МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Київський національний університет ім.Тараса Шевченко

Механіко-математичний факультет

Реферат

***«Numeric and Mathematical Modules»***

Студентки 1 курсу заочної форми навчання

Спеціальності 111 Математика

Корягіної Анастасії

**Зміст**

[**Модуль demical** 3](#_Toc88847174)

[**Модуль statistics** 4](#_Toc88847175)

[**Модуль random** 6](#_Toc88847176)

[**Модуль numbers** 9](#_Toc88847177)

[**Модуль franctions** 11](#_Toc88847178)

[**Висновки** 13](#_Toc88847179)

[**Список використаних джерел** 14](#_Toc88847180)

# **Модуль demical**

Модуль demical-модуль у мові програмування Python, який забезпечує підтримку швидкої арифметики з правильним округленням числа з плаваючою комою.Цей модуль засновано на модулі з плаваючою комою, за принципом, що «комп’ютер має забезпечити ту арифметику, яка працює так само, як та, яку вивчають у школі»

Основою конструкції є:

-контекст;

-десяткове число;

-арифметичні сигнали; [1]

Цей модуль в собі має велику кількість функцій, тут ми опишемо основні, та наведем приклад як вони працюють.

Для того, щоб підключити модуль необхідно на початку коду записати наступну строку: from demical import Demical. Записавши це, ми отримаємо доступ до усіх функції цього модуля.

Основним компонентом для того щоб працювати з цим модулем є клас Demical. Для того щоб використовувати його необхідно створити об’єкт за допомогою конструктора.

**from** decimal **import** Decimal  
x=Decimal(**"0.2"**)

Тепер ми можемо використовувати створене число у арифметичних операціях, також можливо використання цілих чисел, але варто запам’ятати, що використовувати float та Demical у математичних операціях не можна.

Метод для округлення чисел quantize().

number = Decimal(**"0.2356"**)  
number = number.quantize(Decimal(**"1.00"**))  
print(number)  
number = Decimal(**"0.555678"**)  
print(number.quantize(Decimal(**"1.0000"**)))  
  
number = Decimal(**"0.999"**)  
print(number.quantize(Decimal(**"1.0"**)))

Отримаємо:



# **Модуль statistics**

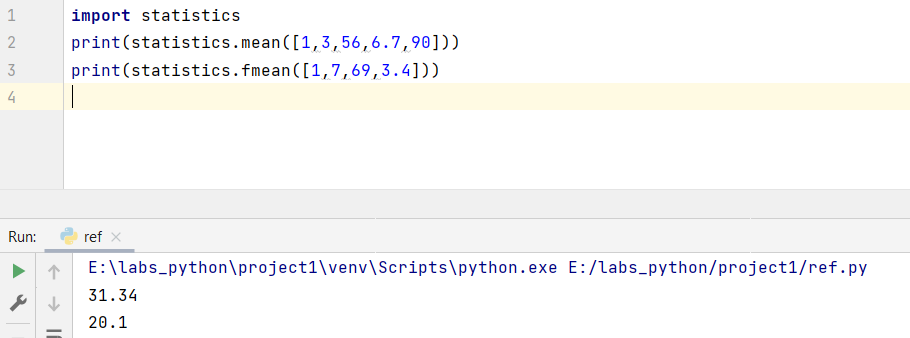
Модуль statistics-це модуль у мові програмування Python,який використовується для представлення функції для розрахунку математичної статистики числових даних. Функції у цьому модулі підтримують наступні типи даних: int, float, Demical та Franction.

Функції модуля як параметри приймають числову послудовніть. Виклик функцій має наступний синтаксис:statistics.функція([параметри(послідовність чисел)]):

-mean() and fmean() ці дві фунції повертають середнє арифметичне число.

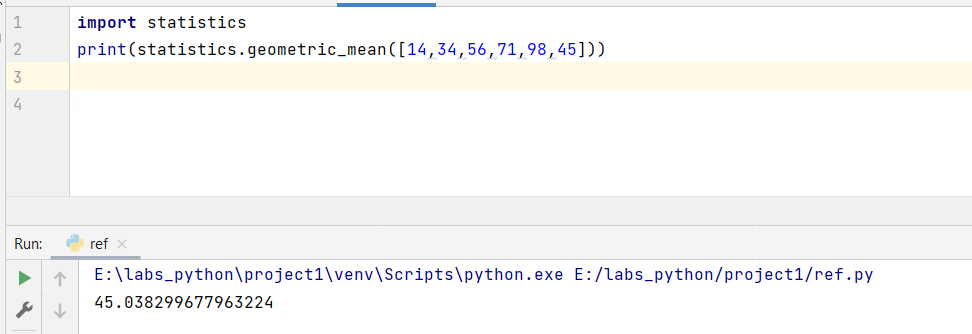
Функція fmean() працює швидше, завжди повертає тип даних float.

