GLUONIX DESIGNER

Contents

[Root/Popup 8](#_Toc198216371)

[Nothing 8](#_Toc198216372)

[Maximize 8](#_Toc198216373)

[Restore 8](#_Toc198216374)

[Minimize 8](#_Toc198216375)

[Restart 8](#_Toc198216376)

[Close 8](#_Toc198216377)

[Hide 8](#_Toc198216378)

[Show 9](#_Toc198216379)

[Grab 9](#_Toc198216380)

[After 9](#_Toc198216381)

[Screen 9](#_Toc198216382)

[Bind 9](#_Toc198216383)

[Config 9](#_Toc198216384)

[Config\_Get 9](#_Toc198216385)

[Add\_Manu 9](#_Toc198216386)

[Add\_Sub\_Menu 10](#_Toc198216387)

[Add\_Separator 10](#_Toc198216388)

[Folder 10](#_Toc198216389)

[File 10](#_Toc198216390)

[Position 10](#_Toc198216391)

[Size 10](#_Toc198216392)

[Locate 10](#_Toc198216393)

[Locate\_Reverse 11](#_Toc198216394)

[Light\_Mode 11](#_Toc198216395)

[Dark\_Mode 11](#_Toc198216396)

[Update\_Color 11](#_Toc198216397)

[Common 12](#_Toc198216398)

[Copy 12](#_Toc198216399)

[Delete 12](#_Toc198216400)

[Hide 12](#_Toc198216401)

[Show 12](#_Toc198216402)

[Focus 12](#_Toc198216403)

[Grab 12](#_Toc198216404)

[Bind 12](#_Toc198216405)

[Config 13](#_Toc198216406)

[Config\_Get 13](#_Toc198216407)

[Position 13](#_Toc198216408)

[Size 13](#_Toc198216409)

[Locate 13](#_Toc198216410)

[Locate\_Reverse 13](#_Toc198216411)

[Canvas/Scroll 14](#_Toc198216412)

[Clear 14](#_Toc198216413)

[Refresh 14](#_Toc198216414)

[Bind\_Item 14](#_Toc198216415)

[Hide\_Item 14](#_Toc198216416)

[Show\_Item 14](#_Toc198216417)

[Delete\_Item 14](#_Toc198216418)

[Delete\_All 15](#_Toc198216419)

[Find\_Near 15](#_Toc198216420)

[Find\_Overlap 15](#_Toc198216421)

[Line 15](#_Toc198216422)

[Polyline 15](#_Toc198216423)

[Add 15](#_Toc198216424)

[Remove 16](#_Toc198216425)

[Pie 16](#_Toc198216426)

[Arc 16](#_Toc198216427)

[Circle 16](#_Toc198216428)

[Rectangle 16](#_Toc198216429)

[Rectangle2 16](#_Toc198216430)

[Oval 17](#_Toc198216431)

[Polygon 17](#_Toc198216432)

[Add 17](#_Toc198216433)

[Remove 17](#_Toc198216434)

[Text 17](#_Toc198216435)

[Set 17](#_Toc198216436)

[Image 18](#_Toc198216437)

[Set 18](#_Toc198216438)

[Initial 18](#_Toc198216439)

[Top 18](#_Toc198216440)

[Reset 18](#_Toc198216441)

[Update 18](#_Toc198216442)

[Update\_All 18](#_Toc198216443)

[Frame 19](#_Toc198216444)

[Clear 19](#_Toc198216445)

[Bar 19](#_Toc198216446)

[Set 19](#_Toc198216447)

[Get 19](#_Toc198216448)

[Button 20](#_Toc198216449)

[Set 20](#_Toc198216450)

[Check 20](#_Toc198216451)

[Set 20](#_Toc198216452)

[Get 20](#_Toc198216453)

[Compound 21](#_Toc198216454)

[Set 21](#_Toc198216455)

[Entry 21](#_Toc198216456)

[Set 21](#_Toc198216457)

[Get 21](#_Toc198216458)

[Image 22](#_Toc198216459)

[Set 22](#_Toc198216460)

[Initial 22](#_Toc198216461)

[Rotate 22](#_Toc198216462)

[Label 22](#_Toc198216463)

[Set 22](#_Toc198216464)

[Roubel 23](#_Toc198216465)

[Set 23](#_Toc198216466)

[Line 23](#_Toc198216467)

[List 24](#_Toc198216468)

[Add 24](#_Toc198216469)

[Remove 24](#_Toc198216470)

[Set 24](#_Toc198216471)

[Reset 24](#_Toc198216472)

[Get 24](#_Toc198216473)

[Clear 24](#_Toc198216474)

[Top 25](#_Toc198216475)

[Variable 25](#_Toc198216476)

[Create 25](#_Toc198216477)

[Get 25](#_Toc198216478)

