



FREE eBook

LEARNING networkx

Free unaffiliated eBook created from
Stack Overflow contributors.

#networkx

Table of Contents

About	1
Chapter 1: Getting started with networkx	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Basic program for displaying nodes in matplotlib using networkx.....	4
Credits	5

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [networkx](#)

It is an unofficial and free networkx ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official networkx.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with networkx

Remarks

This section provides an overview of what networkx is, and why a developer might want to use it.

It should also mention any large subjects within networkx, and link out to the related topics. Since the Documentation for networkx is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Welcome to the world of Graphs. If you have connected data then you might need one of the types of graphs to model those patterns. There are several things that can be done with Graphs like mapping traffic patterns, managing water distribution networks, social media analysis, etc... At it's heart we need to be able to create Nodes and Edges with data associated about each.

NetworkX is a library dealing with Graph Database creation/import/export/manipulation/algorithms/plotting. You can start using several types of network graphs. For example: Facebook might put their users in a plain Graph()

```
import networkx as nx
facebook = nx.Graph()
facebook.add_node('you')
facebook.add_node('me')
if both_agree_to_be_friends('you','me'):
    facebook.add_edge('you','me') #order isn't important here.
```

Facebook would use a regular Graph() because there isn't anything special about the edge between nodes. This Facebook example can only have one edge (friendship) between nodes.

Another type of Graph would be a Directed Graph. Twitter would use a Directed Graph because the nodes have a direction. In Twitter I can follow you but you don't have to follow me. So we could represent that with this code:

```
import networkx as nx
twitter = nx.DiGraph()
twitter.add_node('you')
twitter.add_node('me')
twitter.add_edge('me','you') #order is important here.
```

This is how Twitter might set up 'me' to follow 'you', but not the other way around.

There are MultiGraphs() and MultiDiGraphs() as well just in case you want more than one edge between two nodes. These four types cover a wide variety of problems that can be represented

with Graphs. If you want you can add a dictionary of data to both nodes and edges. The [documentation on this module](#) is incredible. Every algorithm is well researched and well implemented. I hope you enjoy working with it.

Installation instructions: The detailed instructions on installing NetworkX is available [here](#).

As with any other python package, NetworkX can be installed using pip, Miniconda/Anaconda and from source code.

Installing with pip

```
pip install networkx
```

An attempt will be made to find and install an appropriate version of NetworkX that matches your operating system and Python version.

To use pip, you need to have `setuptools` installed.

If you want to install the development version from GitHub, use the command

```
pip install git://github.com/networkx/networkx.git#egg=networkx
```

Miniconda and Anaconda use `conda` for software installation/updates.

NetworkX is [currently installed](#) with [Anaconda](#). [Miniconda](#) doesn't come with NetworkX by default.

You can update/install NetworkX to the latest version with:

```
conda install networkx
```

or if you want to update NetworkX installation then

```
conda update networkx
```

Installing from source

Source file archive

1. Download the source from <https://pypi.python.org/pypi/networkx/> or get the [latest version](#).
2. Unpack and change directory to the source directory (it should have the files README.txt and setup.py).
3. Run `python setup.py install` to build and install
4. (Optional) Run `nosetests` to execute the tests if you have nose installed.

Installing from GitHub

1. Clone the NetworkX repository (see <https://github.com/networkx/networkx/> for options)

```
git clone https://github.com/networkx/networkx.git
```

2. Change directory to NetworkX
3. Run `python setup.py install` to build and install
4. (Optional) Run `nosetests` to execute the tests if you have nose installed.

If you don't have permission to install software on your system, you can install into another directory using the `--user`, `--prefix`, or `--home` flags to `setup.py`.

Requirements To use NetworkX you need Python 2.7, 3.3 or later

Optional Packages

1. NumPy: Provides matrix representation of graphs and is used in some graph algorithms for high-performance matrix computations. (<http://scipy.org/Download>)
2. SciPy: Provides sparse matrix representation of graphs and many numerical scientific tools. (<http://scipy.org/Download>)
3. Matplotlib: Provides flexible drawing of graphs. (<http://matplotlib.sourceforge.net/>)
4. GraphViz in conjunction with either PyGraphviz (<http://pygraphviz.github.io>) or pydotplus (<https://github.com/carlos-jenkins/pydotplus>): provides graph drawing and graph layout algorithms. (<http://graphviz.org>)
5. PyYAML: Required for YAML format reading and writing. (<http://pyyaml.org>)

3: <http://conda.pydata.org/miniconda.html> 2:
<https://networkx.github.io/documentation/development/install.html>

Basic program for displaying nodes in matplotlib using networkx

```
import networkx as nx # importing networkx package
import matplotlib.pyplot as plt # importing matplotlib package and pyplot is for displaying
the graph on canvas
b=nx.Graph()
b.add_node('helloworld')
b.add_node(1)
b.add_node(2)
'''Node can be called by any python-hashable obj like string,number etc'''
nx.draw(b) #draws the networkx graph containing nodes which are declared till before
plt.show() # displays the networkx graph on matplotlib canvas
```

Additional clarification:

```
nx.draw(b,nodelist=[1,'helloworld']) #displays the particular nodes which are given by
nodelist only
nx.draw_networkx(b,nodelist=[1,'helloworld']) #displays the node along with its name given by
us i.e 1, hello respectively
```

Read Getting started with networkx online: <https://riptutorial.com/networkx/topic/5338/getting-started-with-networkx>

Credits

S. No	Chapters	Contributors
1	Getting started with networkx	Back2Basics , Community , Nikhil , Serenity , Unni