

GSET - Introduction to Programming with Python - Summer 2023

Conditionals - Tutorial 6 - Triangles and Quadralaterals

Overview :

Scientists and engineers benefit from the understanding of Trigonometry. In this exercise you are going to use the trigonometry tools available with NUMPY to solve a simple geometry problem and the graphing tools in MATPLOTLIB to display the results. You will also use conditional statements to determine the type of geometry of different examples.

System Requirements:

- **Computer:** A computer is required to complete this tutorial. Any OS should work.
- **Python:** An online Python compiler that supports NumPY and MATPLOTLIB is required([Trinket Python3](#)) or a Python system of your choice with NumPy and MATPLOTLIB installed.

Simple Geometry - Area of a Polygon :

You will be calculating the side length and area of a triangle and a quadrilateral (4 sided polygon). You have been given several formulas on the next page.

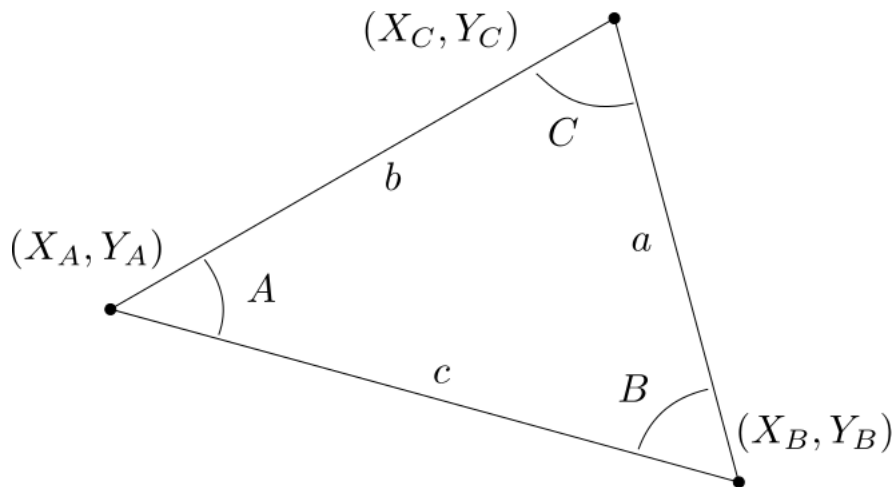
Trigonometric Functions from NumPy :

The trigonometric functions that you need are included in the [NumPy Mathematics functions](#) list. To use NumPy, `import numpy`.

- `numpy.sin(x)` - sine of x
- `numpy.cos(x)` - cosine of x
- `numpy.tan(x)` - tangent of x
- `numpy.asin()` - inverse aka arc-sine
- `numpy.acos()` - inverse aka arc-cosine
- `numpy.atan()` - inverse aka arc-tangent

Equations :

- **Distance Formula:** $D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
- **Heron's Formula(Area):** $A = \sqrt{s(s-a)(s-b)(s-c)}$
- **Perimeter Formula:** $p = a + b + c$ and $s = p/2$



- **Law of Cosines:**

$$a^2 = b^2 + c^2 - 2bc \times \cos(A)$$

$$b^2 = a^2 + c^2 - 2ac \times \cos(B)$$

$$c^2 = a^2 + b^2 - 2ab \times \cos(C)$$

- **Types of Triangles**

- Equilateral
- Right
- Isosceles
- Scalene
- Acute
- Obtuse

Program Minimum Requirements: The program should accomplish the following tasks.

1. Ask the user to enter the 3 vertices of a triangle using the *input()* function. The data should be stored using lists. Give the user instructions for correctly entering the data.
2. Show the vertices as points and the legs of the triangle as lines using MATPLOTLIB. Choose your own markers and linetypes.
3. Calculate the length of each side of the triangle. Show these results in the command window with *print()*.
4. Calculate each of the internal angles of the triangle. Show these results in the command window with *print()*.
5. Calculate the area of the triangle. Show this result in the command window with *print()*.
6. Use one or more conditional statements to determine what type of triangle is formed by the three points. There are six categories shown with the equations. Your program should print the category to the command window.

Optional Advanced Features: Add the following optional features if you have time.

1. Check the input string for the correct number of values and format as requested by *input()* prompt. If there is a problem with the input, print an error message. Skip the calculations and end the program.
2. Check the input string for the correct number of values and format as requested by *input()* prompt. If there is a problem with the input, print an error message. Skip the calculations and repeat the process allowing the user to input a new string. The program should not end until the correct input has been recieved and the calculations have been completed.

Example Code:

```
# Conditionals - GSET - Summer 2023

# prompt the user for input
data=input('''Type the values for the vertices of a triangle
The following format is required. x1, y1, x2, y2, x3, y3:'''')

# get values from user input, store as separate lists
xvals=[float(data.split(',')[0]), float(data.split(',')[2]), float(data.split(',')[4])]
yvals=[float(data.split(',')[1]), float(data.split(',')[3]), float(data.split(',')[5])]
print('xvalues:',xvals)
print('yvalues:',yvals)
```

Testing:

1. Complete the Python code to solve the problem described.
2. Test your code with different inputs. Is the answer correct? How do you know? Are there certain inputs that do not work?
3. Save your code with the download button or use copy and paste. You can view and edit the code in any text editor. Also, save a copy of the program output for your tutorial summary.

Solution Code:

COMING SOON

Tutorial Summary:

Write a brief summary of what you accomplished and what you struggled with the most.

Include the following items in the summary:

- a copy of the output of your program
- a description of what the program does and how to use it

Submission:

Use the appropriate assignment folder on ilearn to submit your program and summary. Submit the following items with your TNTech username in the filenames as shown below.

Files for Tutorial 6 (TNTech Username : twhill21)

- Tutorial Summary: **twhill21_summary6.txt**
- Python Source Code: **twhill21_tutorial6.py**

Tutorial Complete:

Congratulations, after completing *Tutorial_6_Triangles and Quadrilaterals*, you have learned to use conditional statements in Python! You are now ready to start learning about iterative program flow.