Python入門
関数の基本

関数とは

- 引数を与えると、それに基づく計算を行い、結果を返す
  - 数学関数など
- 組み込み関数
  - 特別な指定なしで利用できる
- モジュールを指定して利用する関数
- 自分で定義した関数

数値計算に使える組み込み関数

関数	説明
abs(x)	$\|x\|$
divmod(a,b)	aをbで割った(商、余り)のタプル（値の組）
max(a,b,c,…)	最大値
min(a,b,c,…)	最小値
pow(x,y)	x^y
pow(x,y,z)	$x^y \% z$
round(x,k)	xをk桁に丸める。切り上げ、切り捨ての距離が同じ場合は偶数側になることに注意

```
>>> abs(-10)
10
>>> divmod(10,3)
(3, 1)
>>> divmod(10,2.5)
(4.0, 0.0)
>>> max(3,1,5,10,4)
10
>>> pow(2,8)
256
>>> pow(2,8,3)
1
>>>
>>> round(3.5)
4
>>> round(2.5)
2
>>> round(3.1415,2)
3.14
```

文字列操作

関数	説明
chr(整数)	整数が表すUnicode文字列
ord(一文字)	文字に対するUnicode
len(文字列)	文字列の長さ
str(数値)	数値を文字列化

```
>>> chr(66)
'B'
>>> ord('A')
65
>>> ord('佐')
20304
>>> chr(20305)
'佑'
>>> len('佐賀県')
3
>>> str(100)
'100'
```

モジュール

- 関連する関数などがまとまったライブラリ
- 標準で、math、randomなどのモジュールが配布されている
- 他にも便利なライブラリが多数存在
- モジュール読み込み方法

```
import モジュール1,モジュール2,…
import モジュール1 as 別名
```

mathモジュール

関数	説明
ceil(x)	小数以下切り下げて整数に
copysign(x, y)	xと絶対値が等しく、yと符号の等しい値を返す
fabs(x)	$\|x\|$
factorial(x)	$x!$
floor(x)	小数以下切り上げて整数に
fmod(x,y)	$x\%y$
fsum(iterable)	iterableなデータ列の和
gcd(a,b)	aとbの最大公約数
inf	浮動小数の最大値
nan	浮動小数型の非数

関数	説明
exp(x)	e^x
log(x,b)	$\log_b x$
log2(x)	$\log_2 x$
log10(x)	$\log_{10} x$
sqrt(x)	$\sqrt{x}$
e	e

mathモジュール：三角関数

関数	説明：角度はラジアン
acos(x)	$\mathrm{acos}(x)$ 逆余弦
asin(x)	$\mathrm{asin}(x)$ 逆正弦
atan(x)	$\mathrm{atan}(x)$ 逆正接
atan2(x,y)	原点から(x,y)へのベクトルの角度
cos(θ)	
sin(θ)	
tan(θ)	
degrees(θ)	角度をラジアンから度へ変換
radians(x)	角度を度からラジアンへ変換
pi	π

関数を指定してモジュールを読み込む

- モジュールには多数の関数
- 名前が重複する危険性

```
from モジュール import 関数名 as 別名
```

randomモジュール

- 乱数（数値がでたらめに生成される）を用いる場合
  - シミュレーションなど
- 多数の関数が定義されている
  - randint(a,b):$a \leq x \leq b$の整数乱数を生成
  - random():$[0,1)$の浮動小数乱数を生成
  - gauss(mu,sigma):平均μ、標準偏差σのガウス分布に従う乱数を生成

関数の定義

```
import math
def quadratic(a,b,c):#二次方程式の解
    d = b*b-4*a*c#判別式
    if d >= 0:#実数解
        return ((-b + math.sqrt(d))/(2*a),(b – math.sqrt(d))/(2*a))
    #複素数解
    return ((-b + 1j*math.sqrt(-d))/(2*a),(b - 1j*math.sqrt(-d))/(2*a))

a = 1
b = 3
c = 1
print('二次方程式、実数解')
(x1,x2)=quadratic(a,b,c)
print((x1,x2))
c = 3
print('二次方程式、複素数解')
(x1,x2)=quadratic(a,b,c)
print((x1,x2))
```

関数の定義
引数の数が特定されない場合

```
def average(*args):#引数の個数を指定しない->タプルで受ける
    count=0
    total=0
    for x in args:
        count += 1
        total += x
    return total/count

print(average(1,3,6,8))
```