

Python

Python na hitro

Vladimir Batagelj

Univerza v Ljubljani
FMF, matematika

Seminar za učitelje multiplikatorje, Ljubljana, 2. november 2004

Sredin seminar 1095 in 1096, Ljubljana, 16. in 23. junij 2004

Kazalo

1	Programski jezik za uvod v programiranje	1
5	Podatki v Pythonu	5
14	Krmilni stavki	14
17	Funkcije	17
19	Programiranje	19
25	Risanje v Pythonu	25
27	Slikovni vmesnik	27
28	Pogled na splet	28

Programski jezik za uvod v programiranje

Programski jeziki: Fortran, Algol, Lisp, Basic,

Pascal/(Delphi, Kylix), C, Logo, Elica, Squeak / Smalltalk, Java, Javascript, Scheme, ML, Python.

- enostavnost
- koncepti
- prosta dostopnost

Python

Python je potomec programskega jezika **ABC**, ki so ga v drugi polovici osemdesetih let razvili Leo Guerts, Lambert Meertens, Steven Pemberton na CWI v Amsterdamu.

Python je razvil **Guido van Rossum** v začetku devetdesetih.

Zakaj Python? CP4E – Computer Programming for Everybody.

Na Pythonu temelji **Zope** – orodje za upravljanje z gradivi in njegove nadgradnje **Plone**, **ZopeWiki**, ...

Različica 2.3.4 na <http://www.python.org/>

<http://www.activestate.com/Python.plex>

Slikovni vmesnik IDLE.

Zgledi.

Ti zapiski <http://vlado.fmf.uni-lj.si/seminar/02nov04/python.pdf>.

Python – računalo

```
>>> 3 + 4
7

>>> a = 3+4
>>> a
7
>>> b=(a+3)*2
>>> b
20
>>> a, b = b, a
>>> a, b
(20, 7)
>>> A = 10
>>> a, A
(20, 10)

>>> a = b = 0
>>> a, b
(0, 0)
```

Python loči velike in male črke.

Python – knjižnice

```
>>> sin(0.5)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in ?
    sin(0.5)
NameError: name 'sin' is not defined
```

```
>>> import math
>>> math.sin(0.5)
0.47942553860420301
```

```
>>> from math import sin
>>> sin(0.5)
```

```
>>> pow(2,50)
1125899906842624L
```

Podatki v Pythonu

Števila

```
123, 3.14, 3.2e-12  
12345678901234567L, 3+4j  
0177, 0x3afb
```

```
>>> 0177, 0x3afb  
(127, 15099)
```

```
>>> pow(3+4j, 5)  
(-237-3116j)
```

+, *, /, %

e=E, l=L, j=J

osmiška števila začnejo z ničlo 0, šestnajstiška pa z 0x

Logične vrednosti

Python ne pozna `true` in `false` – uporablja 1 in 0. Ničelni/prazni podatki imajo vlogo `false`, ostali `true`.

`x or y`, `x and y` (le delni izračun, če je izid znan)

```
>>> 3 or 4, 5 and 6, 0 or 3, 5 and 0  
(3, 6, 3, 0)
```

`not x`

```
>>> not not 5  
1
```

`<`, `<=`, `>`, `>=`, `==`, `<>`, `!=`, `is`, `is not`, `in`, `not in`
`x | y`, `x ^ y`, `x & y` (po bitih `or`, `xor`, `and`)
`x << n`, `x >> n` (pomik)

```
>>> 1 << 5  
32
```

knjižnica `math`

```
>>> from math import *  
>>> pi  
3.1415926535897931  
>>> e  
2.7182818284590451
```


Nizi

```
"dober dan", 'koko"s', ''
```

```
>>> print 'koko"s'  
koko"s
```

```
>>> m = 'Ljubljana'
```

```
>>> 'a' in m
```

```
1
```

```
>>> z = "a" "b" + "c"
```

```
>>> z
```

```
'abc'
```

```
>>> z = 3          (spremenljivke so v Pythonu 'kazalci')
```

```
>>> z
```

```
3
```

```
>>> m*3
```

```
'LjubljanaLjubljanaLjubljana'
```

```
>>> len(m)
```

```
9
```

```
>>> m[4]
```

```
'l'
```

```
>>> m[3:5]
```

```
'bl'
```

```
>>> m[:4]
```

```
'Ljub'
```

```
>>> m[:-1]
```

```
'Ljubljan'
```

```
>>> m[-1], m[-2]
```

```
('a', 'n')
```

