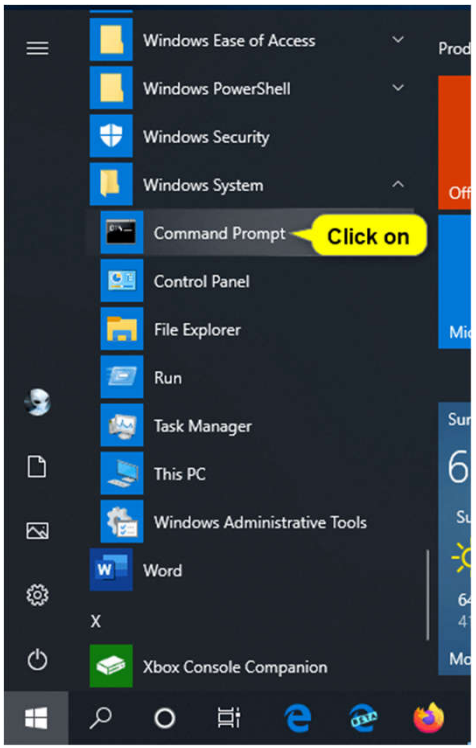
What is NetworkX?

A Python package for the creation and manipulation of complex networks.

- Data structures for graphs, digraphs, and multigraphs
- Many standard graph algorithms
- Network structure and analysis measures
- Generators for classic graphs, random graphs, and synthetic networks

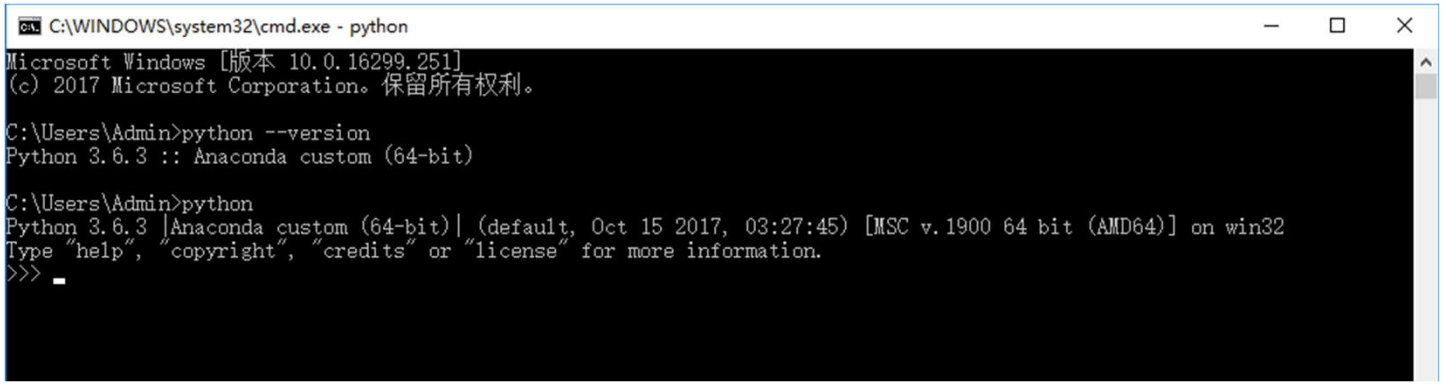
How to install NetworkX

https://networkx.github.io/documentation/stable/install.html



- NetworkX requires Python **3.5**, **3.6**, **3.7**, or **3.8**.
- You should type ***python --version*** in **command prompt**(**power shell** or **命令提示符** or **terminal**) to check the version of your python before installing the package.

- Use command ***pip install networkx*** to install NetworkX.
- Use command ***conda install -c anaconda network*** to install NetworkX if you are using conda environment



Create a graph

Import the networks package

```
>>> import networkx as nx
```

Create an empty graph with no nodes and no edges.

```
>>> G = nx.Graph()
```

Create an empty directed graph.

```
>>> G = nx.DiGraph()
```

Add one or more nodes to the graph G.

```
>>> G.add_node(node)
>>> G.add_nodes_from([node1, node2…])
```

Add one or more edges to the graph G.

```
>>> G.add_edge(u, v)
>>> G.add_edges_from([(u1, v1), (u2, v2)…])
```

https://networkx.github.io/documentation/stable/tutorial.html#creating-a-graph

Algorithms

Build dfs tree from source node.

```
>>> T = nx.dfs_tree(graph, source_node)
```

