



# Moduli – zbirke funkcija

# Moduli – zbirke funkcija

Python ima velik broj ugrađenih funkcija.

Moduli su zbirke funkcija, u kojima su funkcije grupirane na temelju nekih zajedničkih svojstava. Module koristimo prema potrebi.

<https://python.readthedocs.io/en/stable/py-modindex.html>



## Python Module Index

[\\_](#) | [a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#) | [x](#) | [z](#)

<a href="#">__future__</a>	Future statement definitions
<a href="#">__main__</a>	The environment where the top-level script is run.
<a href="#">_dummy_thread</a>	Drop-in replacement for the <code>_thread</code> module.
<a href="#">_thread</a>	Low-level threading API.
<b>a</b>	
<a href="#">abc</a>	Abstract base classes according to PEP 3119.
<a href="#">aifc</a>	Read and write audio files in AIFF or AIFC format.
<a href="#">argparse</a>	Command-line option and argument parsing library.
<a href="#">array</a>	Space efficient arrays of uniformly typed numeric values.
<a href="#">ast</a>	Abstract Syntax Tree classes and manipulation.
<a href="#">asynchat</a>	Support for asynchronous command/response protocols.
<a href="#">asyncio</a>	Asynchronous I/O, event loop, coroutines and tasks.
<a href="#">asyncore</a>	A base class for developing asynchronous socket handling services.
<a href="#">atexit</a>	Register and execute cleanup functions.
<a href="#">audioop</a>	Manipulate raw audio data.
<b>b</b>	
<a href="#">base64</a>	RFC 3548: Base16, Base32, Base64 Data Encodings; Base85 and Ascii85
<a href="#">bdb</a>	Debugger framework.
<a href="#">binascii</a>	Tools for converting between binary and various ASCII-encoded binary representations.
<a href="#">binhex</a>	Encode and decode files in binhex4 format.
<a href="#">bisect</a>	Array bisection algorithms for binary searching.
<a href="#">builtins</a>	The module that provides the built-in namespace.
<a href="#">bz2</a>	Interfaces for bzip2 compression and decompression.
<b>c</b>	
<a href="#">calendar</a>	Functions for working with calendars, including some emulation of the Unix <code>cal</code> program.
<a href="#">cgi</a>	Helpers for running Python scripts via the Common Gateway Interface.
<a href="#">cgi.tb</a>	Configurable traceback handler for CGI scripts.
<a href="#">chunk</a>	Module to read IFF chunks.
<a href="#">cmath</a>	Mathematical functions for complex numbers.
<a href="#">cmd</a>	Build line-oriented command interpreters.
<a href="#">code</a>	Facilities to implement read-eval-print loops.
<a href="#">codecs</a>	Encode and decode data and streams.
<a href="#">codeop</a>	Compile (possibly incomplete) Python code.
<a href="#">collections</a>	Container datatypes
<a href="#">colorsys</a>	Conversion functions between RGB and other color systems.
<a href="#">compileall</a>	Tools for byte-compiling all Python source files in a directory tree.

# Moduli – zbirke funkcija

<b>m</b>	
<code>macpath</code>	<i>Mac OS 9 path manipulation functions.</i>
<code>mailbox</code>	<i>Manipulate mailboxes in various formats</i>
<code>mailcap</code>	<i>Mailcap file handling.</i>
<code>marshal</code>	<i>Convert Python objects to streams of bytes and back (with different constraints).</i>
<code>math</code>	<i>Mathematical functions (sin() etc.).</i>
<code>mimetypes</code>	<i>Mapping of filename extensions to MIME types.</i>
<code>mmap</code>	<i>Interface to memory-mapped files for Unix and Windows.</i>
<code>modulefinder</code>	<i>Find modules used by a script.</i>
<code>msilib (Windows)</code>	<i>Creation of Microsoft Installer files, and CAB files.</i>
<code>msvcrt (Windows)</code>	<i>Miscellaneous useful routines from the MS VC++ runtime.</i>
<code>+ multiprocessing</code>	<i>Process-based parallelism.</i>
<b>n</b>	
<code>netrc</code>	<i>Loading of .netrc files.</i>
<code>nis (Unix)</code>	<i>Interface to Sun's NIS (Yellow Pages) library.</i>
<code>nntplib</code>	<i>NNTP protocol client (requires sockets).</i>
<code>numbers</code>	<i>Numeric abstract base classes (Complex, Real, Integral, etc.).</i>
<b>o</b>	
<code>operator</code>	<i>Functions corresponding to the standard operators.</i>
<code>optparse</code>	<b>Deprecated:</b> <i>Command-line option parsing library.</i>
<code>+ os</code>	<i>Miscellaneous operating system interfaces.</i>
<code>ossaudiodev (Linux, FreeBSD)</code>	<i>Access to OSS-compatible audio devices.</i>