...Nizi

```
>>> napis = """Dober
dan
vsem skupaj"""
>>> napis
'Dober\ndan\nvsem skupaj'
>>> print napis
Dober
dan
vsem skupaj

\n, \r, \v, \t, \f, \0XY, \xXY, \000
\a (bell), \b (backspace), \e (escape)
```

knjižnica regex

Seznami

```
[ ], [ 'b', 'bcd', 3, [ ['x', 1], '3+4' ], a, 7.5 ]  
>>> a = [ 'Nova', 'Gorica' ]  
>>> b = [ 'b', 'bcd', 3, [ ['x', 1], '3+4' ], a, 7.5 ]  
>>> b  
['b', 'bcd', 3, [['x', 1], '3+4'], ['Nova', 'Gorica'], 7.5]  
>>> b[4], b[3], b[-1]  
(['Nova', 'Gorica'], [['x', 1], '3+4'], 7.5)  
>>> b[1:3]  
['bcd', 3]  
>>> len(b)  
6  
>>> 'bcd' in b, 'x' in b  
(1, 0)
```

Seznami – operacije

```
L.append(X), L.sort(), L.index(X), L.reverse()  
del L[i:j]
```

```
>>> a.reverse()  
>>> b  
['b', 'bcd', 3, [['x', 1], '3+4'], ['Gorica', 'Nova'], 7.5]  
>>> b.index(7.5), b.index(a)  
(5, 4)  
>>> del b[3:5]  
>>> b  
['b', 'bcd', 3, 7.5]  
>>> b.sort()  
>>> b  
[3, 7.5, 'b', 'bcd']  
  
>>> a = [1, [2,3]]  
>>> b = a  
>>> c = [1, [2,3]]  
>>> a == b, a == c, a is b, a is c  
(1, 1, 1, 0)  
  
>>> range(4)  
[0, 1, 2, 3]  
>>> range(5,11)  
[5, 6, 7, 8, 9, 10]  
>>> range(7,20,3)  
[7, 10, 13, 16, 19]
```

Slovarji

```
{ } prazen
>>> S = { 'tomo': 'tomaz.pisanski@fmf.uni-lj.si',
          'vlado': 'vladimir.batagelj@uni-lj.si',
          'andrej': 'andrej.mrvar@uni-lj.si' }
>>> S
{'andrej': 'andrej.mrvar@uni-lj.si', 'vlado': 'vladimir.batagelj@uni-lj.si',
 'tomo': 'tomaz.pisanski@fmf.uni-lj.si'}
>>> S['vlado']
'vladimir.batagelj@uni-lj.si'
>>> S.keys()
['andrej', 'vlado', 'tomo']
>>> S.values()
['andrej.mrvar@uni-lj.si', 'vladimir.batagelj@uni-lj.si',
 'tomaz.pisanski@fmf.uni-lj.si']
>>> S['matjaz'] = 'matjaz.zaversnik@fmf.uni-lj.si'
>>> S.has_key('matija'), S.has_key('matjaz')
(0, 1)
>>> S['vlado'] = 'vladimir.batagelj@fmf.uni-lj.si'
>>> S['vlado']
'vladimir.batagelj@fmf.uni-lj.si'
>>> len(S)
4
>>> del S['vlado']
>>> S.has_key('vlado')
0
>>> S[3] = ['a', 395, {1: 'x', 5: 'w'}]
>>> S[3][2][5]
'w'
```

Nabori

So podobni seznamom, le da ne dopuščajo operacij na mestu. Uporabljamo jih, kadar želimo biti gotovi, da se vrednost ne spreminja.

```
>>> a = ( 'a', 'b', 'c', 3, 4, [0, 1], 2004)
>>> a
('a', 'b', 'c', 3, 4, [0, 1], 2004)
>>> a[5]
[0, 1]
>>> len(a)
7
>>> a[2]
'c'
>>> a[2] = 'z'
Traceback (most recent call last):
  File "<pyshell#92>", line 1, in ?
    a[2] = 'z'
TypeError: object doesn't support item assignment
>>> a = a[:2] + ('z',) + a[3:]
>>> a
('a', 'b', 'z', 3, 4, [0, 1], 2004)
```

