

---

# Kivy Cupertino

*Release 0.1.1-beta*

**cmdvmd, arnavs-0**

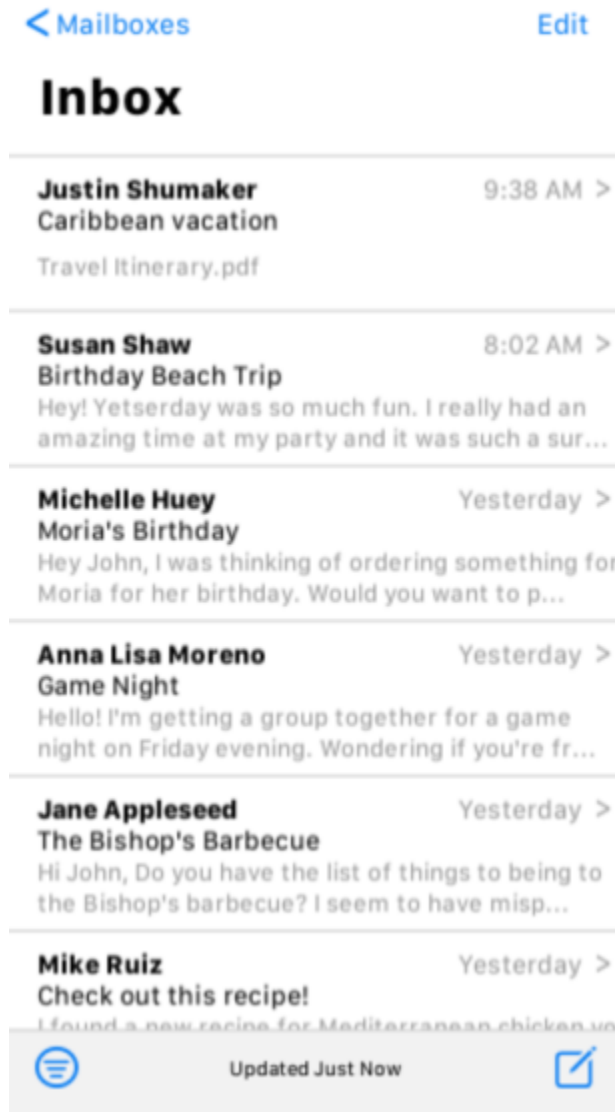
**Dec 20, 2022**



# CONTENTS

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Bar . . . . .	3
1.2	Behavior . . . . .	7
1.3	Button . . . . .	9
1.4	Control . . . . .	15
1.5	Indicator . . . . .	19
1.6	Label . . . . .	21
1.7	Modal . . . . .	23
1.8	Page . . . . .	32
1.9	ScrollView . . . . .	34
1.10	Slider . . . . .	34
1.11	Swipe . . . . .	36
1.12	Switch . . . . .	40
1.13	Symbol . . . . .	42
1.14	Table . . . . .	43
1.15	Textinput . . . . .	46
<b>2</b>	<b>Contributing</b>	<b>51</b>
<b>3</b>	<b>License</b>	<b>53</b>
<b>4</b>	<b>Indices and tables</b>	<b>55</b>
	<b>Python Module Index</b>	<b>57</b>
	<b>Index</b>	<b>59</b>





Kivy Cupertino is a collection of iOS style widgets following [Apple's Human Interface guidelines](#) to be used with the [Kivy](#) cross-platform graphical framework without sacrificing ease of use or application performance.

**Warning:** We are currently in **beta** status; most widgets have been implemented but stability is not guaranteed. We don't recommend using Kivy Cupertino in production.



## INSTALLATION

To install Kivy Cupertino from [PyPi](#), use pip:

```
$ pip install kivycupertino
```

To use newly implemented widgets that have not yet been published to PyPI, install from GitHub (**not recommended for production use**):

```
$ pip install git+https://github.com/cmdvmd/kivy-cupertino.git
```

To install Kivy Cupertino for development use, install with the *dev* extra:

### 1.1 Bar

Bars are generally positioned at the top or bottom of a screen and contain widgets and/or information for easy access by users

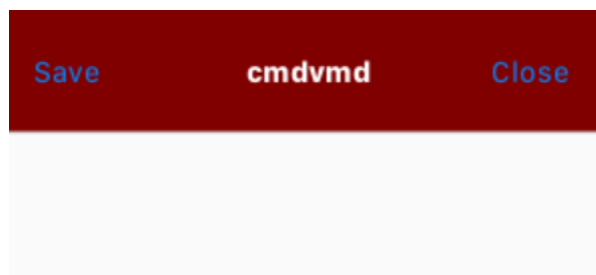
**class** `kivycupertino.uix.bar.CupertinoNavigationBar(*args: Any, **kwargs: Any)`

iOS style Navigation Bar. [CupertinoNavigationBar](#) is a `RelativeLayout` and can accept any number of widgets



**color** = `[0.95, 0.95, 0.95, 1]`

Background color of [CupertinoNavigationBar](#)



**Python**

```
CupertinoNavigationBar(color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoNavigationBar:  
    color: 0.5, 0, 0, 1
```

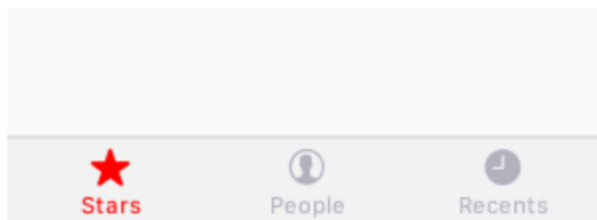
**class** kivycupertino.uix.bar.CupertinoTab(\*args: Any, \*\*kwargs: Any)

iOS style tab to be used with [CupertinoTabBar](#)



**color\_selected** = [0.2, 0.45, 1, 1]

Color of the selected tab of [CupertinoTab](#)

**Python**

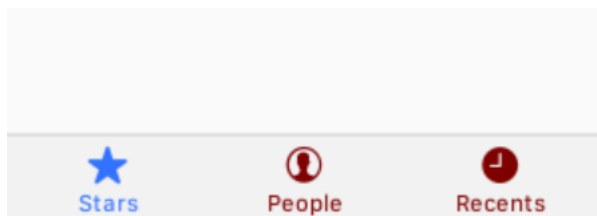
```
CupertinoTabBar(color_selected=(1, 0, 0, 1))
```

**KV**

```
CupertinoTab:  
    color_selected: 1, 0, 0, 1
```

**color\_unselected** = [0.7, 0.7, 0.75, 1]

Color of [CupertinoTabBar](#) when not selected

**Python**

```
CupertinoTab(color_unselected=(0.5, 0, 0, 1))
```

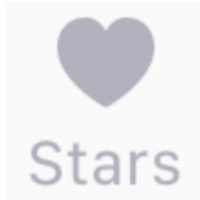
**KV**



```
CupertinoTab:
    color_unselected: 0.5, 0, 0, 1
```

**symbol** = ' '

Symbol of *CupertinoTab*



**Python**

```
CupertinoTab(symbol='hammer_fill')
```

**KV**

```
CupertinoTab:
    symbol: 'hammer_fill'
```

**text** = ' '

Text of *CupertinoTab*



**Python**

```
CupertinoTab(text='Tab')
```

**KV**

```
CupertinoTab:
    text: 'Tab'
```

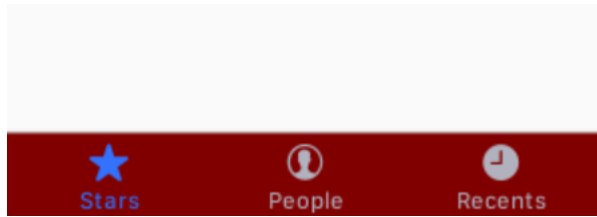
```
class kivycupertino.uix.bar.CupertinoTabBar(*args: Any, **kwargs: Any)
    iOS style tab bar
```

**add\_widget**(*widget*, *index=0*, *canvas=None*)

Add an instance of *CupertinoTab* to *CupertinoTabBar*

**background\_color** = [0.95, 0.95, 0.95, 1]

Background color of *CupertinoTabBar* when selected

**Python**

```
CupertinoTabBar(background_color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoTabBar:
    background_color: 0.5, 0, 0, 1
```

**get\_selected\_tab()**

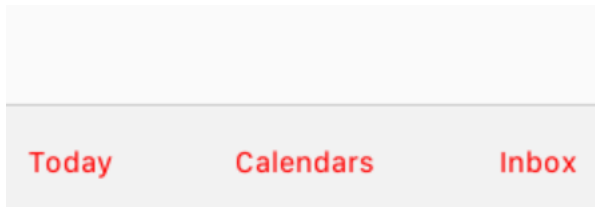
Get the currently selected tab of *CupertinoTabBar*

**Returns**

The selected *CupertinoTab*

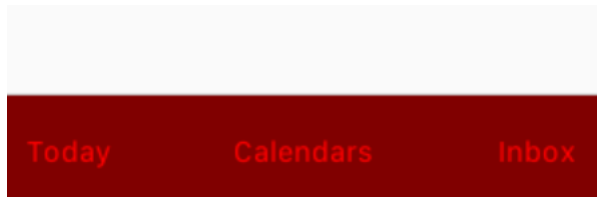
**class** kivycupertino.uix.bar.**CupertinoToolbar**(\*args: Any, \*\*kwargs: Any)

iOS style Toolbar. *CupertinoToolbar* is a *RelativeLayout* and can accept any number of widgets



```
color = [0.95, 0.95, 0.95, 1]
```

Background color of *CupertinoToolbar*

**Python**

```
CupertinoToolbar(color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoToolbar:
    color: 0.5, 0, 0, 1
```

## 1.2 Behavior

Behaviors allow for expanded functionality for existing widgets

---

**Note:** Behaviors can only be used as superclasses for instances of `Widget`

---

**class** `kivycupertino.uix.behavior.CupertinoButtonBehavior(*args: Any, **kwargs: Any)`

Base class for buttons that can only be used with an instance of `kivy.uix.widget.Widget`

**color**

Current color of a widget with *CupertinoButtonBehavior*

**color\_disabled**

Color of widget with *CupertinoButtonBehavior* when disabled



**Python**

```
CupertinoButton(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_down**

Color of widget with *CupertinoButtonBehavior* when pressed

**Python**

```
CupertinoButton(color_down=(1, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    color_down: 1, 0, 0, 1
```

**color\_normal**

Color of widget with *CupertinoButtonBehavior* when not pressed or disabled



**Python**

