pygtkChart

API Documentation

September 28, 2009

Contents

Contents **1**

1 Package pygtk_chart **2**
- 1.1 Modules . . . 2
- 1.2 Variables . . . 2

2 Module pygtk_chart.bar_chart **4**
- 2.1 Functions . . . 4
- 2.2 Variables . . . 4
- 2.3 Class Bar . . . 4
  - 2.3.1 Methods . . . 5
  - 2.3.2 Properties . . . 6
  - 2.3.3 Class Variables . . . 6
- 2.4 Class Grid . . . 7
  - 2.4.1 Methods . . . 7
  - 2.4.2 Properties . . . 9
  - 2.4.3 Class Variables . . . 9
- 2.5 Class BarChart . . . 10
  - 2.5.1 Methods . . . 11
  - 2.5.2 Properties . . . 15
  - 2.5.3 Class Variables . . . 15

3 Module pygtk_chart.basics **16**
- 3.1 Functions . . . 16

4 Module pygtk_chart.chart **18**
- 4.1 Functions . . . 18
- 4.2 Variables . . . 18
- 4.3 Class Chart . . . 19
  - 4.3.1 Methods . . . 20
  - 4.3.2 Properties . . . 23
  - 4.3.3 Class Variables . . . 23
- 4.4 Class Background . . . 24
  - 4.4.1 Methods . . . 24
  - 4.4.2 Properties . . . 26
  - 4.4.3 Class Variables . . . 26
- 4.5 Class Title . . . 26

4.5.1 Methods . . . 27
4.5.2 Properties . . . 27
4.5.3 Class Variables . . . 28
4.6 Class Area . . . 28
4.6.1 Methods . . . 28
4.6.2 Properties . . . 30
4.6.3 Class Variables . . . 30

5 Module pygtk_chart.chart_object 32
5.1 Class ChartObject . . . 32
5.1.1 Methods . . . 32
5.1.2 Properties . . . 34
5.1.3 Class Variables . . . 34

6 Module pygtk_chart.label 35
6.1 Functions . . . 35
6.2 Variables . . . 35
6.3 Class Label . . . 36
6.3.1 Methods . . . 37
6.3.2 Properties . . . 44
6.3.3 Class Variables . . . 44

7 Module pygtk_chart.line_chart 45
7.1 Functions . . . 45
7.2 Variables . . . 47
7.3 Class RangeCalculator . . . 48
7.3.1 Methods . . . 48
7.4 Class LineChart . . . 49
7.4.1 Methods . . . 50
7.4.2 Properties . . . 53
7.4.3 Class Variables . . . 53
7.5 Class Axis . . . 54
7.5.1 Methods . . . 54
7.5.2 Properties . . . 57
7.5.3 Class Variables . . . 57
7.6 Class XAxis . . . 57
7.6.1 Methods . . . 58
7.6.2 Properties . . . 59
7.6.3 Class Variables . . . 59
7.7 Class YAxis . . . 59
7.7.1 Methods . . . 60
7.7.2 Properties . . . 60
7.7.3 Class Variables . . . 61
7.8 Class Grid . . . 61
7.8.1 Methods . . . 62
7.8.2 Properties . . . 64
7.8.3 Class Variables . . . 65
7.9 Class Graph . . . 65
7.9.1 Methods . . . 66
7.9.2 Properties . . . 73
7.9.3 Class Variables . . . 73
7.10 Class Legend . . . 74

7.10.1 Methods . . . 74
7.10.2 Properties . . . 75
7.10.3 Class Variables . . . 76

8 Module pygtk_chart.multi_bar_chart 77
8.1 Variables . . . 77
8.2 Class Bar . . . 77
8.2.1 Methods . . . 78
8.2.2 Properties . . . 78
8.2.3 Class Variables . . . 79
8.3 Class BarGroup . . . 79
8.3.1 Methods . . . 80
8.3.2 Properties . . . 82
8.3.3 Class Variables . . . 82
8.4 Class MultiBarChart . . . 83
8.4.1 Methods . . . 84
8.4.2 Properties . . . 87
8.4.3 Class Variables . . . 88

9 Module pygtk_chart.pie_chart 89
9.1 Functions . . . 89
9.2 Class PieArea . . . 89
9.2.1 Methods . . . 89
9.2.2 Properties . . . 90
9.2.3 Class Variables . . . 90
9.3 Class PieChart . . . 91
9.3.1 Methods . . . 92
9.3.2 Properties . . . 96
9.3.3 Class Variables . . . 97

1 Package pygtk_chart

This package contains four pygtk widgets for drawing simple charts:

- line_chart.LineChart for line charts,
- pie_chart.PieChart for pie charts,
- bar_chart.BarChart for bar charts,
- bar_chart.MultiBarChart for charts with groups of bars.

Version: beta

Author: Sven Festersen, John Dickinson

License: GPL

1.1 Modules

- **bar_chart**: Contains the BarChart widget.
  (Section 2, p. 4)
- **basics**: This module contains simple functions needed by all other modules.
  (Section 3, p. 16)
- **chart**: This is the main module.
  (Section 4, p. 18)
- **chart_object**: This module contains the ChartObject class.
  (Section 5, p. 32)
- **label**: Contains the Label class.
  (Section 6, p. 35)
- **line_chart**: Contains the LineChart widget.
  (Section 7, p. 45)
- **multi_bar_chart**: Contains the MultiBarChart widget.
  (Section 8, p. 77)
- **pie_chart**: Contains the PieChart widget.
  (Section 9, p. 89)

1.2 Variables

Name	Description
__url__	**Value:** `'http://notmyname.github.com/pygtkChart/'`
COLOR_AUTO	**Value:** `0`
COLORS	**Value:** `gdk_color_list_from_file(os.sep.join([os.path.dirname(__f...`
LINE_STYLE_SOLID	**Value:** `0`
LINE_STYLE_DOTTED	**Value:** `1`
LINE_STYLE_DASHED	**Value:** `2`
LINE_STYLE_DASHED_ASY-MMETRIC	**Value:** `3`
POINT_STYLE_CIRCLE	**Value:** `0`
POINT_STYLE_SQUARE	**Value:** `1`

continued on next page

Name	Description
POINT_STYLE_CROSS	**Value:** 2
POINT_STYLE_TRIANGLE_UP	**Value:** 3
POINT_STYLE_TRIANGLE_DOWN	**Value:** 4
POINT_STYLE_DIAMOND	**Value:** 5

2 Module pygtk_chart.bar_chart

Contains the BarChart widget.

Author: John Dickinson (john@johnandkaren.com), Sven Festersen (sven@sven-festersen.de)

2.1 Functions

draw_rounded_rectangle(*context*, *x*, *y*, *width*, *height*, *radius*=0)

Draws a rectangle with rounded corners to context. radius specifies the corner radius in px.

Parameters

`context:` the context to draw on
(type=CairoContext)

`x:` x coordinate of the upper left corner
(type=float)

`y:` y coordinate of the upper left corner
(type=float)

`width:` width of the rectangle in px
(type=float)

`height:` height of the rectangle in px
(type=float)

`radius:` corner radius in px (default: 0)
(type=float.)

2.2 Variables

Name	Description
MODE_VERTICAL	**Value:** `0`
MODE_HORIZONTAL	**Value:** `1`

2.3 Class Bar

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.chart.Area
pygtk_chart.bar_chart.Bar

Known Subclasses: pygtk_chart.multi_bar_chart.Bar

A class that represents a bar on a bar chart.

(section) Properties

The Bar class inherits properties from chart.Area. Additional properties:

- corner-radius (radius of the bar's corners, in px; type: float)

(section) Signals

The Bar class inherits signals from chart.Area.

2.3.1 Methods

__init__(*self*, *name*, *value*, *title=''*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

get_value_label_size(*self*, *context*, *rect*, *mode*, *n*, *bar_padding*)

get_label_size(*self*, *context*, *rect*, *mode*, *n*, *bar_padding*)

set_corner_radius(*self*, *radius*)

Set the radius of the bar's corners in px (default: 0).

Parameters

`radius`: radius of the corners

(type=int in [0, 100].)

get_corner_radius(*self*)

Returns the current radius of the bar's corners in px.

Return Value

int in [0, 100]

Inherited from pygtk_chart.chart.Area(Section 4.6)

get_color(), get_highlighted(), get_label(), get_value(), set_color(), set_highlighted(), set_label(), set_value()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

2.3.2 Properties

Name	Description
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

2.3.3 Class Variables

Name	Description
__gproperties__	**Value:** {"corner-radius":(gobject.TYPE_INT, "bar corner radius", ...
__gtype__	**Value:** <GType pygtk_chart+bar_chart+Bar (168668728)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

2.4 Class Grid

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.bar_chart.Grid

This class represents the grid on BarChart and MultiBarChart widgets.

(section) Properties

bar_chart.Grid inherits properties from ChartObject. Additional properties:

- line-style (the style of the grid lines, type: a line style constant)
- color (the color of the grid lines, type: gtk.gdk.Color)
- show-values (sets whether values should be shown at the grid lines, type: boolean)
- padding (the grid's padding in px, type: int in [0, 100]).

(section) Signals

The Grid class inherits signal from chart_object.ChartObject.

2.4.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_show_values(*self*, *show*)

Set whether values should be shown.

Parameters

`show`: *(type=boolean.)*

get_show_values(*self*)

Returns True if grid values are shown.

Return Value
boolean.

set_color(*self*, *color*)

Set the color of the grid lines.

Parameters
`color:` the grid lines' color
(type=gtk.gdk.Color.)

get_color(*self*)

Returns the current color of the grid lines.

Return Value
gtk.gdk.Color.

set_line_style(*self*, *style*)

Set the style of the grid lines. style has to be one of

- pygtk_chart.LINE_STYLE_SOLID (default)
- pygtk_chart.LINE_STYLE_DOTTED
- pygtk_chart.LINE_STYLE_DASHED
- pygtk_chart.LINE_STYLE_DASHED_ASYMMETRIC

Parameters
`style:` the new line style
(type=one of the constants above.)

get_line_style(*self*)

Returns the current grid's line style.

Return Value
a line style constant.

set_padding(*self*, *padding*)

Set the grid's padding.

Parameters
`padding:` *(type=int in [0, 100].)*

get_padding(*self*)

Returns the grid's padding.

Return Value
int in [0, 100].

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

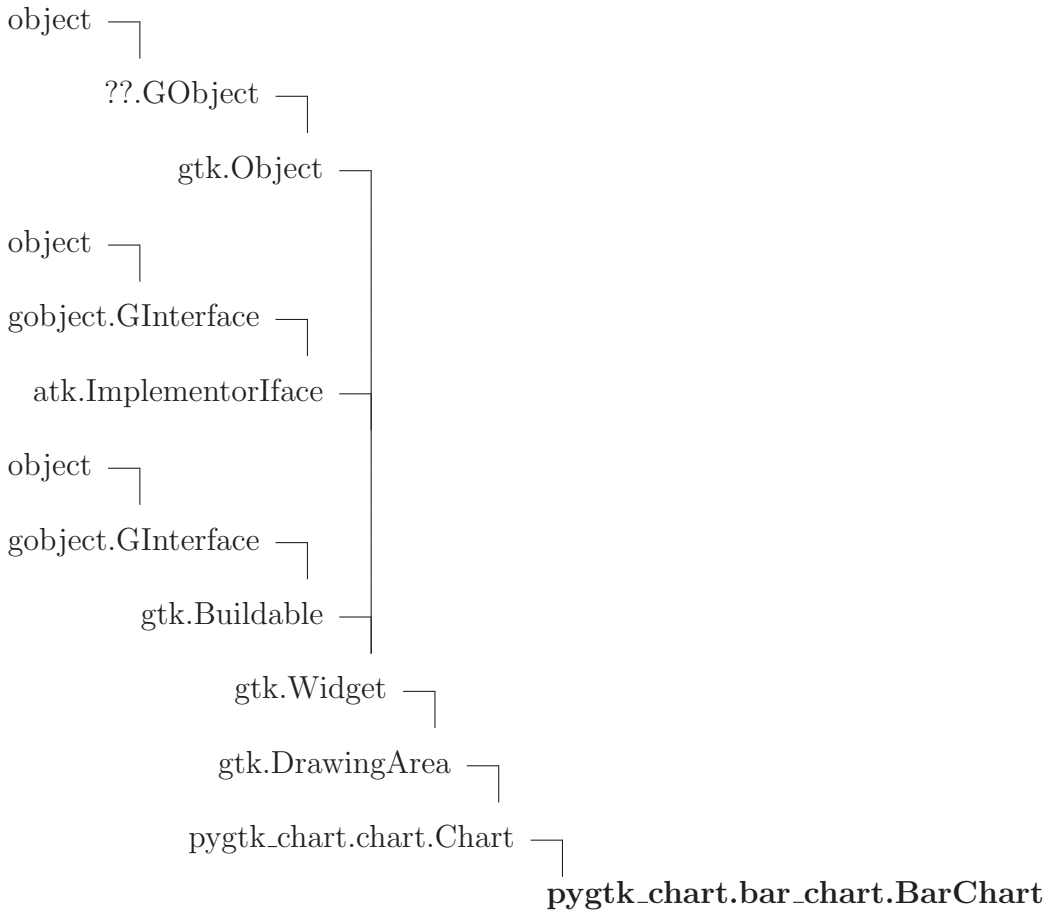
2.4.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

2.4.3 Class Variables

Name	Description
__gproperties__	**Value:** {"show-values":(gobject.TYPE_BOOLEAN, "show values", "Set...
__gtype__	**Value:** <GType pygtk_chart+bar_chart+Grid (168701736)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

2.5 Class BarChart



Known Subclasses: pygtk_chart.multi_bar_chart.MultiBarChart

This is a widget that show a simple BarChart.

