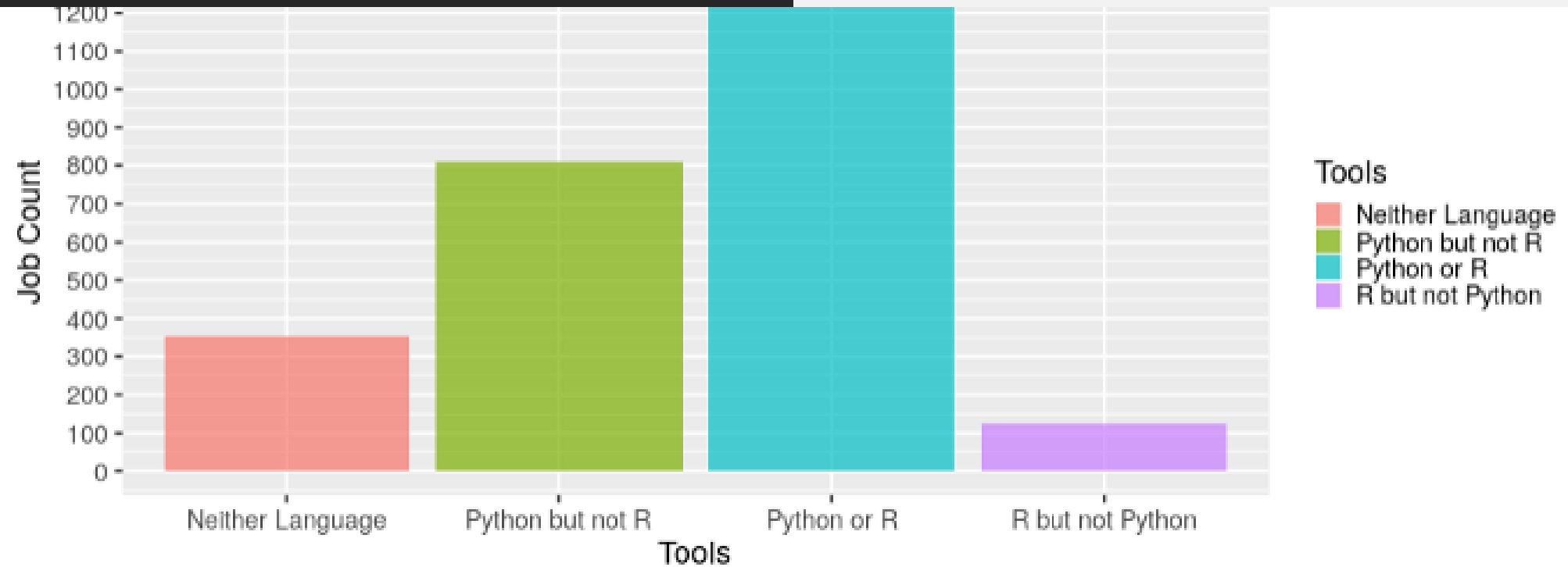


statista



# Python for a Job?

- Required skills for data science roles [Towards data science (2020), Tony Ward]



```

ops.reset_default_graph() # To be able to rerun the model without overwriting tf variables
tf.set_random_seed(1) # To keep results consistent (tensorflow seed)
seed = 3 # To keep results consistent (numpy seed)
(n, n_H0, n_H0, n_C0) = X_train.shape
n_y = Y_train.shape[1]
costs = [] # To keep track of the cost

# Create Placeholders of the correct shape
### START CODE HERE ### (2 line)
X, Y = create_placeholders(n_H0, n_H0, n_C0, n_y)
### END CODE HERE ###

# Initialize parameters
### START CODE HERE ### (1 line)
parameters = initialize_parameters()
### END CODE HERE ###

# Forward propagation: Build the forward propagation in the tensorflow graph
### START CODE HERE ### (2 line)
Z3 = forward_propagation(X, parameters)
### END CODE HERE ###

# Cost function: Add cost function to tensorflow graph
### START CODE HERE ### (1 line)
cost = compute_cost(Z3, Y)
### END CODE HERE ###

# Backpropagation: Define the tensorflow optimizer. Use an AdamOptimizer that minimizes the cost.
### START CODE HERE ### (2 line)
optimizer = tf.train.AdamOptimizer(learning_rate).minimize(cost)
### END CODE HERE ###

# Initialize all the variables globally
init = tf.global_variables_initializer()

# Start the session to compute the tensorflow graph
with tf.Session() as sess:

    # Run the initialization
    sess.run(init)

    # Do the training loop
    for epoch in range(num_epochs):

        minibatch_cost = 0.
        num_minibatches = int(n / minibatch_size) # number of minibatches of size minibatch_size in the train set
        seed = seed + 1
        minibatches = random_mini_batches(X_train, Y_train, minibatch_size, seed)

        for minibatch in minibatches:

            # Select a minibatch
            (minibatch_X, minibatch_Y) = minibatch
            """
            # IMPORTANT: The line that runs the graph on a minibatch.
            # Run the session to execute the optimizer and the cost.
            # The feeddict should contain a minibatch for (X,Y).
            """
            ### START CODE HERE ### (2 line)
            _, temp_cost = sess.run([optimizer, cost], feed_dict={X:minibatch_X, Y:minibatch_Y})
            ### END CODE HERE ###

            minibatch_cost += temp_cost / num_minibatches

        # Print the cost every epoch
        if print_cost == True and epoch % 5 == 0:
            print ("Cost after epoch %i: %f" % (epoch, minibatch_cost))
        if print_cost == True and epoch % 1 == 0:
            costs.append(minibatch_cost)

    # plot the cost
    plt.plot(np.squeeze(costs))
    plt.ylabel('cost')
    plt.xlabel('iterations (per tens)')
    plt.title("Learning rate =" + str(learning_rate))
    plt.show()

    # Calculate the correct predictions
    predict_op = tf.argmax(Z3, 1)
    correct_prediction = tf.equal(predict_op, Y_train)

    # Calculate accuracy
    accuracy = tf.reduce_mean(tf.cast(correct_prediction, "float"))
    print(accuracy)
    train_accuracy = accuracy.eval([X: X_train, Y: Y_train])
    test_accuracy = accuracy.eval([X: X_test, Y: Y_test])
    print("Train Accuracy:", train_accuracy)
    print("Test Accuracy:", test_accuracy)

    return train_accuracy, test_accuracy, parameters

