

# 標準ライブラリとモジュール



初めてのプログラミング

2019年度

只木進一（理工学部）

# 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

# 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
<code>abs(x)</code>	$ x $
<code>divmod(a,b)</code>	$a$ を $b$ で割った(商、余り)のタプル (値の組)
<code>max(a,b,c,...)</code>	最大値
<code>min(a,b,c,...)</code>	最小値
<code>pow(x,y)</code>	$x^y$
<code>pow(x,y,z)</code>	$x^y \% z$
<code>round(x,k)</code>	$x$ 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](#)

# 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

StandardLibraries/basicStringFunctions.ipynb

# 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())

```
#整数部分の桁数
import math
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')
```

# 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
<code>acos(x)</code>	<code>acos(x)</code> 逆余弦
<code>asin(x)</code>	<code>asin(x)</code> 逆正弦
<code>atan(x)</code>	<code>atan(x)</code> 逆正接
<code>atan2(x,y)</code>	原点から(x,y)へのベクトルの角度
<code>cos(<math>\theta</math>)</code>	
<code>sin(<math>\theta</math>)</code>	
<code>tan(<math>\theta</math>)</code>	
<code>degrees(<math>\theta</math>)</code>	角度をラジアンから度へ変換
<code>radians(x)</code>	角度を度からラジアンへ変換
<code>pi</code>	$\pi$

# オブジェクトとメソッド

- ▶ 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

- ➡ 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

# 次回

- ➡ 5章「条件分岐、繰り返し、例外処理」
  - ➡ 例外処理は扱わない