(section) Properties

The BarChart class inherits properties from chart.Chart. Additional properites:

- draw-labels (set wether to draw bar label, type: boolean)
- enable-mouseover (set whether to show a mouseover effect, type: boolean)
- mode (the mode of the bar chart, type: one of MODE_VERTICAL, MODE_HORIZONTAL)
- bar-padding (the sace between bars in px, type: int in [0, 100]).

(section) Signals

The BarChart class inherits signals from chart.Chart. Additional signals:

- bar-clicked: emitted when a bar on the bar chart was clicked callback signature: def bar_clicked(chart, bar).

2.5.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart.Chart.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart.Chart.do_set_property

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the queue_draw() method.

Parameters

`context:` The context to draw on.

(type=cairo.Context)

Overrides: gtk.Widget.draw

draw_basics(*self*, *context*, *rect*)

Draw basic things that every plot has (background, title, ...).

Parameters

`context:` The context to draw on.

(type=cairo.Context)

`rect:` A rectangle representing the charts area.

(type=gtk.gdk.Rectangle)

Overrides: pygtk_chart.chart.Chart.draw_basics

add_bar(*self*, *bar*)

set_bar_padding(*self*, *padding*)

Set the space between two bars in px.

Parameters

`padding`: space between bars in px

(type=int in [0, 100].)

get_bar_padding(*self*)

Returns the space between bars in px.

Return Value

int in [0, 100].

set_mode(*self*, *mode*)

Set the mode (vertical or horizontal) of the BarChart. mode has to be bar_chart.MODE_VERTICAL (default) or bar_chart.MODE_HORIZONTAL.

Parameters

`mode`: the new mode of the chart

(type=one of the mode constants above.)

get_mode(*self*)

Returns the current mode of the chart: bar_chart.MODE_VERTICAL or bar_chart.MODE_HORIZONTAL.

Return Value

a mode constant.

set_draw_labels(*self*, *draw*)

Set whether labels should be drawn on bars.

Parameters

`draw`: *(type=boolean.)*

get_draw_labels(*self*)

Returns True if labels are drawn on bars.

Return Value

boolean.

set_enable_mouseover(*self*, *mouseover*)

Set whether a mouseover effect should be shown when the pointer enters a bar.

Parameters

`mouseover`: *(type=boolean.)*

get_enable_mouseover(*self*)

Returns True if the mouseover effect is enabled.

Return Value

boolean.

Inherited from pygtk_chart.chart.Chart(Section 4.3)

export_png(), export_svg(), get_padding(), set_padding()

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(),

error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(), queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), render_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(),

do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

2.5.2 Properties

Name	Description
Inherited from gtk.Widget allocation, name, parent, requisition, saved_state, state, style, window	
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

2.5.3 Class Variables

Name	Description
__gsignals__	**Value:** {"bar-clicked":(gobject.SIGNAL_RUN_LAST, gobject.TYPE_NON...
__gproperties__	**Value:** {"bar-padding":(gobject.TYPE_INT, "bar padding", "The dis...
__gtype__	**Value:** <GType pygtk_chart+bar_chart+BarChart (168715544)>

3 Module pygtk_chart.basics

This module contains simple functions needed by all other modules.

Author: Sven Festersen (sven@sven-festersen.de)

3.1 Functions

is_in_range(*x*, *(xmin, xmax)*)

Use this method to test whether *xmin* <= *x* <= *xmax*.

Parameters

`xmin:` *(type=number)*

`x:` *(type=number)*

`xmax:` *(type=number)*

intersect_ranges(*range_a*, *range_b*)

get_center(*rect*)

Find the center point of a rectangle.

Parameters

`rect:` The rectangle.

(type=gtk.gdk.Rectangle)

Return Value

A (x, y) tuple specifying the center point.

color_gdk_to_cairo(*color*)

Convert a gtk.gdk.Color to cairo color.

Parameters

`color:` the color to convert

(type=gtk.gdk.Color)

Return Value

a color in cairo format.

color_cairo_to_gdk(*r*, *g*, *b*)

color_rgb_to_cairo(*color*)

Convert a 8 bit RGB value to cairo color.

Parameters

`color:` The color to convert.

(type=a triple of integers between 0 and 255)

Return Value

A color in cairo format.

color_html_to_cairo(*color*)

Convert a html (hex) RGB value to cairo color.

Parameters

`color:` The color to convert.

(type=html color string)

Return Value

A color in cairo format.

color_list_from_file(*filename*)

Read a file with one html hex color per line and return a list of cairo colors.

gdk_color_list_from_file(*filename*)

Read a file with one html hex color per line and return a list of gdk colors.

set_context_line_style(*context*, *style*)

The the line style for a context.

4 Module pygtk_chart.chart

(section) Module Contents

This is the main module. It contains the base classes for chart widgets.

- class Chart: base class for all chart widgets.
- class Background: background of a chart widget.
- class Title: title of a chart.

(section) Colors

All colors that pygtkChart uses are gtk.gdk.Colors as used by PyGTK.

Author: Sven Festersen (sven@sven-festersen.de)

4.1 Functions

init_sensitive_areas()

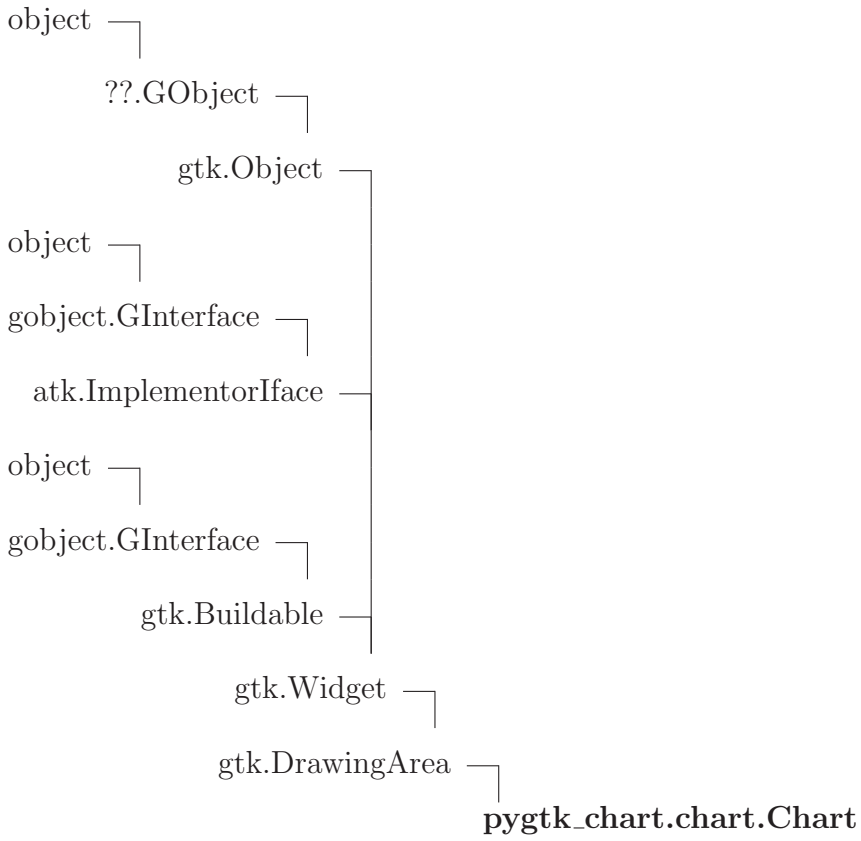
add_sensitive_area(*type*, *coords*, *data*)

get_sensitive_areas(*x*, *y*)

4.2 Variables

Name	Description
COLOR_AUTO	**Value:** `0`
AREA_CIRCLE	**Value:** `0`
AREA_RECTANGLE	**Value:** `1`
CLICK_SENSITIVE_AREAS	**Value:** `[]`

4.3 Class Chart



Known Subclasses: pygtk_chart.bar_chart.BarChart, pygtk_chart.line_chart.LineChart, pygtk_chart.pie_c

This is the base class for all chart widgets.

(section) Properties

The Chart class inherits properties from gtk.DrawingArea. Additional properties:

- padding (the amount of free white space between the chart's content and its border in px, type: int in [0, 100].

(section) Signals

The Chart class inherits signals from gtk.DrawingArea.

4.3.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

do_set_property(*self*, *property*, *value*)

draw_basics(*self*, *context*, *rect*)

Draw basic things that every plot has (background, title, ...).

Parameters

`context`: The context to draw on.

(type=cairo.Context)

`rect`: A rectangle representing the charts area.

(type=gtk.gdk.Rectangle)

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the queue_draw() method.

Parameters

`context`: The context to draw on.

(type=cairo.Context)

Overrides: gtk.Widget.draw

export_svg(*self*, *filename*, *size*=`None`)

Saves the contents of the widget to svg file. The size of the image will be the size of the widget.

Parameters

`filename`: The path to the file where you want the chart to be saved.

(type=string)

`size`: Optional parameter to give the desired height and width of the image.

(type=tuple)

export_png(*self*, *filename*, *size*=`None`)

Saves the contents of the widget to png file. The size of the image will be the size of the widget.

Parameters

`filename`: The path to the file where you want the chart to be saved.

(type=string)

`size`: Optional parameter to give the desired height and width of the image.

(type=tuple)

set_padding(*self*, *padding*)

Set the chart's padding.

Parameters

`padding`: the padding in px

(type=int in [0, 100] (default: 16).)

get_padding(*self*)

Returns the chart's padding.

Return Value

int in [0, 100].

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(),

do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(), error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(), queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), render_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(),

emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(), do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

4.3.2 Properties

Name	Description
Inherited from gtk.Widget allocation, name, parent, requisition, saved_state, state, style, window	
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

4.3.3 Class Variables

Name	Description
__gproperties__	**Value:** {"padding":(gobject.TYPE_INT, "padding", "The chart's pad...
__gtype__	**Value:** <GType pygtk_chart+chart+Chart (168641056)>

4.4 Class Background

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.chart.Background

The background of a chart.

(section) Properties

This class inherits properties from chart_object.ChartObject. Additional properties:

- color (the background color, type: gtk.gdk.Color)
- gradient (the background gradient, type: a pair of gtk.gdk.Color)
- image (path to the background image file, type: string)

(section) Signals

The Background class inherits signals from chart_object.ChartObject.

4.4.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_color(*self*, *color*)

The set_color() method can be used to change the color of the background.

Parameters

`color`: Set the background to be filles with this color.
(type=gtk.gdk.Color)

get_color(*self*)

Returns the background's color.

Return Value
gtk.gdk.Color.

set_gradient(*self*, *color_start*, *color_end*)

Use set_gradient() to define a vertical gradient as the background.

Parameters
`color_start`: The starting (top) color of the gradient.
(type=gtk.gdk.Color)
`color_end`: The ending (bottom) color of the gradient.
(type=gtk.gdk.Color)

get_gradient(*self*)

Returns the gradient of the background or None.

Return Value
A (gtk.gdk.Color, gtk.gdk.Color) tuple or None.

set_image(*self*, *filename*)

The set_image() method sets the background to be filled with an image.