```

## 16 Layer NN: TensorFlow and Keras

```

## Setup the parameters you will use for the two hidden layer neural network.
input_layer_size = 2; # number of the input layer features: In our code we use q^2 and q^3 (normalized)
hidden_layer1_size = 30; # number of nodes in hidden layer1
hidden_layer2_size=30; # number of nodes in hidden layer2
num_labels = 1; # output of the NN: 0(q^2)
# Load Training Data using the dataMonteCarlo.py
print("Loading Data ....")
# save the data to a .mat file for record keeping
(y1,y2,y3)=dnc.dataMonteCarlo()
a={}
a['x_train']=y1
a['y_train']=y2
a['y_n']=y3
sio.savemat('monteData.mat',a)

#using the scipy read the monteData.mat data file
X=y1
y=y2
y_n=y3
(m,l,q)=X.shape
#Defining the output parameters
TTheta1=np.zeros((hidden_layer1_size,input_layer_size+1,q));
TTheta2=np.zeros((hidden_layer2_size,hidden_layer1_size+1,q));
TTheta3=np.zeros((num_labels,hidden_layer2_size+1,q));
#Implementing Neural networks
#loop through q number of neural networks.
for f in range(q):
    Xn=X[:,f]
    yn=y[:,f]
    yn=yn.flatten()
    #Random initialization of the weights
    initial_Theta1 = rbw.randInitializeWeights(input_layer_size, hidden_layer1_size);
    initial_Theta2 = rbw.randInitializeWeights(hidden_layer1_size, hidden_layer2_size);
    initial_Theta3 = rbw.randInitializeWeights(hidden_layer2_size, num_labels);
    #Parameters roll into a column vector initial_nn_params
    initial_nn_params = np.concatenate((initial_Theta1.reshape(initial_Theta1.size, order='F'), initial_Theta2.reshape(initial_Theta2.size, order='F'), initial_Theta3.reshape(
    #this code is compatible with the regularization as well.
    #After figure out a correct regularization parameter use it below.
    lambda_reg = 0 is the default value
    lambda_reg = 0
    nn_params=initial_nn_params
    print("Training Neural Network...")
    #Use the scipy optimize.minimize open source minimization algorithm
    #this can be found at https://github.com/scipy/scipy/blob/master/scipy/optimize/libfgsh.py
    #Minimize a scalar function of one or more variables using the L-BFGS-B algorithm.
    maxiter = 50
    myargs = (input_layer_size, hidden_layer1_size,hidden_layer2_size, num_labels, Xn, yn, lambda_reg)
    results = minimize(mnrf.nnCostFunction, x0=nn_params, args=myargs, options={'disp': True, 'maxiter':maxiter}, method="L-BFGS-B", jac=True)
    nn_params = results["x"]
    #Unroll the parameters from the minimization output
    Theta1p= nn_params[hidden_layer1_size * (input_layer_size + 1):]
    Theta2p=Theta1p[(hidden_layer2_size * (hidden_layer1_size+1):)]
    Theta1 = np.reshape(nn_params[hidden_layer1_size * (input_layer_size + 1):(hidden_layer1_size, input_layer_size + 1), order='F'])
    Theta2 = np.reshape(Theta2p[hidden_layer2_size * (hidden_layer1_size+1):(hidden_layer1_size+1),(hidden_layer2_size, hidden_layer1_size + 1), order='F'])
    Theta3 = np.reshape(Theta2p[num_labels,hidden_layer2_size + 1), order='F')
    a[nnf.nnFit(m,Theta1,Theta2,Theta3,Xn)]

    TTheta1[:,f,:]= Theta1p[:,:]
    TTheta2[:,f,:]= Theta2[:,:]
    TTheta3[:,f,:]= Theta3[:,:]
    #####

e1=TTheta1
e2=TTheta2
e3=TTheta3
e=m
zc={}
zc['TTheta1']=e1
zc['TTheta2']=e2
zc['TTheta3']=e3
zc['e']=e
#save the NN output to a .mat file name :runData.mat
sio.savemat('run_Data.mat',zc)

