pangocairohelpers

Release 0.0.1

Jun 14, 2021

Contents

Helper modules for rendering text using Pango and Cairo.

CHAPTER 1

Documentation

1.1 Overview

1.2 API reference

1.2.1 Pango Helpers

class pangocairohelpers.**LayoutClusters**(*layout: pangocffi.layout.Layout*)

A decomposed representation of `pangocffi.Layout` as clusters (in other words `pangocffi.GlyphItem`)

This class is useful in scenarios where one wants to iterate over each individual cluster (commonly a single glyph or character).

Warning: RTL directional text like Arabic or Hebrew is not supported for now.

get_layout() → pangocffi.layout.Layout

Returns the layout on which this instance is based on

get_clusters() → List[pangocffi.glyph_item.GlyphItem]

Returns a list of `GlyphItem` for each cluster in the layout

get_logical_extents() → List[pangocairohelpers.glyph_extent.GlyphExtent]

Returns a list of `GlyphExtent` for each cluster in the layout

get_max_logical_extent() → Optional[pangocairohelpers.extent.Extent]

Returns the extent of the layout itself

1.2.2 Shapely Helpers

Line Helper

Functions to help with a single line segments.

pangocairohelpers.line_helper.**coords_length**(*coord_a: Tuple[float, float], coord_b: Tuple[float, float]*) → float

Parameters

- **coord_a** – the first coordinate
- **coord_b** – the second coordinate

Returns the length between two coordinates

pangocairohelpers.line_helper.**coords_are_left_to_right**(*coord_a: Tuple[float, float], coord_b: Tuple[float, float]*) → Optional[bool]

Parameters

- **coord_a** – the first coordinate
- **coord_b** – the second coordinate

Returns `True` if the coordinates are going left to right, `False` if right to left. If the line is vertical, `None` is returned.

LineString Helper

Functions to help with Shapely's `LineString` class.

pangocairohelpers.line_string_helper.**left_to_right_length**(*line_string: shapely.geometry.linestring.LineString*) → float

Parameters **line_string** – the `LineString` to measure

Returns the length of all the line segments in `line_string` that go left (-x) to right (+x)

pangocairohelpers.line_string_helper.**right_to_left_length**(*line_string: shapely.geometry.linestring.LineString*) → float

Parameters **line_string** – the `LineString` to measure

Returns the length of all the line segments in `line_string` that go right (+x) to left (-x)

pangocairohelpers.line_string_helper.**interpolated_distance_of_point**(*line_string: shapely.geometry.linestring.LineS point: shapely.geometry.point.Point*) → float

Parameters

- **line_string** – the `LineString` to find the distance on
- **point** – the point to find on the line

Returns the interpolation distance to calculate the position of the point on the line string

pangocairohelpers.line_string_helper.**points_at_distance_from_point_on_line_string**(*line_string: shapely.geom point: shapely.geom distance: float*) → List[shapely.

Parameters

- **line_string** – the `LineString` to find points on
- **point** – the circle's center point to find intersections on the line
- **distance** – the circle's radius to find intersections on the line

Returns a list of points that exist in the `line_string` and intersect the circle at `point` with the radius `distance`

pangocairohelpers.line_string_helper.**next_offset_from_offset_in_line_string**(*line_string: shapely.geometry.line current_offset: float, distance: float*) → Optional[float]

Used to find the next point on a line_string that is at a certain distance away from the current point on the line.

Parameters

- **line_string** – the `LineString` to find the offset on
- **current_offset** – the offset to start at
- **distance** – the distance the next offset should be

Returns the next offset that is `distance` units away from the current offset on the `line_string`

pangocairohelpers.line_string_helper.**angles_at_offsets**(*line_string: shapely.geometry.linestring.LineString*) → List[Tuple[float, float]]

Parameters **line_string** – the `LineString` to read values from

Returns a list of angle values, indexed by the offset within the `line_string`

pangocairohelpers.line_string_helper.**angle_at_offset**(*angles_at_offsets_list: List[Tuple[float, float]], offset: float*) → float

Parameters

- **angles_at_offsets_list** – a list of angle values, indexed by the offset
- **offset** – the offset value to look for

Returns the angle at a specific offset in the `angles_at_offsets`

pangocairohelpers.line_string_helper.**reverse**(*line_string: shapely.geometry.linestring.LineString*) → shapely.geometry.linestring.LineString

Todo :param line_string: :return:

pangocairohelpers.line_string_helper.**substring**(*line_string: shapely.geometry.linestring.LineString*, *start: float*, *end: Optional[float] = None*) → Optional[shapely.geometry.linestring.LineString]

Todo :param line_string: :param start: :param end: :return:

1.2.3 TextPath

class pangocairohelpers.text_path.**TextPath**(*line_string: shapely.geometry.linestring.LineString*, *layout: pangocffi.layout.Layout*)

Renders text similar to the behaviour found in SVG's `<textPath>`.

`line_string` behaves as the baseline for the text in the layout.

Multi-line layouts are not supported and will throw an error.

Left-to-right text is assumed.

text_fits() → bool

Todo

Returns Todo

compute_baseline() → Optional[shapely.geometry.linestring.LineString]

Computes the baseline that the text covers

Returns a linestring representing the baseline of the text

compute_boundaries() → Optional[shapely.geometry.multipolygon.MultiPolygon]

Computes the combined glyph extents for the text path

Returns a union of glyph extents

draw(*context: cairocffi.context.Context*)

Todo

Returns Todo

1.3 Changelog

1.3.1 Version 0.1.0

Released on 2019-??-??.

First PyPI release.

Python Module Index

p

Index