Parameters
`filename`: Path to the file you want to use as background image. If the file does not exists, the background is set to white.
(type=string)

get_image(*self*)

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(),

set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

4.4.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

4.4.3 Class Variables

Name	Description
__gproperties__	**Value:** {`"color":(gobject.TYPE_PYOBJECT, "background color", "The...`
__gtype__	**Value:** <`GType pygtk_chart+chart+Background (168686296)`>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

4.5 Class Title

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.label.Label
pygtk_chart.chart.Title

The title of a chart. The title will be drawn centered at the top of the chart.

(section) Properties

The Title class inherits properties from label.Label.

(section) Signals

The Title class inherits signals from label.Label.

4.5.1 Methods

__init__(*self*, *text*='')
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

Inherited from pygtk_chart.label.Label(Section 6.3)

do_get_property(), do_set_property(), get_allocation(), get_anchor(), get_calculated_dimensions(), get_color(), get_fixed(), get_line_count(), get_max_width(), get_position(), get_real_dimensions(), get_real_position(), get_rotation(), get_size(), get_slant(), get_text(), get_underline(), get_weight(), get_wrap(), set_anchor(), set_color(), set_fixed(), set_max_width(), set_position(), set_rotation(), set_size(), set_slant(), set_text(), set_underline(), set_weight(), set_wrap()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

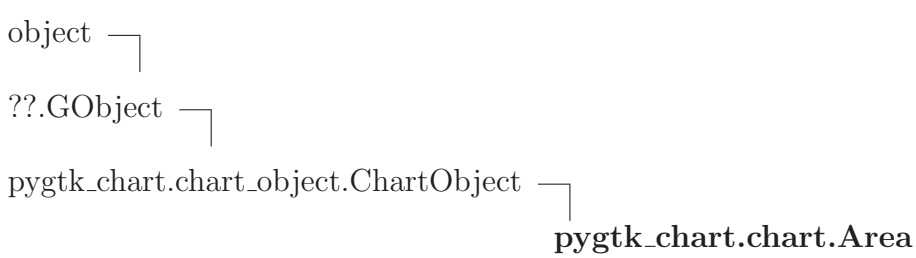
4.5.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

4.5.3 Class Variables

Name	Description
Inherited from pygtk_chart.label.Label (Section 6.3) __gproperties__, __gtype__	
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

4.6 Class Area



Known Subclasses: pygtk_chart.bar_chart.Bar, pygtk_chart.pie_chart.PieArea

This is a base class for classes that represent areas, e.g. the pie_chart.PieArea class and the bar_chart.Bar class.

(section) Properties

The Area class inherits properties from chart_object.ChartObject. Additional properties:

- name (a unique name for the area, type: string, read only)
- value (the value of the area, type: float)
- color (the area's color, type: gtk.gdk.Color)
- label (a label for the area, type: string)
- highlighted (set whether the area should be highlighted, type: boolean).

(section) Signals

The Area class inherits signals from chart_object.ChartObject.

4.6.1 Methods

__**init**__(*self*, *name*, *value*, *title*='')

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_value(*self*, *value*)

Set the value of the area.

Parameters

`value:` *(type=float.)*

get_value(*self*)

Returns the current value of the area.

Return Value

float.

set_color(*self*, *color*)

Set the color of the area.

Parameters

`color:` *(type=gtk.gdk.Color.)*

get_color(*self*)

Returns the current color of the area or COLOR_AUTO.

Return Value

gtk.gdk.Color or COLOR_AUTO.

set_label(*self*, *label*)

Set the label for the area.

Parameters

`label:` the new label

(type=string.)

get_label(*self*)

Returns the current label of the area.

Return Value

string.

set_highlighted(*self*, *highlighted*)

Set whether the area should be highlighted.

Parameters

`highlighted:` *(type=boolean.)*

get_highlighted(*self*)

Returns True if the area is currently highlighted.

Return Value

boolean.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

4.6.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

4.6.3 Class Variables

Name	Description
__gproperties__	**Value:** {"name":(gobject.TYPE_STRING, "area name", "A unique name...
__gtype__	**Value:** <GType pygtk_chart+chart+Area (168686448)>

continued on next page

Name	Description
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

5 Module pygtk_chart.chart_object

This module contains the ChartObject class.

Author: Sven Festersen (sven@sven-festersen.de)

5.1 Class ChartObject

object
??.GObject
pygtk_chart.chart_object.ChartObject

Known Subclasses: pygtk_chart.chart.Area, pygtk_chart.multi_bar_chart.BarGroup, pygtk_chart.line_cha
pygtk_chart.line_chart.Graph, pygtk_chart.line_chart.Grid, pygtk_chart.line_chart.Legend, pygtk_chart.bar_
pygtk_chart.label.Label, pygtk_chart.chart.Background

This is the base class for all things that can be drawn on a chart widget. It emits the signal 'appearance-changed' when it needs to be redrawn.

(section) Properties

ChartObject inherits properties from gobject.GObject. Additional properties:

- visible (sets whether the object should be visible, type: boolean)
- antialias (sets whether the object should be antialiased, type: boolean).

(section) Signals

ChartObject inherits signals from gobject.GObject, Additional signals:

- appearance-changed (emitted if the object needs to be redrawn).

5.1.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

do_set_property(*self*, *property*, *value*)

draw(*self*, *context*, *rect*, **args*)

This method is called by the parent Chart instance. It calls _do_draw.

Parameters

`context:` The context to draw on.

(type=cairo.Context)

`rect:` A rectangle representing the charts area.

(type=gtk.gdk.Rectangle)

set_antialias(*self*, *antialias*)

This method sets the antialiasing mode of the ChartObject. Antialiasing is enabled by default.

Parameters

`antialias:` If False, antialiasing is disabled for this ChartObject.

(type=boolean)

get_antialias(*self*)

Returns True if antialiasing is enabled for the object.

Return Value

boolean.

set_visible(*self*, *visible*)

Use this method to set whether the ChartObject should be visible or not.

Parameters

`visible:` If False, the PlotObject won't be drawn.

(type=boolean)

get_visible(*self*)

Returns True if the object is visble.

Return Value

boolean.

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(),

handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

5.1.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

5.1.3 Class Variables

Name	Description
__gsignals__	**Value:** {"appearance-changed":(gobject.SIGNAL_RUN_LAST, gobject.T...
__gproperties__	**Value:** {"visible":(gobject.TYPE_BOOLEAN, "visibilty of the objec...
__gtype__	**Value:** <GType pygtk_chart+chart_object+ChartObject (168631432)>

6 Module pygtk_chart.label

Contains the Label class.

Author: Sven Festersen (sven@sven-festersen.de)

6.1 Functions

begin_drawing()

finish_drawing()

register_label(*label*)

get_registered_labels()

get_text_pos(*layout*, *pos*, *anchor*, *angle*)

This function calculates the position of bottom left point of the layout respecting the given anchor point.

Return Value
(x, y) pair

6.2 Variables

Name	Description
ANCHOR_BOTTOM_LEFT	**Value:** `0`
ANCHOR_TOP_LEFT	**Value:** `1`
ANCHOR_TOP_RIGHT	**Value:** `2`
ANCHOR_BOTTOM_RIGHT	**Value:** `4`
ANCHOR_CENTER	**Value:** `5`
ANCHOR_TOP_CENTER	**Value:** `6`
ANCHOR_BOTTOM_CENTER	**Value:** `7`
ANCHOR_LEFT_CENTER	**Value:** `8`
ANCHOR_RIGHT_CENTER	**Value:** `9`

continued on next page

Name	Description
UNDERLINE_NONE	**Value:** <enum PANGO_UNDERLINE_NONE of type PangoUnderline>
UNDERLINE_SINGLE	**Value:** <enum PANGO_UNDERLINE_SINGLE of type PangoUnderline>
UNDERLINE_DOUBLE	**Value:** <enum PANGO_UNDERLINE_DOUBLE of type PangoUnderline>
UNDERLINE_LOW	**Value:** <enum PANGO_UNDERLINE_LOW of type PangoUnderline>
STYLE_NORMAL	**Value:** <enum PANGO_STYLE_NORMAL of type PangoStyle>
STYLE_OBLIQUE	**Value:** <enum PANGO_STYLE_OBLIQUE of type PangoStyle>
STYLE_ITALIC	**Value:** <enum PANGO_STYLE_ITALIC of type PangoStyle>
WEIGHT_ULTRALIGHT	**Value:** <enum PANGO_WEIGHT_ULTRALIGHT of type PangoWeight>
WEIGHT_LIGHT	**Value:** <enum PANGO_WEIGHT_LIGHT of type PangoWeight>
WEIGHT_NORMAL	**Value:** <enum PANGO_WEIGHT_NORMAL of type PangoWeight>
WEIGHT_BOLD	**Value:** <enum PANGO_WEIGHT_BOLD of type PangoWeight>
WEIGHT_ULTRABOLD	**Value:** <enum PANGO_WEIGHT_ULTRABOLD of type PangoWeight>
WEIGHT_HEAVY	**Value:** <enum PANGO_WEIGHT_HEAVY of type PangoWeight>
DRAWING_INITIALIZED	**Value:** False
REGISTERED_LABELS	**Value:** []

6.3 Class Label

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.label.Label

Known Subclasses: pygtk_chart.chart.Title

This class is used for drawing all the text on the chart widgets. It uses the pango layout engine.

(section) Properties

The Label class inherits properties from chart_object.ChartObject. Additional properties:

- color (the label's color, type: gtk.gdk.Color)
- text (text to display, type: string)
- position (the label's position, type: pair of float)
- anchor (the anchor that should be used to position the label, type: an anchor constant)
- underline (sets the type of underline, type; an underline constant)
- max-width (the maximum width of the label in px, type: int)
- rotation (angle of rotation in degrees, type: int)
- size (the size of the label's text in px, type: int)
- slant (the font slant, type: a slant style constant)
- weight (the font weight, type: a font weight constant)
- fixed (sets whether the position of the label may be changed dynamicly or not, type: boolean)
- wrap (sets whether the label's text should be wrapped if it's longer than max-width, type: boolean).

(section) Signals

The Label class inherits signals from chart_object.ChartObject.

6.3.1 Methods

__init__(*self*, *position*, *text*, *size*=`None`, *slant*=<`enum PANGO_STYLE_NORMAL of type PangoStyle`>, *weight*=<`enum PANGO_WEIGHT_NORMAL of type PangoWeight`>, *underline*=<`enum PANGO_UNDERLINE_NONE of type PangoUnderline`>, *anchor*=`0`, *max_width*=`99999`, *fixed*=`False`)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

get_calculated_dimensions(*self*, *context*, *rect*)

set_text(*self*, *text*)

Use this method to set the text that should be displayed by the label.

Parameters

`text:` the text to display.

(type=string)

get_text(*self*)

Returns the text currently displayed.

Return Value

string.

set_color(*self*, *color*)

Set the color of the label. color has to be a gtk.gdk.Color.

Parameters

`color:` the color of the label

(type=gtk.gdk.Color.)

get_color(*self*)

Returns the current color of the label.

Return Value

gtk.gdk.Color.

set_position(*self*, *pos*)

Set the position of the label. pos has to be a x,y pair of absolute pixel coordinates on the widget. The position is not the actual position but the position of the Label's anchor point (see `set_anchor` for details).

Parameters

`pos:` new position of the label

(type=pair of (x, y).)

get_position(*self*)

Returns the current position of the label.

Return Value

pair of (x, y).

set_anchor(*self*, *anchor*)

Set the anchor point of the label. The anchor point is the a point on the label's edge that has the position you set with set_position(). anchor has to be one of the following constants:

- label.ANCHOR_BOTTOM_LEFT
- label.ANCHOR_TOP_LEFT
- label.ANCHOR_TOP_RIGHT
- label.ANCHOR_BOTTOM_RIGHT
- label.ANCHOR_CENTER
- label.ANCHOR_TOP_CENTER
- label.ANCHOR_BOTTOM_CENTER
- label.ANCHOR_LEFT_CENTER
- label.ANCHOR_RIGHT_CENTER

The meaning of the constants is illustrated below::

```
    ANCHOR_TOP_LEFT      ANCHOR_TOP_CENTER    ANCHOR_TOP_RIGHT
                     *             *             *
                       ######################
ANCHOR_LEFT_CENTER * #           *           # * ANCHOR_RIGHT_CENTER
                       ######################
                     *             *             *
 ANCHOR_BOTTOM_LEFT    ANCHOR_BOTTOM_CENTER   ANCHOR_BOTTOM_RIGHT
```

The point in the center is of course referred to by constant label.ANCHOR_CENTER.

Parameters

`anchor:` the anchor point of the label

(type=one of the constants described above.)

get_anchor(*self*)

Returns the current anchor point that's used to position the label. See `set_anchor` for details.