[Radio 25](#_Toc198216479)

[Set 25](#_Toc198216480)

[Reset 25](#_Toc198216481)

[Scale 26](#_Toc198216482)

[Set 26](#_Toc198216483)

[Get 26](#_Toc198216484)

[Spinner 26](#_Toc198216485)

[Set 26](#_Toc198216486)

[Get 26](#_Toc198216487)

[Select 27](#_Toc198216488)

[Add 27](#_Toc198216489)

[Remove 27](#_Toc198216490)

[Set 27](#_Toc198216491)

[Get 27](#_Toc198216492)

[Clear 27](#_Toc198216493)

[Sort 27](#_Toc198216494)

[Separator 28](#_Toc198216495)

[Add 28](#_Toc198216496)

[Switch 28](#_Toc198216497)

[Set 28](#_Toc198216498)

[Get 28](#_Toc198216499)

[Text 29](#_Toc198216500)

[Tag 29](#_Toc198216501)

[Add 29](#_Toc198216502)

[Set 29](#_Toc198216503)

[Get 29](#_Toc198216504)

[Tree 30](#_Toc198216505)

[Add 30](#_Toc198216506)

[Edit 30](#_Toc198216507)

[Get 30](#_Toc198216508)

[Get\_All 30](#_Toc198216509)

[Remove 30](#_Toc198216510)

[Remove\_All 30](#_Toc198216511)

[Remove\_Selected 31](#_Toc198216512)

[Current 31](#_Toc198216513)

[Selected 31](#_Toc198216514)

[Child 31](#_Toc198216515)

[Parent 31](#_Toc198216516)

[Index 31](#_Toc198216517)

[Expand 31](#_Toc198216518)

[Select 32](#_Toc198216519)

[Export 32](#_Toc198216520)

[Bind / Bind\_Item 33](#_Toc198216521)

[Bind Specific 33](#_Toc198216522)

[For All 33](#_Toc198216523)

[Config 38](#_Toc198216524)

[Background 38](#_Toc198216525)

[Light\_Background 38](#_Toc198216526)

[Dark\_Background 38](#_Toc198216527)

[Foreground 38](#_Toc198216528)

[Light\_Foreground 39](#_Toc198216529)

[Dark\_Foreground 39](#_Toc198216530)

[Border\_Color 39](#_Toc198216531)

[Light\_Border\_Color 39](#_Toc198216532)

[Dark\_Border\_Color 39](#_Toc198216533)

[Border\_Size 39](#_Toc198216534)

[Resize 40](#_Toc198216535)

[Resize\_Width 40](#_Toc198216536)

[Resize\_Height 40](#_Toc198216537)

[Move 40](#_Toc198216538)

[Move\_Top 40](#_Toc198216539)

[Move\_Left 40](#_Toc198216540)

[Top 40](#_Toc198216541)

[Left 40](#_Toc198216542)

[Width 41](#_Toc198216543)

[Height 41](#_Toc198216544)

[Font\_Size 41](#_Toc198216545)

[Font\_Weight 41](#_Toc198216546)

[Font\_Family 41](#_Toc198216547)

[Scrollbar 41](#_Toc198216548)

[Vertical 42](#_Toc198216549)

[Horizontal 42](#_Toc198216550)

[Last 42](#_Toc198216551)

[Value 42](#_Toc198216552)

[Ridge 42](#_Toc198216553)

[Disable 42](#_Toc198216554)

[Path 43](#_Toc198216555)

[URL 43](#_Toc198216556)

[Array 43](#_Toc198216557)

[Pil 43](#_Toc198216558)

[Photo 43](#_Toc198216559)

[Rotate 43](#_Toc198216560)

[Transparent 44](#_Toc198216561)

[Aspect\_Ratio 44](#_Toc198216562)

[Compound 44](#_Toc198216563)

[Align 44](#_Toc198216564)

[Secure 44](#_Toc198216565)

[Variable 45](#_Toc198216566)

[Minimum 45](#_Toc198216567)

[Maximum 45](#_Toc198216568)

[Increment 45](#_Toc198216569)

[Orient 45](#_Toc198216570)

[Height\_List 45](#_Toc198216571)

[Progress 46](#_Toc198216572)

[Zero 46](#_Toc198216573)

[Radius 46](#_Toc198216574)

[Shadow\_Size 46](#_Toc198216575)

[Shadow\_Color 46](#_Toc198216576)

[Light\_Shadow\_Color 47](#_Toc198216577)

[Dark\_Shadow\_Color 47](#_Toc198216578)

[Background\_Selected 47](#_Toc198216579)

[Foreground\_Selected 47](#_Toc198216580)

[Translucent 47](#_Toc198216581)

[Multiple 47](#_Toc198216582)

[Select\_Background 48](#_Toc198216583)