Приклад роботи:



-geometric\_mean() перетворює об’єкти послідовності у числа з плаваючою крапкою(комою) та повертає середнє геометричне.

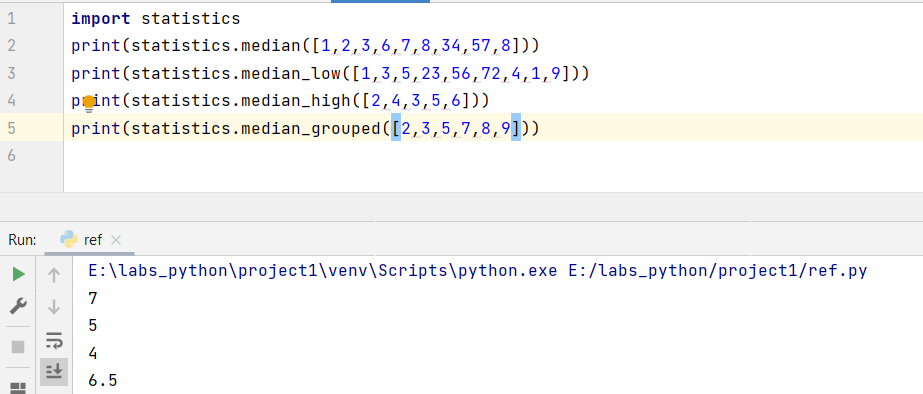
Приклад роботи:



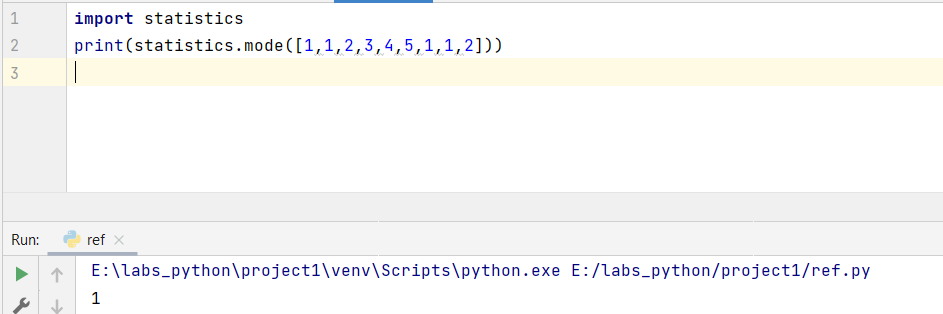
-median() повертає медіану (середнє значення) послідовності, використовує метод «середнє з двох середніх».

Також ще є функції median\_low() повертає найменше середнє значення(найменшу медіану), median\_high() повертає найбільшу медіану, median\_grouped() повертає медіану згрупованих неперервних даних розраховану як 50-й процентиль за допомогою інтерполяції.

Приклад роботи:



-mode() повертає один найбільш поширений елемент з дискретних чи номінальних даних.



-multimode() повертає список найбільш поширениз елементів послідовності у тому порядку, у якому їх було знайдено.

-pstdev() повертає стандарте відхилення числової послідовності ;

-stdev() повертає стандартне відхилення виборки елементів числової послідовності;