Return Value

one of the anchor constants described in `set_anchor`.

set_underline(*self*, *underline*)

Set the underline style of the label. underline has to be one of the following constants:

- label.UNDERLINE_NONE: do not underline the text
- label.UNDERLINE_SINGLE: draw a single underline (the normal underline method)
- label.UNDERLINE_DOUBLE: draw a double underline
- label.UNDERLINE_LOW; draw a single low underline.

Parameters

`underline`: the underline style

(type=one of the constants above.)

get_underline(*self*)

Returns the current underline style. See `set_underline` for details.

Return Value

an underline constant (see `set_underline`).

set_max_width(*self*, *width*)

Set the maximum width of the label in pixels.

Parameters

`width`: the maximum width

(type=integer.)

get_max_width(*self*)

Returns the maximum width of the label.

Return Value

integer.

set_rotation(*self*, *angle*)

Use this method to set the rotation of the label in degrees.

Parameters

`angle`: the rotation angle

(type=integer in [0, 360].)

get_rotation(*self*)

Returns the current rotation angle.

Return Value
integer in [0, 360].

set_size(*self*, *size*)

Set the size of the text in pixels.

Parameters
`size`: size of the text
(type=integer.)

get_size(*self*)

Returns the current size of the text in pixels.

Return Value
integer.

set_slant(*self*, *slant*)

Set the font slant. slat has to be one of the following:

- label.STYLE_NORMAL
- label.STYLE_OBLIQUE
- label.STYLE_ITALIC

Parameters
`slant`: the font slant style
(type=one of the constants above.)

get_slant(*self*)

Returns the current font slant style. See `set_slant` for details.

Return Value
a slant style constant.

set_weight(*self*, *weight*)

Set the font weight. weight has to be one of the following:

- label.WEIGHT_ULTRALIGHT
- label.WEIGHT_LIGHT
- label.WEIGHT_NORMAL
- label.WEIGHT_BOLD
- label.WEIGHT_ULTRABOLD
- label.WEIGHT_HEAVY

Parameters

`weight:` the font weight

(type=one of the constants above.)

get_weight(*self*)

Returns the current font weight. See `set_weight` for details.

Return Value

a font weight constant.

set_fixed(*self*, *fixed*)

Set whether the position of the label should be forced (fixed=True) or if it should be positioned avoiding intersection with other labels.

Parameters

`fixed:` *(type=boolean.)*

get_fixed(*self*)

Returns True if the label's position is forced.

Return Value

boolean

set_wrap(*self*, *wrap*)

Set whether too long text should be wrapped.

Parameters

`wrap:` *(type=boolean.)*

get_wrap(*self*)

Returns True if too long text should be wrapped.

Return Value
boolean.

get_real_dimensions(*self*)

This method returns a pair (width, height) with the dimensions the label was drawn with. Call this method *after* drawing the label.

Return Value
a (width, height) pair.

get_real_position(*self*)

Returns the position of the label where it was really drawn.

Return Value
a (x, y) pair.

get_allocation(*self*)

Returns an allocation rectangle.

Return Value
gtk.gdk.Rectangle.

get_line_count(*self*)

Returns the number of lines.

Return Value
int.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

6.3.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

6.3.3 Class Variables

Name	Description
__gproperties__	**Value:** {"color":(gobject.TYPE_PYOBJECT, "label color", "The colo...
__gtype__	**Value:** <GType pygtk_chart+label+Label (168663000)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

7 Module pygtk_chart.line_chart

Contains the LineChart widget.

Author: Sven Festersen (sven@sven-festersen.de)

7.1 Functions

draw_point(*context*, *x*, *y*, *radius*, *style*)

draw_point_pixbuf(*context*, *x*, *y*, *pixbuf*)

draw_errors(*context*, *rect*, *range_calc*, *x*, *y*, *errors*, *draw_x*, *draw_y*, *xaxis*, *yaxis*, *size*)

separate_data_and_errors(*old_data*)

graph_new_from_function(*func*, *xmin*, *xmax*, *graph_name*, *samples*=`100`, *do_optimize_sampling*=`True`)

Returns a line_chart.Graph with data created from the function y = func(x) with x in [xmin, xmax]. The id of the new graph is graph_name. The parameter samples gives the number of points that should be evaluated in [xmin, xmax] (default: 100). If do_optimize_sampling is True (default) additional points will be evaluated to smoothen the curve.

Parameters

`func:`	the function to evaluate *(type=a function)*
`xmin:`	the minimum x value to evaluate *(type=float)*
`xmax:`	the maximum x value to evaluate *(type=float)*
`graph_name:`	a unique name for the new graph *(type=string)*
`samples:`	number of samples *(type=int)*
`do_optimize_sampling:`	set whether to add additional points *(type=boolean)*

Return Value

line_chart.Graph

optimize_sampling(*func*, *data*)

graph_new_from_file(*filename*, *graph_name*, *x_col*=0, *y_col*=1, *xerror_col*=-1, *yerror_col*=-1)

Returns a line_chart.Graph with point taken from data file filename. The id of the new graph is graph_name.

Data file format: The columns in the file have to be separated by tabs or one or more spaces. Everything after '#' is ignored (comment).

Use the parameters x_col and y_col to control which columns to use for plotting. By default, the first column (x_col=0) is used for x values, the second (y_col=1) is used for y values.

The parameters xerror_col and yerror_col should point to the column in which the x/y error values are. If you do not want to provide x or y error data, omit the paramter or set it to -1 (default).

Parameters

`filename:` path to the data file
(type=string)

`graph_name:` a unique name for the graph
(type=string)

`x_col:` the number of the column to use for x values
(type=int)

`y_col:` the number of the column to use for y values
(type=int)

`xerror_col:` index of the column for x error values
(type=int)

`yerror_col:` index of the column for y error values
(type=int)

Return Value

line_chart.Graph

7.2 Variables

Name	Description
RANGE_AUTO	**Value:** `0`
GRAPH_PADDING	**Value:** `0.0666666666667`
GRAPH_POINTS	**Value:** `1`
GRAPH_LINES	**Value:** `2`

continued on next page

Name	Description
GRAPH_BOTH	**Value:** `3`
COLOR_AUTO	**Value:** `4`
POSITION_AUTO	**Value:** `5`
POSITION_LEFT	**Value:** `6`
POSITION_RIGHT	**Value:** `7`
POSITION_BOTTOM	**Value:** `6`
POSITION_TOP	**Value:** `7`
POSITION_TOP_RIGHT	**Value:** `8`
POSITION_BOTTOM_RIGHT	**Value:** `9`
POSITION_BOTTOM_LEFT	**Value:** `10`
POSITION_TOP_LEFT	**Value:** `11`

7.3 Class RangeCalculator

This helper class calculates ranges. It is used by the LineChart widget internally, there is no need to create an instance yourself.

7.3.1 Methods

__init__(*self*)

add_graph(*self*, *graph*)

get_ranges(*self*, *xaxis*, *yaxis*)

set_xrange(*self*, *xrange*)

set_yrange(*self*, *yrange*)

get_absolute_zero(*self*, *rect*, *xaxis*, *yaxis*)

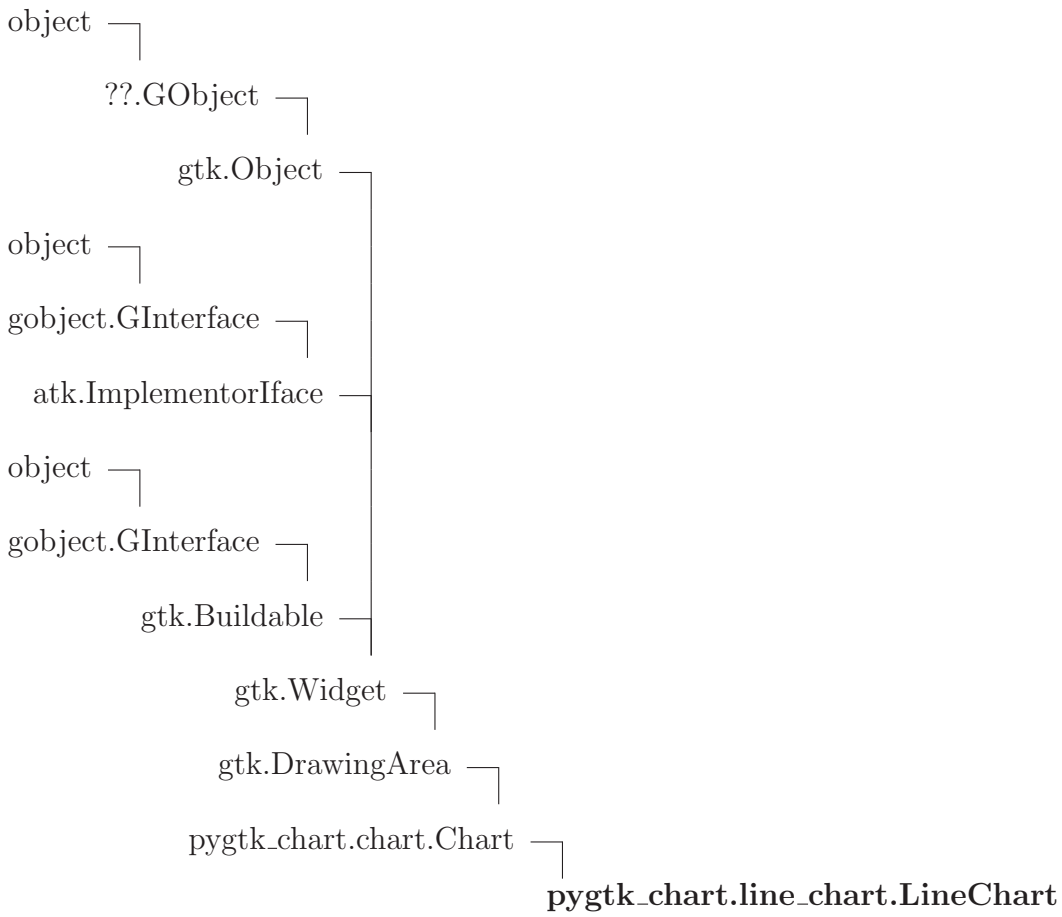
get_absolute_point(*self*, *rect*, *x*, *y*, *xaxis*, *yaxis*)

prepare_tics(*self*, *rect*, *xaxis*, *yaxis*)

get_xtics(*self*, *rect*)

get_ytics(*self*, *rect*)

7.4 Class LineChart



A widget that shows a line chart. The following attributes can be accessed:

- LineChart.background (inherited from chart.Chart)
- LineChart.title (inherited from chart.Chart)
- LineChart.graphs (a dict that holds the graphs identified by their name)
- LineChart.grid
- LineChart.xaxis
- LineChart.yaxis

(section) Properties

LineChart inherits properties from chart.Chart.

(section) Signals

The LineChart class inherits signals from chart.Chart. Additional chart:

- datapoint-clicked (emitted if a datapoint is clicked)
- datapoint-hovered (emitted if a datapoint is hovered with the mouse pointer)

Callback signature for both signals: def callback(linechart, graph, (x, y))

7.4.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

__iter__(*self*)

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the queue_draw() method.

Parameters

`context:` The context to draw on.

(type=cairo.Context)

Overrides: gtk.Widget.draw

add_graph(*self*, *graph*)

Add a graph object to the plot.

Parameters

`graph:` The graph to add.

(type=line_chart.Graph)

remove_graph(*self*, *name*)

Remove a graph from the plot.

Parameters

`name:` The name of the graph to remove.

(type=string)

set_xrange(*self*, *xrange*)

Set the visible xrange. xrange has to be a pair: (xmin, xmax) or RANGE_AUTO. If you set it to RANGE_AUTO, the visible range will be calculated.

Parameters

`xrange`: The new xrange.

(type=pair of numbers)

get_xrange(*self*)

set_yrange(*self*, *yrange*)

Set the visible yrange. yrange has to be a pair: (ymin, ymax) or RANGE_AUTO. If you set it to RANGE_AUTO, the visible range will be calculated.

Parameters

`yrange`: The new yrange.

