# Python
## Modul math

Armando Slaviček

# Modul math

Matematičke funkcije u Pythonu mogu se koristiti uz prethodnu najavu.

Te se naredbe nalaze u datoteci imena math.py.

Naredbe učitavamo u programski kod naredbom *from math import *.*

# Funkcija sqrt(x) vraća realni broj koji predstavlja drugi korijen iz pozitivnog realnog broja x.

```
Python 3.6.0 Shell

File  Edit  Shell  Debug  Options  Window  Help

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from math import *
>>> sqrt(16)
4.0
>>> sqrt(3)
1.7320508075688772
>>> sqrt(153.654)
12.39572506955523
>>> sqrt(-153.654)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    sqrt(-153.654)
ValueError: math domain error
>>>
```

Armando Slaviček

3

Funkcija **pow(x,y)** vraća realni broj koji predstavlja potenciju realnog broja x na realni broj y.



```
Python 3.6.0 Shell

File   Edit   Shell   Debug   Options   Window   Help

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> pow(2,3)
8
>>> pow(2,10)
1024
>>> pow(2.3,10.7)
7421.434270594758
>>> |
```

# Funkcija trunc(x) vraća prirodni broj koji je jednak cijelomu dijelu realnog broja x.

```
Python 3.6.0 Shell

File  Edit  Shell  Debug  Options  Window  Help

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from math import *
>>> trunc(16.456789)
16
>>> trunc(16.4)
16
>>> trunc(16)
16
>>> |
```

# Vrijednost pi unaprijed je definirana.



```
Python 3.6.0 Shell

File  Edit  Shell  Debug  Options  Window  Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> from math import *
>>> pi
3.141592653589793
>>> PI
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    PI
NameError: name 'PI' is not defined
>>> Pi
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    Pi
NameError: name 'Pi' is not defined
>>> # ispis većeg broja decimala broja pi
>>> print('%.25f' % (pi))
3.1415926535897931159979635
>>> print('%.50f' % (pi))
3.14159265358979311599796346854418516159057617187500
>>> |
```

# Funkcije sin(x), cos (x), tan(x), atan(x)

```
Python 3.6.0 Shell

File  Edit  Shell  Debug  Options  Window  Help

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016,
06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for
information.
>>> from math import *
>>> sin (30)
-0.9880316240928618
>>> # računa sin od x u radijanima
>>> sin(radians(30))
0.4999999999999994
>>> cos(45)
0.5253219888177297
>>> cos(radians(45))
0.7071067811865476
>>> tan(56)
-0.6112736881917098
>>> tan(radians(56))
1.482560968512703
>>> atan(-0.6112736881917098)
-0.5486677646162783
>>>
```

## 9.2.3. Trigonometric functions

math. **acos**(*x*)
Return the arc cosine of *x*, in radians.

math. **asin**(*x*)
Return the arc sine of *x*, in radians.

math. **atan**(*x*)
Return the arc tangent of *x*, in radians.

math. **atan2**(*y*, *x*)
Return `atan(y / x)`, in radians. The result is between `-pi` and `pi`. The vector in the plane from the origin to point `(x, y)` makes this angle with the positive X axis. The point of `atan2()` is that the signs of both inputs are known to it, so it can compute the correct quadrant for the angle. For example, `atan(1)` and `atan2(1, 1)` are both `pi/4`, but `atan2(-1, -1)` is `-3*pi/4`.

math. **cos**(*x*)
Return the cosine of *x* radians.

math. **hypot**(*x*, *y*)
Return the Euclidean norm, `sqrt(x*x + y*y)`. This is the length of the vector from the origin to point `(x, y)`.

math. **sin**(*x*)
Return the sine of *x* radians.

math. **tan**(*x*)
Return the tangent of *x* radians.

## 9.2.4. Angular conversion

math. **degrees**(*x*)
Convert angle *x* from radians to degrees.

math. **radians**(*x*)
Convert angle *x* from degrees to radians.

# Izračunaj pomoću trigonometrijskih funkcija

**sin**
```
>>> sin(radians(37))
0.6018150231520483
>>>
>>> degrees(asin(0.6018150231520483))
37.0
```

**tan**
```
>>> tan(radians(123))
-1.5398649638145827
>>> degrees(atan(-1.5398649638145827))
-57.0
>>>
```

**cos**
```
>>> cos(radians(74))
0.27563735581699916
>>> degrees(acos(0.27563735581699916))
74.0
>>>
```
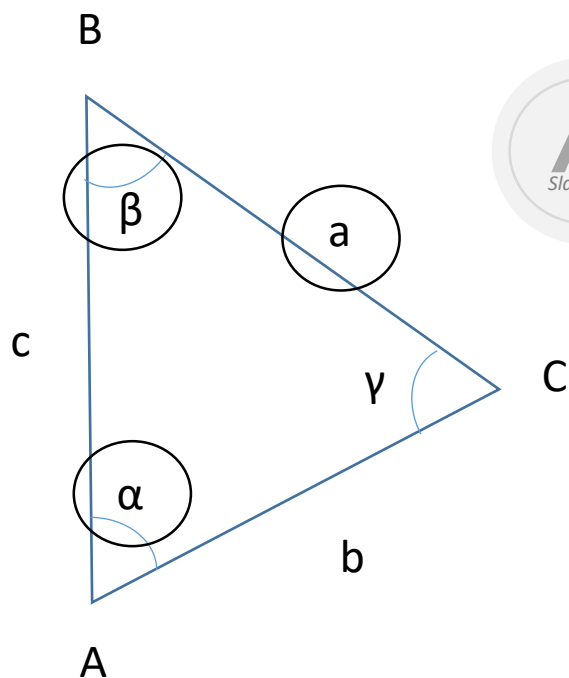
**atan2**
```
>>> degrees(atan2(45.12,23.69))
62.29853627752536
```

# Računanje trokuta po sinusovom poučku

Napiši program u Pythonu koji će iz zadanih veličina izračunati stranice b i c.