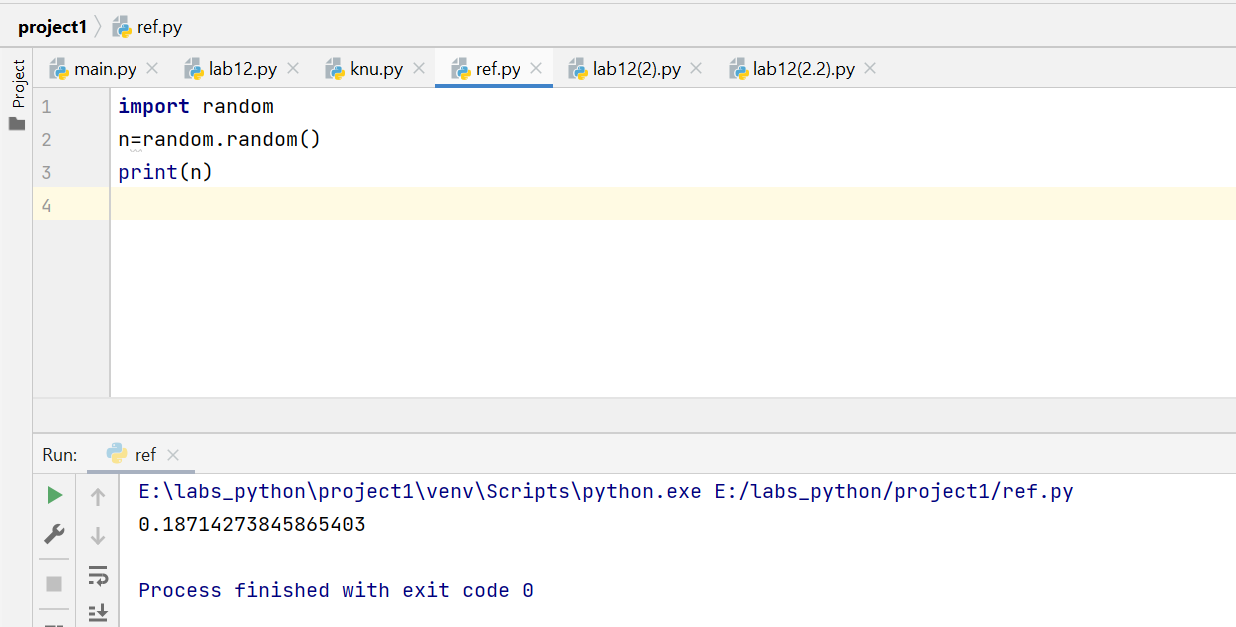
-variance() повертає дисперсію виборки з елементів числової послідовності;

-quantiles() ділить числову послідовність з даними на n неперервних інтервалів з рівною ймовірністю. [2]

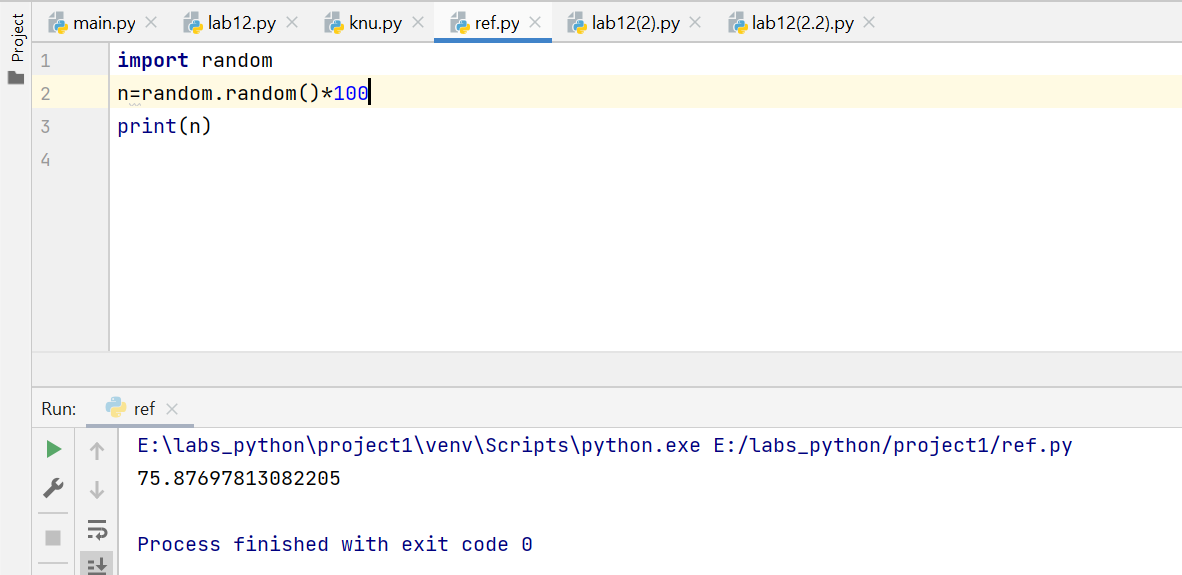
# **Модуль random**

Модуль random-модуль у мові програмування Python,який використовується для генерації випадкових чисел, в собі random має функції, які викликаються наступним чином: **randon.функція**(аргументи).Більшість функцій модуля залежать від основної **randon(),** за допомогою якої генерується випадкове число з плаваючою комою на проміжку [0.0;1.0].