(type=pair of numbers)

get_yrange(*self*)

Inherited from pygtk_chart.chart.Chart(Section 4.3)

do_get_property(), do_set_property(), draw_basics(), export_png(), export_svg(), get_padding(), set_padding()

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(),

do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(), error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(), queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), render_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(),

connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(), do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

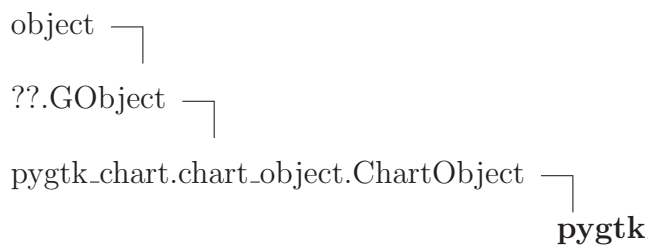
7.4.2 Properties

Name	Description
Inherited from gtk.Widget allocation, name, parent, requisition, saved_state, state, style, window	
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

7.4.3 Class Variables

Name	Description
__gsignals__	**Value:** {"datapoint-clicked":(gobject.SIGNAL_RUN_LAST, gobject.TY...
__gtype__	**Value:** <GType pygtk_chart+line_chart+LineChart (169249432)>
Inherited from pygtk_chart.chart.Chart (Section 4.3) __gproperties__	

7.5 Class Axis



Known Subclasses: pygtk_chart.line_chart.XAxis, pygtk_chart.line_chart.YAxis

This class represents an axis on the line chart.

(section) Properties

The Axis class inherits properties from chart_object.ChartObject. Additional properties:

- label (a label for the axis, type: string)
- show-label (sets whether the axis' label should be shown, type: boolean)
- position (position of the axis, type: an axis position constant)
- show-tics (sets whether tics should be shown at the axis, type: boolean)
- show-tic-lables (sets whether labels should be shown at the tics, type: boolean)
- tic-format-function (a function that is used to format the tic labels, default: str)
- logarithmic (sets whether the axis should use a logarithmic scale, type: boolean).

(section) Signals

The Axis class inherits signals from chart_object.ChartObject.

7.5.1 Methods

__init__(*self*, *range_calc*, *label*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_label(*self*, *label*)

Set the label of the axis.

Parameters

`label:` new label

(type=string.)

get_label(*self*)

Returns the current label of the axis.

Return Value

string.

set_show_label(*self*, *show*)

Set whether to show the axis' label.

Parameters

`show:` *(type=boolean.)*

get_show_label(*self*)

Returns True if the axis' label is shown.

Return Value

boolean.

set_position(*self*, *pos*)

Set the position of the axis. pos hast to be one these constants:
POSITION_AUTO, POSITION_BOTTOM, POSITION_LEFT,
POSITION_RIGHT, POSITION_TOP.

get_position(*self*)

Returns the position of the axis. (see set_position for details).

set_show_tics(*self*, *show*)

Set whether to draw tics at the axis.

Parameters

`show:` *(type=boolean.)*

get_show_tics(*self*)

Returns True if tics are drawn.

Return Value
boolean.

set_show_tic_labels(*self*, *show*)

Set whether to draw tic labels. Labels are only drawn if tics are drawn.

Parameters
`show:` *(type=boolean.)*

get_show_tic_labels(*self*)

Returns True if tic labels are shown.

Return Value
boolean.

set_tic_format_function(*self*, *func*)

Use this to set the function that should be used to label the tics. The function should take a number as the only argument and return a string. Default: str

Parameters
`func:` *(type=function.)*

get_tic_format_function(*self*)

Returns the function currently used for labeling the tics.

set_logarithmic(*self*, *log*)

Set whether the axis should use logarithmic (base 10) scale.

Parameters
`log:` *(type=boolean.)*

get_logarithmic(*self*)

Returns True if the axis uses logarithmic scale.

Return Value
boolean.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

7.5.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

7.5.3 Class Variables

Name	Description
__gproperties__	**Value:** {"label":(gobject.TYPE_STRING, "axis label", "The label o...
__gtype__	**Value:** <GType pygtk_chart+line_chart+Axis (168640912)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

7.6 Class XAxis

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.line_chart.Axis
pygtk_chart.line_chart.XAxis

This class represents the xaxis. It is used by the LineChart widget internally, there is no

need to create an instance yourself.

(section) Properties

The XAxis class inherits properties from Axis.

(section) Signals

The XAxis class inherits signals from Axis.

7.6.1 Methods

__init__(*self*, *range_calc*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

draw(*self*, *context*, *rect*, *yaxis*)

This method is called by the parent Plot instance. It calls _do_draw.

Parameters

`context:` The context to draw on.

`rect:` A rectangle representing the charts area.

Overrides: pygtk_chart.chart_object.ChartObject.draw

Inherited from pygtk_chart.line_chart.Axis(Section 7.5)

do_get_property(), do_set_property(), get_label(), get_logarithmic(), get_position(), get_show_label(), get_show_tic_labels(), get_show_tics(), get_tic_format_function(), set_label(), set_logarithmic(), set_position(), set_show_label(), set_show_tic_labels(), set_show_tics(), set_tic_format_function()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

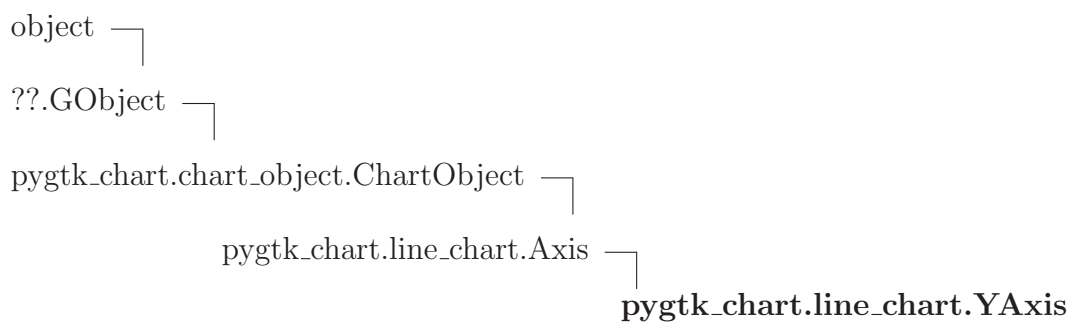
7.6.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

7.6.3 Class Variables

Name	Description
Inherited from pygtk_chart.line_chart.Axis (Section 7.5)	
__gproperties__, __gtype__	
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

7.7 Class YAxis



This class represents the yaxis. It is used by the LineChart widget internally, there is no need to create an instance yourself.

(section) Properties

The YAxis class inherits properties from Axis.

(section) Signals

The YAxis class inherits signals from Axis.

7.7.1 Methods

__init__(*self*, *range_calc*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

draw(*self*, *context*, *rect*, *xaxis*)

This method is called by the parent Plot instance. It calls _do_draw.

Parameters

`context:` The context to draw on.

`rect:` A rectangle representing the charts area.

Overrides: pygtk_chart.chart_object.ChartObject.draw

Inherited from pygtk_chart.line_chart.Axis(Section 7.5)

do_get_property(), do_set_property(), get_label(), get_logarithmic(), get_position(), get_show_label(), get_show_tic_labels(), get_show_tics(), get_tic_format_function(), set_label(), set_logarithmic(), set_position(), set_show_label(), set_show_tic_labels(), set_show_tics(), set_tic_format_function()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

7.7.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	

continued on next page

Name	Description
Inherited from object	
__class__	

7.7.3 Class Variables

Name	Description
Inherited from pygtk_chart.line_chart.Axis (Section 7.5)	
__gproperties__, __gtype__	
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

7.8 Class Grid

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.line_chart.Grid

A class representing the grid of the chart. It is used by the LineChart widget internally, there is no need to create an instance yourself.

(section) Properties

The Grid class inherits properties from chart_object.ChartObject. Additional properties:

- show-horizontal (sets whther to show horizontal grid lines, type: boolean)
- show-vertical (sets whther to show vertical grid lines, type: boolean)
- color (the color of the grid lines, type: gtk.gdk.Color)
- line-style-horizontal (the line style of the horizontal grid lines, type: a line style constant)
- line-style-vertical (the line style of the vertical grid lines, type: a line style constant).

(section) Signals

The Grid class inherits signals from chart_object.ChartObject.

7.8.1 Methods

__init__(*self*, *range_calc*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_draw_horizontal_lines(*self*, *draw*)

Set whether to draw horizontal grid lines.

Parameters

`draw:` *(type=boolean.)*

get_draw_horizontal_lines(*self*)

Returns True if horizontal grid lines are drawn.

Return Value

boolean.

set_draw_vertical_lines(*self*, *draw*)

Set whether to draw vertical grid lines.

Parameters

`draw:` *(type=boolean.)*

get_draw_vertical_lines(*self*)

Returns True if vertical grid lines are drawn.

Return Value

boolean.

set_color(*self*, *color*)

Set the color of the grid.

Parameters

`color:` The new color of the grid.

(type=gtk.gdk.Color)

get_color(*self*)

Returns the color of the grid.

Return Value

gtk.gdk.Color.

set_line_style_horizontal(*self*, *style*)

Set the line style of the horizontal grid lines. style has to be one of these constants:

- pygtk_chart.LINE_STYLE_SOLID (default)
- pygtk_chart.LINE_STYLE_DOTTED
- pygtk_chart.LINE_STYLE_DASHED
- pygtk_chart.LINE_STYLE_DASHED_ASYMMETRIC.

Parameters

`style:` the new line style

(type=one of the constants above.)

get_line_style_horizontal(*self*)

Returns ths current horizontal line style.

Return Value

a line style constant.

set_line_style_vertical(*self*, *style*)

Set the line style of the vertical grid lines. style has to be one of these constants:

- pygtk_chart.LINE_STYLE_SOLID (default)
- pygtk_chart.LINE_STYLE_DOTTED
- pygtk_chart.LINE_STYLE_DASHED
- pygtk_chart.LINE_STYLE_DASHED_ASYMMETRIC.

Parameters

`style`: the new line style

(type=one of the constants above.)

get_line_style_vertical(*self*)

Returns ths current vertical line style.

Return Value

a line style constant.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

7.8.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

7.8.3 Class Variables

Name	Description
__gproperties__	**Value:** {"show-horizontal":(gobject.TYPE_BOOLEAN, "show horizonta...
__gtype__	**Value:** <GType pygtk_chart+line_chart+Grid (170084616)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

7.9 Class Graph

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.line_chart.Graph

This class represents a graph or the data you want to plot on your LineChart widget.

(section) Properties

The Graph class inherits properties from chart_object.ChartObject. Additional properties:

- name (a unique id for the graph, type: string, read only)
- title (the graph's title, type: string)
- color (the graph's color, type: gtk.gdk.Color)
- type (graph type, type: a graph type constant)
- point-size (radius of the datapoints in px, type: int in [1, 100])
- fill-to (set how to fill space under the graph, type: None, Graph or float)
- fill-color (the color of the filling, type: gtk.gdk.Color)
- fill-opacity (the opacity of the filling, type: float in [0, 1])
- show-values (sets whether y values should be shown at the datapoints, type: boolean)
- show-title (sets whether ot show the graph's title, type: boolean)
- line-style (the graph's line style, type: a line style constant)
- point-style (the graph's datapoints' point style, type: a point style constant)
- clickable (sets whether datapoints are sensitive for clicks, type: boolean)
- show-xerrors (sets whether x errors should be shown if error data is available, type:

boolean)

- show-yerrors (sets whether y errors should be shown if error data is available, type: boolean).

(section) Signals

The Graph class inherits signals from chart_object.ChartObject.

7.9.1 Methods

__init__(*self*, *name*, *title*, *data*)

Create a new graph instance. data should be a list of x,y pairs. If you want to provide error data for a datapoint, the tuple for that point has to be (x, y, xerror, yerror). If you want only one error, set the other to zero. You can mix datapoints with and without error data in data.

Parameters

`name:` A unique name for the graph. This could be everything. It's just a name used internally for identification. You need to know this if you want to access or delete a graph from a chart.

(type=string)

`title:` The graphs title. This can be drawn on the chart.

(type=string)

`data:` This is the data you want to be visualized. For detail see description above.

(type=list (see above))

Overrides: object.__init__

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

has_something_to_draw(*self*)

get_x_range(*self*)

Get the the endpoints of the x interval.

Return Value
pair of numbers

get_y_range(*self*)

Get the the endpoints of the y interval.

Return Value
pair of numbers

get_name(*self*)

Get the name of the graph.

Return Value
string

get_title(*self*)

Returns the title of the graph.

Return Value
string

set_title(*self*, *title*)

Set the title of the graph.

Parameters
`title`: The graph's new title.
(type=string)

set_range_calc(*self*, *range_calc*)

get_color(*self*)

Returns the current color of the graph or COLOR_AUTO.

Return Value
gtk.gdk.Color or COLOR_AUTO.

set_color(*self*, *color*)

Set the color of the graph. If set to COLOR_AUTO, the color will be choosen dynamicly.

