

Unit 2: 入力, 出力, 関数

Application: 地平線までの距離

この応用では、与えられた高度から地平線までの距離を決定するプログラムを作成します。

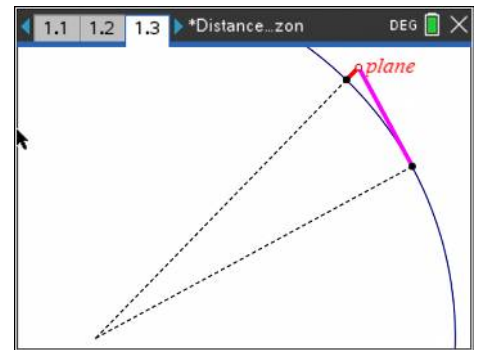
目標

- 追加機能にインポートを使用
- メニューを使ってプログラム作成
- 1つの問題で複数のユニットを使用

飛行機が飛んでいるとき、パイロットは地平線までどれくらい離れていますか。飛行機が高く飛ぶほど、地平線までの距離は遠くなります。飛行機の高度を考慮して、どのようにして地平線までの距離を決定しますか。

地球の半径は3,958.8マイルです。

飛行機の高度をフィートで入力し、地平線までの距離をマイルで生成するプログラムを作成します。



1. 新規Pythonファイルを始めます(これをhorizonと名前を付けます)。

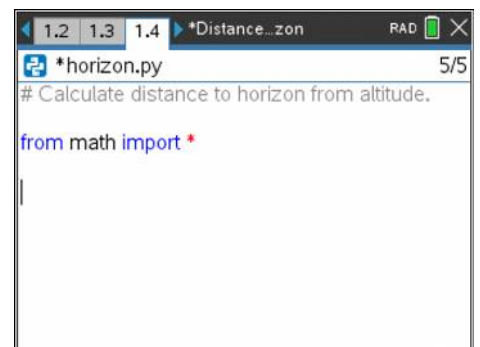
プログラムの目的を説明するコメントから始めます。



2. このプログラムはピタゴラスの定理を使い、Pythonの組み込み関数ではない平方根を必要とします。この関数やその他の関数は、**math**と呼ばれる標準Pythonモジュールにあります。この関数を使うには、数学モジュールをコードにインポートする必要があります。**menu > math** (メニュー>数学)と押して、一番上にあるステートメントを選択します。

from math import *

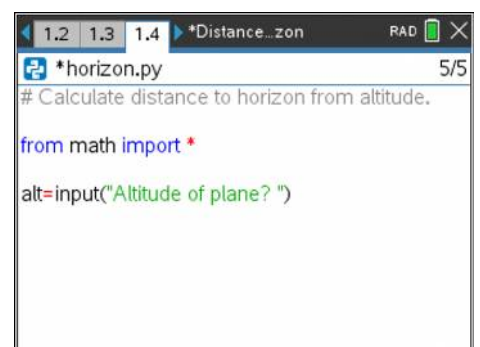
アスタリスク(*)は「すべて」を意味します。メニューを使って、このモジュールのツールを使う方法を確認します。



3. inputステートメントを使って、飛行機の高度を入力します。まず、変数altと=記号を入力します。

つぎに、**menu > Built-ins > I/O** (メニュー>組み込み>I/O)で**input()**関数を確認します。

括弧内のプロンプト(入力要請)については、次のように記述します。“Altitude of plane?”(飛行機の高度?)



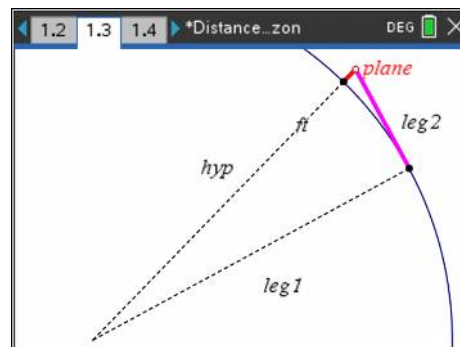
4. 右図から、1つの辺が地球の半径、もう1つの辺が飛行機から地平線までの距離、斜辺が(地球の半径と平面の高度)である直角三角形であることが分かります。ピタゴラスの定理は、次のようになります。

