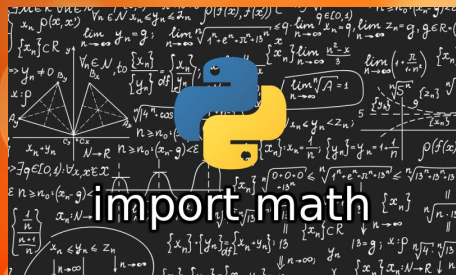


Навчання заради успіху

Мова програмування PYTHON



Модуль Math – математика в Python



Вчитель інформатики В. Д. Партіка



Бібліотека Math в Python забезпечує доступ до деяких популярним математичних функцій і констант, які можна використовувати в кодї для більш складних математичних обчислень

У бібліотеці Math в Python є дві важливі математичні константи

Первой важной математической константой является число Пи (π)

```
import math
print(math.pi)
```

```
>>> %Run 1.py
3.141592653589793
>>>
```

Дану константу можна використовувати для обчислення площі або довжини

```
import math
radius = 2
print("Площа окружности з радіусом 2 дорівнює:", math.pi * radius ** 2)
```

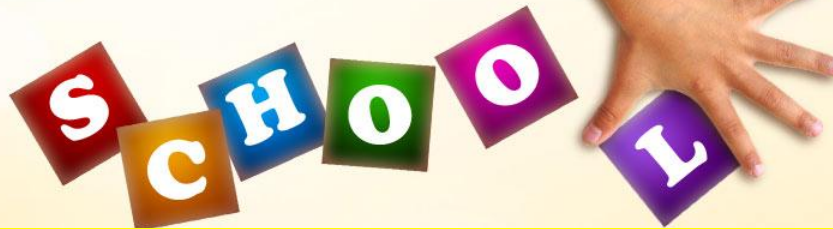
```
>>> %Run 1.py
Площа окружности з радіусом 2 дорівнює: 12.566370614359172
```



Число Ейлера з бібліотеки math

```
import math  
print(math.e)
```

```
>>> %Run 1.py  
2.718281828459045
```



Функції подання чисел.

`ceil ()` і `floor ()` - ціла частина числа.

```
import math
number = 8.10
# виводимо цілу частину числа з округленням до більшого
print("Верхня межа 8.10 це:", math.ceil(number))
# виводимо цілу частину числа з округленням до меншого
print("Нижня межа 8.10 це:", math.floor(number))
```

```
>>> %Run 1.py
Верхня межа 8.10 це: 9
Нижня межа 8.10 це: 8
>>>
```



Функція `fabs ()` - абсолютне

значення

Функція `fabs` використовується для обчислення абсолютного

значення числа

```
# Імпорт модуля math
import math
number = -8.10
# вивід абсолютного значення числа
print(math.fabs(number))
```

```
>>> %Run 1.py
8.1
>>>
```



factorial () - функція факторіала

```
# Імпорт модуля math
import math
number = 5
# вивід факторіала числа
print("факторіал числа",
math.factorial(number))
```

```
>>> %Run 1.py
факторіал числа 120
>>>
```

Якщо хтось намагається використати негативне число, повертається помилка значення (Value Error)



Функція `fmod()` — остаток от деления.

Функція `fmod(x, y)` повертає $x \% y$. Різниця в тому, що вираз $x \% y$ працює тільки з цілими числами, а цю функцію можна використовувати і для чисел з плаваючою точкою.

```
#Импорт модуля math
import math
print(math.fmod(5,2))
print(math.fmod(-5,2))
print(math.fmod(-5.2,2))
print(math.fmod(5.2,2))
```

```
>>> %Run 1.py
1.0
-1.0
-1.20000000000000002
1.20000000000000002
>>>
```