```
CupertinoButton(color_normal=(1, 0, 0, 1))
```

**KV**

```
CupertinoButton:  
    color_normal: 1, 0, 0, 1
```

**disabled**

If widget with *CupertinoButtonBehavior* is disabled

**Python**

```
CupertinoButton(disabled=True)
```

**KV**

```
CupertinoButton:  
    disabled: True
```

**transition\_duration = 0.075**

Duration of the transition of the color of widget with *CupertinoButtonBehavior* when its state changes

**Python**

```
CupertinoButton(transition_duration=0.5)
```

**KV**

```
CupertinoButton:  
    transition_duration: 0.5
```

```
class kivycupertino.uix.behavior.LongPressBehavior(**kwargs)
```

Behavior to detect a long press on a widget

**long\_press\_duration = 1**

Time that constitutes a long press

**Python**

```
ExampleWidget(long_press_duration=2)
```

**KV**

```
ExampleWidget:  
    long_press_duration: 2
```

**on\_long\_press**(*touch*)

Event when a widget is long pressed

**Parameters**

**touch** – The touch on this widget

**class** kivycupertino.uix.behavior.**SelectableBehavior**(\*\*kwargs)

Behavior to detect a selection of a specific widget among all other widgets in parent

**on\_chosen**()

Event when a widget is selected

**on\_selected**(*instance, value*)

Callback when *selected* of a widget with *SelectableBehavior* is changed

**Parameters**

- **instance** – Instance of widget with *SelectableBehavior*
- **value** – Value of *selected*

**refresh**()

Set *selected* of this instance of *SelectableBehavior* to True while setting all other instances in parent to False

---

**Note:** This function is mainly to be used when adding widgets to a parent. Otherwise, setting *selected* to True will achieve this

---

**selected = False**

If a widget is selected in its parent



**Python**

```
ExampleWidget(selected=True)
```

**KV**

```
ExampleWidget:
    selected: True
```

## 1.3 Button

Buttons allow users to execute actions with a single tap

**class** kivycupertino.uix.button.**CupertinoButton**(\*args: Any, \*\*kwargs: Any)

iOS style button

**color\_disabled = [0, 0.35, 0.7, 1]**Background color of *CupertinoButton* when disabled**Python**

```
CupertinoButton(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_down = [0, 0.15, 0.8, 1]**Background color of *CupertinoButton* when pressed**Python**

```
CupertinoButton(color_down=(1, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    color_down: 1, 0, 0, 1
```

**color\_normal = [0, 0.5, 1, 1]**Color of *CupertinoButton* when not pressed or disabled**Python**

```
CupertinoButton(color_normal=(1, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    color_normal: 1, 0, 0, 1
```

**disabled = False**If *CupertinoButton* is disabled**Python**

```
CupertinoButton(disabled=True)
```

**KV**

```
CupertinoButton:
    disabled: True
```

```
font_size = '17sp'
```

text of *CupertinoButton*

**Python**

```
CupertinoButton(font_size='20sp')
```

**KV**

```
CupertinoButton:
    font_size: '20sp'
```

```
text = ' '
```

Text of *CupertinoButton*

**Python**

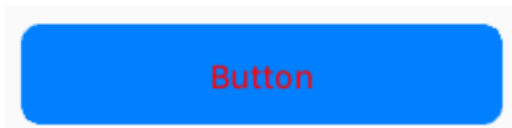
```
CupertinoButton(text='Hello World')
```

**KV**

```
CupertinoButton:
    text: 'Hello World'
```

```
text_color = [1, 1, 1, 1]
```

A ColorProperty defining the color of text of *CupertinoButton*

**Python**

```
CupertinoButton(text_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoButton:
    color_disabled: 1, 0, 0, 1
```

**transition\_duration = 0.075**

Duration of the transition of the color of *CupertinoButton* when its state changes

### Python

```
CupertinoButton(transition_duration=0.5)
```

### KV

```
CupertinoButton:
    transition_duration: 0.5
```

**class** kivycupertino.uix.button.**CupertinoSymbolButton**(\*args: Any, \*\*kwargs: Any)

iOS style button that displays a symbol

**color\_disabled = [0, 0, 0, 0.7]**

Color of *CupertinoSymbolButton* when disabled



### Python

```
CupertinoSymbolButton(color_disabled=(0.5, 0, 0, 1))
```

### KV

```
CupertinoSymbolButton:
    color_disabled: 0.5, 0, 0, 1
```

**color\_down = [0, 0, 0, 0.7]**

Color of *CupertinoSymbolButton* when pressed

### Python

```
CupertinoSymbolButton(color_down=(1, 0, 0, 1))
```

### KV

```
CupertinoSymbolButton:
    color_down: 1, 0, 0, 1
```

**color\_normal = [0, 0, 0, 1]**

Color of *CupertinoSymbolButton* when not pressed or disabled





**Python**

```
CupertinoSymbolButton(color_normal=(1, 0, 0, 1))
```

**KV**

```
CupertinoSymbolButton:
    color_normal: 1, 0, 0, 1
```

**disabled = False**

If *CupertinoSymbolButton* is disabled

**Python**

```
CupertinoSymbolButton(disabled=True)
```

**KV**

```
CupertinoSymbolButton:
    disabled: True
```

**symbol = ''**

Symbol of *CupertinoSymbolButton*

**Python**

```
CupertinoSymbolButton(symbol='wifi')
```

**KV**

```
CupertinoSymbolButton:
    symbol: 'wifi'
```

**transition\_duration = 0.075**

Duration of the transition of the color of *CupertinoSymbolButton* when its state changes

**Python**

```
CupertinoSymbolButton(transition_duration=0.5)
```

**KV**

```
CupertinoSymbolButton:
    transition_duration: 0.5
```

```
class kivycupertino.uix.button.CupertinoSystemButton(*args: Any, **kwargs: Any)
    iOS style System Button
```

**color\_disabled = [0, 0.3, 0.4, 1]**Color of *CupertinoSystemButton* when disabled**System Button****Python**

```
CupertinoSystemButton(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoSystemButton:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_down = [0, 0.15, 0.3, 1]**Color of *CupertinoSystemButton* when pressed**Python**

```
CupertinoSystemButton(color_down=(1, 0, 0, 1))
```

**KV**

```
CupertinoSystemButton:
    color_down: 1, 0, 0, 1
```

**color\_normal = [0.05, 0.5, 0.95, 1]**Color of *CupertinoSystemButton* when not pressed or disabled**System Button****Python**

```
CupertinoSystemButton(color_normal=(1, 0, 0, 1))
```

**KV**

```
CupertinoSystemButton:
    color_normal: 1, 0, 0, 1
```

**disabled = False**If *CupertinoSystemButton* is disabled**System Button****Python**

```
CupertinoSystemButton(disabled=True)
```

**KV**

```
CupertinoSystemButton:
    disabled: True
```

**font\_size = '15sp'**

Font size of the text of *CupertinoSystemButton*

System Button

**Python**

```
CupertinoSystemButton(font_size='20sp')
```

**KV**

```
CupertinoSystemButton:
    font_size: '20sp'
```

**text = ''**

Text of *CupertinoSystemButton*

Send

**Python**

```
CupertinoSystemButton(text='Send')
```

**KV**

```
CupertinoSystemButton:
    text: 'Send'
```

**transition\_duration = 0.075**

Duration of the transition of the color of *CupertinoSystemButton* when its state changes

**Python**

```
CupertinoSystemButton(transition_duration=0.5)
```

**KV**

```
CupertinoSystemButton:
    transition_duration: 0.5
```

## 1.4 Control

Controls allow users to control information on their screen

**class** kivycupertino.uix.control.**CupertinoSegment**(\*args: Any, \*\*kwargs: Any)

iOS style segment to be used with *CupertinoSegmentedControls*

**color = [0, 0, 0, 1]**

Color of text of *CupertinoSegment*

**Python**

```
CupertinoSegment(color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSegment:
    text_color: 1, 0, 0, 1
```

**class** kivycupertino.uix.control.CupertinoSegmentedControls(\*args: Any, \*\*kwargs: Any)  
iOS style Segmented Controls

**add\_widget**(widget, index=0, canvas=None)

Add an instance of [CupertinoSegment](#) to [CupertinoSegmentedControls](#)

**background\_color** = [0.9, 0.9, 0.9, 0.75]

Background color of [CupertinoSegmentedControls](#)

**Python**

```
CupertinoSegmentedControls(background_color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoSegmentedControls:
    background_color: 0.5, 0, 0, 1
```

**color\_selected** = [1, 1, 1, 1]

Background color of selected tab of [CupertinoSegmentedControls](#)

**Python**

```
CupertinoSegmentedControls(color_selected=(1, 0, 0, 1))
```

**KV**

```
CupertinoSegmentedControls:
    color_selected: 1, 0, 0, 1
```

**get\_selected\_segment**()

Get the currently selected segment of [CupertinoSegmentedControls](#)

**Returns**

The selected *CupertinoSegment*

**on\_touch\_move(touch)**

Detect a movement on a segment of *CupertinoSegmentedControls*

**Parameters**

**touch** – Touch on *CupertinoSegmentedControls*

**transition\_duration = 0.1**

Duration of change of selected segment of *CupertinoSegmentedControls*

**Python**

```
CupertinoSegmentedControls(transition_duration=0.5)
```

**KV**

```
CupertinoSegmentedControls:
    transition_duration: 0.5
```

**class** kivycupertino.uix.control.**CupertinoStepper**(\*args: Any, \*\*kwargs: Any)  
iOS style Stepper

**add\_disabled = False**

If add button of *CupertinoStepper* is disabled

**Python**

```
CupertinoStepper(add_disabled=True)
```

**KV**

```
CupertinoStepper:
    add_disabled: True
```

**color\_disabled = [0.8, 0.8, 0.8, 1]**

Background color of button of *CupertinoStepper* when disabled

**Python**

```
CupertinoStepper(add_disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoStepper:  
    add_disabled: True  
    color_disabled: 0.5, 0, 0, 1
```

**color\_down** = [0.7, 0.7, 0.7, 1]

Background color of button of *CupertinoStepper* when valid

#### Python

```
CupertinoStepper(color_down=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoStepper:  
    color_down: 0.5, 0, 0, 1
```

**color\_normal** = [0.9, 0.9, 0.9, 0.75]

Background color of button of *CupertinoStepper* when not valid



#### Python