```

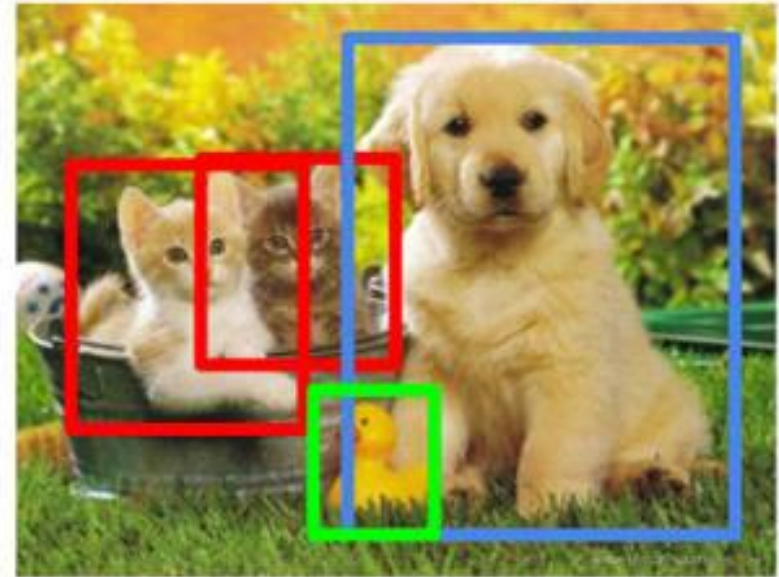
## 2 Layer NN: NumPy and SciPy

## Classification



CAT

## Object Detection



CAT, DOG, DUCK

# Install Python

- [www.Python.org](https://www.python.org)

[www.python.org](https://www.python.org) › [downloads](#) ⋮

## Download Python | Python.org

Download the latest version of Python. **Download Python 3.9.1.** Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other.

You've visited this page 5 times. Last visit: 2/6/21

### Python Releases for Windows

Python 3.9.1 - Python 3.9.0 -  
Python 2.7.18 - Python 3.10.0a4

### Python 3.9.0

Python 3.9.0. Release Date: Oct.  
5, 2020. This is the stable release  
...

### Python 3.8.6

Python 3.8.6. Release Date: Sept.  
24, 2020. This is the sixth ...

[More results from python.org »](#)

### Python 3.7.9

Python 3.7.9. Release Date: Aug.  
17, 2020. Python 3.7.9 is the ...

### Python 3.8.5

Python 3.8.5. Release Date: July  
20, 2020. This is the fifth ...

### PEP 569

PEP: 569. Title: Python 3.8  
Release Schedule. Author ...

[wiki.python.org](https://wiki.python.org) › [moin](#) › [BeginnersGuide](#) › [Download](#) ⋮



Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

## Download the latest version for Windows

Download Python 3.9.1

Looking for Python with a different OS? Python for [Windows](#),  
[Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#),  
[Docker images](#)

Looking for Python 2.7? See below for specific releases



### Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.9	bugfix	2020-10-05	2025-10	PEP 596



Python

PSF

Docs

PyPI

Jobs

Community



Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

## Download

Download Python

Looking for Python

[Linux/UNIX](#), [Mac OS](#)

Want to help test d

[Docker images](#)

Looking for Python

All releases

Source code

Windows

Mac OS X

Other Platforms

License

Alternative Implementations

### Download for Windows

Python 3.9.1

**Note that Python 3.9+ *cannot* be used on Windows 7 or earlier.**

Not the OS you are looking for? Python can be used on many operating systems and environments.

[View the full list of downloads.](#)

## Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.9	bugfix	2020-10-05	2025-10	PEP 596

3.8	bugfix	2019-10-14	2024-10	<a href="#">PEP 569</a>
3.7	security	2018-06-27	2023-06-27	<a href="#">PEP 537</a>
3.6	security	2016-12-23	2021-12-23	<a href="#">PEP 494</a>
2.7	end-of-life	2010-07-03	2020-01-01	<a href="#">PEP 373</a>

### Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
<a href="#">Python 3.8.2</a>	Feb. 24, 2020	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.1</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.6</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.10</a>	Dec. 18, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.9</a>	Nov. 2, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.8</a>	Oct. 29, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 2.7.17</a>	Oct. 19, 2019	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 2.7.5</a>	Oct. 15, 2010	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)

## Install Python 3.7.6 (64-bit)

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



**Install Now**

C:\Users\Genji\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation  
Creates shortcuts and file associations



**Customize installation**

Choose location and features

☒ Install launcher for all users (recommended)

☐ Add Python 3.7 to PATH

Cancel

python  
for  
windows



## Optional Features

☒ Documentation

Installs the Python documentation file.

☒ pip

Installs pip, which can download and install other Python packages.

☒ tk/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.