[Select\_Foreground 48](#_Toc198216584)

[Disable\_Background 48](#_Toc198216585)

[Disable\_Foreground 48](#_Toc198216586)

[Hover\_Background 48](#_Toc198216587)

[Hover\_Foreground 48](#_Toc198216588)

[Hover\_Border\_Color 49](#_Toc198216589)

[Hover\_Shadow\_Color 49](#_Toc198216590)

[Light\_Hover\_Background 49](#_Toc198216591)

[Light\_Hover\_Foreground 49](#_Toc198216592)

[Light\_Hover\_Border\_Color 49](#_Toc198216593)

[Light\_Hover\_Shadow\_Color 49](#_Toc198216594)

[Dark\_Hover\_Background 50](#_Toc198216595)

[Dark\_Hover\_Foreground 50](#_Toc198216596)

[Dark\_Hover\_Border\_Color 50](#_Toc198216597)

[Dark\_Hover\_Shadow\_Color 50](#_Toc198216598)

# Root/Popup

## Nothing

Root.Nothing()

Return → False

## Maximize

Root.Maximize()

Return → Maximized Window

## Restore

Root.Restore()

Return → Restore Window Size

## Minimize

Root.Minimize()

Return → Minimize Window To Taskbar

## Restart

Root.Restart()

Return → Restart Application

## Close

Root.Close()

Return → Close Application

## Hide

Root.Hide()

Return → Hide Application

## Show

Root.Show()

Return → Show Application

## Grab

Root.Grab(Path=False)

Return → Returns and Saves screenshot of Application

## After

Root.After(Delay=1000, Function=lambda : Root.Nothing())

Return → Runs the provided function after milliseconds of delay

## Screen

Root.Screen()

Return → Returns screen size: {'Width':1920, 'Height': 1080}

## Bind

Root.Bind(On\_Click=lambda E: Root.Nothing())

Return → Runs on click function and list of other binds

## Config

Root.Config(Background=’#F5F5F5’)

Return → Sets background color and list of configs

## Config\_Get

Root.Config\_Get(‘Background’)

Return → Background color and list of configs

## Add\_Manu

Root.Add\_Manu(Main=False, Name, Command=False)

Return → Adds Menu Item. If Main is False, will Add to Top Bar, if command is given, No submenu can be added

## Add\_Sub\_Menu

Root.Add\_Sub\_Manu(Main, Name, Command=False)

Return → Adds Sub Menu Item. If command is given, No submenu can be added

## Add\_Separator

Root.Add\_Separator(Main)

Return → Adds horizontal separator in given step of menu

## Folder

Root.Folder(Initial='', Title='', Persistent=True)

Return → Request folder input from user.

## File

Root.File( Initial='', Title='', Multiple=False, Default='.txt', Type=[["Text files", "\*.txt"], ["All files", "\*.\*"]])

Return → Request file input from user

## Position

Root.Position()

Return → Current position of Application [Left, Top]

## Size

Root.Size()

Return → Current size of Application [Width, Height]

## Locate

Root.Locate(Width, Height, Left, Top)

Return → Converts the values from percentage to pixels

## Locate\_Reverse

Root.Locate\_Reverse(Width, Height, Left, Top)

Return → Converts the values from pixels to percentage

## Light\_Mode

Root.Light\_Mode()

Return → Changes full application to Light colors

## Dark\_Mode

Root.Dark\_Mode()

Return → Changes full application to Dark colors

## Update\_Color

Root.Update\_Color()

Return → Updates all widgets for automatic light and dark mode colors

# Common

## Copy

Widget.Copy(Name=False, Main=False)

Return → Creates a copy of widget and inside widgets with a new name, For Canvas items, no arg Name, only Main

## Delete

Widget.Delete()

Return → Delete widget and all inside widgets

## Hide

Widget.Hide()

Return → Hide Widget

## Show

Widget.Show()

Return → Show Widget

## Focus

Widget.Focus()

Return → Bring Widget To Focus {Works on Entry, Button, List & Select}

## Grab

Widget.Grab(Path=False)

Return → Returns and Saves screenshot of Widget

## Bind

Widget.Bind(On\_Click=lambda E: Root.Nothing())

Return → Runs on click function and list of other binds

## Config

Widget.Config(Background=’#F5F5F5’)

Return → Sets background color and list of configs

## Config\_Get

Widget.Config\_Get(‘Background’)

Return → Background color and list of configs

## Position

Widget.Position(Left=False, Top=False)

Return → Current position of Application [Left, Top]

## Size

Widget.Size(Width=False, Height=False)

Return → Current size of Application [Width, Height]

## Locate

Widget.Locate(Width, Height, Left, Top)

Return → Converts the values from percentage to pixels

## Locate\_Reverse

Widget.Locate\_Reverse(Width, Height, Left, Top)

