Cocktailmixer

Erzeugt von Doxygen 1.8.15

Inhaltsverzeichnis

Kapitel 1

Verzeichnis der Namensbereiche

1.1 Liste aller Namensbereiche

Liste aller Namensbereiche mit Kurzbeschreibung:

Kapitel 2

Hierarchie-Verzeichnis

2.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

Kapitel 3

Klassen-Verzeichnis

3.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

Kapitel 4

Datei-Verzeichnis

4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

Kapitel 5

Dokumentation der Namensbereiche

5.1 Cocktailmixer-Namensbereichsreferenz

Klassen

- class Arduino

  Klasse zur Kommunikation mit dem Arduino.
- class Ingredientlist

  Eine Klasse, die die Liste der Zutaten enthält und mehrere Fenster zur Eingabe erstellen kann.
- class IP_Window

  Klasse in der das Fenster mit Eingabefeld für iP und Port erstellt wird.
- class Main_window

  Klasse des Hauptfensters.
- class Recipe

  Klasse zum Erstellen der Rezepte.
- class Secret_settings_window

  Klasse in der das Fenster für die cl-Eingabe des Glases eingestellt werden kann.
- class Settings_window

  Klasse des Einstellungsfensters.

Funktionen

- def update_textsize ()

  Funktion um Textgröße und Schriftart in den Textartpresets zu aktualisieren.
- def decr_size ()

  Funktion um die Textgröße zu verkleinern.
- def incr_size ()

  Funktion um die Textgröße zu vergrößern.
- def create_error_window (error)

  Erstellung einer Fehlermeldung, die bei Buttondruck geschlossen wird.
- def load_file ()

  Funktion zum Laden von Dateien, mit Hilfe der Bibliotheken tkinter_filedialog und pickle.
- def save_file ()

  Funktion um Dateien zu speichern.
- def save_to_file ()

Funktion führt erst das Dialogfenster von tkinter_filedialog aus und danach die Funktion save_file()

- def key_open_settings (event)

  Die Events sind für Tastenkombinationen notwendig und führen die entsprechende Funktion aus.
- def key_open_secret_settings (event)
- def key_save (event)
- def key_save_to (event)
- def key_load (event)
- def key_decr_size (event)
- def key_incr_size (event)

Variablen

- string hlcolor = 'brown4'

  Farbeinstellungen Im folgenden werden Farben in bestimmte Variablen gespeichert, sodass z.B.
- string bgcolor = 'grey12'
- string fgcolor = 'grey70'
- string boxbgcolor = 'grey30'
- string boxfgcolor = 'grey75'
- int textsize = 4

  Starteinstellungen.
- string schriftart = 'Arial'
- string filename = ''
- int number_of_ingredients = 3
- int glas_size = 25
- list recepies = [ ]
- int settings = 0
- int arduino = 0
- ingredientlist = Ingredientlist()

  Objekt wird erstellt.
- mainwindow = Main_window('Cocktails')

  Objekt mit dem Namen mainwindow der Klasse Main_window und mit dem Titel Cocktails wird erstellt.

5.1.1 Dokumentation der Funktionen

5.1.1.1 create_error_window()

```
def Cocktailmixer.create_error_window (
              error )
```

Erstellung einer Fehlermeldung, die bei Buttondruck geschlossen wird.

5.1.1.2 decr_size()

```
def Cocktailmixer.decr_size ( )
```

Funktion um die Textgröße zu verkleinern.

5.1.1.3 incr_size()

```
def Cocktailmixer.incr_size ( )
```

Funktion um die Textgröße zu vergrößern.

5.1.1.4 key_decr_size()

```
def Cocktailmixer.key_decr_size (
              event )
```

5.1.1.5 key_incr_size()

```
def Cocktailmixer.key_incr_size (
              event )
```

5.1.1.6 key_load()

```
def Cocktailmixer.key_load (
              event )
```

5.1.1.7 key_open_secret_settings()

```
def Cocktailmixer.key_open_secret_settings (
              event )
```

5.1.1.8 key_open_settings()

```
def Cocktailmixer.key_open_settings (
              event )
```

Die Events sind für Tastenkombinationen notwendig und führen die entsprechende Funktion aus.

5.1.1.9 key_save()

```
def Cocktailmixer.key_save (
              event )
```

5.1.1.10 key_save_to()

```
def Cocktailmixer.key_save_to (
              event )
```

5.1.1.11 load_file()

```
def Cocktailmixer.load_file ( )
```

Funktion zum Laden von Dateien, mit Hilfe der Bibliotheken tkinter_filedialog und pickle.