Back

Next

Cancel



python  
for  
windows

## 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

C:\Users\Gen\AppData\Local\Programs\Python\Python37

Browse

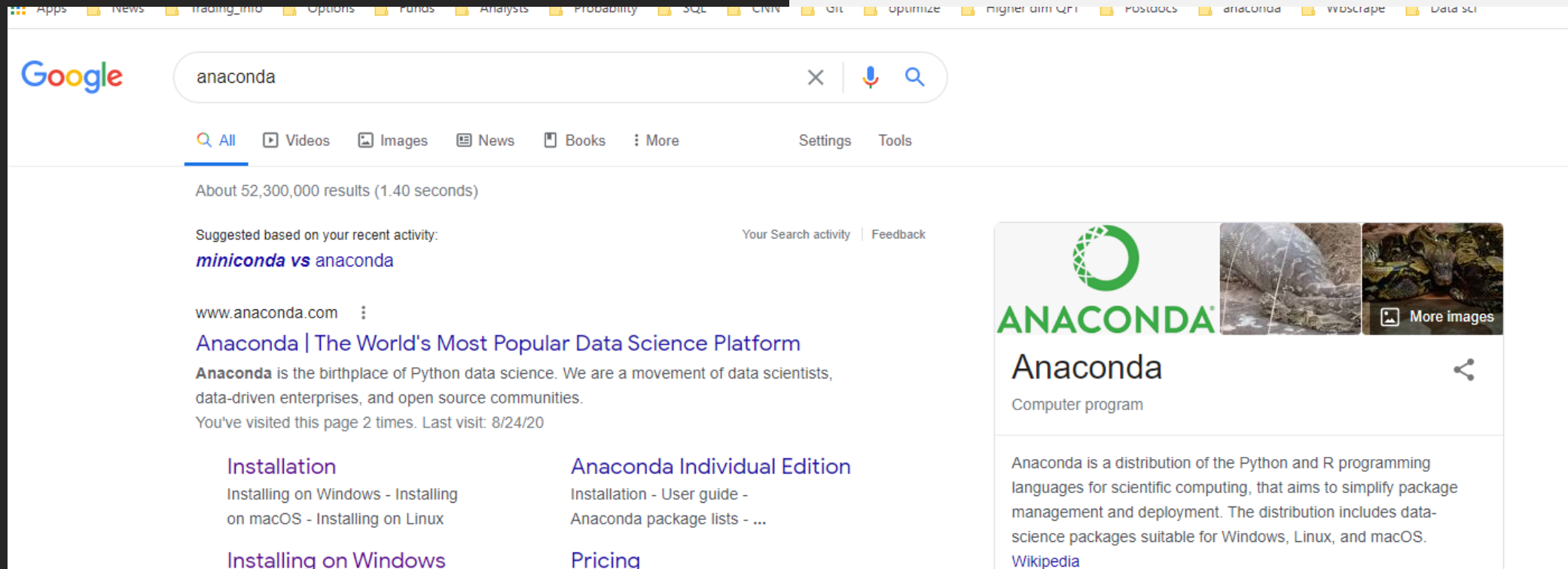
Back

Install

Cancel

# Install anaconda

- Anaconda is an environment manager
- Need anaconda to run Jupyter notebook



The screenshot shows a Google search for 'anaconda'. The search bar at the top contains the word 'anaconda'. Below the search bar, there are tabs for 'All', 'Videos', 'Images', 'News', 'Books', and 'More'. The search results show 'About 52,300,000 results (1.40 seconds)'. A suggested search based on recent activity is 'miniconda vs anaconda'. The first search result is from 'www.anaconda.com' with the title 'Anaconda | The World's Most Popular Data Science Platform'. The description states: 'Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities. You've visited this page 2 times. Last visit: 8/24/20'. Below this, there are links for 'Installation' (with sub-links for Windows, macOS, and Linux) and 'Installing on Windows'. To the right of the main result is a knowledge panel for 'Anaconda', labeled 'Computer program'. It features the Anaconda logo, a snake image, and a 'More images' link. The panel also includes a description: 'Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.' and a 'Wikipedia' link.

Google

anaconda

Q All Videos Images News Books More Settings Tools

About 52,300,000 results (1.40 seconds)

Suggested based on your recent activity: [miniconda vs anaconda](#)

Your Search activity | Feedback

[www.anaconda.com](http://www.anaconda.com)

**Anaconda | The World's Most Popular Data Science Platform**

**Anaconda** is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities.

You've visited this page 2 times. Last visit: 8/24/20

**Installation**

Installing on Windows - Installing on macOS - Installing on Linux

**Installing on Windows**

**Anaconda Individual Edition**

Installation - User guide - Anaconda package lists - ...

**Pricing**

**ANACONDA**

**Anaconda**

Computer program

Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

[Wikipedia](#)

Home

Anaconda Individual Edition

Installation

Installing on Windows

Installing on macOS

Installing on Linux

Installing on Linux POWER

Installing in silent mode

Installing for multiple users

Verifying your installation

Anaconda installer file hashes

Updating from older versions

Uninstalling Anaconda

User guide

Reference

End User License Agreement -  
Anaconda Individual Edition

Anaconda Commercial Edition



## Installing on Windows

### Note

Using Anaconda in a commercial setting? You may need to use [Anaconda Commercial Edition](#). If you have already purchased Commercial Edition, please p  
[Authenticating Commercial Edition](#) section after completing your installation here.

Haven't purchased Commercial Edition yet? Visit <https://anaconda.cloud/register> to get started.

1. [Download the Anaconda installer](#).
2. RECOMMENDED: [Verify data integrity with SHA-256](#). For more information on hashes, see [What about cryptographic hash verification?](#)
3. Double click the installer to launch.

### Note

To prevent permission errors, do not launch the installer from the [Favorites folder](#).

### Note

If you encounter issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If y  
all users, uninstall Anaconda and re-install it for your user only and try again.

4. Click Next.
5. Read the licensing terms and click "I Agree"



# Anaconda Installers

## Windows

Python 3.8

64-Bit Graphical Installer (457 MB)

32-Bit Graphical Installer (403 MB)

## MacOS

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

## Linux

Python 3.8

64-Bit (x86) Installer (529 MB)

64-Bit (Power8 and Power9) Installer (279 MB)

### ADDITIONAL INSTALLERS

The [archive](#) has older versions of Anaconda Individual Edition installers. The





## Welcome to Anaconda3 2020.07 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2020.07 (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



### Select Installation Type

Please select the type of installation you would like to perform for Anaconda3 2020.07 (64-bit).

Install for:

- ☒ Just Me (recommended)
- ☐ All Users (requires admin privileges)

Anaconda, Inc.