```
CupertinoStepper(color_normal=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoStepper:  
    color_normal: 0.5, 0, 0, 1
```

**minus\_disabled** = False

If minus button of *CupertinoStepper* is disabled



#### Python

```
CupertinoStepper(minus_disabled=True)
```

#### KV

```
CupertinoStepper:  
    minus_disabled: True
```

**on\_minus()**

Callback when minus button is valid

**on\_plus()**

Callback when plus button is valid

**text\_color = [0, 0, 0, 1]**

Color of text of button of *CupertinoStepper*



#### Python

```
CupertinoStepper(text_color=(1, 0, 0, 1))
```

#### KV

```
CupertinoStepper:
    text_color: 1, 0, 0, 1
```

## 1.5 Indicator

Indicators help show progress to users

**class** kivycupertino.uix.indicator.**CupertinoActivityIndicator**(\*args: Any, \*\*kwargs: Any)  
iOS style activity indicator

**color = [0.6, 0.6, 0.65, 1]**

Color of the spokes of *CupertinoActivityIndicator*

#### Python

```
CupertinoActivityIndicator(color=(1, 0, 0, 1))
```

#### KV

```
CupertinoActivityIndicator:
    color: 1, 0, 0, 1
```

**duration = 1**

Time for one cycle of *CupertinoActivityIndicator* (in seconds)

#### Python

```
CupertinoActivityIndicator(duration=2)
```

#### KV

```
CupertinoActivityIndicator:
    duration: 2
```

**playing = False**

Color of the spokes of *CupertinoActivityIndicator*

#### Python

```
CupertinoActivityIndicator(playing=True)
```

**KV**

```
CupertinoActivityIndicator:  
    playing: True
```

**spokes = 12**

Amount of spokes of *CupertinoActivityIndicator*

**Python**

```
CupertinoActivityIndicator(spokes=20)
```

**KV**

```
CupertinoActivityIndicator:  
    spokes: 20
```

**class** kivycupertino.uix.indicator.**CupertinoProgressbar**(\*args: Any, \*\*kwargs: Any)  
 iOS style Progress Bar

**color\_selected = [0, 0.5, 1, 1]**

Color of the bar of occurred progress of CupertinoProgressbar

**Python**

```
CupertinoProgressbar(color_selected=(1, 0, 0, 1))
```

**KV**

```
CupertinoProgressbar:  
    color_selected: 1, 0, 0, 1
```

**color\_unselected = [0.7, 0.7, 0.7, 1]**

Color of the bar of not yet occurred progress of CupertinoProgressbar

**Python**

```
CupertinoProgressbar(color_unselected=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoProgressbar:  
    color_unselected: 0.5, 0, 0, 1
```

**on\_value**(instance, value)

Callback when value of *CupertinoProgressbar*

**Parameters**



- **instance** – Instance of *CupertinoProgressbar*
- **value** – Value of *CupertinoProgressbar*

**value = 0**

Amount of progress occurred in interval *[0, 1]* of *CupertinoProgressbar*



**Python**

```
CupertinoProgressbar(value=50)
```

**KV**

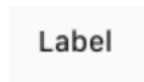
```
CupertinoProgressbar:
    value: 50
```

## 1.6 Label

Labels display text to users

**class** `kivycupertino.uix.label.CupertinoLabel(*args: Any, **kwargs: Any)`

iOS style Label



**bold = False**

If *text CupertinoLabel* is bold



**Python**

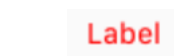
```
CupertinoLabel(bold=True)
```

**KV**

```
CupertinoLabel:
    bold: True
```

**color = [0, 0, 0, 1]**

Color of *text CupertinoLabel*



**Python**

```
CupertinoLabel(color=(1, 0, 0, 1))
```

**KV**

```
CupertinoLabel:
    color: 1, 0, 0, 1
```

**font\_name = 'San Francisco'**

Font of *CupertinoLabel*. To comply with iOS standard, use *San Francisco* or *New York*

Label

**Python**

```
CupertinoLabel(font_name='New York')
```

**KV**

```
CupertinoLabel:
    font_name: 'New York'
```

**font\_size = '15sp'**

Size of the font of *CupertinoLabel*

Label

**Python**

```
CupertinoLabel(font_size='20sp')
```

**KV**

```
CupertinoLabel:
    font_size: '20sp'
```

**italic = False**

If *text* of *CupertinoLabel* is italic

Label

**Python**

```
CupertinoLabel(italic=True)
```

**KV**

```
CupertinoLabel:
    italic: True
```

**text = ''**

A StringProperty defining the text of *CupertinoLabel*

Hello World

**Python**

```
CupertinoLabel(text='Hello World')
```

**KV**

```
CupertinoLabel:
    text: 'Hello World'
```

## 1.7 Modal

Modals help alert users to information

**class** kivycupertino.uix.modal.**CupertinoActionSheet**(\*args: Any, \*\*kwargs: Any)  
iOS style Action Sheet

**action\_height**

Height of *CupertinoModalButton* when added to *CupertinoActionSheet*

**Python**

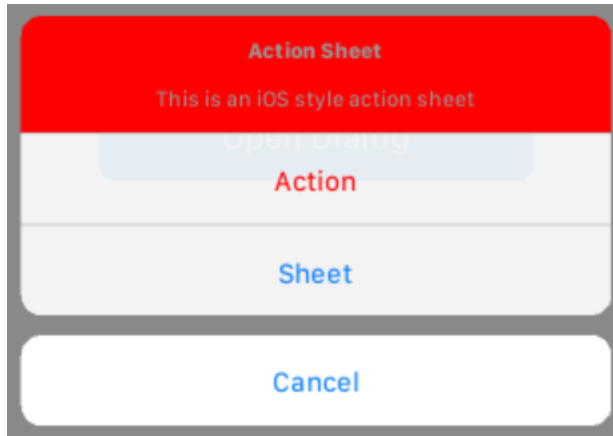
```
CupertinoActionSheet(action_height=75)
```

**KV**

```
CupertinoActionSheet:
    action_height: 75
```

**color\_normal** = [1, 1, 1, 0.9]

Background color of message frame of *CupertinoActionSheet*



#### Python

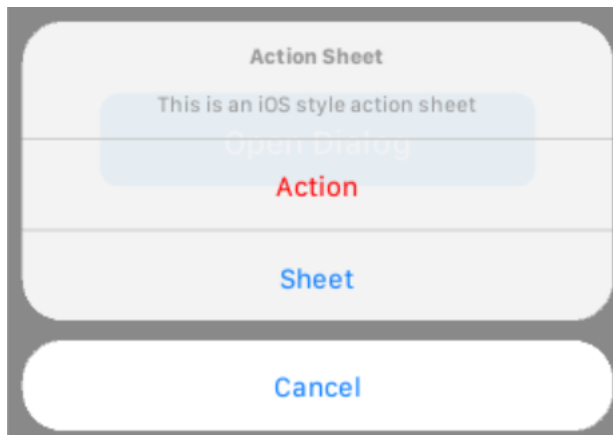
```
CupertinoActionSheet(color_normal=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoActionSheet:
    color_normal: 0.5, 0, 0, 1
```

#### curve

Curve of *CupertinoActionSheet*



#### Python

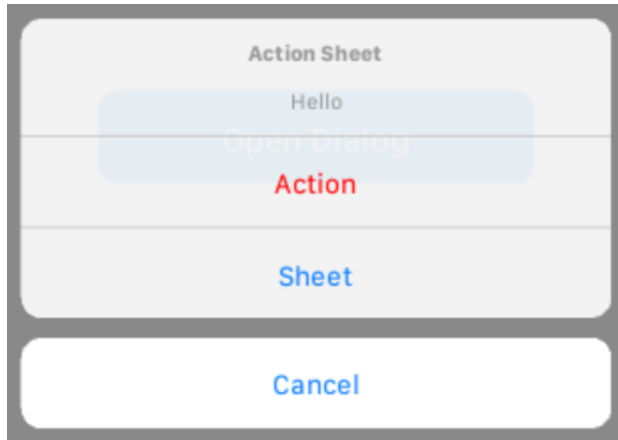
```
CupertinoActionSheet(curve=20)
```

#### KV

```
CupertinoActionSheet:
    curve: 20
```

#### message = ' '

Message shown in message frame of *CupertinoActionSheet*

**Python**

```
CupertinoActionSheet(message='Hello World')
```

**KV**

```
CupertinoActionSheet:
    message: 'Hello World'
```

**spacing**

Spacing between children of *CupertinoActionSheet*

**Python**

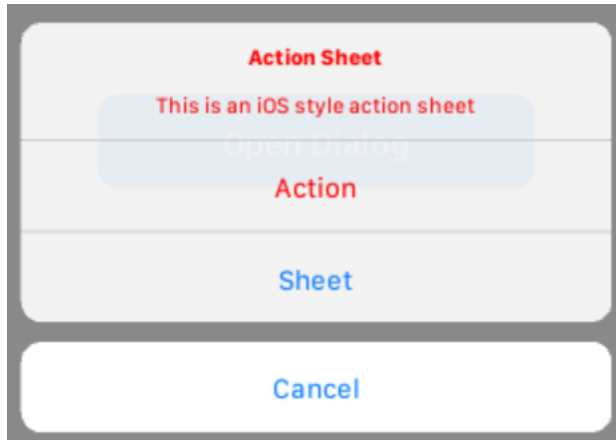
```
CupertinoActionSheet(spacing=5)
```

**KV**

```
CupertinoActionSheet:
    spacing: 5
```

**text\_color = [0.6, 0.6, 0.6, 1]**

Color of *title* and *message* shown in message frame of *CupertinoActionSheet*

**Python**

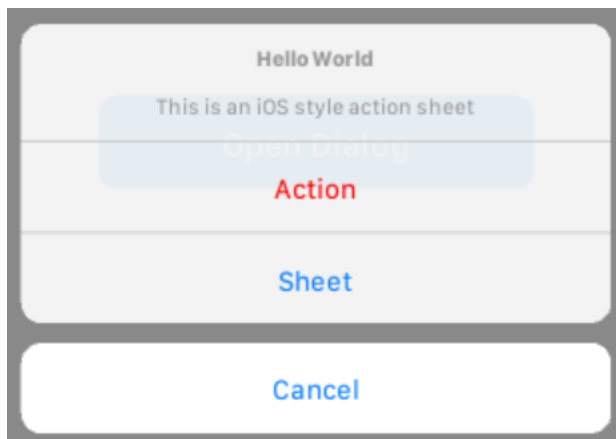
```
CupertinoActionSheet(text_color=[1, 0, 0, 1])
```

**KV**