Return → Converts the values from pixels to percentage

# Canvas/Scroll

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Radius'

Canvas Config Addon: 'Shadow\_Size', ' Shadow\_Color'

Scroll Config Addon: 'Scrollbar', 'Vertical', 'Horizontal', 'Last'

## Clear

Canvas.Clear()

Return → Clears all widgets inside canvas or scroll

## Refresh

Canvas.Refresh()

Return → Update all idle tasks

## Bind\_Item

Canvas.Bind\_Item(Item, On\_Click=lambda E: Root.Nothing())

Return → Runs on click function and list of other binds on an item in canvas

## Hide\_Item

Canvas.Hide\_Item(Item)

Return → Hides an Item

## Show\_Item

Canvas.Show\_Item(Item)

Return → Show an Item

## Delete\_Item

Canvas.Delete\_Item(Item)

Return → Delete an Item

## Delete\_All

Canvas.Delete\_All()

Return → Delete all Items

## Find\_Near

Canvas.Find\_Near(X, Y)

Return → List of all items close to this pixel

## Find\_Overlap

Canvas.Find\_Overlap(X1, Y1, X2, Y2)

Return → List of all the items overlapping provided rectangle

## Line

Config List: 'Outline', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize'

Canvas.Line()

Return → Draw a line with provided parameters and return item

## Polyline

Config List: 'Outline', 'Thickness', 'Resize'

Canvas.Polyline()

Return → Draw a line with provided parameters and return item

### Add

Widget.Add(X, Y)

Return → Adds a new point to polyline

### Remove

Widget.Remove(Index)

Return → Removes a point from polyline

## Pie

Config List: 'Outline', 'Fill', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Start', 'Extent', 'Translucent'

Canvas.Pie()

Return → Draw a pie chart with provided parameters and return item

## Arc

Config List: 'Outline', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Start', 'Extent', 'Translucent'

Canvas.Arc()

Return → Draw an arc with provided parameters and return item

## Circle

Config List: 'Outline', 'Fill', 'Left', 'Top', 'Radius', 'Thickness', 'Resize', 'Translucent'

Canvas.Circle()

Return → Draw a circle with provided parameters and return item

## Rectangle

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize', 'Translucent'

Canvas.Rectangle()

Return → Draw a rectangle with provided parameters and return item

## Rectangle2

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Angle', 'Thickness', 'Resize', 'Translucent'

Canvas.Rectangle2()

Return → Draw a rotated rectangle with provided parameters and return item

## Oval

Config List: 'Outline', 'Fill', 'Width', 'Height', 'Left', 'Top', 'Thickness', 'Resize', 'Translucent'

Canvas.Oval()

Return → Draw an oval with provided parameters and return item

## Polygon

Config List: 'Outline', 'Fill', 'Thickness', 'Resize', 'Translucent'

Canvas. Polygon()

Return → List of points to create polygon, Draw a polygon with provided parameters and return item

### Add

Widget.Add(X, Y)

Return → Adds a new point to polygon

### Remove

Widget.Remove(Index)

Return → Removes a point from polygon

## Text

Config List: 'Width', 'Height', 'Left', 'Top', 'Color', 'Size', 'Value', 'Weight', 'Font', 'Anchor', 'Justify', 'Resize'

Canvas.Text()

Return → Draws a text and return the item

### Set

Widget.Set(Vlaue)

Return → Sets a new text to item

## Image

Config List: 'Width', 'Height', 'Left', 'Top', 'Anchor', 'Url', 'Array', 'Pil', 'Photo', 'Resize'

Canvas.Image()

Return → Draws an image and return item

### Set

Widget.Set(Path)

Return → Sets a new image to item

### Initial

Widget.Initial()

Return → Reset image to initial image loaded at design time

## Top

Scroll.Top()

Return → Moves scrollbar to top

## Reset

Scroll.Reset()

Return → Resets the size of scroll frame and moves or hides scrollbars

## Update

Scroll.Update(Widget)

Return → Updates the size of Scroll region based on provided widget. It is recommended to provide the widget that id most to right and bottom of scroll. You can add an hidden widget so the size given properly to scroll the frame

## Update\_All

Scroll.Update\_All()

Return → Will update the size of scroll frame based on all the widgets in scroll.