# Moduli – zbirke funkcija

r	
<code>random</code>	<i>Generate pseudo-random numbers with various common distributions.</i>
<code>re</code>	<i>Regular expression operations.</i>
<code>readline (Unix)</code>	<i>GNU readline support for Python.</i>
<code>reprlib</code>	<i>Alternate repr() implementation with size limits.</i>
<code>resource (Unix)</code>	<i>An interface to provide resource usage information on the current process.</i>
<code>rlcompleter</code>	<i>Python identifier completion, suitable for the GNU readline library.</i>
<code>runpy</code>	<i>Locate and run Python modules without importing them first.</i>
s	
<code>sched</code>	<i>General purpose event scheduler.</i>
<code>secrets</code>	<i>Generate secure random numbers for managing secrets.</i>
<code>select</code>	<i>Wait for I/O completion on multiple streams.</i>
<code>selectors</code>	<i>High-level I/O multiplexing.</i>
<code>shelve</code>	<i>Python object persistence.</i>
<code>shlex</code>	<i>Simple lexical analysis for Unix shell-like languages.</i>
<code>shutil</code>	<i>High-level file operations, including copying.</i>
<code>signal</code>	<i>Set handlers for asynchronous events.</i>
<code>site</code>	<i>Module responsible for site-specific configuration.</i>
<code>smtplib</code>	<i>A SMTP server implementation in Python.</i>
<code>smtpd</code>	<i>SMTP protocol client (requires sockets).</i>
<code>sndhdr</code>	<i>Determine type of a sound file.</i>
<code>socket</code>	<i>Low-level networking interface.</i>
<code>socketserver</code>	<i>A framework for network servers.</i>
<code>spwd (Unix)</code>	<i>The shadow password database (getspnam() and friends).</i>
<code>sqlite3</code>	<i>A DB-API 2.0 implementation using SQLite 3.x.</i>
<code>ssl</code>	<i>TLS/SSL wrapper for socket objects</i>
<code>stat</code>	<i>Utilities for interpreting the results of os.stat(), os.lstat() and os.fstat().</i>
<code>statistics</code>	<i>mathematical statistics functions</i>
<code>string</code>	<i>Common string operations.</i>
<code>stringprep</code>	<i>String preparation, as per RFC 3453</i>
<code>struct</code>	<i>Interpret bytes as packed binary data.</i>

# Modul math

Modul se prije korištenja treba uvesti s naredbom **import naziv\_modula**  
Funkcija **sqrt** (korijen) je iz **math** modula.

```
>>> import math
>>> math.sqrt(255)
15.968719422671311
>>> math.sqrt(64)
8.0
>>> math.sqrt(2)
1.4142135623730951
>>> math.sqrt(5)
2.23606797749979
>>>
```

# Modul math

Drugi način kako možemo uvesti funkciju iz modula je s naredbom **from naziv\_modula import** funkcija1, funkcija2,...

```
>>> from math import sqrt, sin, cos, tan, atan, fabs
>>> sqrt(25)
5.0
>>> sin(13)
0.4201670368266409
>>> cos(25)
0.9912028118634736
>>> tan(128)
-1.0406148914328552
>>> atan(0.64)
0.5693131911006619
>>> fabs(-25)
25.0
>>>
```

Funkcija **fabs** vraća  
apsolutnu vrijednost broja.