Print the traversing process of dfs from the source node

```
>>> T.edges()
```

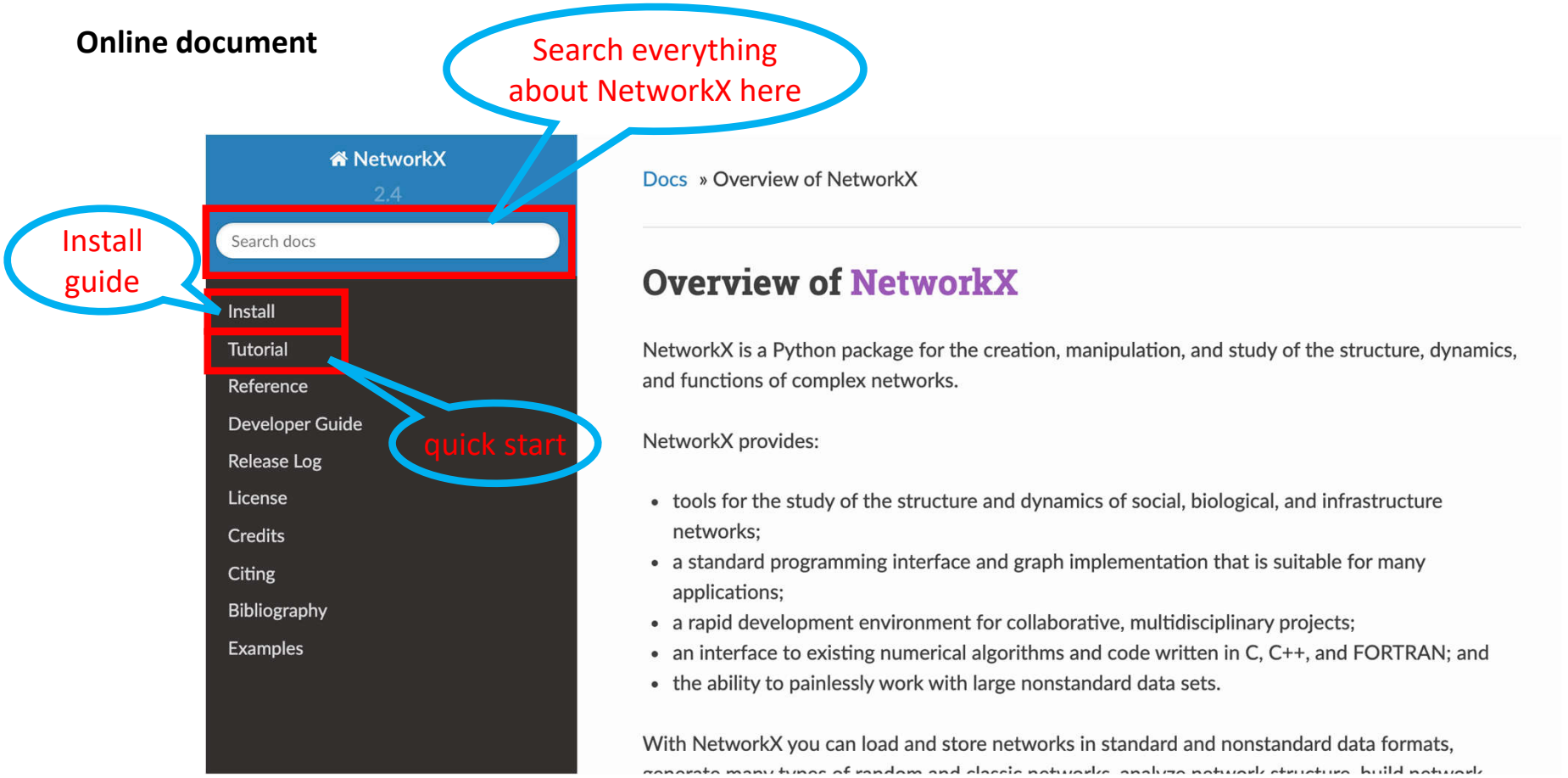
https://networkx.github.io/documentation/stable/reference/algorithms/traversal.html?highlight=bfs%23

Print the shorts path from source node to target node using Dijkstra algorithm

```
>>> dijkstra_path(G, source, target, weight='weight')
```

https://networkx.github.io/documentation/stable/reference/algorithms/shortest_paths.html

Online document



https://networkx.github.io/documentation/stable/index.html