Parameters

`color`: The new color of the graph.

(type=gtk.gdk.Color)

get_type(*self*)

Returns the type of the graph.

Return Value

a type constant (see set_type() for details)

set_type(*self*, *type*)

Set the type of the graph to one of these:

- GRAPH_POINTS: only show points
- GRAPH_LINES: only draw lines
- GRAPH_BOTH: draw points and lines, i.e. connect points with lines

Parameters

`type`: One of the constants above.

get_point_size(*self*)

Returns the radius of the data points.

Return Value

a poisitive integer

set_point_size(*self*, *size*)

Set the radius of the drawn points.

Parameters

`size`: The new radius of the points.

(type=a positive integer in [1, 100])

get_fill_to(*self*)

The return value of this method depends on the filling under the graph. See set_fill_to() for details.

set_fill_to(*self*, *fill_to*)

Use this method to specify how the space under the graph should be filled. fill_to has to be one of these:

- None: dont't fill the space under the graph.
- int or float: fill the space to the value specified (setting fill_to=0 means filling the space between graph and xaxis).
- a Graph object: fill the space between this graph and the graph given as the argument.

The color of the filling is the graph's color with 30% opacity.

Parameters

`fill_to`: *(type=one of the possibilities listed above.)*

get_fill_color(*self*)

Returns the color that is used to fill space under the graph or COLOR_AUTO.

Return Value

gtk.gdk.Color or COLOR_AUTO.

set_fill_color(*self*, *color*)

Set which color should be used when filling the space under a graph. If color is COLOR_AUTO, the graph's color will be used.

Parameters

`color`: *(type=gtk.gdk.Color or COLOR_AUTO.)*

get_fill_opacity(*self*)

Returns the opacity that is used to fill space under the graph.

set_fill_opacity(*self*, *opacity*)

Set which opacity should be used when filling the space under a graph. The default is 0.3.

Parameters

`opacity`: *(type=float in [0, 1].)*

get_show_values(*self*)

Returns True if y values are shown.

Return Value

boolean

set_show_values(*self*, *show*)

Set whether the y values should be shown (only if graph type is GRAPH_POINTS or GRAPH_BOTH).

Parameters

`show:` *(type=boolean)*

get_show_title(*self*)

Returns True if the title of the graph is shown.

Return Value

boolean.

set_show_title(*self*, *show*)

Set whether to show the graph's title or not.

Parameters

`show:` *(type=boolean.)*

add_data(*self*, *data_list*)

Add data to the graph. data_list should be a list of x,y pairs. If you want to provide error data for a datapoint, the tuple for that point has to be (x, y, xerror, yerror). If you want only one error, set the other to zero. You can mix datapoints with and without error data in data_list.

Parameters

`data_list:` *(type=a list (see above).)*

get_data(*self*)

Returns the data of the graph.

Return Value

a list of x, y pairs.

Overrides: ??.GObject.get_data

set_line_style(*self*, *style*)

Set the line style that should be used for drawing the graph (if type is line_chart.GRAPH_LINES or line_chart.GRAPH_BOTH). style has to be one of these constants:

- pygtk_chart.LINE_STYLE_SOLID (default)
- pygtk_chart.LINE_STYLE_DOTTED
- pygtk_chart.LINE_STYLE_DASHED
- pygtk_chart.LINE_STYLE_DASHED_ASYMMETRIC.

Parameters

`style`: the new line style

(type=one of the line style constants above.)

get_line_style(*self*)

Returns the current line style for the graph (see `set_line_style` for details).

Return Value

a line style constant.

set_point_style(*self*, *style*)

Set the point style that should be used when drawing the graph (if type is line_chart.GRAPH_POINTS or line_chart.GRAPH_BOTH). For style you can use one of these constants:

- pygtk_chart.POINT_STYLE_CIRCLE (default)
- pygtk_chart.POINT_STYLE_SQUARE
- pygtk_chart.POINT_STYLE_CROSS
- pygtk_chart.POINT_STYLE_TRIANGLE_UP
- pygtk_chart.POINT_STYLE_TRIANGLE_DOWN
- pygtk_chart.POINT_STYLE_DIAMOND

style can also be a gtk.gdk.Pixbuf that should be used as point.

Parameters

`style`: the new point style

(type=one of the cosnatnts above or gtk.gdk.Pixbuf.)

get_point_style(*self*)

Returns the current point style. See `set_point_style` for details.

Return Value
a point style constant or gtk.gdk.Pixbuf.

set_clickable(*self*, *clickable*)

Set whether the datapoints of the graph should be clickable (only if the datapoints are shown). If this is set to True, the LineChart will emit the signal 'datapoint-clicked' when a datapoint was clicked.

Parameters
`clickable:` *(type=boolean.)*

get_clickable(*self*)

Returns True if the datapoints of the graph are clickable.

Return Value
boolean.

set_show_xerrors(*self*, *show*)

Use this method to set whether x-errorbars should be shown if error data is available.

Parameters
`show:` *(type=boolean.)*

get_show_xerrors(*self*)

Returns True if x-errorbars should be drawn if error data is available.

Return Value
boolean.

set_show_yerrors(*self*, *show*)

Use this method to set whether y-errorbars should be shown if error data is available.

Parameters
`show:` *(type=boolean.)*

get_show_yerrors(*self*)

Returns True if y-errorbars should be drawn if error data is available.

Return Value
boolean.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

7.9.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

7.9.3 Class Variables

Name	Description
__gproperties__	**Value:** {"name":(gobject.TYPE_STRING, "graph id", "The graph's un...
__gtype__	**Value:** <GType pygtk_chart+line_chart+Graph (170159304)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1)	
__gsignals__	

7.10 Class Legend

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.line_chart.Legend

This class represents a legend on a line chart.

(section) Properties

The Legend class inherits properties from chart_object.ChartObject. Additional properties:

- position (the legend's position on the chart, type: a corner position constant).

(section) Signals

The Legend class inherits signals from chart_object.ChartObject.

7.10.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)
Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)
Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

set_position(*self*, *position*)

Set the position of the legend. position has to be one of these position constants:

- line_chart.POSITION_TOP_RIGHT (default)
- line_chart.POSITION_BOTTOM_RIGHT
- line_chart.POSITION_BOTTOM_LEFT
- line_chart.POSITION_TOP_LEFT

Parameters

`position`: the legend's position

(type=one of the constants above.)

get_position(*self*)

Returns the position of the legend. See `set_position` for details.

Return Value

a position constant.

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

7.10.2 Properties

Name	Description
Inherited from ??.GObject	
__grefcount__	
Inherited from object	
__class__	

7.10.3 Class Variables

Name	Description
__gproperties__	**Value:** {"position":(gobject.TYPE_INT, "legend position", "Positi...
__gtype__	**Value:** <GType pygtk_chart+line_chart+Legend (170094128)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

8 Module pygtk_chart.multi_bar_chart

Contains the MultiBarChart widget.

Author: Sven Festersen (sven@sven-festersen.de)

8.1 Variables

Name	Description
MODE_VERTICAL	**Value:** `0`
MODE_HORIZONTAL	**Value:** `1`
COLOR_AUTO	**Value:** `0`
COLORS	**Value:** `gdk_color_list_from_file(os.sep.join([os.path.dirname(__f...`

8.2 Class Bar

object
??.GObject
pygtk_chart.chart_object.ChartObject
pygtk_chart.chart.Area
pygtk_chart.bar_chart.Bar
pygtk_chart.multi_bar_chart.Bar

This is a special version of the bar_chart.Bar class that draws the bars on a MultiBarChart widget.

(section) Properties

This class inherits properties from bar_chart.Bar.

(section) Signals

This class inherits signals from bar_chart.Bar.

8.2.1 Methods

__init__(*self*, *name*, *value*, *title=''*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

get_value_label_size(*self*, *context*, *rect*, *mode*, *bar_count*, *n*, *group_padding*, *bar_padding*)

Overrides: pygtk_chart.bar_chart.Bar.get_value_label_size

get_label_size(*self*, *context*, *rect*, *mode*, *bar_count*, *n*, *group_padding*, *bar_padding*, *label_rotation*)

Overrides: pygtk_chart.bar_chart.Bar.get_label_size

Inherited from pygtk_chart.bar_chart.Bar(Section 2.3)

do_get_property(), do_set_property(), get_corner_radius(), set_corner_radius()

Inherited from pygtk_chart.chart.Area(Section 4.6)

get_color(), get_highlighted(), get_label(), get_value(), set_color(), set_highlighted(), set_label(), set_value()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

8.2.2 Properties

continued on next page

Name	Description

Name	Description
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

8.2.3 Class Variables

Name	Description
Inherited from pygtk_chart.bar_chart.Bar (Section 2.3) __gproperties__, __gtype__	
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

8.3 Class BarGroup

object
└
??.GObject
└
pygtk_chart.chart_object.ChartObject
└
pygtk_chart.multi_bar_chart.BarGroup

This class represents a group of bars on the MultiBarChart widget.

(section) Properties

This class has the following properties:

- name (a unique identifier for the group, type: string)
- title (a title for the group, type: string)
- bar-padding (the space between two bars of the group in px, type: int in [0, 100])
- bars (a list of the bars in the group, read only)
- maximum-value (the maximum value of the bars in the group, read only)
- bar-count (the number of bars in the group, read only).

(section) Signals

The BarGroup class inherits signals from chart_object.ChartObject.

8.3.1 Methods

__init__(*self*, *name*, *title=''*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart_object.ChartObject.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart_object.ChartObject.do_set_property

get_bar_count(*self*)

Returns the number of bars in this group.

Return Value
int in [0, 100].

get_maximum_value(*self*)

Returns the maximum value of the bars in this group.

Return Value
float.

get_bars(*self*)

Returns a list of the bars in this group.

Return Value
list of multi_bar_chart.Bar.

get_name(*self*)

Returns the name (a unique identifier) of this group.

Return Value
string.

set_title(*self*, *title*)

Set the title of the group.

Parameters

`title`: the new title

(type=string.)

get_title(*self*)

Returns the title of the group.

Return Value

string.

get_label(*self*)

Alias for get_title.

Return Value

string.

set_bar_padding(*self*, *padding*)

Set the distance between two bars in this group (in px).

Parameters

`padding`: the padding in px

(type=int in [0, 100].)

get_bar_padding(*self*)

Returns the distance of two bars in the group (in px).

Return Value

int in [0, 100].

add_bar(*self*, *bar*)

Add a bar to the group.

Parameters

`bar`: the bar to add

(type=multi_bar_chart.Bar.)

get_value_label_size(*self*, *context*, *rect*, *mode*, *bar_count*, *n*, *group_padding*, *bar_padding*)

get_label_size(*self*, *context*, *rect*, *mode*, *bar_count*, *n*, *group_padding*, *bar_padding*, *label_rotation*)

get_group_label_size(*self*, *context*, *rect*, *mode*, *rotate_label_horizontal*)