```
CupertinoActionSheet:
    text_color: 1, 0, 0, 1
```

**title = ''**

Title shown in message frame of *CupertinoActionSheet*

**Python**

```
CupertinoActionSheet(title='Hello World')
```

**KV**

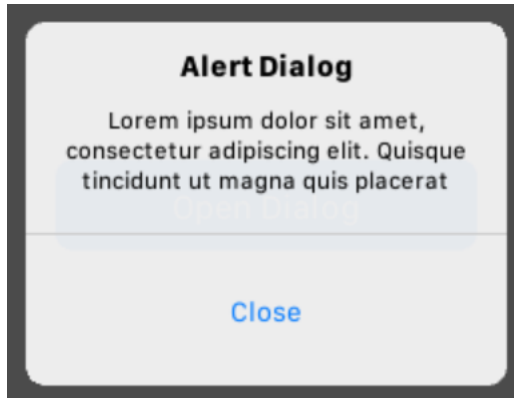
```
CupertinoActionSheet:
    title: 'Hello World'
```

**class** kivycupertino.uix.modal.**CupertinoDialog**(\*args: Any, \*\*kwargs: Any)

iOS style dialog that dynamically adapts to the amount of actions (*CupertinoModalButton*) it has

**action\_height**

Height of *CupertinoModalButton* when added to *CupertinoDialog*

**Python**

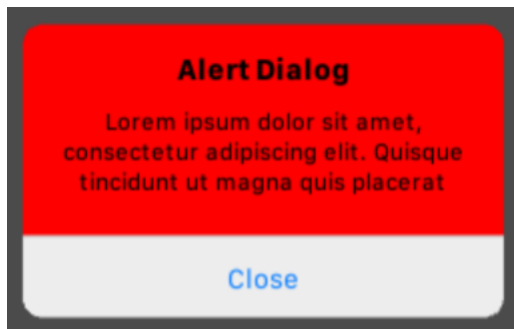
```
CupertinoDialog(action_height=75)
```

**KV**

```
CupertinoDialog:
    action_height: 75
```

**color = [1, 1, 1, 0.9]**

Background color of *CupertinoDialog*

**Python**

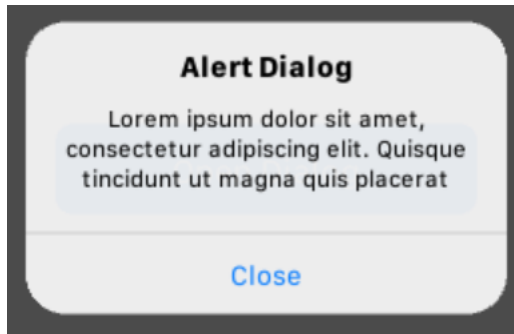
```
CupertinoDialog(color=(1, 0, 0, 1))
```

**KV**

```
CupertinoDialog:
    color: 1, 0, 0, 1
```

**curve**

Curve of *CupertinoDialog*

**Python**

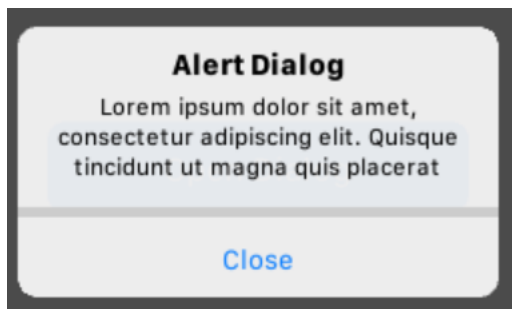
```
CupertinoDialog(curve=20)
```

**KV**

```
CupertinoDialog:
    curve: 20
```

**spacing**

Spacing between children of *CupertinoDialog*

**Python**

```
CupertinoDialog(spacing=5)
```

**KV**

```
CupertinoDialog:
    spacing: 5
```

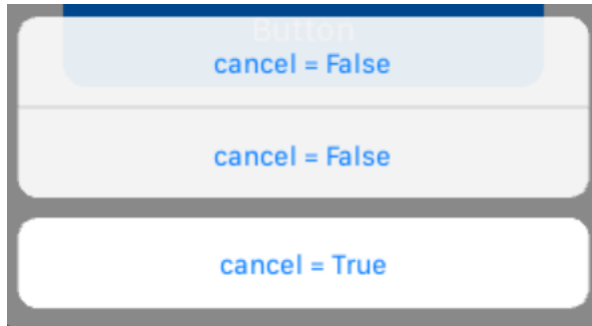
```
class kivycupertino.uix.modal.CupertinoModalButton(*args: Any, **kwargs: Any)
```

Adaptive button to be used in Dialogs

**cancel = False**

If *CupertinoModalButton* should be a cancel button when added to an instance of *CupertinoActionSheet*



**Python**

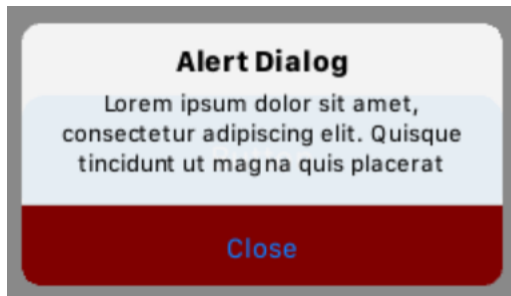
```
CupertinoModalButton(cancel=True)
```

**KV**

```
CupertinoModalButton:
    cancel: True
```

**color\_disabled = [0.8, 0.8, 0.8, 1]**

Background color of *CupertinoModalButton* when disabled

**Python**

```
CupertinoModalButton(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoModalButton:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_down = [0.9, 0.9, 0.9, 0.9]**

Background color of *CupertinoModalButton* when pressed

**Python**

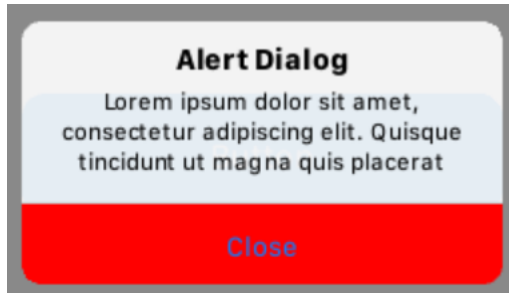
```
CupertinoModalButton(color_down=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoModalButton:
    color_down: 0.5, 0, 0, 1
```

**color\_normal** = [1, 1, 1, 0.9]

Background color of *CupertinoModalButton* when not pressed



#### Python

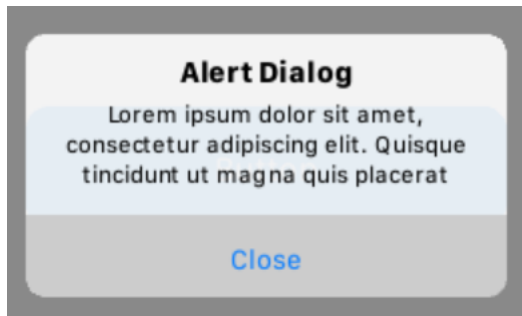
```
CupertinoModalButton(color_normal=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoModalButton:  
    color_normal: 0.5, 0, 0, 1
```

**disabled** = False

If *CupertinoModalButton* is disabled



#### Python

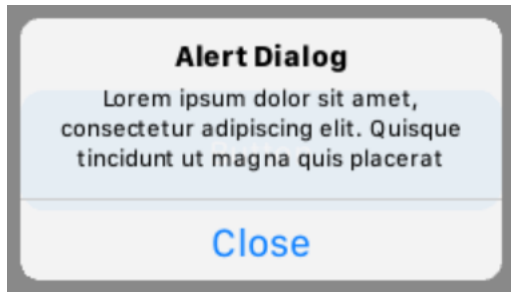
```
CupertinoModalButton(disabled=True)
```

#### KV

```
CupertinoModalButton:  
    disabled: True
```

**font\_size** = '14sp'

Size of text of *CupertinoModalButton*

**Python**

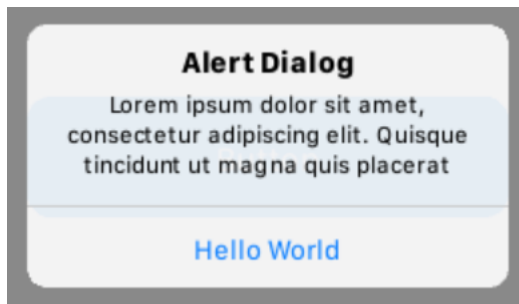
```
CupertinoModalButton(font_size='20sp')
```

**KV**

```
CupertinoModalButton:
    font_size: '20sp'
```

**text** = ' '

Text of *CupertinoModalButton*

**Python**

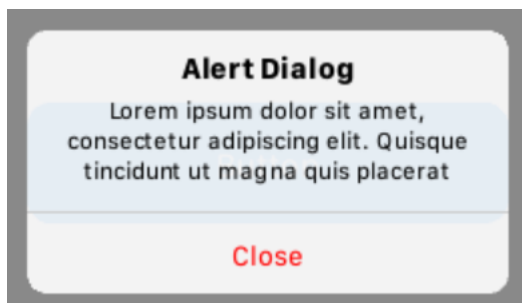
```
CupertinoModalButton(text='Hello World')
```

**KV**

```
CupertinoModalButton:
    text: 'Hello World'
```

**text\_color** = [0.05, 0.5, 1, 1]

Color of the text of *CupertinoModalButton*

**Python**

```
CupertinoModalButton(text_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoModalButton:
    text_color: 1, 0, 0, 1
```

**transition\_duration = 0.075**

Duration of the transition of the color of `CupertinoButton` when its state changes

**Python**

```
CupertinoModalButton(transition_duration=0.5)
```

**KV**

```
CupertinoModalButton:
    transition_duration: 0.5
```

## 1.8 Page

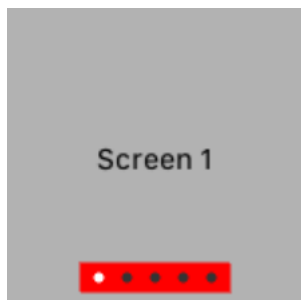
Pages allow for a separation of different features

**class** `kivycupertino.uix.page.CupertinoPageControls`(\*args: Any, \*\*kwargs: Any)

iOS style Page Controls. Will automatically update the number of pages and current page when added to an instance of `CupertinoScreenManager`

**background\_color = [0, 0, 0, 0]**

Background color of `CupertinoPageControls`

**Python**

```
CupertinoPageControls(background_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoPageControls:
    background_color: 1, 0, 0, 1
```