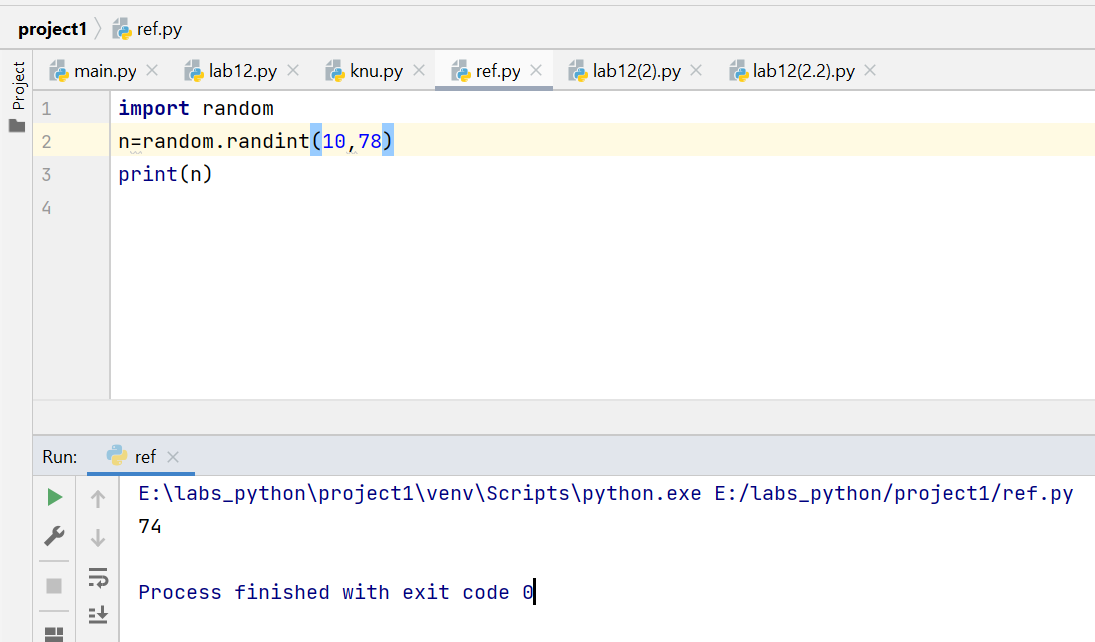
-**random.random**()  
Ця функція повертає випадкове число з плаваючою точкою на проміжку [0.0;1.0].



Якщо необхідно отримати число з більшого діапазону, наприклад [0.0;100.0],необхідно просто виконати множення на необхідну величину( в нашому випадку на 100).



-**random.randint**(min,max) повертає ціле число з проміжку [min;max].