< Back

Next >

Cancel



### Choose Install Location

Choose the folder in which to install Anaconda3 2020.07 (64-bit).

Setup will install Anaconda3 2020.07 (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\ayesh\anaconda3

Browse...

Space required: 2.7GB

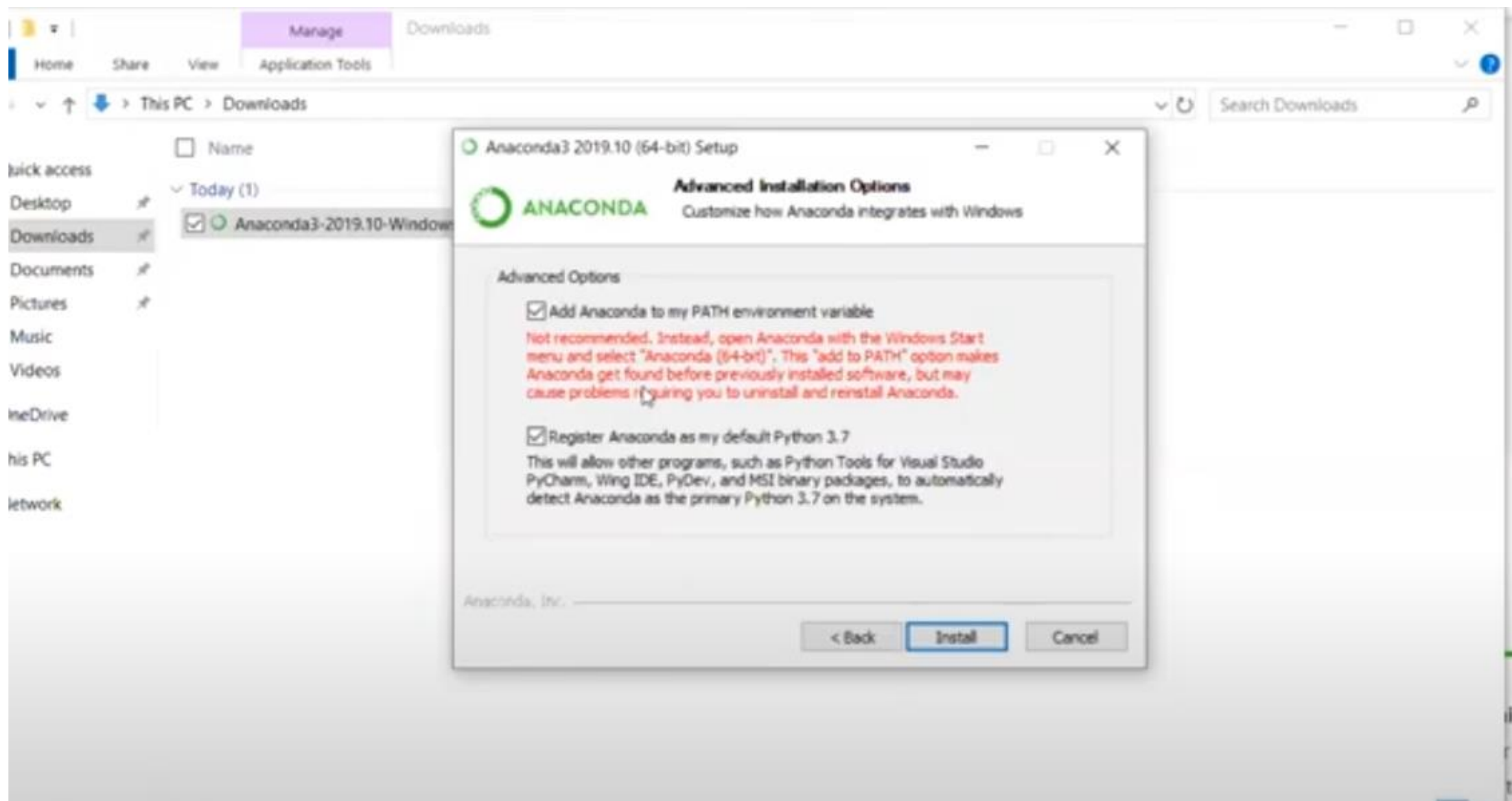
Space available: 247.5GB

Anaconda, Inc.