**color\_selected** = [1, 1, 1, 1]

A ColorProperty defining the color of a dot on *CupertinoPageControls* when not selected

#### Python

```
CupertinoPageControls(color_selected=(1, 0, 0, 1))
```

#### KV

```
CupertinoPageControls:
    color_selected: 1, 0, 0, 1
```

**color\_unselected** = [0.2, 0.2, 0.2, 1]

Color of a dot on *CupertinoPageControls* when not selected

#### Python

```
CupertinoPageControls(color_unselected=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoPageControls:
    color_unselected: 0.5, 0, 0, 1
```

**on\_parent**(*instance, parent*)

Callback when *CupertinoPageControls* is added to *CupertinoScreenManager*

#### Parameters

- **instance** – Instance of class *CupertinoPageControls*
- **parent** – Instance of *CupertinoScreenManager*

**tap** = False

If tapping *CupertinoPageControls* will switch to a screen

#### Python

```
CupertinoPageControls(tap=True)
```

#### KV

```
CupertinoPageControls:
    tap: True
```

**class** kivycupertino.uix.page.CupertinoScreenManager(\*args: Any, \*\*kwargs: Any)

A Screen Manager widget that also accepts an instance of *CupertinoPageControls*

**add\_widget**(*widget, \*args, \*\*kwargs*)

Callback when a Screen or a *CupertinoPageControls* is added to *CupertinoScreenManager*

#### Parameters

- **widget** – Widget to be added to *CupertinoScreenManager*

## 1.9 Scrollview

ScrollViews help show lots of information within a set screen size

**class** kivycupertino.uix.scrollview.CupertinoScrollView(\*args: Any, \*\*kwargs: Any)  
iOS style ScrollView

**bar\_color** = [0.65, 0.65, 0.65, 1]

Color of the bar of *CupertinoScrollView* when scrolling

### Python

```
CupertinoScrollview(bar_color=(1, 0, 0, 1))
```

### KV

```
CupertinoScrollview:  
    bar_color: 1, 0, 0, 1
```

**bar\_inactive\_color** = [0, 0, 0, 0]

Color of the bar of *CupertinoScrollView* when not scrolling

### Python

```
CupertinoScrollview(bar_inactive_color=(0.5, 0, 0, 1))
```

### KV

```
CupertinoScrollview:  
    bar_inactive_color: 0.5, 0, 0, 1
```

## 1.10 Slider

Sliders allow users to choose values

**class** kivycupertino.uix.slider.CupertinoSlider(\*args: Any, \*\*kwargs: Any)  
iOS style slider

**color\_selected** = [0, 0.5, 1, 1]

Color of the bar of occurred progress of *CupertinoSlider*

### Python

```
CupertinoSlider(color_selected=(1, 0, 0, 1))
```

### KV

```
CupertinoSlider:
    color_selected: 1, 0, 0, 1
```

**color\_unselected** = [0.7, 0.7, 0.7, 1]

Color of bar of not yet occurred progress of *CupertinoSlider*

#### Python

```
CupertinoSlider(color_unselected=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoSlider:
    color_unselected: 0.5, 0, 0, 1
```

**max** = 100

Maximum value of *CupertinoSlider*

#### Python

```
CupertinoSlider(max=50)
```

#### KV

```
CupertinoSlider:
    max: 50
```

**min** = 0

Minimum value of *CupertinoSlider*

#### Python

```
CupertinoSlider(min=-50)
```

#### KV

```
CupertinoSlider:
    min: -50
```

**on\_touch\_move**(*touch*)

Callback when *CupertinoSlider* is dragged

#### Parameters

**touch** – Touch on *CupertinoSlider*

**tap** = True

If tapping *CupertinoSlider* can change its *value*

#### Python

```
CupertinoSlider(tap=True)
```

#### KV

```
CupertinoSlider:  
    tap: True
```

**thumb\_color** = [1, 1, 1, 1]

Color of the thumb of *CupertinoSlider*



**Python**

```
CupertinoSlider(thumb_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSlider:  
    thumb_color: 1, 0, 0, 1
```

**value** = 0

Value of *CupertinoSlider*



**Python**

```
CupertinoSlider(value=50)
```

**KV**

```
CupertinoSlider:  
    value: 50
```

## 1.11 Swipe

Swiping allows users to interact with widgets by using hidden actions

**class** kivycupertino.uix.swipe.**CupertinoSwipe**(\*args: Any, \*\*kwargs: Any)

A widget to add swiping functionality to existing Kivy Cupertino widgets

**background\_color** = [1, 1, 1, 1]

Background color of *CupertinoSwipe*



**Python**



```
CupertinoSwipe(background_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSwipe:
    background_color: 1, 0, 0, 1
```

**collapse()**

Callback to reset *CupertinoSwipe* so no actions are visible

**complete\_swipe\_duration = 0.5**

How long after *CupertinoSwipe* is released until swipe is moved to final position (completely expanded or completely collapsed)

**Python**

```
CupertinoSwipe(complete_swipe_duration=1)
```

**KV**

```
CupertinoSwipe:
    complete_swipe_duration: 1
```

**expand(side)**

Callback to completely open a specified side *CupertinoSwipe*

**Parameters**

**side** – The side of *CupertinoSwipe* to expand ('left' or 'right')

**is\_collapsed()**

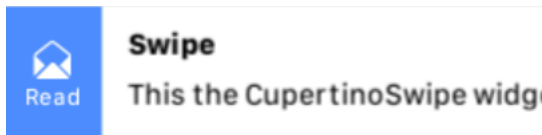
Check if *CupertinoSwipe* is collapsed

**Returns**

If *CupertinoSwipe* is collapsed

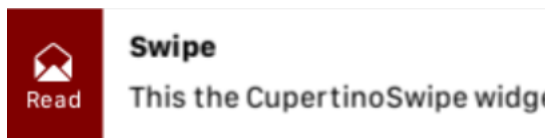
**class** kivycupertino.uix.swipe.**CupertinoSwipeAction**(\*args: Any, \*\*kwargs: Any)

An iOS style action to add to *CupertinoSwipe*



**color\_disabled = [0, 0.3, 0.4, 1]**

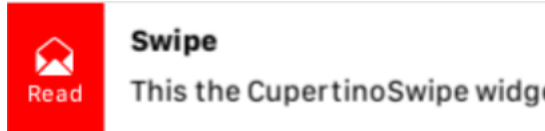
Color of *CupertinoSwipeAction* when disabled

**Python**

```
CupertinoSwipeAction(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoSwipeAction:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_normal = [1, 0, 0, 1]**Color of *CupertinoSwipeAction* when not pressed or disabled**Python**

```
CupertinoSwipeAction(color_normal=(1, 0, 0, 1))
```

**KV**

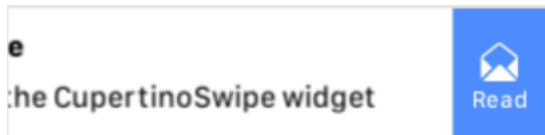
```
CupertinoSwipeAction:
    color_normal: 1, 0, 0, 1
```

**disabled = False**If *CupertinoSwipeAction* is disabled**Python**

```
CupertinoSwipeAction(disabled=True)
```

**KV**

```
CupertinoSwipeAction:
    disabled: True
```

**side = 'left'**Side of *CupertinoSwipe* that *CupertinoSwipeAction* should be shown on**Python**

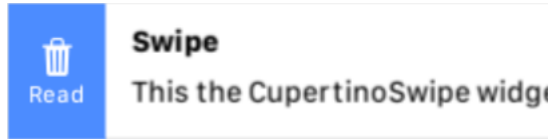
```
CupertinoSwipeAction(side='right')
```

**KV**

```
CupertinoSwipeAction:
    side: 'right'
```

`symbol = ' '`

Text of *CupertinoSwipeAction*



**Python**

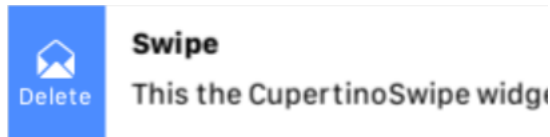
```
CupertinoSwipeAction(symbol='trash_fill')
```

**KV**

```
CupertinoSwipeAction:
    symbol: 'trash_fill'
```

`text = ' '`

Text of *CupertinoSwipeAction*



**Python**

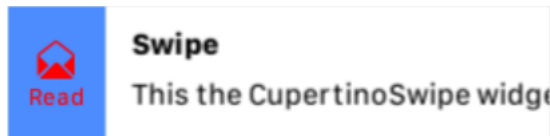
```
CupertinoSwipeAction(text='Delete')
```

**KV**

```
CupertinoSwipeAction:
    text: 'Delete'
```

`text_color = [1, 1, 1, 1]`

Color of text of *CupertinoSwipeAction*



**Python**

```
CupertinoSwipeAction(text_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSwipeAction:
    text_color: 1, 0, 0, 1
```

## 1.12 Switch

Switches allow users to toggle options

**class** kivycupertino.uix.switch.**CupertinoSwitch**(\*args: Any, \*\*kwargs: Any)

iOS style Switch. To comply with iOS standard, keep the width to height ratio of [CupertinoSwitch](#) at 1:0.6

**color\_toggled** = [0.3, 0.85, 0.4, 1]

Background color of [CupertinoSwitch](#) when on

### Python

```
CupertinoSwitch(color_toggled=(1, 0, 0, 1))
```

### KV

```
CupertinoSwitch:
    color_toggled: 1, 0, 0, 1
```

**color\_untoggled** = [0.85, 0.85, 0.85, 1]

Background color of [CupertinoSwitch](#) when off

### Python

```
CupertinoSwitch(color_untoggled=(0.5, 0, 0, 1))
```

### KV

```
CupertinoSwitch:
    color_untoggled: 0.5, 0, 0, 1
```

**on\_toggled**(instance, state)

Callback when state of [CupertinoSwitch](#) is changed

### Parameters

- **instance** – Instance of [CupertinoSwitch](#)
- **state** – If [CupertinoSwitch](#) is toggled

**on\_touch\_move**(touch)

Callback when [CupertinoSwitch](#) is dragged

### Parameters

**touch** – MotionEvent detected on [CupertinoSwitch](#)

**switch\_duration** = 0.1

Duration of color change and thumb movement when state of [CupertinoSwitch](#) is changed

### Python

