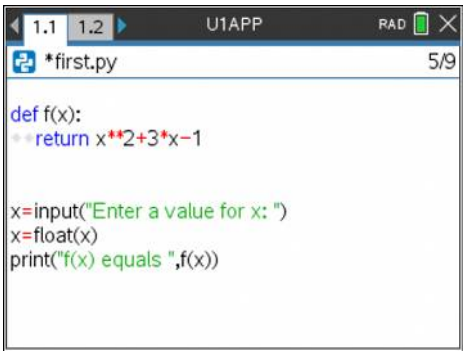
Unit 1: Python入門	Application: 関数は1つより2つ

この応用では，2番目の関数を作成し，2つの関数を使って数学的性質を調べます。

目標

- Pythonファイルの編集
- Pythonファイルのコピー
- 機能の追加
- シェルでの関数式の評価
- 逆関数の作成

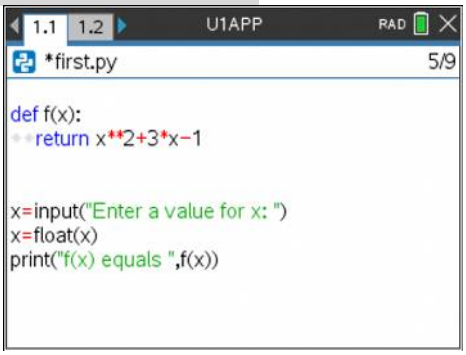
前回のレッスン(ユニット1 スキルビルダー3)では，関数f(x)を定義しました。この応用(Application)では，そのファイルに別の関数を追加し，それらの関数を使っていくつかの式を評価します。



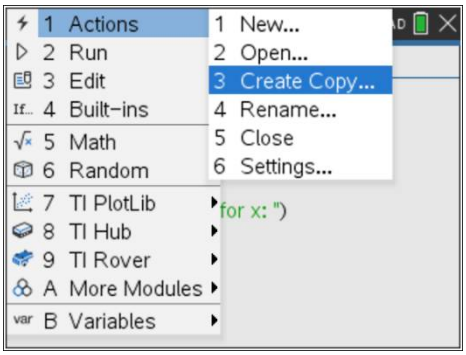
Teacher Tip: この応用はユニット1 スキルビルダー3の発展です。関数を使った式と逆関数の作成についていくつかの数学を強化します。

1. スキルビルダー3のPythonプログラムを使ってTI-Nspireドキュメントから始めます。右図を参照してください。

   doc > File > Save As… (ドキュメント>ファイル>名前を付けて保存…)を押して，別の名前でTI-Nspireドキュメントを保存し，別の名前を使います。右図のタイトルバーは，新しい名前U1APPを示しています。



2. Pythonエディタで**menu > Actions > Create Copy…** (メニュー>アクション>コピーの作成…)を押して，Pythonプログラムのコピーを作成します。
   コピーに別の名前を付けます(既定値では，名前の末尾に1が追加されます)。
   (Create Copy…(コピーの作成...)が使えない場合，プログラムで**ctrl+B**を押して保存します。エディタ上部のPythonファイル名の前にアスタリスクを付けないでください)。
   これにより，ドキュメント内に重複コードを持つ，別のPythonエディタアプリが作成されます(ドキュメントU1APPは，ページ1.1にfirst.py，ページ1.2にfirst1.pyを持ちます)。



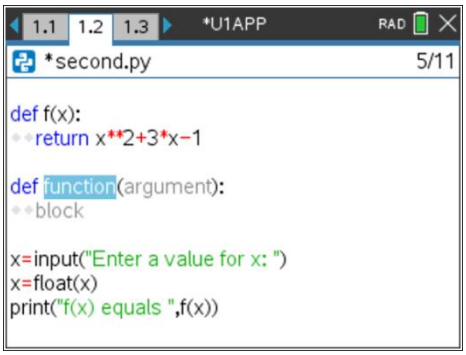
Teacher Tip: Pythonファイルは，それらが作成されたドキュメントに保存されます。TI-Nspireドキュメント内のPythonファイルは，プロブレム番号に関係なく，ドキュメント内のエディタですべて使用できます。ページ(**ctrl+doc**)を挿入して**Add Python** (Pythonの追加)を選択すると，ドキュメントに存在する任意のPythonファイルを**Open**(開く)できます。

ただし，Pythonシェルはプロブレムベースです。各プロブレムには独自のシェル環境があります。プロブレムの1つのシェルを再起動すると，プロブレムのすべてのシェルがリセットされます。

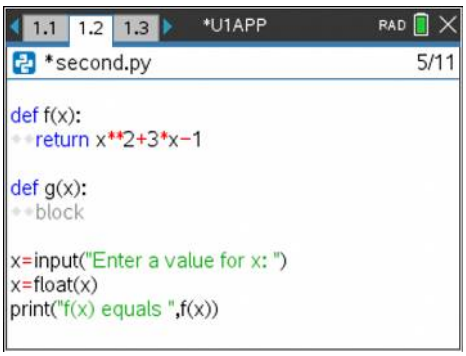
Note: TI-Nspireはコンピュータと同じファイル構造，すなわちフォルダがあって，その中にドキュメントを作成・保存するという構造です。ドキュメントは複数のプロブレム(Problem)からなり，各プロブレムには複数のページを作成することができます。1ページが1画面にあたります。1つのドキュメントには最大30プロブレム，1つのプロブレムには最大50ページ作成できます。定義された変数や関数は，作成されたプロブレム内のみ有効です。