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

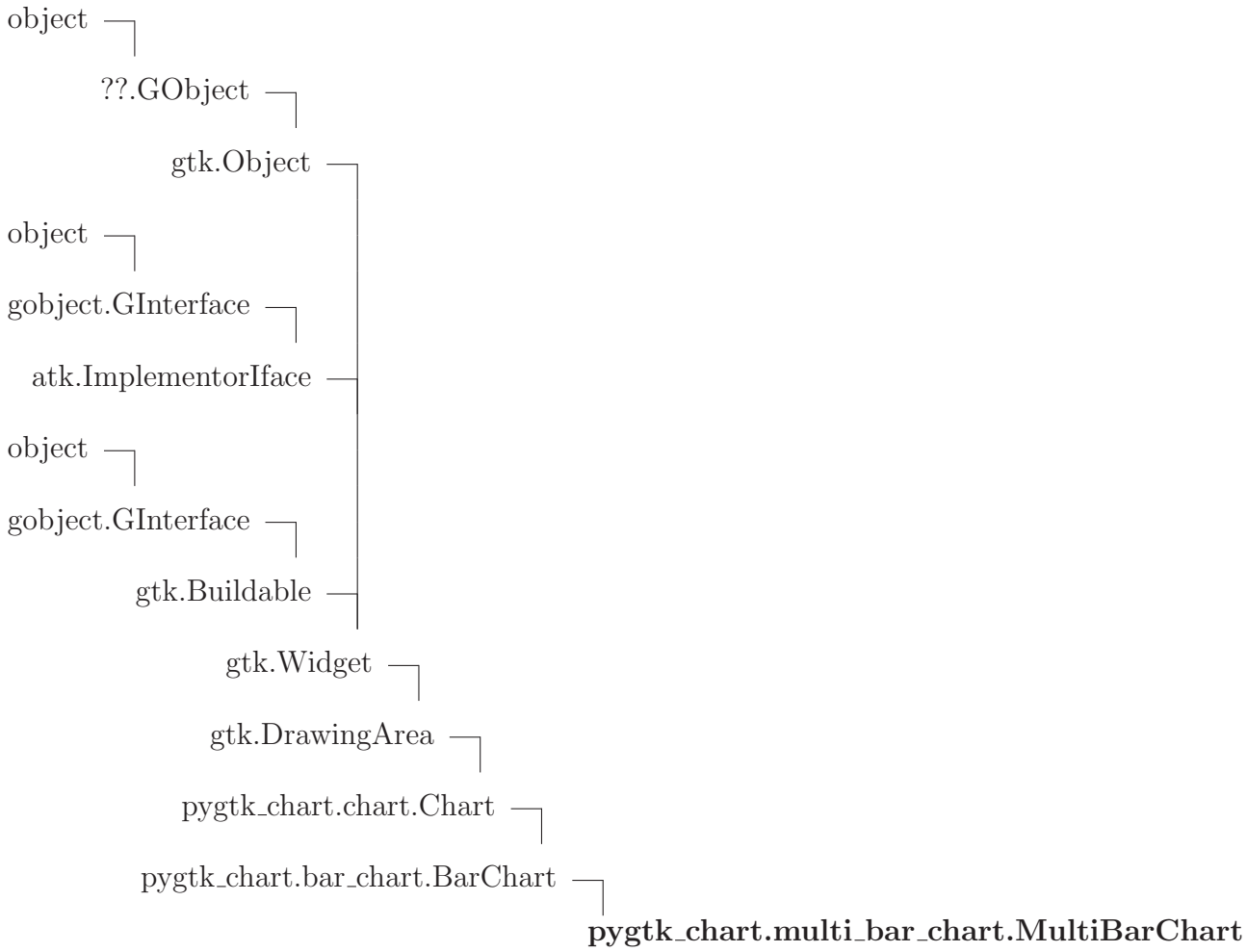
8.3.2 Properties

Name	Description
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

8.3.3 Class Variables

Name	Description
__gproperties__	**Value:** {"name":(gobject.TYPE_STRING, "group name", "A unique ide...
__gtype__	**Value:** <GType pygtk_chart+multi_bar_chart+BarGroup (168660840)>
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

8.4 Class MultiBarChart



The MultiBarChart widget displays groups of bars. Usage: create multi_bar_chart.BarGroups and add multi_bar_chart.Bars. The add the bar groups to MultiBarChart.

(section) Properties

The MultiBarChart class inherits properties from bar_chart.BarChart (except bar-padding). Additional properties:

- group-padding (the space between two bar groups in px, type: int in [0, 100], default: 16)
- label-rotation (the angle (in degrees) that should be used to rotate bar labels in vertical mode, type: int in [0, 360], default: 300)
- rotate-group-labels (sets whether group labels should be roteated by 90 degrees in horizontal mode, type: boolean, default: False).

(section) Signals

The MultiBarChart class inherits the signal 'bar-clicked' from bar_chart.BarChart. Additional signals:

- group-clicked: emitted when a bar is clicked, callback signature: def group_clicked(chart, group, bar).

8.4.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart.Chart.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart.Chart.do_set_property

set_group_padding(*self*, *padding*)

Set the amount of free space between bar groups (in px, default: 16).

Parameters

`padding`: the padding
(type=int in [0, 100].)

get_group_padding(*self*)

Returns the amount of free space between two bar groups (in px).

Return Value

int in [0, 100].

set_label_rotation(*self*, *angle*)

Set the abgle (in degrees) that should be used to rotate the bar labels in vertical mode (defualt: 300 degrees).

Parameters

`angle`: *(type=int in [0, 360].)*

get_label_rotation(*self*)

Returns the angle by which bar labels are rotated in vertical mode.

Return Value
int in [0, 350].

set_rotate_group_labels(*self*, *rotate*)

Set wether the groups' labels should be rotated by 90 degrees in horizontal mode (default: False).

Parameters
`rotate:` *(type=boolean.)*

get_rotate_group_labels(*self*)

Returns True if group labels should be rotated by 90 degrees in horizontal mode.

Return Value
boolean.

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the queue_draw() method.

Parameters
`context:` The context to draw on.

(type=cairo.Context)

Overrides: gtk.Widget.draw

add_group(*self*, *group*)

Add a BarGroup to the chart.

Parameters
`group:` *(type=multi_bar_chart.BarGroup.)*

add_bar(*self*, *bar*)

Alias for add_group. This method is deprecated. Use add_group instead.

Overrides: pygtk_chart.bar_chart.BarChart.add_bar

Inherited from pygtk_chart.bar_chart.BarChart(Section 2.5)

draw_basics(), get_bar_padding(), get_draw_labels(), get_enable_mouseover(), get_mode(), set_bar_padding(), set_draw_labels(), set_enable_mouseover(), set_mode()

Inherited from pygtk_chart.chart.Chart(Section 4.3)

export_png(), export_svg(), get_padding(), set_padding()

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(), error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(),

queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), render_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(), do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

8.4.2 Properties

Name	Description
Inherited from gtk.Widget allocation, name, parent, requisition, saved_state, state, style, window	
Inherited from ??.GObject	

continued on next page

Name	Description
__grefcount__	
Inherited from object __class__	

8.4.3 Class Variables

Name	Description
__gsignals__	**Value:** {"group-clicked":(gobject.SIGNAL_RUN_LAST, gobject.TYPE_N...
__gproperties__	**Value:** {"group-padding":(gobject.TYPE_INT, "group padding", "The...
__gtype__	**Value:** <GType pygtk_chart+multi_bar_chart+MultiBarChart (1687275...

9 Module pygtk_chart.pie_chart

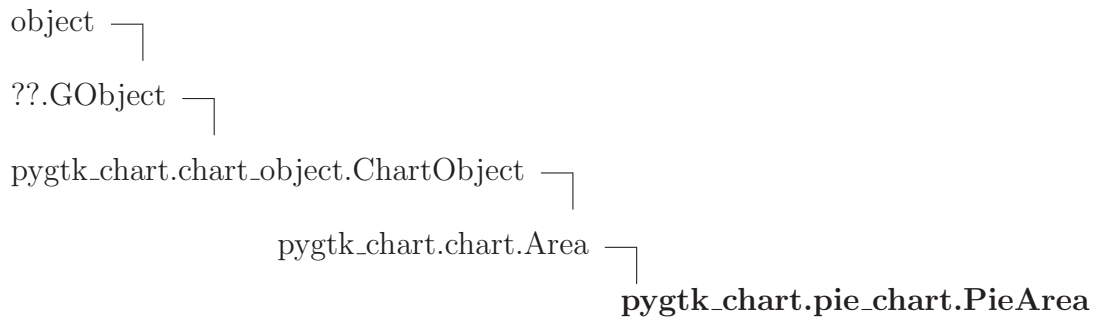
Contains the PieChart widget.

Author: Sven Festersen (sven@sven-festersen.de)

9.1 Functions

draw_sector(*context*, *cx*, *cy*, *radius*, *angle*, *angle_offset*)

9.2 Class PieArea



This class represents the sector of a pie chart.

(section) Properties

The PieArea class inherits properties from chart.Area.

(section) Signals

The PieArea class inherits signals from chart.Area.

9.2.1 Methods

__**init**__(*self*, *name*, *value*, *title*='')

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

Inherited from pygtk_chart.chart.Area(Section 4.6)

do_get_property(), do_set_property(), get_color(), get_highlighted(), get_label(), get_value(), set_color(), set_highlighted(), set_label(), set_value()

Inherited from pygtk_chart.chart_object.ChartObject(Section 5.1)

draw(), get_antialias(), get_visible(), set_antialias(), set_visible()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

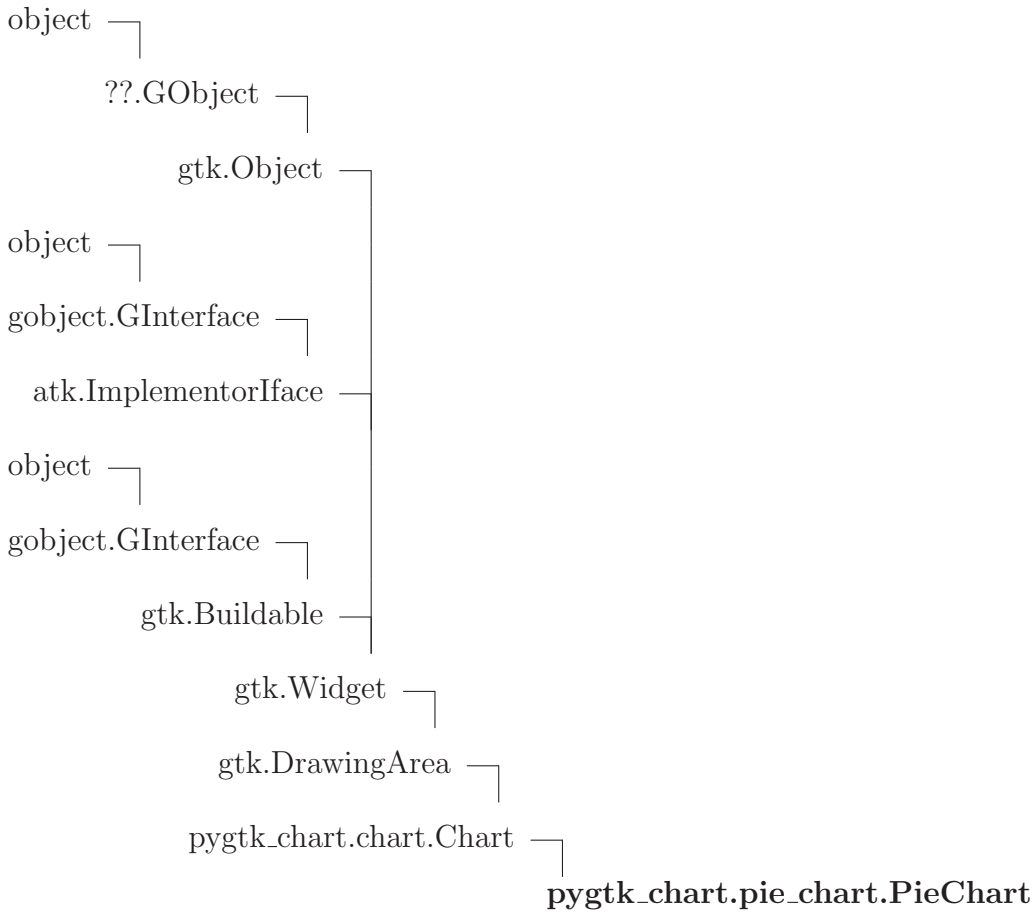
9.2.2 Properties

Name	Description
Inherited from ??.GObject __grefcount__	
Inherited from object __class__	

9.2.3 Class Variables

Name	Description
Inherited from pygtk_chart.chart.Area (Section 4.6) __gproperties__, __gtype__	
Inherited from pygtk_chart.chart_object.ChartObject (Section 5.1) __gsignals__	

9.3 Class PieChart



This is the pie chart class.

(section) Properties

The PieChart class inherits properties from chart.Chart. Additional properties:

- rotate (the angle that the pie chart should be rotated by in degrees, type: int in [0, 360])
- draw-shadow (sets whther to draw a shadow under the pie chart, type: boolean)
- draw-labels (sets whether to draw area labels, type: boolean)
- show-percentage (sets whether to show percentage in area labels, type: boolean)
- show-values (sets whether to show values in area labels, type: boolean)
- enable-scroll (sets whether the pie chart can be rotated by scrolling with the mouse wheel, type: boolean)
- enable-mouseover (sets whether a mouse over effect should be added to areas, type: boolean).

(section) Signals

The PieChart class inherits signals from chart.Chart. Additional signals:

- area-clicked (emitted when an area is clicked)

callback signature: def callback(piechart, area).

9.3.1 Methods

__init__(*self*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

do_get_property(*self*, *property*)

Overrides: pygtk_chart.chart.Chart.do_get_property

do_set_property(*self*, *property*, *value*)

Overrides: pygtk_chart.chart.Chart.do_set_property

draw(*self*, *context*)

Draw the widget. This method is called automatically. Don't call it yourself. If you want to force a redrawing of the widget, call the queue_draw() method.

Parameters

`context`: The context to draw on.