Функція `fsum()` — точна сума `float`

Обчислює точну суму значень з плаваючою точкою в ітеріруемому об'єкті і суму списку або діапазону даних.

```
# Імпорт модуля math
import math
# сума списку
numbers=[.1,.2,.3,.4,.5,.6,.7,.8,8.9]
print("сума", numbers, ":", math.fsum(numbers))
# сума діапазона
print("сума чисел від 1 до 10:", math.fsum(range(1,11)))
```

```
>>> %Run 1.py
сума [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 8.9] : 12.5
сума чисел від 1 до 10: 55.0
```




Функція `exp()`

Ця функція приймає один параметр у вигляді дробового числа і повертає e^x .

```
# Імпорт модуля math
import math
print("e в ступені 5", math.exp(5))
print("e в ступені 2.5", math.exp(2.5))
```

```
>>> %Run 1.py
e в ступені 5: 148.4131591025766
e в ступені 2.5: 12.182493960703473
>>>
```



Функція `log ()` - логарифм

числа

Функція `log (x [, base])` знаходить логарифм числа x за основою e (за замовчуванням). `base`- параметр опціональний. Якщо потрібно обчислити логарифм з певною основою, його

потрібно вказати

```
# Імпорт модуля math
import math
# логарифм з основою e
print(math.log(2))
# логарифм з вказаною основою(2)
print(math.log(64,2))
```

```
>>> %Run 1.py
0.6931471805599453
6.0
>>>
```



Функція $\log_{10}()$

Обчислює логарифм за основою 10.

```
# Імпорт модуля math
import math
print(math.log10(1000))
```

```
>>> %Run 1.py
3.0
>>>
```



Функція `pow ()` - ступінь числа

Используется для нахождения степени числа. Синтаксис функции `pow (Base, Power)`. Она принимает два аргумента: основание и степень.

```
# Импорт модуля math
import math
print (math.pow (5, 4) )
print (math.pow (2, 10) )
```

```
>>> %Run 1.py
625.0
1024.0
>>>
```



Функція `sqrt ()` - квадратний корінь числа

Ця функція використовується для знаходження квадратного кореня числа. Вона приймає число як аргумент і знаходить його квадратний корінь.

```
# Імпорт модуля math
import math
print(math.sqrt(256))
print(math.sqrt(1024))
print(math.sqrt(625))
```

```
>>> %Run 1.py
16.0
32.0
25.0
>>>
```



Тригонометричні функції

В Python есть следующие тригонометрические функции.

```
import math
# функція синусу
print("синус  $\pi/2$  :", math.sin(math.pi/2))
# функція косинуса
print("косинус 0 :", math.cos(0))
# функція тангенса
print("тангенс  $\pi/4$  :", math.tan(math.pi/4))
# функція арксинуса
print("арксинус 0 :", math.acos(0))
# функція арккосинуса
print("арккосинус 1 :", math.acos(1))
# функція арктангенса
print("арктангенс 0.5 :", math.atan(0.5))
```



```
>>> %Run 1.py
синус PI/2 : 1.0
косинус 0 : 1.0
тангенс PI/4 : 0.99999999999999999999
арксинус 0 : 1.5707963267948966
арккосинус 1 : 0.0
арктангенс 0.5 : 0.4636476090008061
>>>
```



Функція перетворення кутів

`degrees ()` : конвертує радіани в градуси;
`radians ()` : конвертує градуси в радіани;

```
# Імпорт модуля math
import math
print(math.degrees(1.57))
print(math.radians(90))
```

```
>>> %Run 1.py
89.95437383553924
1.5707963267948966
>>>
```




Приклад:

```
import math
A = int(input("введіть число A="))
B = int(input("введіть число B="))
A1 = math.pow(A, 2)
B1 = math.pow(B, 3)
S = A1+B1
P = math.sqrt(A1 * B1)
M = max(A1, B1, S, P)
print (M)
```

```
>>> %Run 1.py
введіть число A=5
введіть число B=4
89.0
```

Python - це маленький початок великих справ.



Дякую за увагу!