# Frame

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height'

## Clear

Frame.Clear()

Return → Clears all widgets inside frame

# Bar

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Progress', 'Zero'

## Set

Bar.Set(Value=10)

Return → Sets the bar to specific position

## Get

Bar.Set()

Return → The current position of bar

# Button

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Value', 'Ridge', 'Disable', 'Disable\_Foreground', 'Active\_Background', 'Active\_Foreground'

## Set

Button.Set(Value=’Click Me’)

Return → Sets the name of the button

# Check

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height'

## Set

Check.Set(Check=True)

Return → Sets the current state of check box

## Get

Check.Set()

Return → The current state of check button

# Compound

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Value', 'Path', 'Url', 'Array', 'Pil', 'Rotate', 'Transparent', 'Compound', 'Aspect\_Ratio'

## Set

Compound.Set(Path, Value)

Return → Sets the image path or name of compound button

# Entry

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Align', 'Disable', 'Secure' , 'Disable\_Background', 'Disable\_Foreground', 'Select\_Background', 'Select\_Foreground'

## Set

Entry.Set(Value)

Return → Sets the value to entry

## Get

Entry.Get()

Return → The value to entry

# Image

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Path', 'Url', 'Array', 'Pil', 'Rotate', 'Transparent', 'Aspect\_Ratio'

## Set

Image.Set(Path)

Return → Sets the image path and reloads image. If Url is True, Path is Url & if Array is True, Path is cv2 Frame Array (RGB)

## Initial

Image.Initial()

Return → Reset image to initial image loaded at design time

## Rotate

Image.Rotate(Value=10)

Return → Rotates image in given angle degree values

# Label

Config List: 'Background', 'Foreground', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Align', 'Value'

## Set

Label.Set(Value)

Return → Sets text value of label

# Roubel

Config List: 'Background', 'Foreground', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Value', 'Radius', 'Shadow\_Size', 'Shadow\_Color'

## Set

Label.Set(Value)

Return → Sets text value of label

# Line

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height'

# List

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Disable', 'Scrollbar', 'Vertical', 'Select\_Foreground', 'Select\_Background', 'Multiple'

## Add

List.Add(Value)

Return → Adds the value to list

## Remove

List.Remove(Value)

Return → Removes the value from list

## Set

List.Set(Value)

Return → Sets the value to list

## Reset

List.Reset()

Return → Resets List Selection

## Get

List.Get()

Return → The current value to list

## Clear

List.Clear()

Return → Delete all the values in list

## Top

List.Top()

Return → Scroll list to top

# Variable

## Create

Variable = Gluonix.Variable()

Return → Creates a variable object for Radio Button

## Get

Variable.Get()

Return → Returns current values of variable set by radio buttons

# Radio

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Value', 'Variable'

## Set

Radio.Set()

Return → Activates current radio and unset all the other radios for same variable

## Reset

Radio.Reset()

Return → Unset radio

# Scale

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Increment', 'Minimum', 'Maximum', 'Orient', 'Disable'

## Set

Scale.Set(Value)

Return → Set scale to given value

## Get

Scale.Get()

Return → Current values of scale

# Spinner

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Align', 'Increment', 'Minimum', 'Maximum', 'Disable'

## Set

Spinner.Set(Value)

Return → Set spinner to given value

## Get

Spinner.Get()

Return → Current values of spinner

# Select

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize\_Font', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Font\_Family', 'Disable'

## Add

Select.Add(Value)

Return → Adds the value to select

## Remove

Select.Remove(Value)

Return → Removes the value from select

## Set

Select.Set(Value)

Return → Sets the value to select

## Get

Select.Get()

Return → The current value to select

## Clear

Select.Clear()

Return → Delete all the values in select

## Sort

Select.Sort()

Return → Sorts list values A - Z

# Separator

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height'

## Add

Separator.Add(Frame)

Return → Adds frame to control the movement and size, Need two frames and open provided, frames can’t be changed.

# Switch

Config List: 'Background', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height'

## Set

Check.Set(Check=True)

Return → Sets the current state of check box

## Get

Check.Get()

Return → The current state of check button

# Text

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Disable'

## Tag

Text.Tag(Name, Font\_Size=12, Font\_Weight='normal', Font='Times New Roman', Foreground="#000000", Background="#FFFFFF")

Return → Creates a custom tag to use in text

## Add

Text.Add(Value, Tag=’Default’)

Return → Sets the current state of check box

## Set

Text.Set(Value)

Return → Sets values to text area

## Get

Text.Get()

Return → The values of text area

# Tree

Config List: 'Background', 'Foreground', 'Border\_Color', 'Border\_Size', 'Resize', 'Resize\_Width', 'Resize\_Height', 'Move', 'Move\_Left', 'Move\_Top', 'Left', 'Top', 'Width', 'Height', 'Font\_Size', 'Font\_Weight', 'Background\_Selected', 'Foreground\_Selected '

## Add

Tree.Add(Name, Parent=’’, Index=’end’, Value=[], ID=None, Path=None)

Return → Create a tree item with specific value. Path for Image file.

