

Python Basic 02

Math functions, string operations

Carlos 2023 Fall

Math Functions

Math Functions

Extreme value, normalize, operation

Function name	Description
<code>abs (x)</code>	Returns the absolute value of x
<code>min (x1 ,x2 , . . .)</code>	Returns the minimum of x1, x2, ...
<code>max (x1 ,x2 , . . .)</code>	Returns the maximum of x1, x2, ...
<code>pow (x ,y)</code>	Returns x^y

Math Functions

Convert

Function name	Description
<code>hex(x)</code>	Returns the hex (十六進位) of x, x must be decimal (十進位)
<code>oct(x)</code>	Returns the oct (八進位) of x, x must be decimal (十進位)
<code>bin(x)</code>	Returns the bin (二進位) of x, x must be decimal (十進位)

Math Functions

Example(1)

```
1  a = -1
2  numbers = [23, -1, 5, -4, 7]
3  b = 24
4  # abs of a
5  print('a = ', a)
6  print('numbers = ', numbers)
7  print('abs of a = ', abs(a))
8  # min of numbers
9  print('min of numbers = ', min(numbers))
10 # max of numbers
11 print('max of numbers = ', max(numbers))
12 # hex of b
13 print('b = ', b)
14 print('hex of b = ', hex(b))
15 # oct of b
16 print('oct of b = ', oct(b))
17 # bin of b
18 print('bin of b = ', bin(b))
```

ex_math1.py

```
● (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_math1.py
a = -1
numbers = [23, -1, 5, -4, 7]
abs of a = 1
min of numbers = -4
max of numbers = 23
b = 24
hex of b = 0x18
oct of b = 0o30
bin of b = 0b11000
```

ex_math1.py output

Math Functions

Rounding

Function name	Description
<code>int(x)</code>	Returns the integer value of x, truncates floating part to zero (無條件捨去)
<code>round(x, [precision])</code>	Returns the round of x to a given precision in decimal digits (四捨五入)

Math Functions

Example(2)

```
1 a = 2.71828
2 # int of a
3 print('a = ', a)
4 print('int(a) = ', int(a))
5 # round a to integer
6 print('round(a) = ', round(a))
7 # round a with 2 digit percision
8 print('round(a, 2) = ', round(a, 2))
```

ex_math2.py

```
● (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_math2.py
a = 2.71828
int(a) = 2
round(a) = 3
round(a, 2) = 2.72
```

ex_math2.py output

Module: math

Intro

- ‘math’ is a python module
- To use ‘math’, must use ‘import’ to include the module to your code
- Functions in ‘math’ will be like ‘math.function () ’

Module: math

Commonly used functions

Function name	Description
<code>math.ceil(x)</code>	Returns ceil of x
<code>math.floor(x)</code>	Returns floor of x
<code>math.fabs(x)</code>	Returns absolute floating number of x
<code>math.factorial(x)</code>	Returns x!
<code>math.gcd(x, y)</code>	Returns the greatest common divisor of x and y
<code>math.exp(x)</code>	Returns e^x
<code>math.log(x, [base])</code>	Returns $\log_{\text{base}} x$
<code>math.sqrt(x)</code>	Returns square root of x

Module: math

Example(3)

```
1 import math
2
3 a = -24.134
4 b = 5
5 c = 15
6 # ceil of a
7 print('ceil(a) = ', math.ceil(a))
8 # floor of a
9 print('floor(a) = ', math.floor(a))
10 # fabs of a
11 print('fabs(a) = ', mathfabs(a))
12 # factorial of b
13 print('factorial(b) = ', math.factorial(b))
14 # gcd of b, c
15 print('gcd(b, c) = ', math.gcd(b, c))
```

ex_math3.py

```
● (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_math3.py
ceil(a) = -24
floor(a) = -25
fabs(a) = 24.134
factorial(b) = 120
gcd(b, c) = 5
```

ex_math3.py output

Exercise - 1

Using math module

Let $x = \log_2 5$ and $y = \log_2 3$

- A. What are the values of $\log_2 5$ and $\log_2 3$
- B. Compute $2^{x^2+y^2+1}$

Module: random

Intro

- ‘random’ is a python module
- ‘random’ provides several functions to generate random numbers

