

Unit 1: Python入門

Skill Builder 3: Python関数の紹介

このレッスンでは、関数を定義し、その関数を使って式を評価します。Pythonでのインデント(字下げ)の目的と、エディタでのインラインプロンプト(行内入力要請)の支援を体験します。

目標

- 関数の定義
- 式の評価に関数を使用
- input()関数

関数は、数学と同様、Pythonプログラミングでも大きな役割を果たします。関数はさまざまな種類の値を生成するため使用でき、複雑なプロセスをより小さく、管理しやすい部分に分割するのに役立つPythonサブルーチンとして機能できます。

プログラムは問題を解決するための段階的なレシピ、つまりアルゴリズム(algorithm, 問題を解くための計算手順や処理手順)です。すべてのプログラムはその下にあるアルゴリズムです。

1. ユーザーがxの値を入力できるプログラムを作成すると、プログラムはその値を使って関数を評価します。  $f(x)=x^2 + 3x - 1$ .

まず、**menu > Built-ins > Function > def function()** (メニュー>組み込み>関数>def function())を使って関数を定義します。関数テンプレートが表示され、インラインプロンプトの関数、引数、ブロックが提供されます。独自のコードに置き換える必要があります。



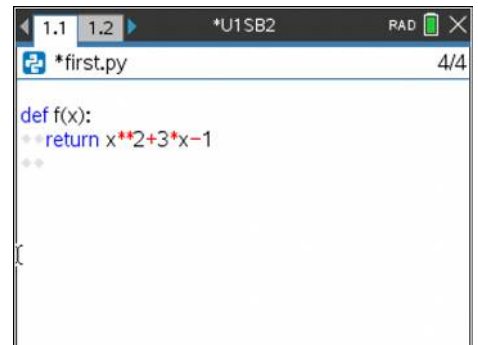
```
def function(argument):
  ==block
  [
```

2. **function(関数)**が選択されていることに注意します。文字fを入力して**function**をfに置き換え、Tabを押して**argument(引数)**をxに置き換え、もう一度Tabを押して**block(ブロック)**を次のように変更します。

**return x\*\*2 + 3\*x - 1**

**return**は、**menu > Built-ins > Function** (メニュー>組み込み>関数)にあります。

**x\*\*2**は、**x<sup>2</sup>**または**x\*x**のPython表記です。  
そして、**3x**ではなく**3\*x**と入力します。



```
def f(x):
  ==return x**2+3*x-1
  ==
  [
```

ブロックが2つの◆◆でインデント(字下げ)されていることに注意してください。薄い灰色のひし形は、より複雑なプログラムでインデントの量を追跡するためのプレースホルダー(あとから入力される文字・値の代わりに、仮に入力されている文字や値)です。Pythonではインデントは非常に重要です。

関数の定義が完了しました。このプログラムを実行すると、この関数定義はすぐには実行されませんが、シェルはそれが存在することを「認識」します。

3. `input()`文を使って、ユーザーがxの数値を入力できるようにします。新しい空白行の先頭にバックスペース([del])して、次のように記述します。

```
x=input("Enter a value for x: ")
```

キーパッドの=キーを使います。

`input()`は、**menu > Built-ins > I/O** (メニュー>組み込み>I/O)を押します。または、入力します。

一重引用符または二重引用符(**Ctrl+x**(かけ算記号))のいずれかを使用できます。

引用符で囲まれた緑色のテキストは、キーパッドから1文字ずつ入力します。

コロン(:)は句読点キーにあります。必須ではありませんが、プロンプト(入力要請)の見栄えが良くなります。

```
1.1 1.2 *U1SB2 RAD 5/5
*first.py
def f(x):
    return x**2+3*x-1
x=input("Enter a value for x: ")
```

4. 次の行に`print()`文を記述します。

```
print("f(x) equals ", f(x))
```

`Print()`は、**menu > Built-ins > I/O** (メニュー>組み込み>I/O)にあることを思い出してください。

引用符で囲まれた`f(x)`は`f(x)`として出力されますが、コンマの後の`f(x)`は関数呼び出しであって、関数によって返される値(戻り値)に置き換えられます。

```
1.1 1.2 *U1SB2 RAD 6/6
*first.py
def f(x):
    return x**2+3*x-1
x=input("Enter a value for x: ")
print("f(x) equals ", f(x))
```

5. **Ctrl+R**を押してプログラムを実行します。プロンプトで、xの数値を入力し、**Enter**を押します。

```
1.1 1.2 *U1SB2 RAD 3/3
Python Shell
>>>#Running first.py
>>>from first import *
Enter a value for x: |
```

6. (おそらくは)最初のランタイムエラーへようこそ。メッセージは長いのですが、重要なのは最後の2行です。最後の行番号と`TypeError`(エラータイプ)に注意してください: "can't convert..."(変換できません...)

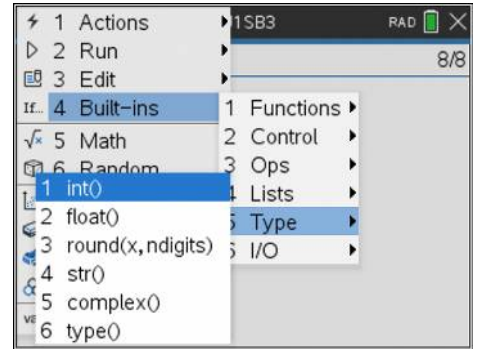
`input()`関数は数値ではなく文字列を返すため、エラーが発生します。文字列を数値に変換する必要があります。

Pythonには、**int**(整数), **str**(文字列), **float**(10進数の数値), **complex**(複素数), **bool**(ブール値はTrueまたはFalse)の5つのタイプのデータがあります。

```
1.1 1.2 U1SB2 RAD 10/10
Python Shell
>>>#Running first.py
>>>from first import *
Enter a value for x: 4
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
  File "/python/doc16/first.py", line 7, in <module>
  File "/python/doc16/first.py", line 3, in f
TypeError: can't convert 'int' object to str implicitly
>>>|
```

7. エラーの修正 : Pythonにはいくつかの変換関数が組み込まれています。それらは **menu > Built-ins > Type** (メニュー>組み込み>タイプ) にあります。

それらは, **int()**, **float()**, **str()**, **complex()** です。



8. 文字列xを数値に変換するには, print文の前に次を置きます。

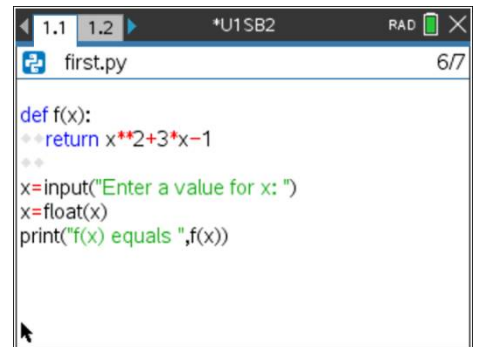
**x=float(x)**

または

**x=int(x)**

両方を試して, 違いを確認してください。

この例では**float()**を選択しました。なぜですか。



9. プログラムを再び実行すると, 意図したとおりに動作します。



10. チャレンジ問題 : 次のように, 関数の括弧内のxを(文字xではなく)xの値で表示するには, コードをどのように変更すればよいですか。

**f(4.0) equals 27.0**

