triangleLib Documentation

Release 1.0

Uwe Hernandez Acosta

May 30, 2018

CONTENTS

CHAPTER

ONE

EXAMPLE: GUIDE.RST — THE TRIANGLELIB GUIDE

Whether you need to test the properties of triangles, or learn their dimensions, `trianglelib` does it all!

1.1 Special triangles

There are two special kinds of triangle for which `trianglelib` offers special support.

Equilateral triangle All three sides are of equal length.

Isosceles triangle Has at least two sides that are of equal length.

These are supported both by simple methods that are available in the *`trianglelib.utils`* module, and also by a pair of methods of the main *`Triangle`* class itself.

1.2 Triangle dimensions

The library can compute triangle perimeter, area, and can also compare two triangles for equality. Note that it does not matter which side you start with, so long as two triangles have the same three sides in the same order!

```
>>> from trianglelib.shape import Triangle
>>> t1 = Triangle(3, 4, 5)
>>> t2 = Triangle(4, 5, 3)
>>> t3 = Triangle(3, 4, 6)
>>> print t1 == t2
True
>>> print t1 == t3
False
>>> print t1.area()
6.0
>>> print t1.scale(2.0).area()
24.0
```

1.3 Valid triangles

Many combinations of three numbers cannot be the sides of a triangle. Even if all three numbers are positive instead of negative or zero, one of the numbers can still be so large that the shorter two sides could not actually meet to make a closed figure. If c is the longest side, then a triangle is only possible if:

$$a + b > c\partial_x\psi(x)$$

While the documentation for each function in the *utils* module simply specifies a return value for cases that are not real triangles, the *Triangle* class is more strict and raises an exception if your sides lengths are not appropriate:

```
>>> from trianglelib.shape import Triangle
>>> Triangle(1, 1, 3)
Traceback (most recent call last):
  ...
ValueError: one side is too long to make a triangle
```

If you are not sanitizing your user input to verify that the three side lengths they are giving you are safe, then be prepared to trap this exception and report the error to your user.

1.4 The subMod

This sub module is for testing the includings of several sub docs

CHAPTER

TWO

THE TRIANGLELIB API REFERENCE

2.1 The "shape" module

class triangelib.shape.**Triangle**(*a*, *b*, *c*)
A triangle is a three-sided polygon.

<Here is the description of instantiation.>

is_similar(*triangle*)
Return whether this triangle is similar to another triangle.

is_equilateral()
Return whether this triangle is equilateral.

is_isosceles()
Return whether this triangle is isoceles.

perimeter()
Return the perimeter of this triangle.

area()
Return the area of this triangle.

scale(*factor*)
Return a new triangle, *factor* times the size of this one.

2.2 The "utils" module

Routines to test triangle properties without explicit instantiation.

triangelib.utils.**compute_area**(*a*, *b*, *c*)
Return the area of the triangle with side lengths *a*, *b*, and *c*.

If the three lengths provided cannot be the sides of a triangle, then the area 0 is returned.

triangelib.utils.**compute_perimeter**(*a*, *b*, *c*)
Return the perimeer of the triangle with side lengths *a*, *b*, and *c*.

If the three lengths provided cannot be the sides of a triangle, then the perimeter 0 is returned.

triangelib.utils.**is_equilateral**(*a*, *b*, *c*)
Return whether lengths *a*, *b*, and *c* are an equilateral triangle.

triangelib.utils.**is_isosceles**(*a*, *b*, *c*)
Return whether lengths *a*, *b*, and *c* are an isosceles triangle.

trianglelib.utils.**is_triangle**(*a*, *b*, *c*)
Return whether lengths *a*, *b*, *c* can be the sides of a triangle.

CHAPTER

THREE

THE "SUBMOD" MODULE

This sub module is for testing the includings of several sub docs

3.1 The 'subFunc' Module

this is the test function for a sub module

`trianglelib.subMod.subFunc.`**`wrapper`**(*func*)
this function wraps 'func'

PYTHON MODULE INDEX

t

INDEX