Die gespeicherten Listen und Werte werden an den entsprechenden Plätzen eingefügt. (Umkehrung von save_file)

5.1.1.12 save_file()

```
def Cocktailmixer.save_file ( )
```

Funktion um Dateien zu speichern.

Die Funktion kann dazu auf alle relevanten Listen zugreifen und speichert Namen und Mengen der Rezepte in neuen Listen. Anschließend werden alle wichtigen Werte und Listen in eine neue Liste (save_list) gespeichert. Ist kein Dateiname vorhanden wird ein Dialogfenster von tkinter_filedialog geöffnet. Anschließend wird die Liste mittels der Bibliothek pickle gespeichert.

5.1.1.13 save_to_file()

```
def Cocktailmixer.save_to_file ( )
```

Funktion führt erst das Dialogfenster von tkinter_filedialog aus und danach die Funktion save_file()

5.1.1.14 update_textsize()

```
def Cocktailmixer.update_textsize ( )
```

Funktion um Textgröße und Schriftart in den Textartpresets zu aktualisieren.

Parameter

global	text Variable für die Größe des normalen Textes.
global	headline Variable für die Größe von Überschriften.
global	mainheadline Variable für die Größe von Hauptüberschriften.

5.1.2 Variablen-Dokumentation

5.1.2.1 arduino

```
int Cocktailmixer.arduino = 0
```

5.1.2.2 bgcolor

```
string Cocktailmixer.bgcolor = 'grey12'
```

5.1.2.3 boxbgcolor

```
string Cocktailmixer.boxbgcolor = 'grey30'
```

5.1.2.4 boxfgcolor

```
string Cocktailmixer.boxfgcolor = 'grey75'
```

5.1.2.5 fgcolor

```
string Cocktailmixer.fgcolor = 'grey70'
```

5.1.2.6 filename

```
string Cocktailmixer.filename = ''
```

5.1.2.7 glas_size

```
int Cocktailmixer.glas_size = 25
```

5.1.2.8 hlcolor

```
string Cocktailmixer.hlcolor = 'brown4'
```

Farbeinstellungen Im folgenden werden Farben in bestimmte Variablen gespeichert, sodass z.B.

Buttons einheitlich aussehen und die Farbe auf Wunsch leicht geändert werden kann.

Parameter

hlcolor	Überschriftenfarbe

5.1.2.9 ingredientlist

```
Cocktailmixer.ingredientlist = Ingredientlist()
```

Objekt wird erstellt.

5.1.2.10 mainwindow

```
Cocktailmixer.mainwindow = Main_window('Cocktails')
```

Objekt mit dem Namen mainwindow der Klasse Main_window und mit dem Titel Cocktails wird erstellt.

5.1.2.11 number_of_ingredients

```
int Cocktailmixer.number_of_ingredients = 3
```

5.1.2.12 recepies

```
list Cocktailmixer.recepies = []
```

5.1.2.13 schriftart

```
string Cocktailmixer.schriftart = 'Arial'
```

5.1.2.14 settings

```
int Cocktailmixer.settings = 0
```

5.1.2.15 textsize

```
int Cocktailmixer.textsize = 4
```

Starteinstellungen.

Parameter

textsize	Die Textgröße ist standardmäßig bei Programmstart auf 4 eingestellt und wird später mit unterschiedlichen Faktoren für Überschriften oder normalen Texten multipliziert, sodass sich durch die Änderung der Variable textsize die Schriftgröße ändert, aber das Verhältnis gleich bleibt.

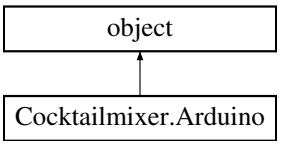
Kapitel 6

Klassen-Dokumentation

6.1 Cocktailmixer.Arduino Klassenreferenz

Klasse zur Kommunikation mit dem Arduino.

Klassendiagramm für Cocktailmixer.Arduino:



Öffentliche Methoden

- def __init__ (self, host, port)

  Initiator der Klasse (Verbindungsaufbau)
- def send_command (self, command)

  Funktion die den Text (in utf8 codiert) an den Arduino sendet.
- def close (self)

  Verbindung wird wieder geschlossen.