## Edit

Tree.Edit(Name=False, Value=False, Tag=False, ID=False)

Return → Edits parameters of on focus tree item or by providing the id of item

## Get

Tree.Get(ID=False)

Return → The values of on focus tree item or by providing the id of item

## Get\_All

Tree.Get\_All(ID=False)

Return → The everything of on focus tree item or by providing the id of item

## Remove

Tree.Remove(ID)

Return → Deletes given item

## Remove\_All

Tree.Remove\_All()

Return → Deletes all items in the tree

## Remove\_Selected

Tree.Remove\_Selected()

Return → Deletes all selected items in the tree

## Current

Tree.Current()

Return → The item in focus

## Selected

Tree.Selected()

Return → List of selected items

## Child

Tree.Child(ID)

Return → List of all children of item

## Parent

Tree.Parent(ID)

Return → Parent of given item

## Index

Tree.Index(ID)

Return → Index of an item

## Expand

Tree.Expand(ID)

Return → Expand the current tree item

## Select

Tree.Select(ID)

Return → Makes specific item in focus

## Export

Tree.Export(Path)

Return → Exports tree structure to text file ‘.txt’

# Bind / Bind\_Item

## Bind Specific

Cursor\_Hand

Cursor\_Loading

Cursor\_Resize\_Vertical

Cursor\_Resize\_Horizontal

Cursor\_Arrow

Cursor

On\_Show (**No Event**)

On\_Hide (**No Event**)

On\_Close (Root & Popup) (**No Event**)

On\_Resize (Root, Popup, Frame, Canvas & Scroll) (**No Event**)

On\_Change (List, Select, Check, Switch & Radio) (**No Event**)

## For All

On\_Configure

On\_Destroy

On\_Expose

On\_Visibility

On\_Motion

On\_Click

On\_Release

On\_Double\_Click

On\_Triple\_Click

On\_Middle\_Click

On\_Middle\_Release

On\_Middle\_Double\_Click

On\_Middle\_Triple\_Click

On\_Right\_Click

On\_Right\_Release

On\_Right\_Double\_Click

On\_Right\_Triple\_Click

On\_Drag

On\_Middle\_Drag

On\_Right\_Drag

On\_Mouse\_Wheel

On\_Hover\_In

On\_Hover\_Out

On\_Key

On\_Key\_Release

On\_Focus\_Out

On\_Map

On\_Unmap

On\_Copy

On\_Cut

On\_Paste

On\_Undo

On\_Redo

On\_Control\_Click

On\_Control\_Release

On\_Control\_Double\_Click

On\_Control\_Triple\_Click

On\_Control\_Middle\_Click

On\_Control\_Middle\_Release

On\_Control\_Middle\_Double\_Click

On\_Control\_Middle\_Triple\_Click

On\_Control\_Right\_Click

On\_Control\_Right\_Release

On\_Control\_Right\_Double\_Click

On\_Control\_Right\_Triple\_Click

On\_Control\_Drag

On\_Control\_Middle\_Drag

On\_Control\_Right\_Drag

On\_Control\_Mouse\_Wheel

On\_Control\_Hover\_In

On\_Control\_Hover\_Out

On\_Alt\_Click

On\_Alt\_Release

On\_Alt\_Double\_Click

On\_Alt\_Triple\_Click

On\_Alt\_Middle\_Click

On\_Alt\_Middle\_Release

On\_Alt\_Middle\_Double\_Click

On\_Alt\_Middle\_Triple\_Click

On\_Alt\_Right\_Click

On\_Alt\_Right\_Release

On\_Alt\_Right\_Double\_Click

On\_Alt\_Right\_Triple\_Click

On\_Alt\_Drag

On\_Alt\_Middle\_Drag

On\_Alt\_Right\_Drag

On\_Alt\_Mouse\_Wheel

On\_Alt\_Hover\_In

On\_Alt\_Hover\_Out

On\_Shift\_Click

On\_Shift\_Release

On\_Shift\_Double\_Click

On\_Shift\_Triple\_Click

On\_Shift\_Middle\_Click

On\_Shift\_Middle\_Release

On\_Shift\_Middle\_Double\_Click

On\_Shift\_Middle\_Triple\_Click

On\_Shift\_Right\_Click

On\_Shift\_Right\_Release

On\_Shift\_Right\_Double\_Click

On\_Shift\_Right\_Triple\_Click

On\_Shift\_Drag

On\_Shift\_Middle\_Drag

On\_Shift\_Right\_Drag

On\_Shift\_Mouse\_Wheel

On\_Shift\_Hover\_In

On\_Shift\_Hover\_Out

# Config

Multiple values in config can be set and get at same time

Widget.Config(Background=’#F5F5F5’, Foreground=’red’, …..)

Widget.Config\_Get(‘Background’, ‘Foreground’, …..)