Branje in izpis, datoteke

Branje in izpis z/v delovno okno sta enostavna:

```
>>> x = input("Vnesi stevilo: ")
Vnesi stevilo: 5
>>> print "Njegov kvadrat je", x*x
Njegov kvadrat je 25
>>>
```

osnovni ukazi za delo z datotekami pa so:

```
lst = open('izpis.lst', 'w')
dat = open('podatki.dat', 'r')

s = dat.read()      cela datoteka
s = dat.read(n)    n zlogov
s = dat.readline()
s = dat.readlines()
lst.write(s)
lst.writelines(L)
lst.close()
```

Krmilni stavki

Stavek nadaljujemo v novo vrsto z \ Tudi vsebina [] se lahko razteza čez več vrstic. Več stavkov v vrstici ločimo s ;

```
>>> a = 3; b = 4
>>> a
3
```

označuje vrstično pojasnilo

Obvezno zamikanje

... Krmilni stavki

```
if p1 :
    stavki1
elif p2 :
    stavki2
else:
    stavki

for i in S:
    stavki1
else:
    stavki2          # "ce ni bil narejen break

while p :
    stavki1
else:
    stavki2

break                # prekini zanko
continue            # na za"cetek zanke
pass                # prazni stavek

try:
    stavki1
except :
    raise stanje
```

...Krmilni stavki

```
>>> a = 'ljubljana'
>>> for i in a :
    if i < 'm' : print i
    else: break
else:
    print a
```

```
l
j
>>> a = 'abceda'
>>> for i in a :
    if i < 'm' : print i
    else: break
else:
    print a
```

```
a
b
e
c
e
d
a
abceda
```

Funkcije

```
def ime(p1,p2,...,pn):  
    global v1, v2, ...,vk  
    stavki  
    return vrednost
```

Imena - pravilo LGB
Local, Global, Built-in

```
return lahko vrne tudi nabor  
    return v1, v2, v3
```

V p_i lahko uporabimo tudi obliko $p=v$ kjer je v privzeta vrednost. To lahko uporabimo tudi pri klicu.

*name nabor prestalih mestnih dejanskih argumentov

**name slovar prestalih imenovanih dejanskih argumentov

Funkcije – lambda

```
lambda p1, p2, ..., pn: izraz
```

```
>>> f = lambda x, y : x*x + y*y
```

```
>>> f(3,4)
```

```
25
```

```
>>> (lambda x, y : x*x + y*y)(3,4)
```

```
25
```

```
>>> apply(f, (3,4))
```

```
25
```

```
>>> map((lambda x: x*x + x + 41), range(41))
```

```
[41, 43, 47, 53, 61, 71, 83, 97, 113, 131, 151, 173,  
197, 223, 251, 281, 313, 347, 383, 421, 461, 503, 547,  
593, 641, 691, 743, 797, 853, 911, 971, 1033, 1097,  
1163, 1231, 1301, 1373, 1447, 1523, 1601, 1681]
```

Programiranje

Pogovorni način

```
>>> print 'dober dan'  
dober dan
```

Uporaba funkcije

```
>>> ime = 'Janez'  
>>> def pozdrav():  
    print 'dober dan', ime
```

```
>>> pozdrav()  
dober dan Janez
```

Pripravimo to na datoteki pozdrav0.py

```
ime = 'Janez'  
def pozdrav():  
    print 'dober dan', ime
```

Program na datoteki

Sedaj lahko zahtevamo

```
>>> import pozdrav0
>>> ime
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in ?
    ime
NameError: name 'ime' is not defined
>>> pozdrav0.ime
'Janez'
>>> pozdrav0.pozdrav()
dober dan Janez
```

Poskusimo znova

```
>>> from pozdrav0 import *
>>> ime
'Janez'
>>> pozdrav()
dober dan Janez
```

...Program na datoteki

Poglejmo še izboljšano različico `pozdrav1.py`

```
#!/usr/bin/python
ime = 'Janez'
def pozdrav():
    """Pozdrav() izpise lep pozdrav
    osebi navedeni v spremenljivki ime.

    V. Batagelj, junij 2004"""
    print 'dober dan', ime
if __name__ == '__main__':
    pozdrav()
else:
    print pozdrav.__doc__
```

