

Unit 2: 入力, 出力, 関数

Skill Builder 2: ヘロンの公式

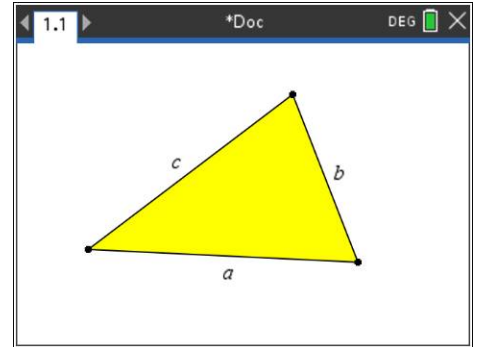
このレッスンでは、プログラムテンプレートを使って、値を評価して返す関数を作成し、数学の演算子を調べます。

目標

- プログラムテンプレート(Type:)の使用
- ヘロンの公式の関数を作成

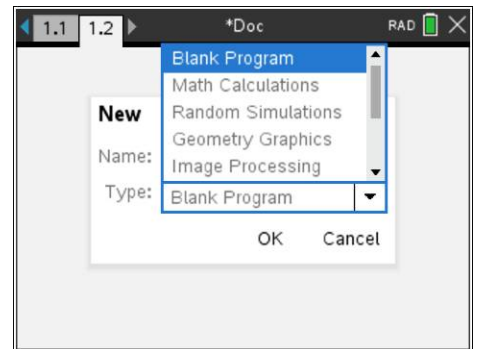
3辺の長さを与えられた三角形があります。面積は求められますか。はい、ヘロンの公式を使っています。😊

このレッスンでは、3辺の長さを使って三角形の面積を決定する関数を作成し、その関数を使うプログラムを作成します。



1. このプログラムは、前のプログラムと同様、sqrt()関数を必要とします。今回は、プロジェクトに最も一般的に必要な関数をプリロードする、Pythonテンプレートを使います。

Add Python > New (Pythonの追加>新規)を選択してPythonファイルの名前を入力すると(名前は'area'とします)、Name:の下にType:というラベルのフィールドがあります。既定値は'Blank Program (空白プログラム)'です。右側のポップアップ矢印をクリックすると、利用可能なプログラミングプロジェクトが表示されます(たくさんあります)。

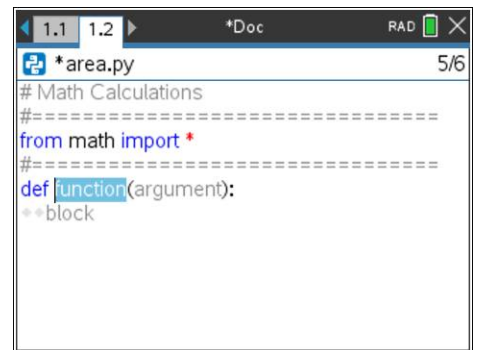


このプロジェクトでは、Type: **Math Calculations**(数学計算)を選択してから**OK**をクリックまたは**enter**を押します。

2. **Math Calculations**(数学計算)テンプレートは、次のステートメントを提供します。

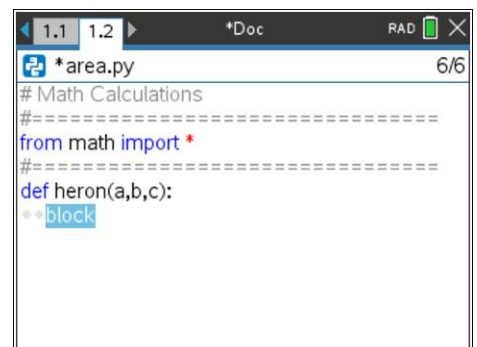
from math import *

つぎに、**menu > Built-ins > Function**(メニュー>組み込み>関数)から**def function()**ステートメントを選択します。



3. Heronという名前の関数を作成します。三角形の3辺を表すa, b, cの3つの引数を用意します。

ブロックのコードについては、次のステップをご覧ください。



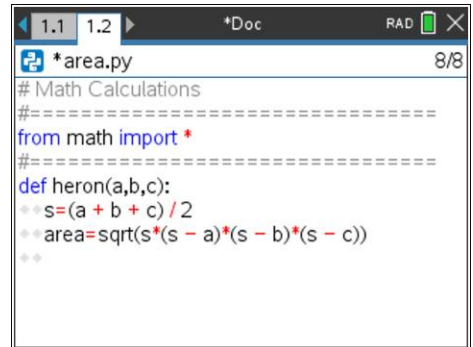
4. **Heron's Formula**(ヘロンの公式)は、2ステップの計算です。
まず、周囲の長さsの半分を計算します。

$$s = (a + b + c) / 2$$

そして、面積は次のようになります。

$$\text{area} = \sqrt{s(s - a)(s - b)(s - c)}$$

この関数ブロックの2つのステートメントがインデント(字下げ)されていることに注意してください。



```

1.1 1.2 *Doc RAD 8/8
*area.py
# Math Calculations
#=====
from math import *
#=====
def heron(a,b,c):
    s=(a + b + c) / 2
    area=sqrt(s*(s - a)*(s - b)*(s - c))
    
```