Öffentliche Attribute

- socket

6.1.1 Ausführliche Beschreibung

Klasse zur Kommunikation mit dem Arduino.

6.1.2 Beschreibung der Konstruktoren und Destruktoren

6.1.2.1 __init__()

```
def Cocktailmixer.Arduino.__init__ (
              self,
              host,
              port )
```

Initiator der Klasse (Verbindungsaufbau)

Parameter

host	iP-Adresse des Netzwerks
port	Passwort zum Verbinden

6.1.3 Dokumentation der Elementfunktionen

6.1.3.1 close()

```
def Cocktailmixer.Arduino.close (
              self )
```

Verbindung wird wieder geschlossen.

6.1.3.2 send_command()

```
def Cocktailmixer.Arduino.send_command (
              self,
              command )
```

Funktion die den Text (in utf8 codiert) an den Arduino sendet.

Parameter

command	Text der gesendet werden soll.

6.1.4 Dokumentation der Datenelemente

6.1.4.1 socket

```
Cocktailmixer.Arduino.socket
```

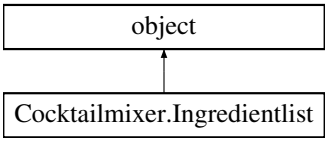
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.2 Cocktailmixer.Ingredientlist Klassenreferenz

Eine Klasse, die die Liste der Zutaten enthält und mehrere Fenster zur Eingabe erstellen kann.

Klassendiagramm für Cocktailmixer.Ingredientlist:



Öffentliche Methoden

- def __init__ (self)

  Liste der Zutaten wird initialisiert.
- def show_in_settings (self, master)

  Erstellung der graphischen Oberfläche für Zutaten in den Einstellungen.
- def show_in_main (self, master)

  Zeigt die Zutaten im Hauptfenster bei 'Dein Cocktail' an.
- def save (self)

  Zutaten werden aus den Felder für Zutaten im Einstellungsfenster kopiert und in der Liste der Zutaten gespeichert.
- def add_ingredient (self)

  Funktion zum Hinzufügen einer Zutat bei Button-Druck '+', solange bis es 10 sind.
- def del_ingredient (self)

  Funktion zum Löschen einer Zutat bei Button-Druck '-', solange es mehr als 0 sind.

Öffentliche Attribute

- ilist
- iframe_s
- ilabel
- ientrys
- editframe
- delbutton
- addbutton
- iframe_m
- inames

6.2.1 Ausführliche Beschreibung

Eine Klasse, die die Liste der Zutaten enthält und mehrere Fenster zur Eingabe erstellen kann.

6.2.2 Beschreibung der Konstruktoren und Destruktoren

6.2.2.1 __init__()

```
def Cocktailmixer.Ingredientlist.__init__ (
              self )
```

Liste der Zutaten wird initialisiert.

6.2.3 Dokumentation der Elementfunktionen

6.2.3.1 add_ingredient()

```
def Cocktailmixer.Ingredientlist.add_ingredient (
              self )
```

Funktion zum Hinzufügen einer Zutat bei Button-Druck '+', solange bis es 10 sind.

6.2.3.2 del_ingredient()

```
def Cocktailmixer.Ingredientlist.del_ingredient (
              self )
```

Funktion zum Löschen einer Zutat bei Button-Druck '-', solange es mehr als 0 sind.

6.2.3.3 save()

```
def Cocktailmixer.Ingredientlist.save (
              self )
```

Zutaten werden aus den Felder für Zutaten im Einstellungsfenster kopiert und in der Liste der Zutaten gespeichert.

6.2.3.4 show_in_main()

```
def Cocktailmixer.Ingredientlist.show_in_main (
              self,
              master )
```

Zeigt die Zutaten im Hauptfenster bei 'Dein Cocktail' an.

6.2.3.5 show_in_settings()

```
def Cocktailmixer.Ingredientlist.show_in_settings (
              self,
              master )
```

Erstellung der graphischen Oberfläche für Zutaten in den Einstellungen.

Dafür wird eine Überschrift ('Zutaten'), Felder zur Eingabe mit der Anzahl der Zutaten, sowie die Buttons '+' und '-' zum Hinzufügen einer neuen Zutat zur Liste ertelllt.