< Back

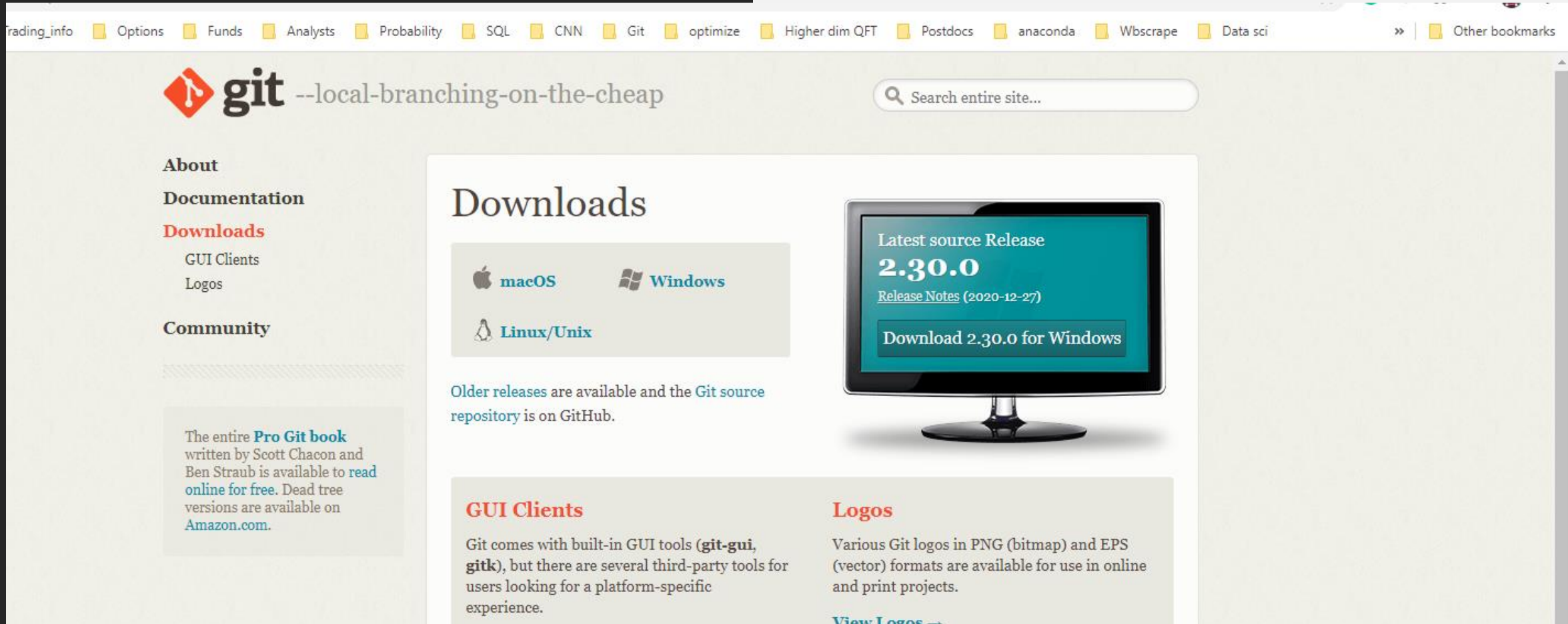
Next >

Cancel



# Install Git

Download it at [git-scm.com](https://git-scm.com)



The screenshot shows the Git website homepage. At the top, there's a navigation bar with various links like 'trading\_info', 'Options', 'Funds', 'Analysts', 'Probability', 'SQL', 'CNN', 'Git', 'optimize', 'Higher dim QFT', 'Postdocs', 'anaconda', 'Wbscrape', 'Data sci', and 'Other bookmarks'. Below this, the Git logo is prominently displayed with the tagline '--local-branching-on-the-cheap'. A search bar is located to the right of the logo. On the left side, there's a sidebar with links for 'About', 'Documentation', 'Downloads', 'GUI Clients', 'Logos', and 'Community'. The main content area features a 'Downloads' section with icons for macOS, Windows, and Linux/Unix. To the right of this, there's a monitor graphic displaying the latest source release '2.30.0' and a button to 'Download 2.30.0 for Windows'. Below the 'Downloads' section, there's a note about older releases being available on GitHub. At the bottom, there are sections for 'GUI Clients' and 'Logos'.

trading\_info Options Funds Analysts Probability SQL CNN Git optimize Higher dim QFT Postdocs anaconda Wbscrape Data sci Other bookmarks

**git** --local-branching-on-the-cheap

Search entire site...

**About**  
**Documentation**  
**Downloads**  
GUI Clients  
Logos  
**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

macOS Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

**GUI Clients**  
Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

**Logos**  
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.  
[View Logos →](#)

Latest source Release  
**2.30.0**  
[Release Notes \(2020-12-27\)](#)  
Download 2.30.0 for Windows

## Downloads

GUI Clients

Logos

## Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

## Your download is starting...

You are downloading the latest (2.23.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 2 months ago, on

[Click here to download manually.](#)



## Now What?



GUI Clients

Logos

Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

**Your download is starting...**

You are downloading the latest (2.23.0) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released 2 months ago, on

[Click here to download manually.](#)



**Now What?**

Now that you have downloaded Git, it's time to start using it.

# Git

- 
- After finishing the installation
    - Close the command prompt and open it again
    - Type git init to see whether it is working properly

```
C:\Users\ayesh>git init
✓Reinitialized existing Git repository in C:/Users/ayesh/.git/

C:\Users\ayesh>_
```