Will return a dictionary of requested configurations

## Background

Sets the background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Background=’#F5F5F5’)

## Light\_Background

Sets the light mode background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Background=’#F5F5F5’)

## Dark\_Background

Sets the dark mode background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Background=’#F5F5F5’)

## Foreground

Sets the foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’

Widget.Config(Foreground=’#F5F5F5’)

## Light\_Foreground

Sets the light mode foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Foreground=’#F5F5F5’)

## Dark\_Foreground

Sets the dark mode foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Foreground=’#F5F5F5’)

## Border\_Color

Sets the border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. This feature does not work on LITE objects.

Widget.Config(Border\_Color=’#000000’)

## Light\_Border\_Color

Sets the light mode border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Border\_Color =’#F5F5F5’)

## Dark\_Border\_Color

Sets the dark mode border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Border\_Color =’#F5F5F5’)

## Border\_Size

Sets the border size of widget. In pixel values. This feature does not work on LITE objects.

Widget.Config(Border\_Size=2)

## Resize

Defines if the widget can be resized when main root window resizes.

Widget.Config(Resize=True)

## Resize\_Width

Defines if the widget can be resized only width wise main root window resizes.

Widget.Config(Resize\_Width=True)

## Resize\_Height

Defines if the widget can be resized only height wise main root window resizes.

Widget.Config(Resize\_Height=True)

## Move

Defines if the widget can be moved when main root window resizes.

Widget.Config(Move=True)

## Move\_Top

Defines if the widget can be moved only up and down main root window resizes.

Widget.Config(Move\_Top=True)

## Move\_Left

Defines if the widget can be resized only left and right main root window resizes.

Widget.Config(Move\_Left=True)

## Top

Changes the top position of widget. Only accepted in pixels.

Widget.Config(Top=100)

## Left

Changes the left position of widget. Only accepted in pixels.

Widget.Config(Left=150)

## Width

Changes the width of widget. Only accepted in pixels.

Widget.Config(Width=100)

\* If the inside widget is provided in percentage, changed the main widgets will affect the inside widget after the application restarts

## Height

Changes the height of widget. Only accepted in pixels.

Widget.Config(Height=100)

\* If the inside widget is provided in percentage, changed the main widgets will affect the inside widget after the application restarts

## Font\_Size

Font size for the widgets which support text. Only accepted in pixels.

Widget.Config(Font\_Size=20)

## Font\_Weight

Font weight for the widgets which support text. ‘normal’, ‘bold’

Widget.Config(Font\_Weight=’normal’)

## Font\_Family

Font family for the widgets which support text. Only tKinter specific.

'Times New Roman', 'Helvetica', etc.

Widget.Config(Font\_Family= 'Times New Roman')

## Scrollbar

Width of scrollbar. Only accepted in pixels.

Widget.Config(Scrollbar=20)

\* Exclusive to scroll frame

## Vertical

If vertical scroll bar should be displayed at initial state of frame.

Widget.Config(Vertical=True)

\* Exclusive to scroll frame

## Horizontal

If horizontal scroll bar should be displayed at initial state of frame.

Widget.Config(Horizontal=True)

\* Exclusive to scroll frame

## Last

Define the last widget of scrolls frame, the widget which is most right and bottom of frame.

Widget.Config(Last=Widget)

\* Exclusive to scroll frame

## Value

Sets text value for widgets that support text.

Widget.Config(Value=’Click Here’)

## Ridge

If the button should have ridged border

Widget.Config(Ridge=True)

\* Exclusive to button

## Disable

Disbales the function of widget

Widget.Config(Disbale=True)

\* Exclusive to button, list, entry, select, spinner, scale, text

## Path

Gives the image path for widget

Widget.Config(Path=’./Image.png’)

\* Exclusive to image, compound, canvas image

## URL

Defines if image path given is a http url

Widget.Config(Url=True)

\* Exclusive to image, compound, canvas image

## Array

Defines if image path is cv2 array

Widget.Config(Array=True)

\* Exclusive to image, compound, canvas image

## Pil

Defines if image path is pillow image

Widget.Config(Pil=True)

\* Exclusive to image, compound, canvas image

## Photo

Defines if image path is tk photo image

Widget.Config(Photo=True)

\* Exclusive to canvas image

## Rotate

Rotates the image to specific degree of angle

Widget.Config(Rotate=90)

\* Exclusive to image, compound, canvas image

## Transparent

Defines if the provided image support to be transparent in nature.

Widget.Config(Transparent=True)

\* Exclusive to image, compound, canvas image

## Aspect\_Ratio

If the given image should keep aspect ration when fitting to current widget.