Parameter

master	Zeigt Zutatenfenster mit Eingabefeldern im Einstellungsmenü an

6.2.4 Dokumentation der Datenelemente

6.2.4.1 addbutton

`Cocktailmixer.Ingredientlist.addbutton`

6.2.4.2 delbutton

`Cocktailmixer.Ingredientlist.delbutton`

6.2.4.3 editframe

`Cocktailmixer.Ingredientlist.editframe`

6.2.4.4 ientrys

`Cocktailmixer.Ingredientlist.ientrys`

6.2.4.5 iframe_m

`Cocktailmixer.Ingredientlist.iframe_m`

6.2.4.6 iframe_s

`Cocktailmixer.Ingredientlist.iframe_s`

6.2.4.7 ilabel

`Cocktailmixer.Ingredientlist.ilabel`

6.2.4.8 ilist

`Cocktailmixer.Ingredientlist.ilist`

6.2.4.9 inames

`Cocktailmixer.Ingredientlist.inames`

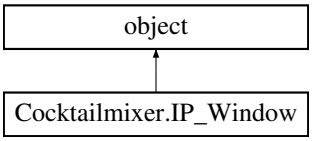
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.3 Cocktailmixer.IP_Window Klassenreferenz

Klasse in der das Fenster mit Eingabefeld für iP und Port erstellt wird.

Klassendiagramm für Cocktailmixer.IP_Window:



Öffentliche Methoden

- def __init__ (self, title)

  Initialisierung des Fensters IP.
- def done (self)

  Funktion die ausgeführt wird wenn 'Fertig'-Button im iP-Window gedrückt wird.

Öffentliche Attribute

- window
- ip_text
- ip_entry
- port_text
- port_entry
- button

6.3.1 Ausführliche Beschreibung

Klasse in der das Fenster mit Eingabefeld für iP und Port erstellt wird.

6.3.2 Beschreibung der Konstruktoren und Destruktoren

6.3.2.1 __init__()

```
def Cocktailmixer.IP_Window.__init__ (
              self,
              title )
```

Initialisierung des Fensters IP.

6.3.3 Dokumentation der Elementfunktionen

6.3.3.1 done()

```
def Cocktailmixer.IP_Window.done (
              self )
```

Funktion die ausgeführt wird wenn 'Fertig'-Button im iP-Window gedrückt wird.

Einträge für iP und host werden an die Klasse Arduino übergeben und das Fenster zerstört.

6.3.4 Dokumentation der Datenelemente

6.3.4.1 button

```
Cocktailmixer.IP_Window.button
```

6.3.4.2 ip_entry

```
Cocktailmixer.IP_Window.ip_entry
```

6.3.4.3 ip_text

`Cocktailmixer.IP_Window.ip_text`

6.3.4.4 port_entry

`Cocktailmixer.IP_Window.port_entry`

6.3.4.5 port_text

`Cocktailmixer.IP_Window.port_text`

6.3.4.6 window

`Cocktailmixer.IP_Window.window`

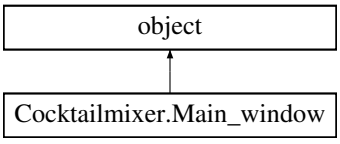
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.4 Cocktailmixer.Main_window Klassenreferenz

Klasse des Hauptfensters.

Klassendiagramm für Cocktailmixer.Main_window:



Öffentliche Methoden

- def __init__ (self, title)

  Initialisierung des Rahmens des Hauptfensters und Ausführung der Optionen die es beinhaltet.
- def create_menubar (self)

  Die Menübar wird erstellt.
- def key_combinations (self)

  Funktion der Definition der Tastenkombinationen für Tastenkombinationen.
- def create_hlframe (self, title)

  Überschrift im Hauptfenster wird erstellt.
- def create_bodyframe (self)

  Die Rezepte aus Namen und Mengen werden im Hauptfenster erstellt.
- def create_botframe (self)

  Erstellung des Bereichs 'Dein Cocktail' im Hauptfenster.
- def open_settings (self)

  Funktion zum erstellen des Einstellungsfensters.
- def open_secret_settings (self)

  Funktion zum erstellen des Einstellungsfensters für die Glasgröße.
- def open_IP_Window (self)

  Funktion zum erstellen des IP-Fensters.
- def mix_userrecepie (self)

  Funktion die ausgeführt wird wenn Mischen-Button im Hauptfenster, 'Dein Cocktail' gedrückt wird.