*Zadatak 1.*

α = 40°33'29''
β = 69°54'22''
γ = 69°32'07''
a = 230.25

| a : b = sin α: sin β | → | b = (a · sin β) / sin α |

| a : c = sin α: sin γ | → | c = (a · sin γ) / sin α |

*Rješenje:*
*b = 332.56*
*c = 331.76*

*Zadatak 2.*

α = 29°17'02''
β = 56°56'05''
γ = 93°46'53''
a = 331.77

*Rješenje:*
*b = 568.43*
*c = 676.80*

*Zadatak 3.*

α = 40°16'29''
β = 85°33'17''
γ = 54°10'14''
a = 438.71

*Rješenje:*
*b = 676.60*
*c = 550.22*

# Trilateracija
## (Lučni presjek)

Tb

$q$

$d_b$

$d_{AB}$

$h$

$p$

T($y_T$,$x_T$)

$d_a$

Ta

Napiši program u Pythonu koji će izračunati koordinate točke T.
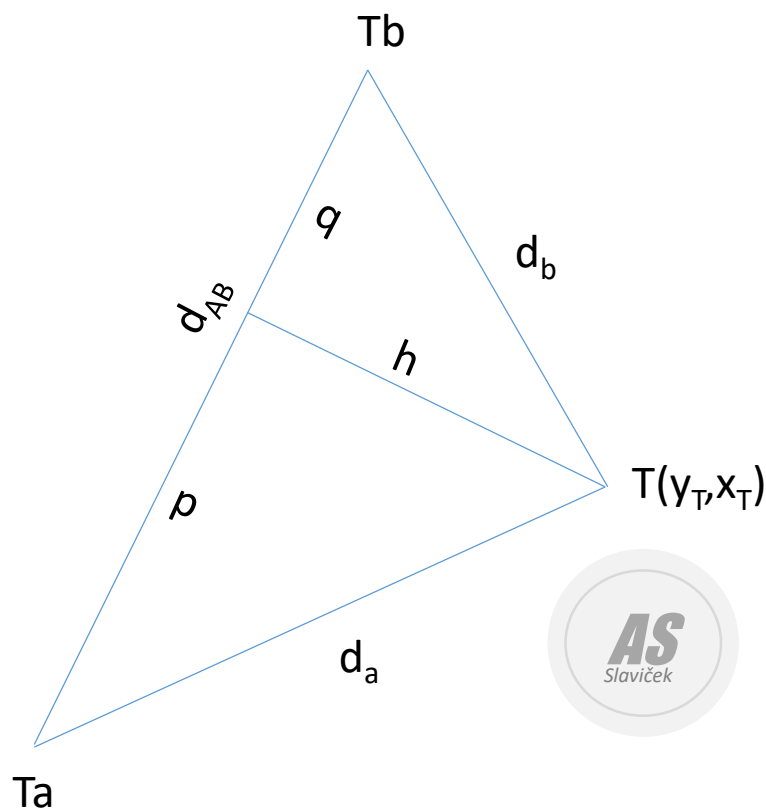
```
# Točka T određena je lučnim presjekom, tj. odmjeranjem dužina da i db
# od točaka Ta i Tb čije su koordinate poznate.
# Unesi dužine: da i db.
# Unesi: Točka A(ya,xa,) i točke B(yb,xb)
# dy=yb-ya, dx=xb-xa
# dAB=sqrt(dy**2+dx**2)
# (a+b)/2=1/2 sqrt((yb-ya)**2+(xb-xa)**2)
# (a-b)/2=(da+db)*(da-db)/2dAB
# a= (a+b)/2+(a-b)/2
# b=(a+b)/2-(a-b)/2
# h=sqrt(da**2-a**2) ili sqrt(db**2-b**2)
# unesi predznak za h - za h je predznak pozitivan ako točka T leži
# lijevo od pravca TaTb, a negativan ako leži desno od njega
# p=(yb-ya)/(a+b)
# q=(xb-xa)/(a+b)
# yt=ya+a*p-h*q
# xt=xa+a*q+h*p
```

LUČNI PRESJEK - PRIMJER
ya= 8116.03, xa= 2358.01
yb= 6942.00, xb= 1625.80
da= 755.77
db= 975.10
Točka se nalazi lijevo od pravca TaTb? Lijevo=1, Desno= -1 =-1
p= -0.8485042326072416
q = -0.5291885932704861
a= 554.638
b= 829.009
yt = 7373.739
xt = 2500.112