Module: random

Functions

Function name	Description
<code>random.randint(x, y)</code>	Returns random integer in range [x, y]
<code>random.random()</code>	Returns random number in range [0, 1)
<code>random.shuffle(x)</code>	Shuffle list x
<code>random.choice(x)</code>	Returns the random choice of list x

Module: random

Example(4)

```
1 import random
2
3 numbers = [1, 2, 3, 4, 5, 6, 7]
4 # randint(2, 100)
5 print('randint(2, 100) = ', random.randint(2, 100))
6 # random()
7 print('random() = ', random.random())
8 # shuffle numbers
9 print('numbers = ', numbers)
10 random.shuffle(numbers)
11 print('numbers (after shuffle) = ', numbers)
12 # choice numbers
13 print('choice(numbers) = ', random.choice(numbers))
```

ex_random.py

- (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_random.py
randint(2, 100) = 28
random() = 0.6378501389675645
numbers = [1, 2, 3, 4, 5, 6, 7]
numbers (after shuffle) = [5, 6, 3, 1, 2, 4, 7]
choice(numbers) = 1

ex_random.py output

String Operations

String Operations

Attributes

- Each character has its Unicode
- String's length

String Operations

Attributes

Function name	Description
<code>ord(c)</code>	Returns the unicode of single character c
<code>chr(i)</code>	Returns the unicode string of index i
<code>len(s)</code>	Returns the length of s

String Operations

Example(5)

```
1 # unicode of 'A'  
2 print('ord(A) = ', ord('A'))  
3 # unicode string of index 65  
4 print('chr(65) = ', chr(65))
```

ex_unicode.py

- (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_unicode.py
ord(A) = 65
chr(65) = A

ex_unicode.py output

String Operations

Escape sequence

- A string that start with ‘\’

Escape sequence	Meaning
\\\	Print \
\'	Print ‘
\”	Print “
\n	Linefeed
\r	Carriage return
\t	Horizontal tab
\b	Backspace
\a	Bell

Commonly used escape sequence

String Operations

Example(6)

```
1 string = 'Hello, Python!'
2 # \
3 print('\'', string, '\'')
4 # \n
5 print('\n')
6 # \t
7 print('\t', string)
```

ex_escapeSequence.py

- (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_escapeSequence.py
' Hello, Python! '

Hello, Python!

ex_escapeSequence.py output

String Operations

Operator

Operator	Meaning
<code>+</code>	Concatenate strings
<code>*</code>	Duplicate string
<code>>, <, >=, <=, ==, !=</code>	Compare given strings' unicode (not length!!)
<code>in</code>	Determine whether string A is exists in string B
<code>not in</code>	Determine whether string A is not exists in string B

Operation on string

String Operations

Example(7)

```
1  a = 'Hello'
2  b = ', Python'
3  c = '!'
4  d = 'lo'
5  e = 'Hello'
6  f = 'Hello1'
7  g = 'hello'
8  # concatenation
9  print(a + b + c)
10 # duplication
11 print(3 * a)
12 # comparison
13 print('Is a > e? ans: ', a > e)
14 print('Is a > f? ans: ', a > f)
15 print('Is a > g? ans: ', a > g)
16 print('Is d in a? ans: ', d in a)
17 print('Is d not in a? ans: ', d not in a)
```

ex_stringOperation.py

```
● (.venv) kaiyang@Kais-MacBook-Pro Class_Example % python3 ex_stringOperation.py
Hello, Python!
HelloHelloHello
Is a > e? ans:  True
Is a > f? ans:  False
Is a > g? ans:  False
Is d in a? ans:  True
Is d not in a? ans:  False
```

ex_stringOperation.py output

Exercise - 2

【輸入輸出、數據處理】

- 輸入：a, b, c 三個數
- 輸出：算數平均數、幾何平均數

$$\frac{a + b + c}{3}$$

算數平均數

$$\sqrt[3]{a \cdot b \cdot c}$$

幾何平均數

```
Input a: 3  
Input b: 3  
Input c: 3  
The arithmetic mean is: 3.0  
The geometric mean is : 3.0
```

Sample output 01

```
Input a: 3  
Input b: 4  
Input c: 5  
The arithmetic mean is: 4.0  
The geometric mean is : 3.9148676411688634
```

Sample output 02