3. 新規のPythonファイル名は**second.py**です。

   つぎに，関数f(x)の下に2番目の関数テンプレートを追加します。
   空白行で，**menu > Built-ins > Functions (**メニュー>組み込み>関数)を押し，**def function()**を選択します。
   この場合も，関数の構文には，**def**行の最後にコロン(:)が含まれています。これは，次のコードが関数の定義であり，ブロックがインデントされていることを示しています。



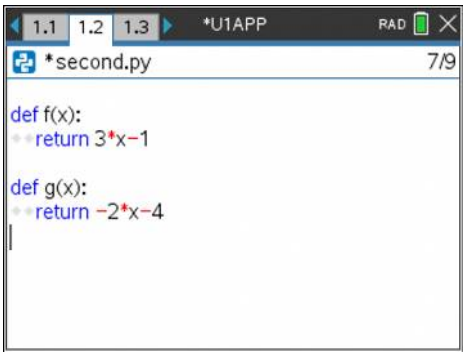
4. 2番目の関数にg(x)という名前を付けます。



5. 関数**f(x)**から**x**2+**を削除して，**f(x) = 3 * x - 1**になるようにf(x)を変更します。
   g(x)を定義します。

```
def g(x):
  ◆◆return -2*x - 4
```

   プログラム下部にある3行のコードを削除し，2つの関数のみを残します。



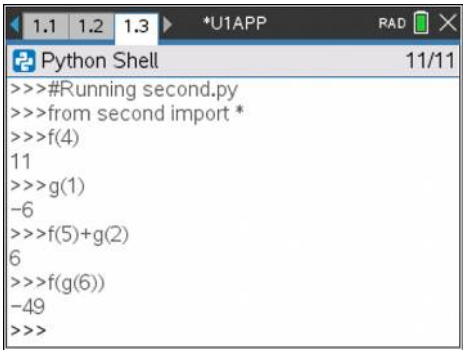
Teacher Tip: このレッスンの後半で，2つの1次関数を使って逆関数を調べます。

キーパッドの負の符号キー(⊖)と引き算キー(−)はどちらも，Pythonエディタでは同じことを行います。つまり，文脈に応じて負の符号または引き算を実行します。すなわち，TI-Nspireは負の符号と引き算をキーにより区別しますが，Pythonエディタは区別しません。

6. **ctrl+R**を押して，式f(1)+g(1)を入力します。

   次のような2つの関数を使って，他の式を試してください。

   f(4)+g(1), f(5)+g(2), (f+g)(4), f(g(6))



Teacher Tip: シェルのテキスト(履歴)をクリアするには，シェルで**menu > Tools > Clear History** (メニュー>ツール>履歴のクリア)を押します。

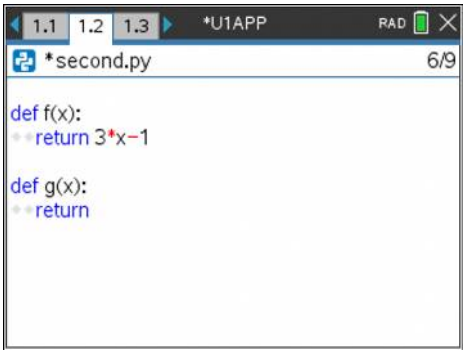
(f+g)(x)は有効なPython式ではありません。エラーメッセージを作成します。

TI-Nspireドキュメントのシェル履歴は，ドキュメントと一緒に保存されません。ドキュメントを閉じると，すべてのシェルテキストはクリアされます。シェル履歴を保存するには，テキストをすべて選択してからコピーし(**ctrl+A, ctrl+C**)，ドキュメントのNotesアプリに貼り付けます(**ctrl+V**)。

次の最後のステップは，関数の合成と逆関数を含むため，初期の学習者には上級かもしれません。しかし，関数の合成と逆関数に精通している学生にとって，それはやりがいのある演習です。

7. **合成関数と逆関数**

   準備はいいですか。エディタに戻り，つねにf(g(x)) = g(f(x))になるようにg(x)を変更します(xの値に関係なく)。関数を，注意を払ってテストしてください。



Teacher Tip: ステップ7では，f(x)の逆関数を記述する必要があります。引用した例では，これはg(x)＝(x＋1)/3です。
[訳注] f(x)＝3x−1，g(x)が1次関数のとき，すべての実数xについてf(g(x))＝g(f(x))を満たす関数g(x)は，g(x)＝ax＋(1−a)/2 (aは任意の実数)となります。g(x)＝ax＋b(a≠0)とおいて，f(x)＝3x−1とともにf(g(x))＝g(f(x))に代入すれば求められます。この**Teacher Tip**の説明は，f(x)の逆関数を考えてa＝1/3の特別な場合だけ求めています。問題の条件f(g(x))＝g(f(x))は，逆関数でもこの条件は成り立つという確認だけしか使っていません。ここでは，教材全体の流れを考慮し，原文のままにしてあります。