(type=cairo.Context)

Overrides: gtk.Widget.draw

add_area(*self*, *area*)

get_pie_area(*self*, *name*)

Returns the PieArea with the id 'name' if it exists, None otherwise.

Parameters

`name`: the id of a PieArea

(type=string)

Return Value

a PieArea or None.

set_rotate(*self*, *angle*)

Set the rotation angle of the PieChart in degrees.

Parameters

`angle:` angle in degrees 0 - 360

(type=integer.)

get_rotate(*self*)

Get the current rotation angle in degrees.

Return Value

integer.

set_draw_shadow(*self*, *draw*)

Set whether to draw the pie chart's shadow.

Parameters

`draw:` *(type=boolean.)*

get_draw_shadow(*self*)

Returns True if pie chart currently has a shadow.

Return Value

boolean.

set_draw_labels(*self*, *draw*)

Set whether to draw the labels of the pie areas.

Parameters

`draw:` *(type=boolean.)*

get_draw_labels(*self*)

Returns True if area labels are shown.

Return Value

boolean.

set_show_percentage(*self*, *show*)

Set whether to show the percentage an area has in its label.

Parameters

`show:` *(type=boolean.)*

get_show_percentage(*self*)

Returns True if percentages are shown.

Return Value
boolean.

set_enable_scroll(*self*, *scroll*)

Set whether the pie chart can be rotated by scrolling with the mouse wheel.

Parameters
`scroll:` *(type=boolean.)*

get_enable_scroll(*self*)

Returns True if the user can rotate the pie chart by scrolling.

Return Value
boolean.

set_enable_mouseover(*self*, *mouseover*)

Set whether a mouseover effect should be shown when the pointer enters a pie area.

Parameters
`mouseover:` *(type=boolean.)*

get_enable_mouseover(*self*)

Returns True if the mouseover effect is enabled.

Return Value
boolean.

set_show_values(*self*, *show*)

Set whether the area's value should be shown in its label.

Parameters
`show:` *(type=boolean.)*

get_show_values(*self*)

Returns True if the value of a pie area is shown in its label.

Return Value
boolean.

Inherited from pygtk_chart.chart.Chart(Section 4.3)

draw_basics(), export_png(), export_svg(), get_padding(), set_padding()

Inherited from gtk.DrawingArea

size()

Inherited from gtk.Widget

activate(), add_accelerator(), add_events(), add_mnemonic_label(), can_activate_accel(), child_focus(), child_notify(), class_path(), create_pango_context(), create_pango_layout(), destroy(), do_button_press_event(), do_button_release_event(), do_can_activate_accel(), do_client_event(), do_composited_changed(), do_configure_event(), do_delete_event(), do_destroy_event(), do_direction_changed(), do_drag_begin(), do_drag_data_delete(), do_drag_data_get(), do_drag_data_received(), do_drag_drop(), do_drag_end(), do_drag_leave(), do_drag_motion(), do_enter_notify_event(), do_event(), do_expose_event(), do_focus(), do_focus_in_event(), do_focus_out_event(), do_get_accessible(), do_grab_broken_event(), do_grab_focus(), do_grab_notify(), do_hide(), do_hide_all(), do_hierarchy_changed(), do_key_press_event(), do_key_release_event(), do_leave_notify_event(), do_map(), do_map_event(), do_mnemonic_activate(), do_motion_notify_event(), do_no_expose_event(), do_parent_set(), do_popup_menu(), do_property_notify_event(), do_proximity_in_event(), do_proximity_out_event(), do_realize(), do_screen_changed(), do_scroll_event(), do_selection_clear_event(), do_selection_get(), do_selection_notify_event(), do_selection_received(), do_selection_request_event(), do_show(), do_show_all(), do_show_help(), do_size_allocate(), do_size_request(), do_state_changed(), do_style_set(), do_unmap(), do_unmap_event(), do_unrealize(), do_visibility_notify_event(), do_window_state_event(), drag_begin(), drag_check_threshold(), drag_dest_add_image_targets(), drag_dest_add_text_targets(), drag_dest_add_uri_targets(), drag_dest_find_target(), drag_dest_get_target_list(), drag_dest_get_track_motion(), drag_dest_set(), drag_dest_set_proxy(), drag_dest_set_target_list(), drag_dest_set_track_motion(), drag_dest_unset(), drag_get_data(), drag_highlight(), drag_source_add_image_targets(), drag_source_add_text_targets(), drag_source_add_uri_targets(), drag_source_get_target_list(), drag_source_set(), drag_source_set_icon(), drag_source_set_icon_name(), drag_source_set_icon_pixbuf(), drag_source_set_icon_stock(), drag_source_set_target_list(), drag_source_unset(), drag_unhighlight(), ensure_style(), error_bell(), event(), freeze_child_notify(), get_accessible(), get_action(), get_activate_signal(), get_allocation(), get_ancestor(), get_child_requisition(), get_child_visible(), get_clipboard(), get_colormap(), get_composite_name(), get_direction(), get_display(), get_events(), get_extension_events(), get_has_tooltip(), get_modifier_style(), get_name(), get_no_show_all(), get_pango_context(), get_parent(), get_parent_window(), get_pointer(), get_root_window(), get_screen(), get_settings(), get_size_request(), get_snapshot(), get_style(), get_tooltip_markup(), get_tooltip_text(), get_tooltip_window(), get_toplevel(), get_visual(), get_window(), grab_add(), grab_default(), grab_focus(), grab_remove(), has_screen(), hide(), hide_all(), hide_on_delete(), input_shape_combine_mask(), intersect(), is_ancestor(), is_composited(), is_focus(), keynav_failed(), list_mnemonic_labels(), map(), menu_get_for_attach_widget(), mnemonic_activate(), modify_base(), modify_bg(), modify_cursor(), modify_fg(), modify_font(), modify_style(), modify_text(), path(), queue_clear(), queue_clear_area(), queue_draw(), queue_draw_area(), queue_resize(), queue_resize_no_redraw(), rc_get_style(), realize(), region_intersect(), remove_accelerator(), remove_mnemonic_label(), ren-

der_icon(), reparent(), reset_rc_styles(), reset_shapes(), selection_add_target(), selection_add_targets(), selection_clear_targets(), selection_convert(), selection_owner_set(), selection_remove_all(), send_expose(), set_accel_path(), set_activate_signal(), set_app_paintable(), set_child_visible(), set_colormap(), set_composite_name(), set_direction(), set_double_buffered(), set_events(), set_extension_events(), set_has_tooltip(), set_name(), set_no_show_all(), set_parent(), set_parent_window(), set_redraw_on_allocate(), set_scroll_adjustments(), set_sensitive(), set_set_scroll_adjustments_signal(), set_size_request(), set_state(), set_style(), set_tooltip_markup(), set_tooltip_text(), set_tooltip_window(), set_uposition(), set_usize(), shape_combine_mask(), show(), show_all(), show_now(), size_allocate(), size_request(), style_get_property(), thaw_child_notify(), translate_coordinates(), trigger_tooltip_query(), unmap(), unparent(), unrealize()

Inherited from gtk.Object

do_destroy(), flags(), remove_data(), remove_no_notify(), set_flags(), unset_flags()

Inherited from ??.GObject

__cmp__(), __copy__(), __deepcopy__(), __delattr__(), __gdoc__(), __gobject_init__(), __hash__(), __new__(), __repr__(), __setattr__(), chain(), connect(), connect_after(), connect_object(), connect_object_after(), disconnect(), disconnect_by_func(), emit(), emit_stop_by_name(), freeze_notify(), get_data(), get_properties(), get_property(), handler_block(), handler_block_by_func(), handler_disconnect(), handler_is_connected(), handler_unblock(), handler_unblock_by_func(), notify(), props(), set_data(), set_properties(), set_property(), stop_emission(), thaw_notify(), weak_ref()

Inherited from atk.ImplementorIface

ref_accessible()

Inherited from gtk.Buildable

add_child(), construct_child(), do_add_child(), do_construct_child(), do_get_internal_child(), do_parser_finished(), do_set_name(), get_internal_child(), parser_finished()

Inherited from object

__getattribute__(), __reduce__(), __reduce_ex__(), __str__()

9.3.2 Properties

Name	Description
Inherited from gtk.Widget allocation, name, parent, requisition, saved_state, state, style, window	
Inherited from ??.GObject __grefcount__	
Inherited from object	

continued on next page

Name	Description
__class__	

9.3.3 Class Variables

Name	Description
__gproperties__	**Value:** {"rotate":(gobject.TYPE_INT, "rotation", "The angle to ro...
__gsignals__	**Value:** {"area-clicked":(gobject.SIGNAL_RUN_LAST, gobject.TYPE_NO...
__gtype__	**Value:** <GType pygtk_chart+pie_chart+PieChart (171517840)>

Index

pygtk_chart *(package)*, 2–3
pygtk_chart.bar_chart *(module)*, 4–15
pygtk_chart.bar_chart.Bar *(class)*, 4–6
pygtk_chart.bar_chart.BarChart *(class)*, 9–15
pygtk_chart.bar_chart.draw_rounded_rectangle *(function)*, 4
pygtk_chart.bar_chart.Grid *(class)*, 6–9
pygtk_chart.basics *(module)*, 16–17
pygtk_chart.basics.color_cairo_to_gdk *(function)*, 16
pygtk_chart.basics.color_gdk_to_cairo *(function)*, 16
pygtk_chart.basics.color_html_to_cairo *(function)*, 17
pygtk_chart.basics.color_list_from_file *(function)*, 17
pygtk_chart.basics.color_rgb_to_cairo *(function)*, 16
pygtk_chart.basics.gdk_color_list_from_file *(function)*, 17
pygtk_chart.basics.get_center *(function)*, 16
pygtk_chart.basics.intersect_ranges *(function)*, 16
pygtk_chart.basics.is_in_range *(function)*, 16
pygtk_chart.basics.set_context_line_style *(function)*, 17
pygtk_chart.chart *(module)*, 18–31
pygtk_chart.chart.add_sensitive_area *(function)*, 18
pygtk_chart.chart.Area *(class)*, 28–31
pygtk_chart.chart.Background *(class)*, 23–26
pygtk_chart.chart.Chart *(class)*, 18–23
pygtk_chart.chart.get_sensitive_areas *(function)*, 18
pygtk_chart.chart.init_sensitive_areas *(function)*, 18
pygtk_chart.chart.Title *(class)*, 26–28
pygtk_chart.chart_object *(module)*, 32–34
pygtk_chart.chart_object.ChartObject *(class)*, 32–34
pygtk_chart.label *(module)*, 35–44
pygtk_chart.label.begin_drawing *(function)*, 35
pygtk_chart.label.finish_drawing *(function)*, 35
pygtk_chart.label.get_registered_labels *(function)*, 35
pygtk_chart.label.get_text_pos *(function)*, 35
pygtk_chart.label.Label *(class)*, 36–44
pygtk_chart.label.register_label *(function)*, 35
pygtk_chart.line_chart *(module)*, 45–76
pygtk_chart.line_chart.Axis *(class)*, 53–57
pygtk_chart.line_chart.draw_errors *(function)*, 45
pygtk_chart.line_chart.draw_point *(function)*, 45
pygtk_chart.line_chart.draw_point_pixbuf *(function)*, 45
pygtk_chart.line_chart.Graph *(class)*, 65–73
pygtk_chart.line_chart.graph_new_from_file *(function)*, 46
pygtk_chart.line_chart.graph_new_from_function *(function)*, 45
pygtk_chart.line_chart.Grid *(class)*, 61–65
pygtk_chart.line_chart.Legend *(class)*, 73–76
pygtk_chart.line_chart.LineChart *(class)*, 49–53
pygtk_chart.line_chart.optimize_sampling *(function)*, 46
pygtk_chart.line_chart.RangeCalculator *(class)*, 48–49
pygtk_chart.line_chart.separate_data_and_errors *(function)*, 45
pygtk_chart.line_chart.XAxis *(class)*, 57–

59
pygtk_chart.line_chart.YAxis *(class)*, 59–61
pygtk_chart.multi_bar_chart *(module)*, 77–88
pygtk_chart.multi_bar_chart.Bar *(class)*, 77–79
pygtk_chart.multi_bar_chart.BarGroup *(class)*, 79–82
pygtk_chart.multi_bar_chart.MultiBarChart *(class)*, 82–88
pygtk_chart.pie_chart *(module)*, 89–97
pygtk_chart.pie_chart.draw_sector *(function)*, 89
pygtk_chart.pie_chart.PieArea *(class)*, 89–90
pygtk_chart.pie_chart.PieChart *(class)*, 90–97