Unit 1: Python入門

Skill Builder 2: 編集，変数，式

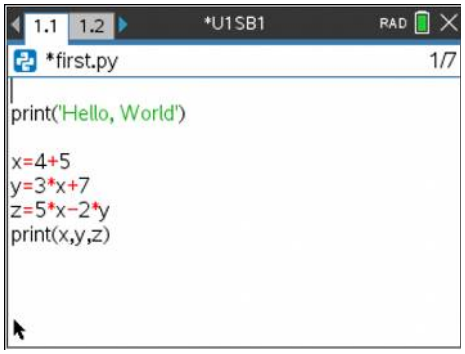
このレッスンでは，Pythonエディタで作業し，メニューを調べ，いくつかの数学の演算を使い，色の強調表示を体験し，特別なキーパッドの変更に注意し，さまざまなデータ型を操作します。

目標

- メニューを探索
- 演算子の使用
- 基本的なタイプの変数について学習
- 代入文(assignment statement)＝を使用
- エディタでキーボードの違いを体験

---

1. スキルビルダー1で作成したTI-Nspireのドキュメントを使うか，新規ドキュメントを開始してPythonエディタを追加します。
print('Hello, World')は前のレッスンから残っています。エディタで，次のステートメント(文)を記述します。

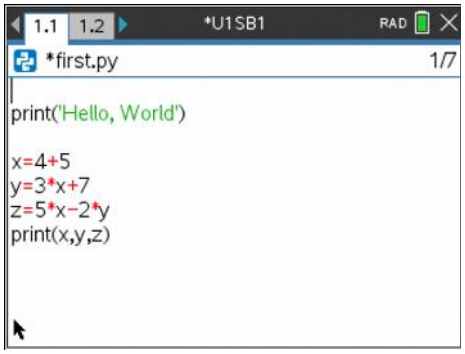


x = 4 + 5
y = 3*x + 7
z = 5*x - 2*y
print(x, y, z)

カンマ(,)は，キーパッドの文字Oの左側にあります。
これらのステートメントは，まだ結果を出さないことに注意してください。

Teacher Tip: 演算子は赤色で強調表示されます。さらに，乗法記号を使う必要があります。PythonはTI-Nspireのように乗法記号の省略をサポートしていません。5xは構文エラーです。5*xが正しいものです。x5は変数名になります。'^'はPython演算子です。ただし，5^2=7，6^2=9，5^1=4です。これはビット単位のXORを表します。^の使用は避けてください。ただし，準備してください。

2. 変数は，値を保存する文字または単語です。＝は代入演算子です。式の結果を，左側の変数名に格納します。

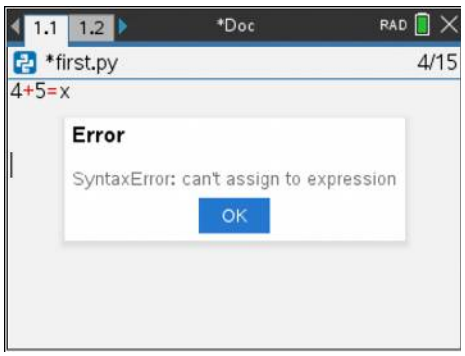


x = 4 + 5

これは，xという名前の変数に値9を格納します。変数xはz=5*x−2*yなど，他の式で使用でき，xの値が使われます。

Note: Pythonでは大文字と小文字は区別されます。Xとxは異なる変数です。

3. 次の式を試すとどうなりますか。

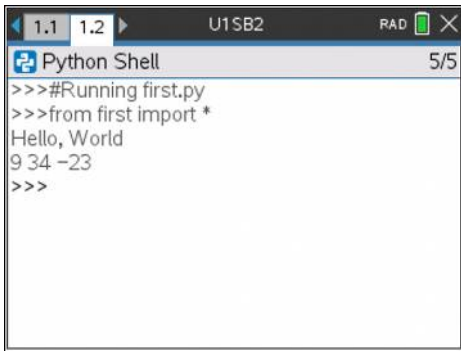


4 + 5 = x

Ctrl+Bと押します。
これは許されません。式は右側になければいけません。

Teacher Tip: 識別方法：Pythonでは大文字と小文字は区別されます。**X**は**x**とは異なる変数です。Pythonプログラマーは，大文字または小文字の変数，あるいは ”HoursWorked” など，2つの単語をくっつけてプログラムを読みやすくできます。また，Variable_Nameのようにアンダースコア(_)文字が使え，提供されたツールでは多く使われています。