Öffentliche Attribute

- window
- menubar
- mainmenu
- filemenu
- viewmenu
- hlframe
- headline
- bodyframe
- botframe
- bothl
- botbody
- botbodyleft
- botbodyright
- userentrys
- mix_button
- secret_settings
- ip_window
- userrecepi
- total_amount
- send_text

6.4.1 Ausführliche Beschreibung

Klasse des Hauptfensters.

6.4.2 Beschreibung der Konstruktoren und Destruktoren

6.4.2.1 __init__()

```
def Cocktailmixer.Main_window.__init__ (
              self,
              title )
```

Initialisierung des Rahmens des Hauptfensters und Ausführung der Optionen die es beinhaltet.

6.4.3 Dokumentation der Elementfunktionen

6.4.3.1 create_bodyframe()

```
def Cocktailmixer.Main_window.create_bodyframe (
              self )
```

Die Rezepte aus Namen und Mengen werden im Hauptfenster erstellt.

6.4.3.2 create_botframe()

```
def Cocktailmixer.Main_window.create_botframe (
              self )
```

Erstellung des Bereichs 'Dein Cocktail' im Hauptfenster.

6.4.3.3 create_hlframe()

```
def Cocktailmixer.Main_window.create_hlframe (
              self,
              title )
```

Überschrift im Hauptfenster wird erstellt.

6.4.3.4 create_menubar()

```
def Cocktailmixer.Main_window.create_menubar (
              self )
```

Die Menübar wird erstellt.

Dafür werden verschiedene Menüpunkte erstellt, die nach unten ausklappbar sind und Optionen anbieten. Beim Anklicken verweisen sie auf die vorher erstellten Funktionen load_file, save_file etc.

6.4.3.5 key_combinations()

```
def Cocktailmixer.Main_window.key_combinations (
              self )
```

Funktion der Definition der Tastenkombinationen für Tastenkombinationen.

Sie verweisen auf die Events, welche dann die Funktionen ausführen.

6.4.3.6 mix_userrecepie()

```
def Cocktailmixer.Main_window.mix_userrecepie (
              self )
```

Funktion die ausgeführt wird wenn Mischen-Button im Hauptfenster, 'Dein Cocktail' gedrückt wird.

String mit Mengen der Zutaten wird an den Arduino gesendet, wenn die Gesamtmenge kleiner als die Glasgröße ist.

6.4.3.7 open_IP_Window()

```
def Cocktailmixer.Main_window.open_IP_Window (
              self )
```

Funktion zum erstellen des IP-Fensters.

6.4.3.8 open_secret_settings()

```
def Cocktailmixer.Main_window.open_secret_settings (
              self )
```

Funktion zum erstellen des Einstellungsfensters für die Glasgröße.

6.4.3.9 open_settings()

```
def Cocktailmixer.Main_window.open_settings (
              self )
```

Funktion zum erstellen des Einstellungsfensters.

6.4.4 Dokumentation der Datenelemente

6.4.4.1 bodyframe

```
Cocktailmixer.Main_window.bodyframe
```

6.4.4.2 botbody

```
Cocktailmixer.Main_window.botbody
```

6.4.4.3 botbodyleft

```
Cocktailmixer.Main_window.botbodyleft
```

6.4.4.4 botbodyright

```
Cocktailmixer.Main_window.botbodyright
```

6.4.4.5 botframe

```
Cocktailmixer.Main_window.botframe
```

6.4.4.6 bothl

```
Cocktailmixer.Main_window.bothl
```

6.4.4.7 filemenu

Cocktailmixer.Main_window.filemenu

6.4.4.8 headline

Cocktailmixer.Main_window.headline

6.4.4.9 hlframe

Cocktailmixer.Main_window.hlframe

6.4.4.10 ip_window

Cocktailmixer.Main_window.ip_window

6.4.4.11 mainmenu

Cocktailmixer.Main_window.mainmenu

6.4.4.12 menubar

Cocktailmixer.Main_window.menubar

6.4.4.13 mix_button

Cocktailmixer.Main_window.mix_button

6.4.4.14 secret_settings

Cocktailmixer.Main_window.secret_settings

6.4.4.15 send_text

```
Cocktailmixer.Main_window.send_text
```

6.4.4.16 total_amount

```
Cocktailmixer.Main_window.total_amount
```

6.4.4.17 userentrys

```
Cocktailmixer.Main_window.userentrys
```

6.4.4.18 userrecepi

```
Cocktailmixer.Main_window.userrecepi
```

6.4.4.19 viewmenu

```
Cocktailmixer.Main_window.viewmenu
```

6.4.4.20 window

```
Cocktailmixer.Main_window.window
```

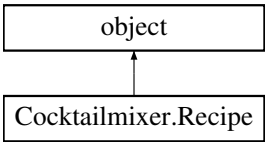
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.5 Cocktailmixer.Recipe Klassenreferenz

Klasse zum Erstellen der Rezepte.