# Anaconda environment

- Information:
  - <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- `conda create -n myenv python=3.7`(or whatever version)

```
base) D:\python_teaching>conda create -n test1 python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

WARNING conda.gateways.disk.delete:unlink_or_rename_to_trash(139): Could not remove or rename C:\Users\ayesh\anaconda3\p
s\request-2.83.1-0.tar.bz2. Please remove this file manually (you may need to reboot to free file handles)

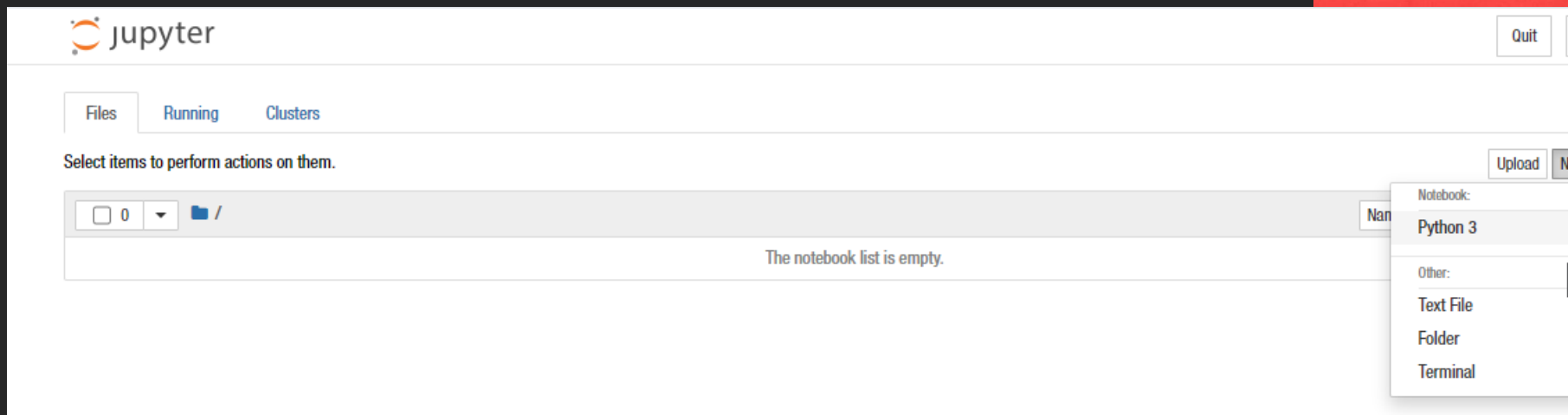
# Package Plan ##

environment location: C:\Users\ayesh\anaconda3\envs\test1

added / updated specs:
- python=3.7
```

# Install packages

- conda install numpy
- conda install matplotlib
- Open: Jupyter notebook



# Numeric Expressions

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Power
%	Remainder

# Number Types

Numbers have two main types

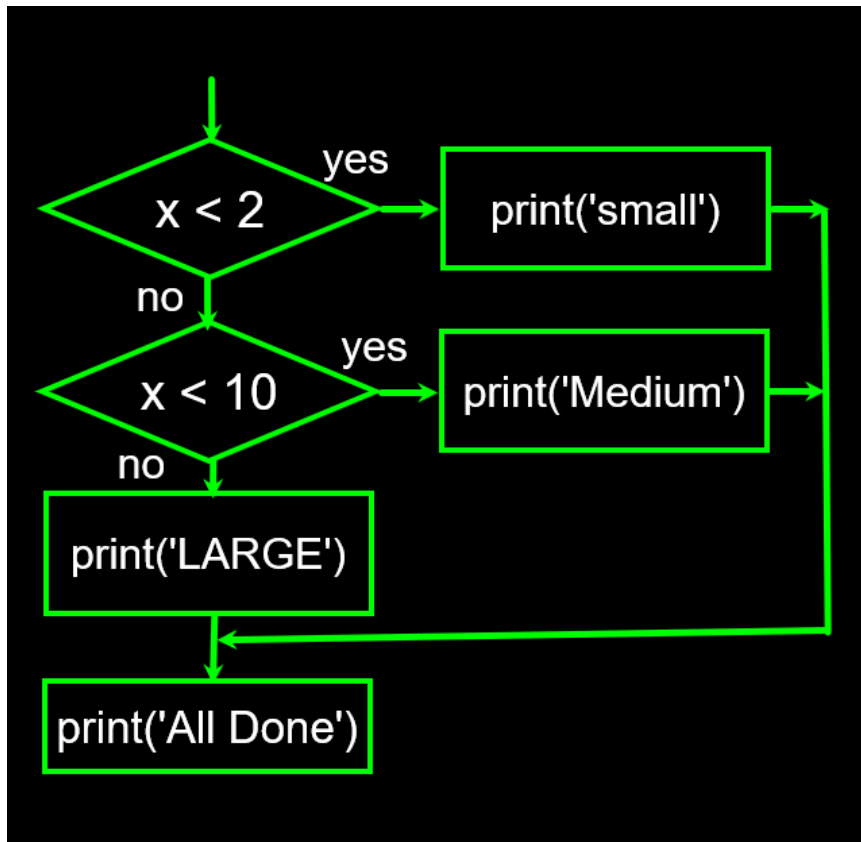
```
graph TD; A[Numbers have two main types] --> B["- Integers are whole numbers:  
-14, -2, 0, 1, 100, 401233"]; B --> C["- Floating Point Numbers have decimal parts: -2.5, 0.0, 98.6, 14.0"]; C --> D["There are other number types - they are variations on float and integer"];
```

- Integers are whole numbers:  
-14, -2, 0, 1, 100, 401233

- Floating Point Numbers have decimal parts: -2.5, 0.0, 98.6, 14.0

There are other number types - they are variations on float and integer

# Conditional Statements



```
In [15]: x = input('Enter a number: ')
```

Enter a number: 7

```
In [19]: x = int(x)
if x < 2:
    print( "You entered "+str(x)+", is small number")
elif x < 10:
    print( "You entered "+str(x)+", is medium number")
else:
    print("You entered "+str(x)+", is lage number")
```