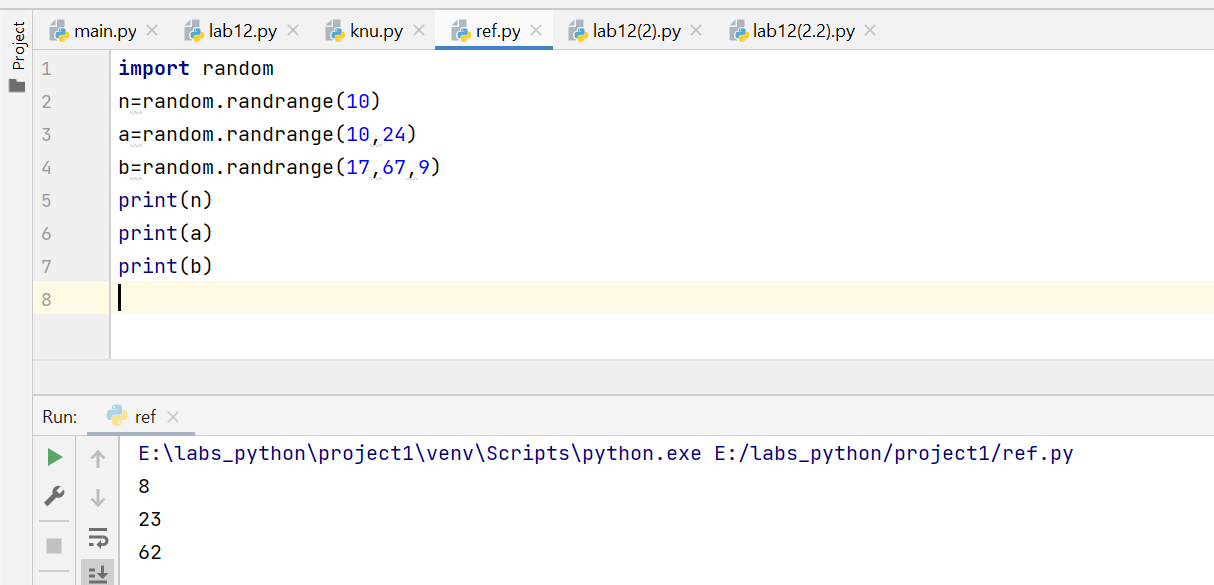
На цьому прикладі видно, що функція повертає саме ЦІЛЕ число, можна задавати проміжок який вам необхідно.

-**random.randrange**() повертає випадкове число з вказаного набору, має в собі три форми: **randrange**(stop),**randrange**(start,stop),**randrange**(start,stop,step). Між ними є помітна різниця.

-**randrange**(stop) виконує вивід випадково числа з діапазону чисел від 0 до вказаного stop числа.

-**randrange**(start,stop) виконує вивід випадкового числа з діапазону від start числа до stop числа.

-**randrange**(start,stop,step) виконує вивід випадково числа з діапазону від start числа до stop числа, які між собою відрізняються на step число(або з шагом у step).



**-random.weibullvariate**(alpha, beta)-розподіл Вейбула;

- **random.choice**(sequence) випадковий елемент непустої послідовності;

Також цей модуль містить в собі функції за допомогою яких можна працювати зі списками. Це функції **shuffle**(), яка випадковим способом сортує список та функція **choise**(), повертає випадковий елемент зі списку.[3]

# **Модуль numbers**

Модуль numbers-модуль у мові програмування Python, який визначає ієрархію числових абстрактних базових класів, які поступово визначають більше операцій.

-class numbers.Number

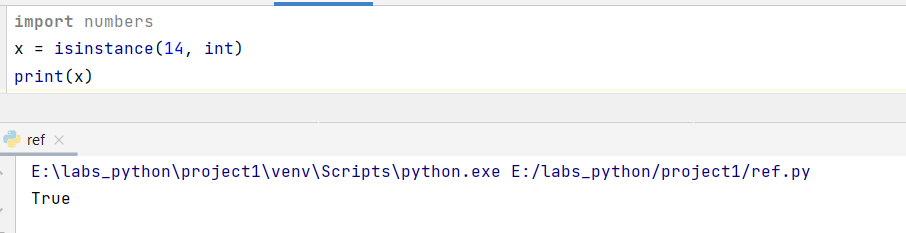
Корінь числової ієрархії. Якщо вам потрібно перевірити, чи є число, яке ви ввели якимось з типів (наприклад int, float, string або іншими) використовується функція isinstance(),яка приймає два параметри.[4]

Синтаксис:

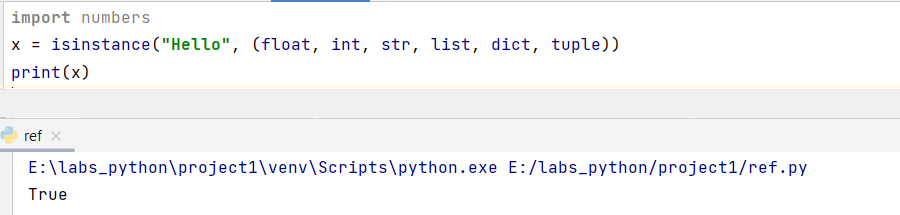
isinstance(object, type)

Приклад роботи:

Для того, щоб перевірити чи є число 14 типом int. Якщо це істина отримаємо “True”, якщо ні-“False”.