Klassendiagramm für Cocktailmixer.Recipe:



Öffentliche Methoden

- def __init__ (self)

  Liste mit Namen der Rezepte und Mengen der Zutaten wird initialisiert.
- def show_in_settings (self, master)

  Vorhandene Namen und Mengen werden im Einstellungsfenster angezeigt, wird dieses geöffnet.
- def show_in_main (self, master)

  Vohandene Namen und Mengen werden im Hauptfenster angezeigt.
- def save (self)

  Namen und Mengen werden aus den Boxen des Einstellungsfensters gespeichert.
- def send (self)

  Bei Knopfdruck 'Mischen' im Hauptfenster werden die Mengen der Zutaten in einem String gespeichert und an den Arduino gesendet.
- def delete (self)

  Funktion zum Löschen einer Spalte (eines kompletten Rezepts) im Einstellungsfenster.
- def give_total_amount (self)

  Mengen aller Zutaten eines Rezepts werden aufaddiert und in total_amount gespeichert.

Öffentliche Attribute

- name
- amounts
- rframe
- rname
- rspinbox
- delete_button
- innerframe
- rnamelabel
- ramount
- rbutton
- send_text
- total_amount

6.5.1 Ausführliche Beschreibung

Klasse zum Erstellen der Rezepte.

6.5.2 Beschreibung der Konstruktoren und Destruktoren

6.5.2.1 __init__()

```
def Cocktailmixer.Recipe.__init__ (
              self )
```

Liste mit Namen der Rezepte und Mengen der Zutaten wird initialisiert.

6.5.3 Dokumentation der Elementfunktionen

6.5.3.1 delete()

```
def Cocktailmixer.Recipe.delete (
              self )
```

Funktion zum Löschen einer Spalte (eines kompletten Rezepts) im Einstellungsfenster.

6.5.3.2 give_total_amount()

```
def Cocktailmixer.Recipe.give_total_amount (
              self )
```

Mengen aller Zutaten eines Rezepts werden aufaddiert und in total_amount gespeichert.

(um später mit glas_size zu vergleichen).

6.5.3.3 save()

```
def Cocktailmixer.Recipe.save (
              self )
```

Namen und Mengen werden aus den Boxen des Einstellungsfensters gespeichert.

6.5.3.4 send()

```
def Cocktailmixer.Recipe.send (
              self )
```

Bei Knopfdruck 'Mischen' im Hauptfenster werden die Mengen der Zutaten in einem String gespeichert und an den Arduino gesendet.

6.5.3.5 show_in_main()

```
def Cocktailmixer.Recipe.show_in_main (
              self,
              master )
```

Vohandene Namen und Mengen werden im Hauptfenster angezeigt.

6.5.3.6 show_in_settings()

```
def Cocktailmixer.Recipe.show_in_settings (
              self,
              master )
```

Vorhandene Namen und Mengen werden im Einstellungsfenster angezeigt, wird dieses geöffnet.

6.5.4 Dokumentation der Datenelemente

6.5.4.1 amounts

```
Cocktailmixer.Recipe.amounts
```

6.5.4.2 delete_button

```
Cocktailmixer.Recipe.delete_button
```

6.5.4.3 innerframe

```
Cocktailmixer.Recipe.innerframe
```

6.5.4.4 name

```
Cocktailmixer.Recipe.name
```

6.5.4.5 ramount

```
Cocktailmixer.Recipe.ramount
```

6.5.4.6 rbutton

```
Cocktailmixer.Recipe.rbutton
```

6.5.4.7 rframe

`Cocktailmixer.Recipe.rframe`

6.5.4.8 rname

`Cocktailmixer.Recipe.rname`

6.5.4.9 rnamelabel

`Cocktailmixer.Recipe.rnamelabel`

6.5.4.10 rspinbox

`Cocktailmixer.Recipe.rspinbox`

6.5.4.11 send_text

`Cocktailmixer.Recipe.send_text`

6.5.4.12 total_amount

`Cocktailmixer.Recipe.total_amount`

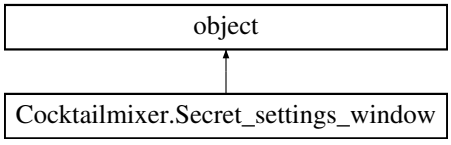
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.6 Cocktailmixer.Secret_settings_window Klassenreferenz

Klasse in der das Fenster für die cl-Eingabe des Glases eingestellt werden kann.

