標準ライブラリとモジュール

初めてのプログラミング

2020年度

只木進一（理工学部）

関数とは
Functions

- 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
  - 式で表せなくてもよい
- 引数を与えると、そのデータを加工して返す

Pythonの関数
Functions in Python

- 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
  - 沢山ある
- 自分で定義した関数

サンプルプログラムの取得

- GitHubRepositoryを指定
  - https://github.com/first-programming-saga/StandardLibraries

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

StandardLibraries/basicFunctions.ipynb

基本的関数

基本的関数の例を示す。これらは、モジュールのimportが不要である。

絶対値

$$\mathrm{abs}(x) = |x|$$

In [ ]:
```
abs(-10)
```

徐算の商と余り $(a, b) = \mathrm{divmod}(x, y)$

$$a = \mathrm{int}(x/y)$$
$$b = x \mod y$$

In [ ]:
```
divmod(10,3)
```

In [ ]:
```
divmod(10,2.5)
```

最大値

In [ ]:
```
max(3,1,5,10,4)
```

べき乗

$$\mathrm{pow}(x, y) = x^y$$

In [ ]:
```
pow(2,8)
```

$$\mathrm{pow}(x, y, z) = x^y \mod z$$

In [ ]:
```
pow(2,8,3)
```

まるめ。距離が等しい場合には、偶数側に丸めること注意が必要。

In [ ]:
```
round(3.5)
```

In [ ]:
```
round(2.5)
```

二番目の引数を与えて、丸める桁を指定することもできる。

In [ ]:
```
round(3.1415,2)
```

課題

与えられた数値の中から、最小値を求める例を作成しなさい

文字操作関数
string functions

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

StandardLibraries/basicStringFunctions.ipynb

文字に関する基本的関数

文字に関する基本的な関数の例。

文字コードから文字へ

In [ ]:
```
chr(66)
```

文字から文字コードへ

In [ ]:
```
ord('A')
```

In [ ]:
```
ord('佐')
```

In [ ]:
```
chr(20305)
```

文字列の長さ。len()は、他のオブジェクトについても長さを返す。

In [ ]:
```
len('佐賀県')
```

数値に対応した文字列を返す

In [ ]:
```
str(100)
```

文字列に対応した整数値を返す

In [ ]:
```
int('256')
```

文字列に対応した浮動小数点値を返す

In [ ]:
```
float('9.12')
```

文字コードからアルファベット小文字を表示する

In [ ]:
```
ca = 97#小文字'a'のアスキーコード
for i in range(ca,ca+26):
    print(chr(i))
```

課題

文字コードを指定することで、アルファベット大文字を表示しなさい

モジュール (modules)

- 関連する関数や定数などをまとめたもの
  - pythonと一緒に配布されているもの
  - 後からインストールするもの
- import モジュール
- from モジュール import 関数
- asで別名を付けることも可能

mathモジュール

- 様々な数学関数
- 整数への切り上げ(ceil())、整数への切り下げ(floor())、最大公約数(gcd())、平方根(sqrt())

StandardLibraries/mathFunctions.ipynb

数学関数

これらはmathモジュールが必要である。

In [ ]:
```
import math
```

mathモジュールを使った例

xに対して、その常用対数を求め($y = \log_{10} x$)、整数へと切り下げて1を加える。これにより、xの整数部分の桁数がわかる。

In [ ]:
```
x = 101
y = math.log10(x)
print(f'log_10({x}) = {y}')
n = math.floor(y)+1
print(f'{x}の整数部分は{n}桁です')#文字列中に数値を埋めこむ。~{x}~は変数xの値を埋めこむことを表す
```

二つの整数の最大公約数

In [ ]:
```
math.gcd(21,36)
```

指数関数、対数関数、平方根、三角関数が使える

【課題】

小数以下の切り下げ、切り上げの例を作成しなさい。

In [ ]:
```

```

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)	$\log_b x$
log(x)	$\ln x = \log_e x$
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)	acos(x) 逆余弦
asin(x)	asin(x) 逆正弦
atan(x)	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)	余弦
sin(θ)	正弦
tan(θ)	正接
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

オブジェクトとメソッド Objects and Methods

- pythonでは、データやデータの塊をオブジェクトと呼ぶ
- オブジェクトには、操作方法（メソッド）が付随している

文字列のメソッド
Methods for strings

- 大文字小文字変換
- 含まれる文字の数
- 文字列を発見
- 文字列を置き換え
- 余分な文字を取り去る
- 文字列差し込み
- 注意：immutableであること

StandardLibraries/stringFunctions.ipynb

文字列を操作する基本的な関数

元の文字列は変更されていないことに注意する。

In [ ]:
```
a = 'Apple Pie'
```

In [ ]:
```
a.upper()
```

In [ ]:
```
a.lower()
```

In [ ]:
```
#元の文字列が変更されていないことを確認
a
```

In [ ]:
```
b = "saga university"
```

In [ ]:
```
b.capitalize()
```

In [ ]:
```
b.title()
```

部分文字列

インデクスを用いて、部分文字列を取り出すことができる

In [ ]:
```
a[0:4]
```

In [ ]:
```
a[:4]
```

In [ ]:
```
a[4:]
```

検索と置換

In [ ]:
```
print(b.find('y'))#'y'を探索
```

In [ ]:
```
print(b.find('s',0))#'s'を先頭から探索
```

In [ ]:
```
print(b.find('z'))#含まれていない文字を探すと
```

In [ ]:
```
print(b.rfind('s'))#後ろから探索
```

In [ ]:
```
c = b.replace('a','A')
print(c)
```

次回

- 5章「条件分岐、繰り返し、例外処理」
  - 条件分岐とwhile