Python STL

Release 0.0.1

Apr 21, 2017

Contents

`stl` is a Python library for reading and writing 3D geometry data written in both the binary and ASCII variants of the STL ("STereo Lithography") format.

STL is commonly used in preparing solid figures for 3D printing and other kinds of automatic manufacturing, and is a popular export format for 3D CAD applications.

(This library has nothing to do with the C++ Standard Template Library.)

Contents:

CHAPTER 1

Reading STL Files

STL files can be read using the functions in the main `stl` module.

`stl.`**`read_ascii_file`**(*file*)
Read an STL file in the *ASCII* format.

Takes a `file`-like object (supporting a `read` method) and returns a *`stl.Solid`* object representing the data from the file.

If the file is invalid in any way, raises `stl.ascii.SyntaxError`.

`stl.`**`read_binary_file`**(*file*)
Read an STL file in the *binary* format.

Takes a `file`-like object (supporting a `read` method) and returns a *`stl.Solid`* object representing the data from the file.

If the file is invalid in any way, raises `stl.binary.FormatError`.

`stl.`**`read_ascii_string`**(*data*)
Read geometry from a `str` containing data in the STL *ASCII* format.

This is just a wrapper around `read_ascii_file()` that first wraps the provided string in a `StringIO.StringIO` object.

`stl.`**`read_binary_string`**(*data*)
Read geometry from a `str` containing data in the STL *binary* format.

This is just a wrapper around `read_binary_file()` that first wraps the provided string in a `StringIO.StringIO` object.

CHAPTER 2

Writing STL Files

In order to write an STL file you must first construct a valid *stl.Solid* object containing the data that is to be written.

Files can then be written using *stl.Solid.write_ascii()* and *stl.Solid.write_binary()* respectively.

CHAPTER 3

Data Types

The following data types, with *stl.Solid* as the root, are used to represent data read from or to be written to an STL file.

class stl.**Solid**(*name=None*, *facets=None*)
A solid object; the root element of an STL file.

add_facet(**args*, ***kwargs*)
Append a new facet to the object. Takes the same arguments as the *stl.Facet* type.

write_ascii(*file*)
Write this object to a file in STL *ascii* format.

`file` must be a file-like object (supporting a `write` method), to which the data will be written.

write_binary(*file*)
Write this object to a file in STL *binary* format.

`file` must be a file-like object (supporting a `write` method), to which the data will be written.

class stl.**Facet**(*normal*, *vertices*)
A facet (triangle) from a *stl.Solid*.

class stl.**Vector3d**(*x*, *y*, *z*)
Three-dimensional vector.

Used to represent both normals and vertices of *stl.Facet* objects.

This is a subtype of `tuple`, so can also be treated like a three-element tuple in (`x`, `y`, `z`) order.

x
The X value of the vector, which most applications interpret as the left-right axis.

y
The Y value of the vector, which most applications interpret as the in-out axis.

z
The Z value of the vector, which most applications interpret as the up-down axis.

CHAPTER 4

Indices and tables

- genindex
- search

Index