# Modul math

Ako se žele uvesti sve funkcije iz nekog modula koristimo naredbu:

```
>>> from math import *
>>> sqrt(81)
9.0
>>> sin(45)
0.8509035245341184
>>> sin(radians(45))
0.7071067811865475
>>> cos(60)
-0.9524129804151563
>>> cos(radians(60))
0.5000000000000001
>>> tan(37.456789)
-0.24718005688289044
>>> tan(radians(37.456789))
0.7661294557649995
>>>
```

# Modul math

```
*Izracunaj hor duljinu.py - D:/_0_SKOLA - GTS/_1_PREDAVANJA_GEOSKOLA/1_GEOINFO_GIS_PIS_ACAD_PYTHON/2r/_PYTHON/MODULI/Izracunaj hor duljinu.py (3.6.3)*
File Edit Format Run Options Window Help
# Program: ASlavicek
# Izračunaj horizontalnu duljinu ako je mjereno:
# Vertikalni kut i kosa duljina.
from math import *
print('Unesi vertikalni kut u sljedećem formatu: stup,minute,sek')
stup,mnt,sek=eval(input('Unesi vertikalni kut - npr 89,12,46 = '))
ver_kut=stup+mnt/60+sek/3600
d_kosa=float(input('Unesi duljinu= '))
d_hor=d_kosa*sin(radians(ver_kut))
print('Horizontalna duljina iznosi = ',round(d_hor,3))
```



# Modul math

```
Izracunaj hor duljinu_ukupna hor_duljina.py - D:/_0_SKOLA - GTS/_1_PREDAVANJA_GEOSKOLA/1_GEOINFO_GIS_PIS_ACAD_PYTHON/2r/_PYTHON/MODULI/Izracunaj hor duljinu_ukupn...
File Edit Format Run Options Window Help
# Program: ASlavicek
# Izračunaj horizontalnu duljinu ako je mjereno:
# Vertikalni kut i kosa duljina.
# Omogući u programu višekratno računanje.
# Izračunaj ukupnu horizontalnu duljinu.
# -----
from math import *
duljina=int(input('Koliko hor. duljina treba izračunati? '))
h_duljina=0
for i in range(duljina):
    print('Unesi vertikalni kut u sljedećem formatu: stup,minute,sek')
    stup,mnt,sek=eval(input('Unesi vertikalni kut - npr 89,12,46 = '))
    ver_kut=stup+mnt/60+sek/3600
    d_kosa=float(input('Unesi duljinu= '))
    d_hor=d_kosa*sin(radians(ver_kut))
    print('Horizontalna duljina iznosi = ',round(d_hor,3))
    h_duljina=h_duljina+d_hor
print('-----')
print('Ukupna suma horizontalnih duljina iznosi', round(h_duljina,3))
Ln: 21 Col: 0
```



# Modul math

## TAHIMETRIJA

Računanje koordinata točaka polarnom metodom - "online obrazac".

Autor: Armando Slaviček, 2021.; mail: [slavicek@geoskola.hr](mailto:slavicek@geoskola.hr)



### RAČUNANJE KOORDINATA TOČAKA POLARNOM METODOM

A. Slaviček

Staja -lište	Broj točk e	Orijentacija ( $\beta$ )			Vertikalni kut			$d_{kosa}$ (m)	$d_{hor} = d_k * \sin V_k$	$v_i = v + \beta_i$			$\Delta E_i = d_i * \sin v_i$	$\Delta N_i = d_i * \cos v_i$	$\Delta h = d_{kl} * \cos V_k$	$E_n = E_{ST} + \Delta E_i$	$N_n = N_{ST} + \Delta N_i$	$H_n = H_{ST} + \Delta h + i - s$	Broj točke
		°	'	"	°	'	"			°	'	"							
PT1															5,567,981.03	5,074,589.70	179.76	PT1	
Ori.	PT2	0	00	00						158	30	48			5,567,997.24	5,074,548.52	179.12	PT2	
$i =$	5	7	47	55	89	22	32	27.052	27.050	166	18	43	6.40	-26.28	0.29	5,567,987.43	5,074,563.42	179.75	5
1.70	6	297	42	10	88	58	00	18.984	18.981	96	12	58	18.87	-2.06	0.34	5,567,999.90	5,074,587.64	179.80	6
$s =$	7	294	19	55	88	57	27	18.483	18.480	92	50	43	18.46	-0.92	0.34	5,567,999.49	5,074,588.78	179.80	7
2.00	8	293	07	40	87	56	32	17.722	17.711	91	38	28	17.70	-0.51	0.64	5,567,998.73	5,074,589.19	180.10	8
	9	296	01	13	87	44	47	15.823	15.811	94	32	01	15.76	-1.25	0.62	5,567,996.79	5,074,588.45	180.08	9
	10	298	29	38	88	43	49	15.476	15.472	97	00	26	15.36	-1.89	0.34	5,567,996.39	5,074,587.81	179.80	10
<a href="mailto:slavicek@geoskola.hr">slavicek@geoskola.hr</a>																			



© Armando Slaviček, 2021.