4. **ctrl + R**を押して，このコードを実行します。まったく同じ式を使った場合，次のように表示されます。

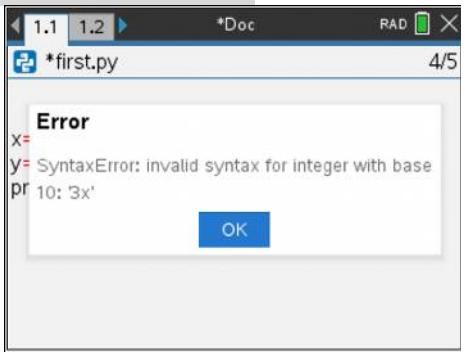


Hello, World
9 34 -23
>>>

つまり，xは9，yは34，zは−23です。確認してみましょう。
Ctrl+左矢印を押してエディタに戻り，他の式を試してください。

Teacher Tip: プログラミングエラーには3つの種類があります。インタプリタエラーは通常‘syntax error’(構文エラー)と表示され，ランタイムエラー(プログラムの実行時に発生するエラー)はコンピュータによって検出されます。3番目のエラーはプログラマーの頭の中にあります。たとえば，誤った式(演算順序の誤用など)または論理構造(‘or’が必要な場合に‘and’を使うなど)を入力します。**例**：5+3/7+1 vs. (5+3)/(7+1) (演算の順序)。どちらも適切に評価されますが，結果は異なります。プログラマーは何を意図していたのでしょう。

5. 次をやってみましょう。

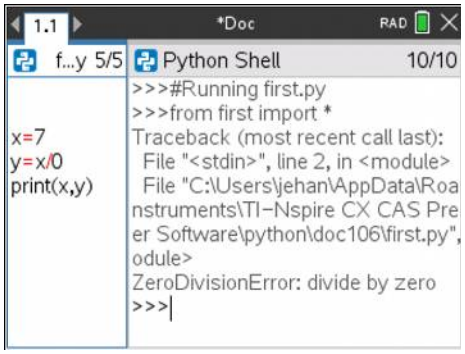


x = 7
y = 3x
print(x, y)

このコードを実行すると何が得られますか。

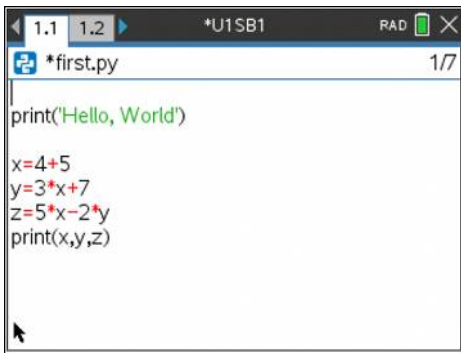
コンピュータプログラミングの世界へようこそ!
‘Syntax Error’(構文エラー)はコードのテキスト内のエラーです。英語の先生はそれを文法上の誤りと言います。TI-Nspireでは許されていても，Pythonでは‘3x’は許されません。3*xでなければいけません。

6. プログラミングエラーには3つの種類があります。インタプリタエラーは通常‘syntax error’(構文エラー)と表示され，ランタイムエラー(プログラムの実行時に発生するエラー)はコンピュータによって検出されます。3番目のエラーは，プログラマーの頭の中にあります。たとえば，誤った式(演算の順序の誤用など)または論理構造(‘or’が必要な場合に‘and’を使うなど)を入力します。**例**：5+3/7+1 vs. (5+3)/(7+1) (演算の順序)。どちらも適切に評価されますが，結果は異なります。プログラマーは何を意図していたのでしょう。



7. コードをステップ1の3つの式とprint()ステートメントに戻します。出力の表示を改善するには，各値にメッセージ(x=など)を追加します。

print(“x=”,x,” y=”,y,” z=”,z)

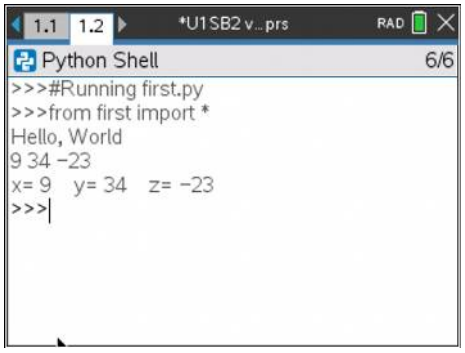


引用符内のy=とz=の前にいくつかの余白があります。
余白はインタプリタによって無視されますが，引用符で囲まれている場合は，記述どおりに出力されます。

このステートメントを入力するときは，十分注意してください。コンマと引用符の位置は非常に重要です。いずれかの記号が間違った位置にある場合，**Ctrl+R**を押すと‘Syntax Error’(構文エラー)メッセージが表示されます。エラーを見つけて修正する必要があります。エラーは通常，カーソルの近くまたは真上にあります。

8. このプログラムを実行すると，次のように表示されます。

x= 9 y= 34 z= -23



Teacher Tip: Pythonで出力をフォーマットする方法は他にもあります。このコースでは取り上げていない.format()機能などです。String concatenation(文字列結合，ある文字列の後ろに別の文字列をつなげて1つの文字列にする処理)は，出力を改善するもう1つの方法です。