Klassendiagramm für Cocktailmixer.Secret_settings_window:



Öffentliche Methoden

- def __init__ (self, title)

  Initialisierung des Fensters Glasgröße.
- def done (self)

  Funktion die ausgeführt wird wenn 'Fertig'-Button im Glasgröße-Fenster gedrückt wird.

Öffentliche Attribute

- window
- headline
- text
- size_spinbox
- button

6.6.1 Ausführliche Beschreibung

Klasse in der das Fenster für die cl-Eingabe des Glases eingestellt werden kann.

6.6.2 Beschreibung der Konstruktoren und Destruktoren

6.6.2.1 __init__()

```
def Cocktailmixer.Secret_settings_window.__init__ (
              self,
              title )
```

Initialisierung des Fensters Glasgröße.

6.6.3 Dokumentation der Elementfunktionen

6.6.3.1 done()

```
def Cocktailmixer.Secret_settings_window.done (
              self )
```

Funktion die ausgeführt wird wenn 'Fertig'-Button im Glasgröße-Fenster gedrückt wird.

Parameter

global	glas_size Wert aus der Spinbox wird als Integer in der Variable glas_size gespeichert.

6.6.4 Dokumentation der Datenelemente

6.6.4.1 button

`Cocktailmixer.Secret_settings_window.button`

6.6.4.2 headline

`Cocktailmixer.Secret_settings_window.headline`

6.6.4.3 size_spinbox

`Cocktailmixer.Secret_settings_window.size_spinbox`

6.6.4.4 text

`Cocktailmixer.Secret_settings_window.text`

6.6.4.5 window

`Cocktailmixer.Secret_settings_window.window`

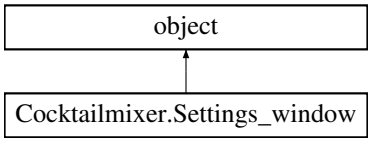
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

6.7 Cocktailmixer.Settings_window Klassenreferenz

Klasse des Einstellungsfensters.

Klassendiagramm für Cocktailmixer.Settings_window:



Öffentliche Methoden

- def __init__ (self, title)

  Fenster wird als Nebenfenster initialisiert.
- def create_hlframe (self, title)

  Überschriftszeile im Einstellungsfenster.
- def create_bodyframe (self)

  Listen und Werte aus dem Hauptfenster werden ins Einstellungsfensterkopiert und das Fenster ertsellt.
- def create_botframe (self)

  Die Buttons 'Neu' und 'Fertig' im Einstellungsfenster werden erstellt.
- def save_all (self)

  Die eingegebenen Zutaten und Rezepte werden gespeichert.
- def update_main (self)

  Funktion zerstört altes 'Cocktails'-Fenster und baut das neue auf, wenn self.done ausgeführt wird.
- def done (self)

  Funktion aktualisiert das Hauptfenster, wenn die eingegebenen cl-Angaben kleiner als die Größe des Glases ist.
- def add (self)

  Funktion zum Hinzufügen eines neuen Rezepts beim Druck auf den Button 'Neu'.

Öffentliche Attribute

- window
- hlframe
- headline
- bodyframe
- botframe
- add_button
- done_button
- max_amount

6.7.1 Ausführliche Beschreibung

Klasse des Einstellungsfensters.

6.7.2 Beschreibung der Konstruktoren und Destruktoren

6.7.2.1 __init__()

```
def Cocktailmixer.Settings_window.__init__ (
              self,
              title )
```

Fenster wird als Nebenfenster initialisiert.

6.7.3 Dokumentation der Elementfunktionen

6.7.3.1 add()

```
def Cocktailmixer.Settings_window.add (
              self )
```

Funktion zum Hinzufügen eines neuen Rezepts beim Druck auf den Button 'Neu'.

6.7.3.2 create_bodyframe()

```
def Cocktailmixer.Settings_window.create_bodyframe (
              self )
```

Listen und Werte aus dem Hauptfenster werden ins Einstellungsfensterkopiert und das Fenster ertsellt.