```
CupertinoSwitch(switch_duration=0.5)
```

**KV**

```
CupertinoSwitch:
    switch_duration: 0.5
```

**thumb\_color = [1, 1, 1, 1]**

Color of thumb of *CupertinoSwitch*

**Python**

```
CupertinoSwitch(thumb_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSwitch:
    thumb_color: 1, 0, 0, 1
```

**thumb\_padding**

Amount of padding around thumb of *CupertinoSwitch* in interval [0, 1] as a percentage of the height of *CupertinoSwitch*

**Python**

```
CupertinoSwitch(thumb_padding=0.1)
```

**KV**

```
CupertinoSwitch:
    thumb_padding: 0.1
```

**toggled = False**

If *CupertinoSwitch* is on

**Python**

```
CupertinoSwitch(toggled=True)
```

**KV**

```
CupertinoSwitch:
    toggled: True
```

## 1.13 Symbol

Symbols help portray an action with a simple symbol. To view all symbols in Kivy Cupertino, visit [Framework7](#) or run the Symbols program

**class** `kivycupertino.uix.symbol.CupertinoSymbol(*args: Any, **kwargs: Any)`

Display an iOS style symbol.



`color = [0, 0, 0, 1]`

Color of *CupertinoSymbol*



**Python**

```
CupertinoSymbol(color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSymbol:
    color: 1, 0, 0, 1
```

**on\_symbol**(*instance*, *symbol*)

Callback when symbol of *CupertinoSymbol* is changed

**Parameters**

- **instance** – Instance of *CupertinoSymbol*
- **symbol** – Symbol to be displayed

**symbol** = ' '

Symbol to be displayed by *CupertinoSymbol*.



**Python**

```
CupertinoSymbol(symbol='alarm_fill')
```

**KV**

```
CupertinoSymbol:
    symbol: 'alarm_fill'
```

## 1.14 Table

Tables help organize data and information for users to view and interact with

**class** `kivycupertino.uix.table.CupertinoClickableTableCell(*args: Any, **kwargs: Any)`

iOS style clickable Cell for Table View. `CupertinoClickableTableCell` is a `RelativeLayout` and can accept any number of widgets

**color\_disabled** = `[0.8, 0.8, 0.8, 1]`

Background color of `CupertinoClickableTableCell` when disabled



**Python**

```
CupertinoTableCell(disabled=True, color_disabled=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoClickableTableCell:
    disabled: True
    color_disabled: 0.5, 0, 0, 1
```

**color\_down** = `[0.9, 0.9, 0.9, 0.9]`

Background color of `CupertinoClickableTableCell` when pressed

**Python**

```
CupertinoClickableTableCell(color_down=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoClickableTableCell:
    color_down: 0.5, 0, 0, 1
```

**color\_normal** = `[1, 1, 1, 1]`

Background color of `CupertinoClickableTableCell` when not pressed or disabled



**Python**

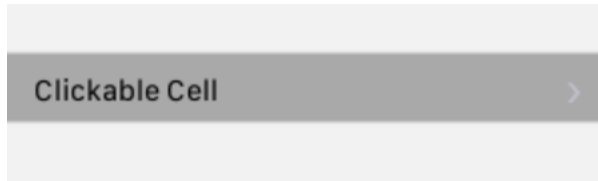
```
CupertinoClickableTableCell(color_normal=(1, 0, 0, 1))
```

**KV**

```
CupertinoClickableTableCell:
    color_normal: 1, 0, 0, 1
```

**disabled = False**

If *CupertinoClickableTableCell* is disabled



**Python**

```
CupertinoClickableTableCell(disabled=True)
```

**KV**

```
CupertinoClickableTableCell:
    disabled: True
```

**transition\_duration = 0.075**

Duration of the transition of the color of CupertinoButton when its state changes

**Python**

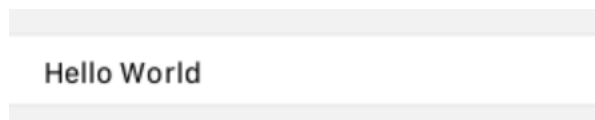
```
CupertinoClickableTableCell(transition_duration=0.5)
```

**KV**

```
CupertinoClickableTableCell:
    transition_duration: 0.5
```

**class** kivycupertino.uix.table.CupertinoTableCell(\*args: Any, \*\*kwargs: Any)

iOS style Cell for Table View. *CupertinoTableCell* is a *RelativeLayout* and can accept any number of widgets



**color = [1, 1, 1, 1]**

Background color of *CupertinoTableCell*



**Python**

```
CupertinoTableCell(color=(1, 0, 0, 1))
```

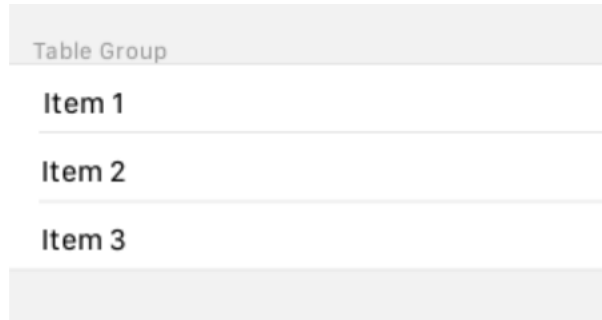
**KV**



```
CupertinoNavigationBar:
    CupertinoTableCell: 1, 0, 0, 1
```

**class** kivycupertino.uix.table.**CupertinoTableGroup**(\*args: Any, \*\*kwargs: Any)

iOS style table group



**add\_widget**(widget, index=0, canvas=None)

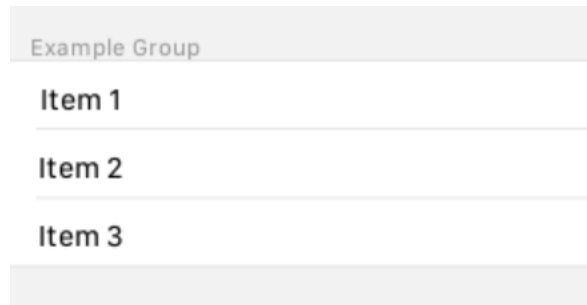
Add an instance of [CupertinoTableCell](#) to [CupertinoTableGroup](#)

#### Parameters

- **widget** – Instance of [CupertinoTableCell](#) to be added to [CupertinoTableGroup](#)
- **index** – Index at which [CupertinoTableCell](#) will be inserted into children of [CupertinoTableGroup](#)
- **canvas** – Canvas at which [CupertinoTableCell](#) will be inserted into [CupertinoTableGroup](#)

**text** = ' '

Background color of [CupertinoClickableTableCell](#) when disabled



#### Python

```
CupertinoTableGroup(text='Example Group')
```

#### KV

```
CupertinoTableGroup:
    text: 'Example Group'
```

**text\_color** = [0.6, 0.6, 0.6, 1]

Background color of [CupertinoClickableTableCell](#) when disabled

**Table Group**

Item 1

Item 2

Item 3

**Python**

```
CupertinoTableGroup(text_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoTableGroup:
    text_color: 1, 0, 0, 1
```

## 1.15 Textinput

Text fields allow users to enter input

**class** kivycupertino.uix.textinput.**CupertinoSearchBar**(\*args: Any, \*\*kwargs: Any)  
iOS style search bar

**background\_color** = [0.85, 0.85, 0.85, 0.7]

Background color of *CupertinoSearchBar*

**Python**

```
CupertinoSearchBar(background_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSearchBar:
    background_color: 1, 0, 0, 1
```

**color\_down** = [0.4, 0.4, 0.4, 1]

Color of the clear button of *CupertinoSearchBar* when pressed

**Python**

```
CupertinoSearchBar(color_down=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoSearchBar:
    color_down: 0.5, 0, 0, 1
```

**cursor\_color = [0.25, 0.5, 0.95, 1]**

Color of cursor of *CupertinoTextView*

#### Python

```
CupertinoSearchBar(background_color=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoSearchBar:
    background_color: 0.5, 0, 0, 1
```

**foreground\_color = [0, 0, 0, 1]**

Text color of *CupertinoSearchBar*

#### Python

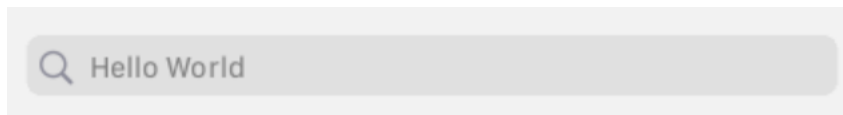
```
CupertinoSearchBar(foreground_color=(1, 0, 0, 1))
```

#### KV

```
CupertinoSearchBar:
    foreground_color: 1, 0, 0, 1
```

**hint\_text = 'Search'**

Text of hint of *CupertinoTextField*



#### Python

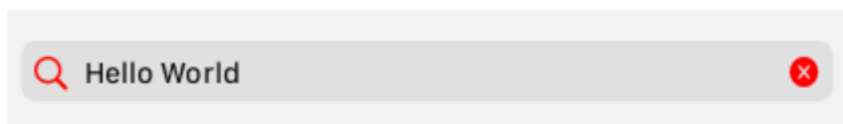
```
CupertinoSearchBar(hint_text='Hello World')
```

#### KV

```
CupertinoSearchBar:
    hint_text: 'Hello World'
```

**symbol\_color = [0.55, 0.55, 0.6, 1]**

Color of the symbols of *CupertinoSearchBar*



#### Python

```
CupertinoSearchBar(symbol_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoSearchBar:
    symbol_color: 1, 0, 0, 1
```

**text** = ''Text of *CupertinoTextField***Python**

```
CupertinoSearchBar(text='Hello World')
```

**KV**

```
CupertinoSearchBar:
    text: 'Hello World'
```

**class** kivycupertino.uix.textinput.**CupertinoTextField**(\*args: Any, \*\*kwargs: Any)

iOS style Text Field to be used for single-line input

**background\_color** = [0, 0, 0, 0]Background color of *CupertinoTextField*Text Field**Python**

```
CupertinoTextField(background_color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoTextField:
    background_color: 0.5, 0, 0, 1
```

**cursor\_color** = [0.25, 0.5, 0.95, 1]Color of cursor of *CupertinoTextField***Python**

```
CupertinoTextField(cursor_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoTextField:
    cursor_color: 1, 0, 0, 1
```

**foreground\_color = [0, 0, 0, 1]**

Text color of *CupertinoTextField*

#### Python