$$(\text{leg1})^2 + (\text{leg2})^2 = \text{hyp}^2$$

$$\text{または } \text{radius}^2 + \text{leg2}^2 = (\text{radius} + \text{alt})^2$$

leg2について解くと、

$$\text{leg2} = \text{sqrt}((\text{radius} + \text{alt})^2 - \text{radius}^2)$$



5. 次の式をプログラムに入力します。

$$\text{leg2} = \text{sqrt}((\text{radius} + \text{alt})^2 - \text{radius}^2)$$

括弧に注意し、2乗には必ず**2を使います。また、変数radiusを使って地球の半径を表し、3958.8に設定します。

$$\text{radius} = 3958.8$$

プログラムをテストする前に、さらに2つのことを処理する必要があります。

6. input()が文字列を生成することを思い出します。数値が必要なので、float関数を入力ステートメントに追加します。

$$\text{alt} = \text{float}(\text{input}(\text{"Altitude of plane? "}))$$

また、高度はフィート単位ですが、半径はマイル単位です。ステートメントを使って高度をマイルに変換します。

$$\text{alt} = \text{alt} / 5280$$

```

1.2 1.3 1.4 *Distance...zon DEG 7/9
# Calculate distance to horizon from altitude.
from math import *
alt=input("Altitude of plane? ")
radius=3958.8
leg2 = sqrt((radius+alt)**2 -radius**2)

```

7. 準備はほぼ完了しましたが、まだ、答えを印刷する必要があります。

print(leg2)

これは答えを出しますが、より分かりやすいメッセージは次のようになります。

print('Distance to horizon: ', leg2, 'miles.')

準備ができればctrl+Rを押して、プログラムを実行します。さまざまな高度を試してください。

ドキュメントを保存することを忘れないでください。

```

1.2 1.3 1.4 *U2SB1 ...zon RAD 1/15
# Calculate distance to horizon from altitude.
from math import *
alt=float(input("Altitude of plane? "))
alt=alt/5280
radius=3958.8
leg2 = sqrt((radius+alt)**2 -radius**2)

```

```

1.2 1.3 1.4 *U2SB1 ...zon RAD 8/9
# Calculate distance to horizon from altitude.
from math import *
alt=float(input("Altitude of plane? "))
alt=alt/5280
radius=3958.8
leg2 = sqrt((radius+alt)**2 -radius**2)
print("Distance to horizon: ',leg2,'miles')

```

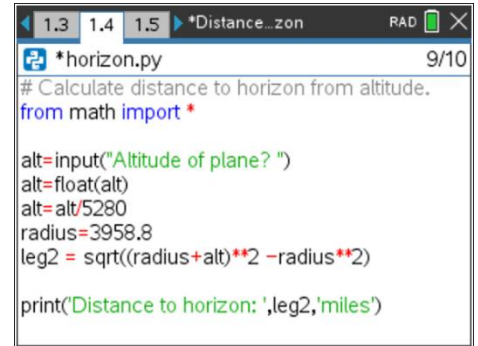
8. まとめ：最終プログラムのステートメントの順序は重要ですが、プログラムに入力される順序は重要ではありません。これがテキストエディタの力です。このプログラムと、このユニットの他のすべてのプログラムは、ステートメントを上から下に処理する言語の「シーケンス」構造の例です。

プログラムを実行すると、プログラム内のコメント、余分なスペース、空白行は無視されます。importステートメントは、TI-Nspire Pythonプログラミング言語に含まれている個別のモジュールから追加の関数を取り込みます。

9. **質問1**：国際宇宙ステーションは地球の表面から254マイル上にあります。乗組員から地平線までどのくらいの距離ですか。

質問2：海を見渡す海水の端のビーチに立っている場合、地平線はどのくらい離れていますか。

ヒント：地面からの目の高さ(フィート単位)を入力します。



```
1.3 1.4 1.5 *Distance...zon RAD 9/10
*horizon.py
# Calculate distance to horizon from altitude.
from math import *

alt=input("Altitude of plane? ")
alt=float(alt)
alt=alt/5280
radius=3958.8
leg2 = sqrt((radius+alt)**2 -radius**2)

print('Distance to horizon: ',leg2,'miles')
```