You entered 7, is medium number

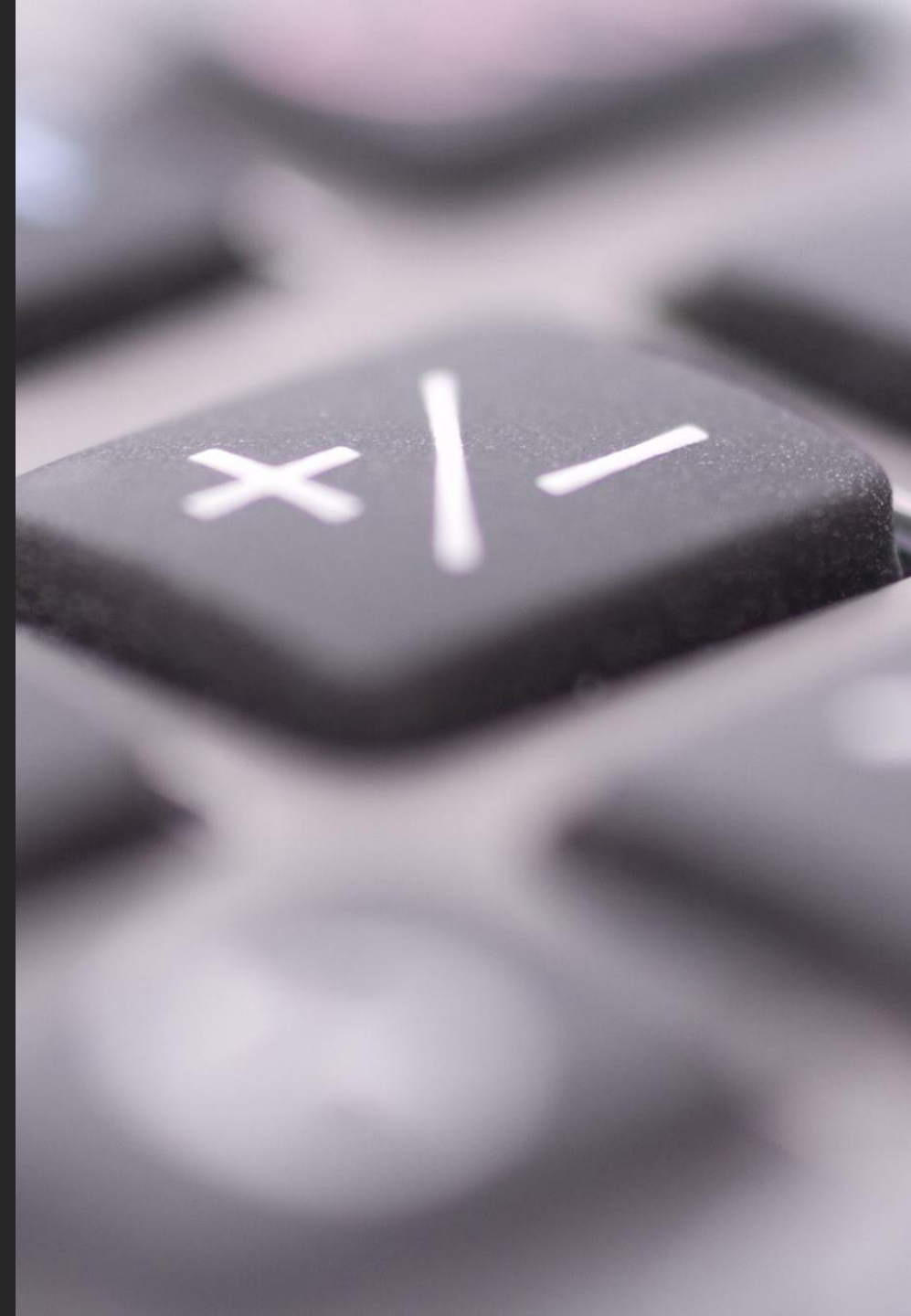
# Functions

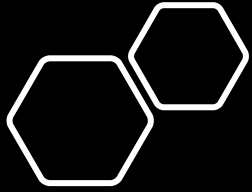
- There are two kinds of functions
  - Built-in functions that are provided as part of Python
    - Ex: `print()`, `input()`, `type()`, `float()`, `int()` ...
  - Functions that we define ourselves

```
def myConditionalEval(x):  
    if x < 2:  
        print( "You entered "+str(x)+" , is a small number")  
    elif x< 10:  
        print( "You entered "+str(x)+" , is a medium number")  
    else:  
        print("You entered "+str(x)+" , is a lage number")
```

```
myConditionalEval(6)
```

```
You entered 6, is s medium number
```





# Definite Loops

- For loop
  - Runs over finite set of elements once
- Structure
  - Starts with **for** construct

```
x = [1,2,3]
for i in x:
    print("the number is "+str(i))
```

```
the number is 1
the number is 2
the number is 3
```

---

# Data structures: Lists

- List: allows inserting multiple values to a variable

```
In [ ]: uokSubjects = ['AMAT', 'PMAT', 'PHYS', 'STAT', 'CHEM']
```

- In matlab list starts with index = 1
- In Python list starts with index = 0
  - Ex: In matlab index = 1 is 'AMAT',  
print(uokSubjects[1]) = AMAT
  - Ex: In Python index = 0 is 'AMAT',  
print(uokSubjects[0]) = AMAT



# Data structures: Dictionaries



- 
- List: Like a can of pringles
    - Neatly stacked pringles at specific position
  - Dictionary: like a bag of different colored chips
    - No specific order but easy to pick up a specific-colored chip

# Dictionaries

```
lst = list()
>>> lst.append(21)
>>> lst.append(183)
>>> print(lst)
[21, 183]
>>> lst[0] = 23
>>> print(lst)
[23, 183]
```

```
ddd = dict()
>>> ddd['age'] = 21
>>> ddd['course'] = 182
>>> print(ddd)
{'course': 182, 'age': 21}
>>> ddd['age'] = 23
>>> print(ddd)
{'course': 182, 'age': 23}
```