...Program na datoteki

```
>>> import pozdravl
Pozdrav()   izpise lep pozdrav
            osebi navedeni v spremenljivki   ime.
```

```
    V. Batagelj, junij 2004
```

```
>>> pozdravl.pozdrav()
dober dan Janez
```

in v ukaznem načinu

```
D:\Python\2.3>python pozdravl.py
dober dan Janez
```

```
D:\Python\2.3>
```


...Program na datoteki

```
>>> import sys
>>> sys.path
['D:\\Python\\2.3\\Lib\\idlelib', 'C:\\WINNT\\system32\\python23.zip',
'D:\\Python\\2.3', 'D:\\Python\\2.3\\DLLs', 'D:\\Python\\2.3\\lib',
'D:\\Python\\2.3\\lib\\plat-win', 'D:\\Python\\2.3\\lib\\lib-tk',
'D:\\Python\\2.3\\lib\\site-packages']
>>> sys.path.append('D:\\vlado\\work\\Python\\seminar')
>>> sys.path
['D:\\Python\\2.3\\Lib\\idlelib', 'C:\\WINNT\\system32\\python23.zip',
'D:\\Python\\2.3', 'D:\\Python\\2.3\\DLLs', 'D:\\Python\\2.3\\lib',
'D:\\Python\\2.3\\lib\\plat-win', 'D:\\Python\\2.3\\lib\\lib-tk',
'D:\\Python\\2.3\\lib\\site-packages', 'D:\\vlado\\work\\Python\\seminar']
>>> import pozdravl
Pozdrav() izpise lep pozdrav
    osebi navedeni v spremenljivki ime.

    V. Batagelj, junij 2004
>>> pozdravl.pozdrav()
dober dan Janez
>>>
```

Po popravkih

```
reload(pozdrav)
```

Podatki so lahko program

Velika moč jezikov, ki temeljijo na tolmačenju je, da je meja med podatki in programom prehodna. V Pythonu to omogočata ukaza `exec` in `eval` (ter `execfile`). Na primer:

```
>>> ukazi = 'b = "ha"; c = (b+"-")*10+b'
>>> ukazi
'b = "ha"; c = (b+"-")*10+b'
>>> exec ukazi
>>> c
'ha-ha-ha-ha-ha-ha-ha-ha-ha-ha'
>>> from math import *
>>> f = 'sin(x)+2*cos(3*x)'
>>> for i in range(10):
        x = i/10.; print i, x, eval(f)
```