Widget.Config(Aspect\_Ratio=True)

\* Exclusive to image, compound, canvas image

## Compound

Location of text on the image. ‘left’, ‘right’, ‘top’, ‘bottom’, ‘center’

Widget.Config(Compound=’center’)

\* Exclusive to compound

## Align

Location of text in widget

Widget.Config(Align=’center’)

\* Entry: ‘left’, ‘right’, ‘center’

\* Spinner: ‘left’, ‘right’, ‘center’

\* Label: ‘n’, ‘e’, ‘w’, ‘s’, ‘ne’, ‘nw’, ‘se’, ‘sw’, ‘center’

## Secure

Hides the entry text with \*

Widget.Config(Secure=True)

\* Exclusive to Entry

## Variable

Provides the variable to multiple radio buttons to store the radio value.

Widget.Config(Variable=Temp\_Variable)

\* Exclusive to Radio

## Minimum

Minimum value for widget

Widget.Config(Minimum=0)

\* Exclusive to Scale, Spinner

## Maximum

Maximum value for widget

Widget.Config(Maximum=100)

\* Exclusive to Scale, Spinner

## Increment

Increment step for widget

Widget.Config(Increment=1)

\* Exclusive to Scale, Spinner

## Orient

Orientation of the widget

Widget.Config(Orient=’Horizontal’)

\* Exclusive to Scale, Seperator

## Height\_List

Height of the list when on click widget

Widget.Config(Height\_List=500)

\* Exclusive to Select

## Progress

Initial value for the widget

Widget.Config(Progress=10)

\* Exclusive to Bar

## Zero

Position of widget. ‘Left’, ‘Right’, ‘Top’, ‘Bottom’

Widget.Config(Zero=’Left’)

\* Exclusive to Bar

## Radius

Radius for rounded corners.

Widget.Config(Radius=20)

\* Exclusive to Canvas, Roubel

## Shadow\_Size

Applies shadow size to all sides.

Widget.Config(Shadow\_Size=10)

\* Exclusive to Canvas, Roubel

## Shadow\_Color

Sets the shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Shadow\_Color=’#FF0000’)

\* Exclusive to Canvas, Roubel

## Light\_Shadow\_Color

Sets the light mode shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Shadow\_Color =’#F5F5F5’)

## Dark\_Shadow\_Color

Sets the dark mode shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Shadow\_Color =’#F5F5F5’)

## Background\_Selected

Background color of current selected item.

Widget.Config(Background\_Selected=’#FFFFFF’)

\* Exclusive to Tree

## Foreground\_Selected

Foreground color of current selected item.

Widget.Config(Foreground\_Selected=’#000000’)

\* Exclusive to Tree

## Translucent

Makes Fill Color Translucent.

Widget.Config(Translucent=True)

\* Exclusive to Canvas Item Rectangle, Rectangle2, Circle, Oval, Arc, Pie, & Polygon

## Multiple

Makes Multiple Selection.

Widget.Config(Multiple=True)

\* Exclusive to List

## Select\_Background

Sets the select text background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Select\_Background=’#F5F5F5’)

## Select\_Foreground

Sets the select text foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’

Widget.Config(Select\_Foreground=’#F5F5F5’)

## Disable\_Background

Sets the disbaled background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Disable\_Background=’#F5F5F5’)

## Disable\_Foreground

Sets the disbaled foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’

Widget.Config(Disable\_Foreground=’#F5F5F5’)

## Hover\_Background

Sets the hover background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Hover\_Background=’#F5F5F5’)

## Hover\_Foreground

Sets the hover foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Hover\_Foreground=’#F5F5F5’)

## Hover\_Border\_Color

Sets the hover border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Hover\_Border\_Color=’#F5F5F5’)

## Hover\_Shadow\_Color

Sets the hover shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Hover\_Shadow\_Color=’#F5F5F5’)

## Light\_Hover\_Background

Sets the light mode hover background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Hover\_Background=’#F5F5F5’)

## Light\_Hover\_Foreground

Sets the light mode hover foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Hover\_Foreground=’#F5F5F5’)

## Light\_Hover\_Border\_Color

Sets the light mode hover border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Hover\_Border\_Color=’#F5F5F5’)

## Light\_Hover\_Shadow\_Color

Sets the light mode hover shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Light\_Hover\_Shadow\_Color=’#F5F5F5’)

## Dark\_Hover\_Background

Sets the dark mode hover background of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Hover\_Background=’#F5F5F5’)

## Dark\_Hover\_Foreground

Sets the dark mode hover foreground of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Hover\_Foreground=’#F5F5F5’)

## Dark\_Hover\_Border\_Color

Sets the dark mode hover border color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Hover\_Border\_Color=’#F5F5F5’)

## Dark\_Hover\_Shadow\_Color

Sets the dark mode hover shadow color of widget. Can be simple HTML values like ‘red’, ‘green’, etc. or any Hex color value ‘#F5F5F5’. False, it will take the background of the Main Frame of the Widget

Widget.Config(Dark\_Hover\_Shadow\_Color=’#F5F5F5’)