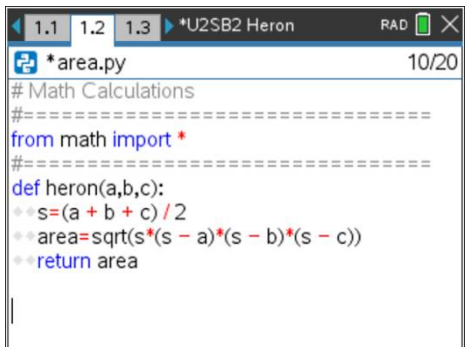
5. **return**ステートメントを指定して関数を終了します

return area

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

数学同様、関数は引数を持ち、値を生成します。returnステートメントは、値を使うメインプログラムに値を送信するため必要です。

重要 : delまたはshift + tabを使って、挿入カーソルを新しい行の先頭に戻します。分かりやすくするために、1行以上空けることもできます。



```

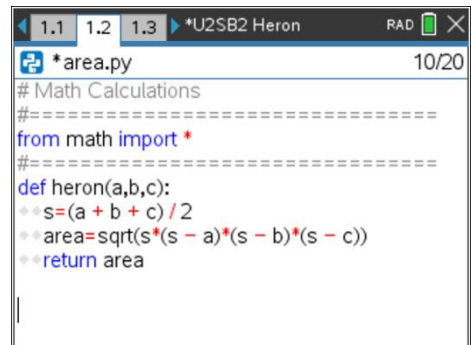
1.1 1.2 1.3 *U2SB2 Heron RAD 10/20
*area.py
# Math Calculations
#=====
from math import *
#=====
def heron(a,b,c):
    s=(a + b + c) / 2
    area=sqrt(s*(s - a)*(s - b)*(s - c))
    return area
    
```

6. 最後に、次のステートメントを使ってメインプログラムを完成します。

input() (3辺a, b, cに3回)

print() (面積をprint(プリント))

次のステップに進む前に、自分で書いてみましょう。



```

1.1 1.2 1.3 *U2SB2 Heron RAD 10/20
*area.py
# Math Calculations
#=====
from math import *
#=====
def heron(a,b,c):
    s=(a + b + c) / 2
    area=sqrt(s*(s - a)*(s - b)*(s - c))
    return area
    
```

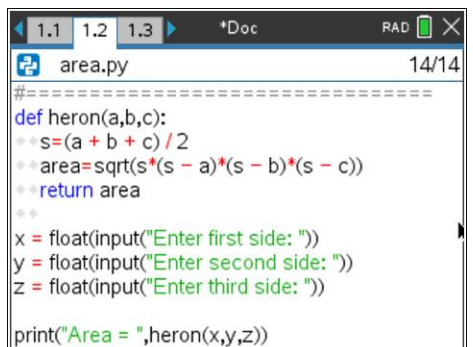
7. 3つの**input()**ステートメントは、3辺の長さの入力を要求し、各辺の長さを数値に変換して(文字式なので)、値を3つの変数x, y, zに格納します。

x = float(input("Enter first side: "))

他の2つはよく似ています。

print()ステートメントは、引数として3つの変数x, y, zを使い、関数heronの値を出力します。

print("Area = ", heron(x,y,z))



```

1.1 1.2 1.3 *Doc RAD 14/14
area.py
#=====
def heron(a,b,c):
    s=(a + b + c) / 2
    area=sqrt(s*(s - a)*(s - b)*(s - c))
    return area
    x = float(input("Enter first side: "))
    y = float(input("Enter second side: "))
    z = float(input("Enter third side: "))
    print("Area = ",heron(x,y,z))
    
```

8. 3, 4, 5など, すでに面積が分かっている数を使ってプログラムをテストします。面積が6であるのはなぜですか。他に, 3つの辺が与えられたとき, 計算しやすい面積がありますか。

作業内容を保存することを忘れないでください。



```
>>>#Running area.py
>>>from area import *
Enter first side: 3
Enter second side: 4
Enter third side: 5
Area = 6.0
>>>
```