Napiši programe koji će računati:

- Horizontalnu duljinu
- Smjerni kut
- Koordinatne razlike ( $\Delta E$ ,  $\Delta N$ )
- $\Delta h$
- E, N i H neke točke

# Moduli random

Funkcije za generiranje slučajnih brojeva.

**randint (a,b)** – vraća slučajni broj n koji je  $a \leq n \leq b$

```
>>>
>>> from random import *
>>> randint(0,20)
1
>>> randint(0,20)
2
>>> randint(0,20)
11
>>> randint(0,20)
17
>>> # slučajni cijeli broj u intervalu [0,20]
>>> .
```



# Modul random

**random ()** – vraća slučajni realni broj  $n$  koji je  $0.0 \leq n < 1.0$

```
>>> random ()
0.84310542597082
>>> random ()
0.49493221391364595
>>> random ()
0.3689900773587699
>>> random ()
0.36669170020503505
>>> random ()
0.2127785575693547
>>>
```

# Moduli – zbirke funkcija

**uniform (a,b)** – vraća slučajni realni broj  $n$  koji je  $a \leq n \leq b$   
ako je  $a \leq b$  ili je  $b \leq n \leq a$  ako je  $b < a$

```
>>> uniform(10, 50)
46.754826497980545
>>> uniform(10, 50)
16.110916946170857
>>> uniform(10, 50)
13.850432684940298
>>> uniform(10, 50)
43.53364693269026
>>> uniform(10, 50)
26.143487363386818
>>>
```



# Moduli – zbirke funkcija

**sample (N,k)** – vraća listu od k jedinstvenih elemenata iz liste N

```
>>> # definicija liste koja sadrži 50 brojeva
>>> # od 0 do 49
>>> brojevi = [i for i in range (50)]
>>> sample (brojevi,7)
[46, 43, 21, 12, 38, 24, 40]
>>> sample (brojevi,3)
[34, 39, 4]
>>> sample (brojevi,15)
[26, 1, 6, 27, 16, 18, 8, 48, 37, 17, 35, 44, 30, 2, 20]
>>> sample (brojevi,21)
[36, 8, 10, 3, 42, 37, 9, 0, 14, 38, 17, 44, 20, 35, 28, 33, 43,
22, 46, 29, 39]
>>> |
```



# Modul random

```
# ASlavicek
# Pogodi broj
from random import *
print('Pogodi broj u intervalu: 0-20')
zamisljen_broj=randint(0,20)
broj=int(input('Traženi broj je = '))

while zamisljen_broj != broj:
    print('Nisi pogodio')
    broj=int(input('Trazeni broj je = '))

print('Pogodio si!, Bravo!!!')
```



# Modul random

```
# ASlavicek
# Pogodi broj
from random import *
print('Pogodi broj u intervalu: 0-20')
zamisljen_broj=randint(0,20)
broj=int(input('Traženi broj je = '))

while zamisljen_broj != broj:
    print('Nisi pogodio')
    broj=int(input('Trazeni broj je = '))

print('Pogodio si!, Bravo!!!')
```





# Modul random

```
# ASlavicek
# Pogodi broj. Omogući korisniku pomoć pri rješavanju.
# Je li traženi broj manji ili veći od traženog?
from random import *
print('Pogodi broj u intervalu: 0-100')
zamisljen_broj=randint(0,100)
broj=int(input('Traženi broj je = '))

while zamisljen_broj != broj:
    print('Nisi pogodilo')
    if zamisljen_broj>broj:
        print("Broj je veći.")
    else:
        print("Broj je manji.")

    broj=int(input('Trazeni broj je = '))

print('Pogodio si!, Bravo!!!')
```





*Pravoslavijem na pažnji.*

*A. Slavicek.*

