

Moduł math w Pythonie

M@я3k Pцđ€£kØ

Programowanie w Pythonie

Moduł math

- Język Python pozwala na korzystanie z modułów, które zawierają wyspecjalizowane funkcje do wykorzystania w danym programie.
 - Poszerzają możliwości standardowego poziomu Pythona
- Moduł **math** zawiera różne funkcje i operacje matematyczne przydatne w różnych działaniach.
- W celu użycia go należy użyć na początku programu polecenia `import math`.

```
'''  
Funkcje matematyczne  
'''  
import math
```

Moduł math

```
'''  
Funkcje matematyczne  
'''  
import math  
print ('Funkcje matematyczne')  
  
x = 6.5  
  
print (math.floor (x))
```

Spis treści

- Zaokrąglanie
- Potęgi i logarytmy
- Funkcje trygonometryczne
- Konwersja stopni i radianów
- Silnia
- Wartość bezwzględna
- Największy wspólny dzielnik 2 liczb całkowitych
- Stałe w module math

Zaokrąglanie systemowe

- Język Python posiada funkcję `round()`, która zaokrągla do najbliższej liczby całkowitej.

```
print(round(3.14159))           #3
```

```
print(round(3.7))               #4
```

- Funkcja `round()` może przyjąć dwa argumenty: liczbę do zaokrąglenia i opcjonalnie liczbę miejsc po przecinku

```
print(round(3.14159, 2))        #3.14
```

```
print(round(2.718281828459, 4)) #2,7183
```

- Granicą jest połowa 0.5, zaokrąglane w górę.

```
print(round(3.5))               #4
```

```
print(round(3.499999999999999)) #3
```

Zaokrąglanie w górę i w dół

- Moduł **math** pozwala na ustalenie jak ma wyglądać zaokrąglanie.
- Funkcja `math.ceil(x)` zaokrągla w górę do najbliższej liczby całkowitej lub równej x
- Funkcja `math.floor(x)` zaokrągla w dół do najbliższej liczby całkowitej lub równej x

```
print (math.ceil (3.14159) )    #4
```

```
print (math.floor (3.14159) )   #3
```

Zaokrąglanie

```
print(round(3.14159))  
print(round(3.7))  
  
print(round(3.14159, 2))  
print(round(2.718281828459, 4))  
  
print(round(3.5))  
print(round(3.499999999999999))
```

```
3  
4  
  
3.14  
2.7183  
  
4  
3
```

```
print (math.ceil (3.14159))  
print (math.floor (3.14159))
```

```
4  
3
```

Potęgi i logarytmy

- Eksponenta

`math.exp(x)` $\#e^{*}x = e^x$

- Logarytm naturalny

`math.log(x)` $\#\ln(x)$

- Logarytm dziesiętny

`math.log10(x)` $\#\log_{10}(x)$

- Logarytm o podstawie 2

`math.log2(x)` $\#\log_2(x)$

- Potęga x^y

`math.pow(x, y)` $\#x^y$

- Pierwiastek kwadratowy

`math.sqrt(x)` $\#\sqrt{x}$

Potęgi i logarytmy

```
print (math.exp(1))  
print (math.log (2.712))  
print (math.log10 (1000))  
print (math.log2 (32))  
print ((math.sqrt(81)))
```

```
2.718281828459045  
0.9976863700781492  
3.0  
5.0  
9.0
```

Funkcje trygonometryczne

- Możliwe jest użycie funkcji sinus, cosinus, tangens oraz ich arcusów i wersji hiperpoblicznych.
- Argumentem jest stopień podany w radianach.

Postać w Pythonie	Funkcja trygonometryczna
<code>math.sin(x)</code>	Sinus
<code>math.cos(x)</code>	Cosinus
<code>math.tan(x)</code>	Tangens

Konwersja stopni i radianów

- Ze stopni należy go przeliczyć ze wzoru:

$$\text{Kąt w radianach} = \text{kąt w stopniach} * \frac{\pi}{180}$$

- Zamiana stopni na radiany.

`math.radians(x)`

- Zamiana radiany na stopnie.

`math.degrees(x)`

Funkcje trygonometryczne

```
stopien = 90  
radian = stopien * math.pi/180  
  
print (math.cos(radian))  
  
print (math.sin(radian))  
  
print (math.tan(radian))
```

```
6.123233995736766e-17  
  
1.0  
  
1.633123935319537e+16
```

```
stopien = 90  
  
print (math.cos(math.radians(stopien)))  
  
print (math.sin(math.radians(stopien)))  
  
print (math.tan(math.radians(stopien)))
```

Wartość bezwzględna

- Wartość bezwzględna dla liczb całkowitych, zmiennoprzecinkowych i zespolonych

`abs ()`

- Dla liczb zmiennoprzecinkowych, można użyć funkcji `math.fabs()`, którą zoptymalizowano pod kątem większej precyzji.
 - Nie działa na liczbach zespolonych

`math.fabs (x)`

Wartość bezwzględna

```
print (abs (100))  
print (abs (-100))  
  
print (abs (36.1245))  
print (abs (-36.1245))  
  
print (math.fabs (-144))  
print (math.fabs (-144.1234567890))
```

```
100  
100  
  
36.1245  
36.1245  
  
144.0  
144.123456789
```

Silnia

- Silnia

- $x = 1 * 2 * 3 * \dots * x$

`math.factorial(x)` $\#x!$

$10! = 3\,628\,800$

```
print (math.factorial(10) )
```

```
3628800
```

Największy wspólny dzielnik 2 liczb całkowitych

- Największy wspólny dzielnik 2 liczb całkowitych

`math.gcd(x, y)`

```
print (math.gcd(15, 20))  
print (math.gcd(15, 24))  
print (math.gcd(15, 60))
```

```
5  
3  
15
```

Ćwiczenie 1

1. Równanie liniowe z jedną niewiadomą
 - Napisz program rozwiązujący równanie : $ax + b = 0$
 - Program wczytuje **a** i **b**, a potem oblicza wartość **x**.
2. Napisz program wczytujący liczbę zmiennoprzecinkową i podający między jakimi liczbami całkowitymi się ona znajduje.
3. Napisz program wczytujący liczbę **x** i wyliczający dla niej silnię **x!**.

Ćwiczenie 2

1. Napisz program wyliczający największy wspólny dzielnik dla następujących par liczb:
 - a) 36 i 225
 - b) 120 i 175
 - c) 32 i 81
 - d) 16 i 90
 - e) 121 i 330
 - f) 75 i 500
2. Wczytaj przyprostokątne **a** i **b**. Następnie oblicz przeciwprostokątną **c** korzystając z twierdzenia Pitagorasa: $c = \sqrt{a^2 + b^2}$

Ćwiczenie 3

1. Napisz program wyliczający cotangens kąta 45 stopni.
 - Do obliczeń wykorzystaj właściwości trygonometryczne **tangens** i **cotangens**.
 - Stopnie przelicz na radiany.
2. Napisz program sprawdzający wzór na jedynkę trygonometryczną.
3. Masz trójkąt prostokątny z którego znasz jeden bok przyprostokątny **a** i przyległy do niego kąt ostry **alfa**. Oblicz drugi bok przyprostokątny, przeciwprostokątną i pole trójkąta.

Powtórzenie

1. Co umożliwia moduł `math`?
2. Jak zrealizować zaokrąglanie liczb?
3. Jakimi metodami można uzyskać pierwiastki?
4. Jakie funkcje trygonometryczne realizuje moduł `math`?
5. Jak przeliczyć stopnie kąta na radiany i odwrotnie?
6. Jak uzyskać moduł liczby?