# Pomoć oko funkcija

Python funkcija help dohvaća službenu dokumentaciju predane funkcije (koja često sadrži i primjere korištenja). Primjer:

help(networkx.Graph)

# Potrebni importovi

import networkx as nx

import matplotlib.pyplot as plt

import numpy as np

import itertools

import random

from simulation import Simulation

# Inicijalizacija mreže

graph = nx.Graph()

graph = nx.DiGraph()

# Dodavanje čvorova i veza

graph.add\_node()

graph.add\_nodes\_from()

graph.add\_edge()

graph.add\_edges\_from()

# Iteriranje po čvorovima ili vezama

graph.nodes()

graph.edges()

# Dodavanje i pristupanje atributima čvorova i veza

nx.set\_node\_attributes(graph, value, attribute)

nx.get\_node\_attributes(graph, attribute)

nx.set\_edge\_attributes(graph, value, attribute)

nx.get\_edge\_attributes(graph, attribute)

# Prikaz mreže

nx.draw(graph)

nx.draw\_networkx(graph, node\_color, labels, pos)

# Podmreže

nx.subgraph(graph, subgraph\_nodes)

nx.find\_cliques(graph)

nx.connected\_components(graph)

# Stupnjevi čvorova

graph.degree()

graph.in\_degree()

graph.out\_degree()

# Čitanje i pisanje

nx.read\_edgelist(path)

nx.write\_edgelist(graph, path)

# Svojstvene vrijednosti mreže

graph.number\_of\_nodes()

graph.number\_of\_edges()

nx.degree\_assortativity\_coefficient(graph)

nx.average\_shortest\_path\_length(graph)

nx.diameter(graph)

nx.average\_clustering(graph)

nx.degree\_centrality(graph)

nx.betweenness\_centrality(graph)

nx.is\_connected(graph)

nx.density(graph)

# K-jezgrena dekompozicija

nx.k\_core(graph, k)

nx.k\_shell(graph, k)

# Slučajni modeli

nx.gnp\_random\_graph(n\_nodes, probability)

nx.gnm\_random\_graph(n\_nodes, n\_edges)

nx.watts\_strogatz\_graph(n\_nodes, n\_neighbours\_to\_join, rewiring\_probability)

nx.barabasi\_albert\_graph(n\_nodes, n\_connections)

# Kombinacije elemenata

itertools.combinations(iterator, combination\_length)

# Simulacija

sim = Simulation(G, initial\_state, state\_transition, name)

sim.state()

sim.run()

sim.plot()