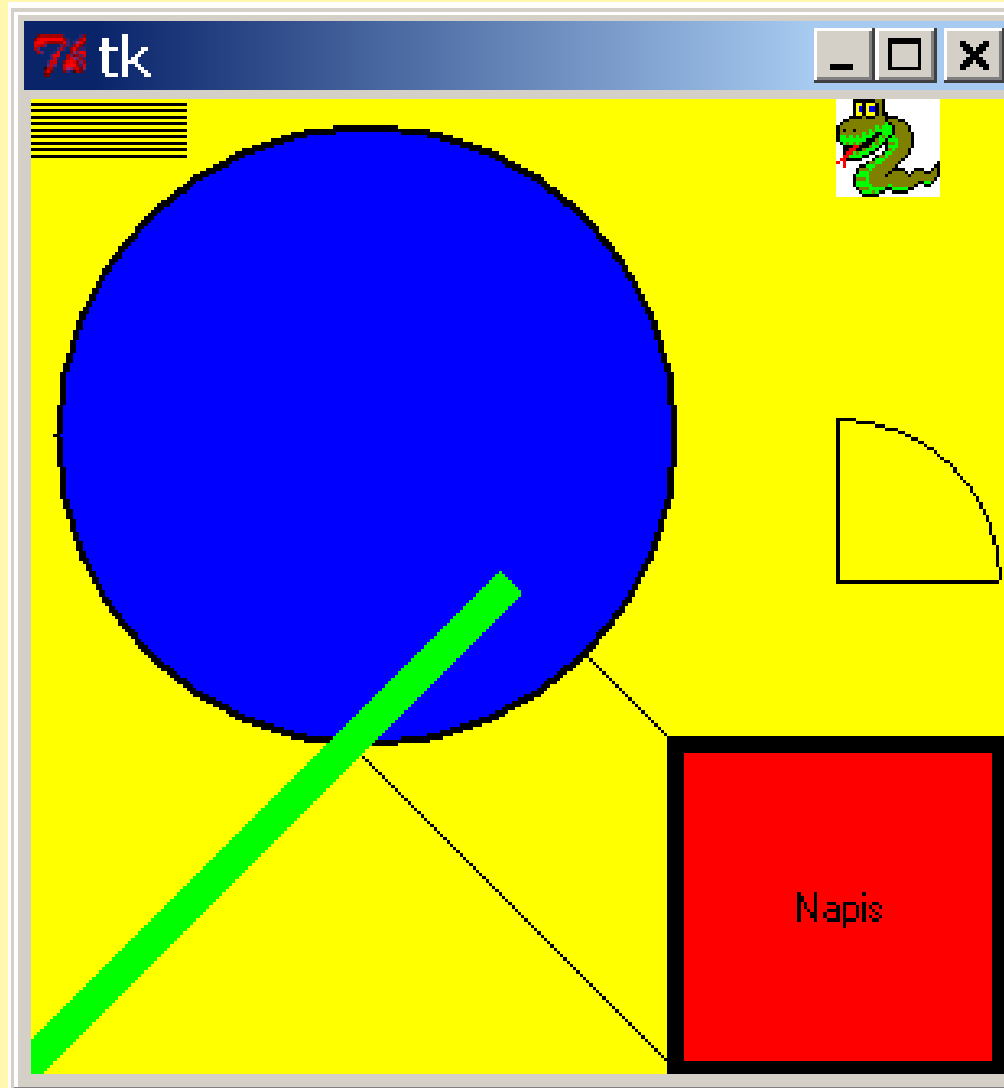
```
0 0.0 2.0
1 0.1 2.0105063949
2 0.2 1.84934056061
3 0.3 1.5387401432
4 0.4 1.11413385126
5 0.5 0.62089994194
6 0.6 0.110238284009
7 0.7 -0.365474521962
8 0.8 -0.757431340183
9 0.9 -1.02481737441
>>>
```

Risanje v Pythonu

```
from Tkinter import *
canvas=Canvas(width=300,height=300,bg='yellow')
canvas.pack(expand=YES,fill=BOTH)
canvas.create_line(100,100,200,200)
canvas.create_line(100,200,200,300)
for i in range(1,20,2):
    canvas.create_line(0,i,50,i)
canvas.create_oval(10,10,200,200,width=2,fill='blue')
canvas.create_arc(200,200,300,100)
canvas.create_rectangle(200,200,300,300,width=5,fill='red')
canvas.create_line(0,300,150,150,width=10,fill='green')
canvas.create_text(250,250,text='Napis')
piton=PhotoImage(file='D:/vlado/work/python/seminar/pyLogo.gif')
canvas.create_image(250,0,image=piton,anchor=NW)
mainloop()
```

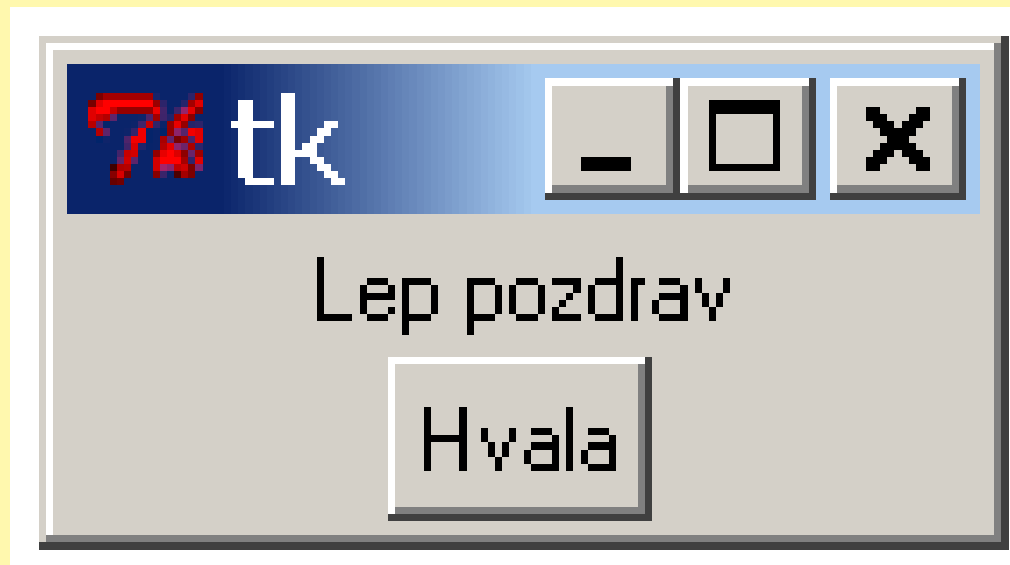
Slika, glej naslednjo prosojnico, ustvari šele ukaz `mainloop()`.