Ще один приклад, перевірка чи є “Hello” одним з типів (float, int, str, list, tuple, dict).[5]



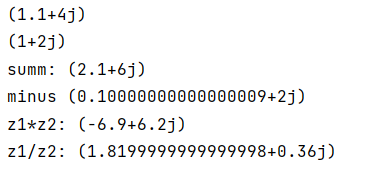
class numbers.Complex

Підкласи цього типу описують комплексні числа і включають операції, які працюють із вбудованим складним типом.Перетворення на складні та логічні, реальні, уявні, +, -, \*, /, \*\*, abs(), conjugate(), ==, and !=. Усе окрім - і! [4]

Запишемо комплексні числа та відтворимо з ними наступні операції: +, -, \*, /

z1=complex(1.1,4)  
print(z1)  
z2=complex(1,2)  
print(z2)  
print(**"summ:"**,z1+z2,**"\nminus"**,z1-z2,**"\nz1\*z2:"**,z1\*z2,**"\nz1/z2:"**,z1/z2)

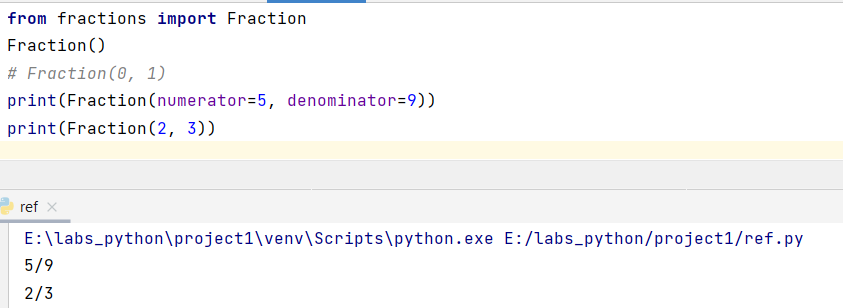
Отримаємо наступне:



# **Модуль franctions**

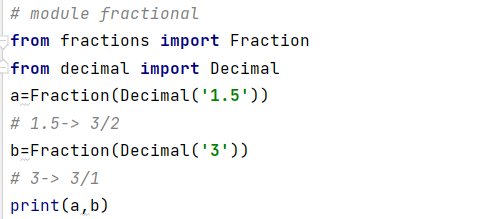
Модуль franctions-модуль у мові програмування Python, який забезпечує підтримку арифметики раціональних чисел.

Для того, щоб створити дріб необхідно ввести чисельник та знаменик denominator.



Для запису чисельника або знаменника можна також використовувати інші екземпляри Franction().

Можливо з раціонального числа перетворити у десяткове за допомогою Franction(Demical()).



У модулі також підтримаються математичні(арифметичні операції) над раціональними числами. Але потрібно запам’ятати, що не можливо виконати дію над Franction() та Demical() у одному математичному виразі![6]

a=Fraction(3,7)  
b=Fraction(7,19)  
print(**"a+b:"**,a+b,**"\n a-b:"**,a-b,**"\n a\*b:"**,a\*b,**"\n a/b+3:"**,a/b+3,**"\n a\*\*b:"**,a\*\*b)

# **Висновки**

Numeric and Mathematical Modules містять у собі наступні модулі:

- numbers;

- math;

-demical;

-fractions;

-itertools;

-functools;

-random;

-operator;

-cmath;

Використання цього модуля корисне для того, щоб працювати з числами, коли у задачі необхідно використання комплексного чи раціонального числа, коли необхідно виводити випадкове число, для побудови графіків. Це широко використовується при розв’язувані математичних задач.

# **Список використаних джерел**

1.Dialogi.tech:[Електроне джерело]-режим доступу:

<https://digitology.tech/docs/python_3/library/decimal.html>

2. DOCS Python3:[Електроне джерело]-режим доступу:

<https://docs-python.ru/standart-library/modul-statistics-python/>

3.METANIT:[Електроне джерело]-режим доступу:  
<https://metanit.com/python/tutorial/6.1.php>

4.Python docs:[Електроне джерело]-режим доступу:

<https://docs.python.org/3/library/numbers.html>

5.W3School:[Електроне джерело]-режим доступу:

<https://www.w3schools.com/python/ref_func_isinstance.asp>

6.DOCS Python3:[Електроне джерело]-режим доступу:

<https://docs-python.ru/standart-library/modul-fractions-python/>