```
CupertinoTextField(foreground_color=(1, 0, 0, 1))
```

#### KV

```
CupertinoTextField:
    foreground_color: 1, 0, 0, 1
```

**hint\_text = ''**

Text of hint of *CupertinoTextField*

Hello World

#### Python

```
CupertinoTextField(hint_text='Hello World')
```

#### KV

```
CupertinoTextField:
    hint_text: 'Hello World'
```

**class** kivycupertino.uix.textinput.**CupertinoTextView**(\*args: Any, \*\*kwargs: Any)

iOS style Text View for multiline input

**background\_color = [0, 0, 0, 0]**

Background color of *CupertinoTextView*

Hello  
World

#### Python

```
CupertinoTextField(background_color=(0.5, 0, 0, 1))
```

#### KV

```
CupertinoTextField:
    background_color: 0.5, 0, 0, 1
```

**cursor\_color = [0.25, 0.5, 0.95, 1]**  
Color of cursor of *CupertinoTextView*

**Python**

```
CupertinoTextView(background_color=(0.5, 0, 0, 1))
```

**KV**

```
CupertinoTextView:  
    background_color: 0.5, 0, 0, 1
```

**foreground\_color = [0, 0, 0, 1]**  
Text color of *CupertinoTextView*

**Python**

```
CupertinoTextField(foreground_color=(1, 0, 0, 1))
```

**KV**

```
CupertinoTextField:  
    foreground_color: 1, 0, 0, 1
```

## CONTRIBUTING

To help improve this project, [fork our repository](#) and install Kivy Cupertino with the dev extra:

```
$ pip install kivycupertino[dev]
```

Visit the [projects page](#) to view pending projects you may be able to work on





## LICENSE

Kivy Cupertino is licensed under the MIT license

Copyright (c) 2021 cmdvmd  
Portions Copyright (c) 2021 arnavs-0  
Copyright (c) 2010-2020 Kivy Team and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### k

- `kivycupertino.uix.bar`, 3
- `kivycupertino.uix.behavior`, 7
- `kivycupertino.uix.button`, 9
- `kivycupertino.uix.control`, 15
- `kivycupertino.uix.indicator`, 19
- `kivycupertino.uix.label`, 21
- `kivycupertino.uix.modal`, 23
- `kivycupertino.uix.page`, 32
- `kivycupertino.uix.scrollview`, 34
- `kivycupertino.uix.slider`, 34
- `kivycupertino.uix.swipe`, 36
- `kivycupertino.uix.switch`, 40
- `kivycupertino.uix.symbol`, 42
- `kivycupertino.uix.table`, 43
- `kivycupertino.uix.textinput`, 46



## INDEX

### A

**action\_height** (kivycupertino.uix.modal.CupertinoActionSheet attribute), 23

**action\_height** (kivycupertino.uix.modal.CupertinoDialog attribute), 26

**add\_disabled** (kivycupertino.uix.control.CupertinoStepper attribute), 17

**add\_widget()** (kivycupertino.uix.bar.CupertinoTabBar method), 5

**add\_widget()** (kivycupertino.uix.control.CupertinoSegmentedControls method), 16

**add\_widget()** (kivycupertino.uix.page.CupertinoScreenManager method), 33

**add\_widget()** (kivycupertino.uix.table.CupertinoTableGroup method), 45

### B

**background\_color** (kivycupertino.uix.bar.CupertinoTabBar attribute), 5

**background\_color** (kivycupertino.uix.control.CupertinoSegmentedControls attribute), 16

**background\_color** (kivycupertino.uix.page.CupertinoPageControls attribute), 32

**background\_color** (kivycupertino.uix.swipe.CupertinoSwipe attribute), 36

**background\_color** (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 46

**background\_color** (kivycupertino.uix.textinput.CupertinoTextField attribute), 48

**background\_color** (kivycuper-

тино.uix.textinput.CupertinoTextView attribute), 49

**bar\_color** (kivycupertino.uix.scrollview.CupertinoScrollView attribute), 34

**bar\_inactive\_color** (kivycupertino.uix.scrollview.CupertinoScrollView attribute), 34

**bold** (kivycupertino.uix.label.CupertinoLabel attribute), 21

### C

**cancel** (kivycupertino.uix.modal.CupertinoModalButton attribute), 28

**collapse()** (kivycupertino.uix.swipe.CupertinoSwipe method), 37

**color** (kivycupertino.uix.bar.CupertinoNavigationBar attribute), 3

**color** (kivycupertino.uix.bar.CupertinoToolbar attribute), 6

**color** (kivycupertino.uix.behavior.CupertinoButtonBehavior attribute), 7

**color** (kivycupertino.uix.control.CupertinoSegment attribute), 15

**color** (kivycupertino.uix.indicator.CupertinoActivityIndicator attribute), 19

**color** (kivycupertino.uix.label.CupertinoLabel attribute), 21

**color** (kivycupertino.uix.modal.CupertinoDialog attribute), 27

**color** (kivycupertino.uix.symbol.CupertinoSymbol attribute), 42

**color** (kivycupertino.uix.table.CupertinoTableCell attribute), 44

**color\_disabled** (kivycupertino.uix.behavior.CupertinoButtonBehavior attribute), 7

**color\_disabled** (kivycupertino.uix.button.CupertinoButton attribute), 9

**color\_disabled** (kivycupertino.uix.button.CupertinoSymbolButton at-

<i>tribute</i> ), 12		<i>color_normal</i> (kivycupertino.uix.modal.CupertinoActionSheet <i>attribute</i> ), 23
<i>color_disabled</i> (kivycupertino.uix.button.CupertinoSystemButton <i>attribute</i> ), 13		<i>color_normal</i> (kivycupertino.uix.modal.CupertinoModalButton <i>attribute</i> ), 29
<i>color_disabled</i> (kivycupertino.uix.control.CupertinoStepper <i>attribute</i> ), 17		<i>color_normal</i> (kivycupertino.uix.swipe.CupertinoSwipeAction <i>attribute</i> ), 38
<i>color_disabled</i> (kivycupertino.uix.modal.CupertinoModalButton <i>attribute</i> ), 29		<i>color_normal</i> (kivycupertino.uix.table.CupertinoClickableTableCell <i>attribute</i> ), 43
<i>color_disabled</i> (kivycupertino.uix.swipe.CupertinoSwipeAction <i>attribute</i> ), 37		<i>color_selected</i> (kivycupertino.uix.bar.CupertinoTab <i>attribute</i> ), 4
<i>color_disabled</i> (kivycupertino.uix.table.CupertinoClickableTableCell <i>attribute</i> ), 43		<i>color_selected</i> (kivycupertino.uix.control.CupertinoSegmentedControls <i>attribute</i> ), 16
<i>color_down</i> (kivycupertino.uix.behavior.CupertinoButtonBehavior <i>attribute</i> ), 7		<i>color_selected</i> (kivycupertino.uix.indicator.CupertinoProgressbar <i>attribute</i> ), 20
<i>color_down</i> (kivycupertino.uix.button.CupertinoButton <i>attribute</i> ), 10		<i>color_selected</i> (kivycupertino.uix.page.CupertinoPageControls <i>attribute</i> ), 32
<i>color_down</i> (kivycupertino.uix.button.CupertinoSymbolButton <i>attribute</i> ), 12		<i>color_selected</i> (kivycupertino.uix.slider.CupertinoSlider <i>attribute</i> ), 34
<i>color_down</i> (kivycupertino.uix.button.CupertinoSystemButton <i>attribute</i> ), 14		<i>color_toggled</i> (kivycupertino.uix.switch.CupertinoSwitch <i>attribute</i> ), 40
<i>color_down</i> (kivycupertino.uix.control.CupertinoStepper <i>attribute</i> ), 18		<i>color_unselected</i> (kivycupertino.uix.bar.CupertinoTab <i>attribute</i> ), 4
<i>color_down</i> (kivycupertino.uix.modal.CupertinoModalButton <i>attribute</i> ), 29		<i>color_unselected</i> (kivycupertino.uix.indicator.CupertinoProgressbar <i>attribute</i> ), 20
<i>color_down</i> (kivycupertino.uix.table.CupertinoClickableTableCell <i>attribute</i> ), 43		<i>color_unselected</i> (kivycupertino.uix.page.CupertinoPageControls <i>attribute</i> ), 33
<i>color_down</i> (kivycupertino.uix.textinput.CupertinoSearchBar <i>attribute</i> ), 46		<i>color_unselected</i> (kivycupertino.uix.slider.CupertinoSlider <i>attribute</i> ), 35
<i>color_normal</i> (kivycupertino.uix.behavior.CupertinoButtonBehavior <i>attribute</i> ), 7		<i>color_untoggled</i> (kivycupertino.uix.switch.CupertinoSwitch <i>attribute</i> ), 40
<i>color_normal</i> (kivycupertino.uix.button.CupertinoButton <i>attribute</i> ), 10		<i>complete_swipe_duration</i> (kivycupertino.uix.swipe.CupertinoSwipe <i>attribute</i> ), 37
<i>color_normal</i> (kivycupertino.uix.button.CupertinoSymbolButton <i>attribute</i> ), 12		CupertinoActionSheet (class in kivycupertino.uix.modal), 23
<i>color_normal</i> (kivycupertino.uix.button.CupertinoSystemButton <i>attribute</i> ), 14		CupertinoActivityIndicator (class in kivycupertino.uix.indicator), 19
<i>color_normal</i> (kivycupertino.uix.control.CupertinoStepper <i>attribute</i> ), 18		CupertinoButton (class in kivycupertino.uix.button), 9
		CupertinoButtonBehavior (class in kivycupertino.uix.behavior), 7
		CupertinoClickableTableCell (class in kivycuper-



- tino.uix.table*), 43
- CupertinoDialog (class in *kivycupertino.uix.modal*), 26
- CupertinoLabel (class in *kivycupertino.uix.label*), 21
- CupertinoModalButton (class in *kivycupertino.uix.modal*), 28
- CupertinoNavigationBar (class in *kivycupertino.uix.bar*), 3
- CupertinoPageControls (class in *kivycupertino.uix.page*), 32
- CupertinoProgressbar (class in *kivycupertino.uix.indicator*), 20
- CupertinoScreenManager (class in *kivycupertino.uix.page*), 33
- CupertinoScrollView (class in *kivycupertino.uix.scrollview*), 34
- CupertinoSearchBar (class in *kivycupertino.uix.textinput*), 46
- CupertinoSegment (class in *kivycupertino.uix.control*), 15
- CupertinoSegmentedControls (class in *kivycupertino.uix.control*), 16
- CupertinoSlider (class in *kivycupertino.uix.slider*), 34
- CupertinoStepper (class in *kivycupertino.uix.control*), 17
- CupertinoSwipe (class in *kivycupertino.uix.swipe*), 36
- CupertinoSwipeAction (class in *kivycupertino.uix.swipe*), 37
- CupertinoSwitch (class in *kivycupertino.uix.switch*), 40
- CupertinoSymbol (class in *kivycupertino.uix.symbol*), 42
- CupertinoSymbolButton (class in *kivycupertino.uix.button*), 12
- CupertinoSystemButton (class in *kivycupertino.uix.button*), 13
- CupertinoTab (class in *kivycupertino.uix.bar*), 4
- CupertinoTabBar (class in *kivycupertino.uix.bar*), 5
- CupertinoTableCell (class in *kivycupertino.uix.table*), 44
- CupertinoTableGroup (class in *kivycupertino.uix.table*), 45
- CupertinoTextField (class in *kivycupertino.uix.textinput*), 48
- CupertinoTextView (class in *kivycupertino.uix.textinput*), 49
- CupertinoToolbar (class in *kivycupertino.uix.bar*), 6
- cursor\_color (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 47
- cursor\_color (kivycupertino.uix.textinput.CupertinoTextField attribute), 48
- cursor\_color (kivycupertino.uix.textinput.CupertinoTextView attribute), 49
- curve (kivycupertino.uix.modal.CupertinoActionSheet attribute), 24
- curve (kivycupertino.uix.modal.CupertinoDialog attribute), 27
- ## D
- disabled (kivycupertino.uix.behavior.CupertinoButtonBehavior attribute), 8
- disabled (kivycupertino.uix.button.CupertinoButton attribute), 10
- disabled (kivycupertino.uix.button.CupertinoSymbolButton attribute), 13
- disabled (kivycupertino.uix.button.CupertinoSystemButton attribute), 14
- disabled (kivycupertino.uix.modal.CupertinoModalButton attribute), 30
- disabled (kivycupertino.uix.swipe.CupertinoSwipeAction attribute), 38
- disabled (kivycupertino.uix.table.CupertinoClickableTableCell attribute), 44
- duration (kivycupertino.uix.indicator.CupertinoActivityIndicator attribute), 19
- ## E
- expand() (kivycupertino.uix.swipe.CupertinoSwipe method), 37
- ## F
- font\_name (kivycupertino.uix.label.CupertinoLabel attribute), 22
- font\_size (kivycupertino.uix.button.CupertinoButton attribute), 11
- font\_size (kivycupertino.uix.button.CupertinoSystemButton attribute), 14
- font\_size (kivycupertino.uix.label.CupertinoLabel attribute), 22
- font\_size (kivycupertino.uix.modal.CupertinoModalButton attribute), 30
- foreground\_color (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 47
- foreground\_color (kivycupertino.uix.textinput.CupertinoTextField attribute), 48
- foreground\_color (kivycupertino.uix.textinput.CupertinoTextView attribute), 50
- ## G
- get\_selected\_segment() (kivycupertino.uix.control.CupertinoSegmentedControls method), 16

`get_selected_tab()` (kivycupertino.uix.bar.CupertinoTabBar method), 6

## H

`hint_text` (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 47

`hint_text` (kivycupertino.uix.textinput.CupertinoTextField attribute), 49

## I

`is_collapsed()` (kivycupertino.uix.swipe.CupertinoSwipe method), 37

`italic` (kivycupertino.uix.label.CupertinoLabel attribute), 22

## K

`kivycupertino.uix.bar`  
module, 3

`kivycupertino.uix.behavior`  
module, 7

`kivycupertino.uix.button`  
module, 9

`kivycupertino.uix.control`  
module, 15

`kivycupertino.uix.indicator`  
module, 19

`kivycupertino.uix.label`  
module, 21

`kivycupertino.uix.modal`  
module, 23

`kivycupertino.uix.page`  
module, 32

`kivycupertino.uix.scrollview`  
module, 34

`kivycupertino.uix.slider`  
module, 34

`kivycupertino.uix.swipe`  
module, 36

`kivycupertino.uix.switch`  
module, 40

`kivycupertino.uix.symbol`  
module, 42

`kivycupertino.uix.table`  
module, 43

`kivycupertino.uix.textinput`  
module, 46

## L

`long_press_duration` (kivycupertino.uix.behavior.LongPressBehavior attribute), 8

`LongPressBehavior` (class in kivycupertino.uix.behavior), 8

## M

`max` (kivycupertino.uix.slider.CupertinoSlider attribute), 35

`message` (kivycupertino.uix.modal.CupertinoActionSheet attribute), 24

`min` (kivycupertino.uix.slider.CupertinoSlider attribute), 35

`minus_disabled` (kivycupertino.uix.control.CupertinoStepper attribute), 18

`module`

`kivycupertino.uix.bar`, 3

`kivycupertino.uix.behavior`, 7

`kivycupertino.uix.button`, 9

`kivycupertino.uix.control`, 15

`kivycupertino.uix.indicator`, 19

`kivycupertino.uix.label`, 21

`kivycupertino.uix.modal`, 23

`kivycupertino.uix.page`, 32

`kivycupertino.uix.scrollview`, 34

`kivycupertino.uix.slider`, 34

`kivycupertino.uix.swipe`, 36

`kivycupertino.uix.switch`, 40

`kivycupertino.uix.symbol`, 42

`kivycupertino.uix.table`, 43

`kivycupertino.uix.textinput`, 46

## O

`on_chosen()` (kivycupertino.uix.behavior.SelectableBehavior method), 9

`on_long_press()` (kivycupertino.uix.behavior.LongPressBehavior method), 8

`on_minus()` (kivycupertino.uix.control.CupertinoStepper method), 18

`on_parent()` (kivycupertino.uix.page.CupertinoPageControls method), 33

`on_plus()` (kivycupertino.uix.control.CupertinoStepper method), 18

`on_selected()` (kivycupertino.uix.behavior.SelectableBehavior method), 9

`on_symbol()` (kivycupertino.uix.symbol.CupertinoSymbol method), 42

`on_toggled()` (kivycupertino.uix.switch.CupertinoSwitch method), 40

**on\_touch\_move()** (kivycupertino.uix.control.CupertinoSegmentedControls method), 17  
**on\_touch\_move()** (kivycupertino.uix.slider.CupertinoSlider method), 35  
**on\_touch\_move()** (kivycupertino.uix.switch.CupertinoSwitch method), 40  
**on\_value()** (kivycupertino.uix.indicator.CupertinoProgressbar method), 20

**P**

**playing** (kivycupertino.uix.indicator.CupertinoActivityIndicator attribute), 19

**R**

**refresh()** (kivycupertino.uix.behavior.SelectableBehavior method), 9

**S**

**SelectableBehavior** (class in kivycupertino.uix.behavior), 9

**selected** (kivycupertino.uix.behavior.SelectableBehavior attribute), 9

**side** (kivycupertino.uix.swipe.CupertinoSwipeAction attribute), 38

**spacing** (kivycupertino.uix.modal.CupertinoActionSheet attribute), 25

**spacing** (kivycupertino.uix.modal.CupertinoDialog attribute), 28

**spokes** (kivycupertino.uix.indicator.CupertinoActivityIndicator attribute), 20

**switch\_duration** (kivycupertino.uix.switch.CupertinoSwitch attribute), 40

**symbol** (kivycupertino.uix.bar.CupertinoTab attribute), 5

**symbol** (kivycupertino.uix.button.CupertinoSymbolButton attribute), 13

**symbol** (kivycupertino.uix.swipe.CupertinoSwipeAction attribute), 39

**symbol** (kivycupertino.uix.symbol.CupertinoSymbol attribute), 42

**symbol\_color** (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 47

**T**

**tap** (kivycupertino.uix.page.CupertinoPageControls attribute), 33

**tap** (kivycupertino.uix.slider.CupertinoSlider attribute), 35

**text** (kivycupertino.uix.bar.CupertinoTab attribute), 5

**text** (kivycupertino.uix.button.CupertinoButton attribute), 11

**text** (kivycupertino.uix.button.CupertinoSystemButton attribute), 15

**text** (kivycupertino.uix.label.CupertinoLabel attribute), 22

**text** (kivycupertino.uix.modal.CupertinoModalButton attribute), 31

**text** (kivycupertino.uix.swipe.CupertinoSwipeAction attribute), 39

**text** (kivycupertino.uix.table.CupertinoTableGroup attribute), 45

**text** (kivycupertino.uix.textinput.CupertinoSearchBar attribute), 48

**text\_color** (kivycupertino.uix.button.CupertinoButton attribute), 11

**text\_color** (kivycupertino.uix.control.CupertinoStepper attribute), 18

**text\_color** (kivycupertino.uix.modal.CupertinoActionSheet attribute), 25

**text\_color** (kivycupertino.uix.modal.CupertinoModalButton attribute), 31

**text\_color** (kivycupertino.uix.swipe.CupertinoSwipeAction attribute), 39

**text\_color** (kivycupertino.uix.table.CupertinoTableGroup attribute), 45

**thumb\_color** (kivycupertino.uix.slider.CupertinoSlider attribute), 36

**thumb\_color** (kivycupertino.uix.switch.CupertinoSwitch attribute), 41

**thumb\_padding** (kivycupertino.uix.switch.CupertinoSwitch attribute), 41

**title** (kivycupertino.uix.modal.CupertinoActionSheet attribute), 26

**toggled** (kivycupertino.uix.switch.CupertinoSwitch attribute), 41

**transition\_duration** (kivycupertino.uix.behavior.CupertinoButtonBehavior attribute), 8

**transition\_duration** (kivycupertino.uix.button.CupertinoButton attribute), 12

**transition\_duration** (kivycupertino.uix.button.CupertinoSymbolButton attribute), 13

**transition\_duration** (kivycupertino.uix.button.CupertinoSystemButton attribute), 15

*tribute*), 15  
transition\_duration (kivycupertino.uix.control.CupertinoSegmentedControls  
*attribute*), 17  
transition\_duration (kivycupertino.uix.modal.CupertinoModalButton *at-*  
*tribute*), 32  
transition\_duration (kivycupertino.uix.table.CupertinoClickableTableCell  
*attribute*), 44

## V

value (kivycupertino.uix.indicator.CupertinoProgressbar  
*attribute*), 21  
value (kivycupertino.uix.slider.CupertinoSlider *at-*  
*tribute*), 36