...Risanje v Pythonu



Slikovni vmesnik

```
from Tkinter import *  
win = Frame()  
win.pack()  
Label(win, text='Lep pozdrav').pack(side=TOP)  
Button(win, text='Hvala', command=win.quit).pack(side=BOTTOM)  
win.mainloop()
```



Pogled na spleť


```
import urllib, urlparse, string
url1 = 'http://www.amazon.com/exec/obidos/tg/detail/-/'
book = '0761956042'
url2 = '?v=glance'
url = url1 + book + url2
page = urllib.urlopen(url)
a = page.read()
lstr = '<a name=same_store></a>'
rstr = '<a name=cross_store></a>'
lind = string.index(a,lstr) + len(lstr)
rind = string.index(a,rstr,lind)
b = a[lind:rind]
lref = '<a href='
rref = '>'
linr = string.index(b,lref) + len(lref)
rinr = string.index(b,rref,linr)
url = b[linr:rinr]
b = b[rinr+1:]
rtit = '</a>\n'
rint = string.index(b,rtit)
lint = rint + len(rtit)
titl = b[:rint]
laut = '\n'
lina = string.index(b,laut,lint)
auth = b[lint:lina]
b = b[lina+1:]
```

... Pogled na splet

```
>>> url
'/exec/obidos/tg/detail/-/0761963391?v=glance'
>>> titl
'Social Network Analysis: A Handbook'
>>> auth
'by John P Scott'
>>> b
'<li class=small>\n<a href=/exec/obidos/tg/deta
il/-/0195160371?v=glance>Theories of Communicat
ion Networks</a>\nby Peter R. Monge, Noshir S.
Contractor\n<li class=small>\n<a href=/exec/obi
dos/tg/detail/-/0803943032?v=glance>Advances in
Social Network Analysis : Research in the Socia
l and Behavioral Sciences (SAGE Focus Editions)
</a>\nby Stanley Wasserman (Editor), Joseph Gal
askiewicz (Editor)\n<li class=small>\n<a href=
/exec/obidos/tg/detail/-/0393041425?v=glance>Six
Degrees: The Science of a Connected Age</a>\nby
Duncan J. Watts\n<li class=small>\n<a href=/exe
c/obidos/tg/detail/-/0393041530?v=glance>Nexus:
Small Worlds and the Groundbreaking Science of
Networks</a>\nby Mark Buchanan\n</ul>\n'
>>>
```

Amazon

Introducing Social Networks (Introducing Statistical Methods series)
by [Alain Degenne](#), [Michel Forse](#)



List Price: \$41.95
Price: **\$41.95** & This item ships for **FREE with Super Saver Shipping**. [See details.](#)

Availability: Usually ships within 24 hours

21 used & new from **\$40.16**


[Look inside this book](#) Edition: Paperback

Other Editions:	List Price:	Our Price:	Other Offers:
Hardcover	\$115.00	\$115.00	Order it used!

[See more product details](#)

Better Together

Buy this book with [Social Network Analysis](#) by Stanley Wasserman, et al today!



Buy Together Today: \$74.90

Customers who bought this book also bought:

- [Social Network Analysis: A Handbook](#) by John P Scott ([Rate it](#))
- [Theories of Communication Networks](#) by Peter R. Monge, Noshir S. Contractor ([Rate it](#))
- [Advances in Social Network Analysis : Research in the Social and Behavioral Sciences \(SAGE Focus Editions\)](#) by Stanley Wasserman (Editor), Joseph Galaskiewicz (Editor) ([Rate it](#))
- [Six Degrees: The Science of a Connected Age](#) by Duncan J. Watts ([Rate it](#))
- [Nexus: Small Worlds and the Groundbreaking Science of Networks](#) by Mark Buchanan ([Rate it](#))