6.7.3.3 create_botframe()

```
def Cocktailmixer.Settings_window.create_botframe (
              self )
```

Die Buttons 'Neu' und 'Fertig' im Einstellungsfenster werden erstellt.

6.7.3.4 create_hlframe()

```
def Cocktailmixer.Settings_window.create_hlframe (
              self,
              title )
```

Überschriftszeile im Einstellungsfenster.

6.7.3.5 done()

```
def Cocktailmixer.Settings_window.done (
              self )
```

Funktion aktualisiert das Hauptfenster, wenn die eingegebenen cl-Angaben kleiner als die Größe des Glases ist.

Sonst wird das 'Error'-Fenster aufgerufen. (Bei Knopfdruck auf 'Fertig').

6.7.3.6 save_all()

```
def Cocktailmixer.Settings_window.save_all (
              self )
```

Die eingegebenen Zutaten und Rezepte werden gespeichert.

6.7.3.7 update_main()

```
def Cocktailmixer.Settings_window.update_main (
              self )
```

Funktion zerstört altes 'Cocktails'-Fenster und baut das neue auf, wenn self.done ausgeführt wird.

6.7.4 Dokumentation der Datenelemente

6.7.4.1 add_button

```
Cocktailmixer.Settings_window.add_button
```

6.7.4.2 bodyframe

```
Cocktailmixer.Settings_window.bodyframe
```

6.7.4.3 botframe

```
Cocktailmixer.Settings_window.botframe
```

6.7.4.4 done_button

```
Cocktailmixer.Settings_window.done_button
```

6.7.4.5 headline

Cocktailmixer.Settings_window.headline

6.7.4.6 hlframe

Cocktailmixer.Settings_window.hlframe

6.7.4.7 max_amount

Cocktailmixer.Settings_window.max_amount

6.7.4.8 window

Cocktailmixer.Settings_window.window

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Cocktailmixer.py

Kapitel 7

Datei-Dokumentation

7.1 Cocktailmixer.py-Dateireferenz

Klassen

- class Cocktailmixer.Arduino

  Klasse zur Kommunikation mit dem Arduino.
- class Cocktailmixer.Ingredientlist

  Eine Klasse, die die Liste der Zutaten enthält und mehrere Fenster zur Eingabe erstellen kann.
- class Cocktailmixer.Recipe

  Klasse zum Erstellen der Rezepte.
- class Cocktailmixer.Main_window

  Klasse des Hauptfensters.
- class Cocktailmixer.Settings_window

  Klasse des Einstellungsfensters.
- class Cocktailmixer.Secret_settings_window

  Klasse in der das Fenster für die cl-Eingabe des Glases eingestellt werden kann.
- class Cocktailmixer.IP_Window

  Klasse in der das Fenster mit Eingabefeld für iP und Port erstellt wird.

Namensbereiche

- Cocktailmixer

Funktionen

- def Cocktailmixer.update_textsize ()

  Funktion um Textgröße und Schriftart in den Textartpresets zu aktualisieren.
- def Cocktailmixer.decr_size ()

  Funktion um die Textgröße zu verkleinern.
- def Cocktailmixer.incr_size ()

  Funktion um die Textgröße zu vergrößern.
- def Cocktailmixer.create_error_window (error)

  Erstellung einer Fehlermeldung, die bei Buttondruck geschlossen wird.

- def Cocktailmixer.load_file ()

  Funktion zum Laden von Dateien, mit Hilfe der Bibliotheken tkinter_filedialog und pickle.
- def Cocktailmixer.save_file ()

  Funktion um Dateien zu speichern.
- def Cocktailmixer.save_to_file ()

  Funktion führt erst das Dialogfenster von tkinter_filedialog aus und danach die Funktion save_file()
- def Cocktailmixer.key_open_settings (event)

  Die Events sind für Tastenkombinationen notwendig und führen die entsprechende Funktion aus.
- def Cocktailmixer.key_open_secret_settings (event)
- def Cocktailmixer.key_save (event)
- def Cocktailmixer.key_save_to (event)
- def Cocktailmixer.key_load (event)
- def Cocktailmixer.key_decr_size (event)
- def Cocktailmixer.key_incr_size (event)

Variablen

- string Cocktailmixer.hlcolor = 'brown4'

  Farbeinstellungen Im folgenden werden Farben in bestimmte Variablen gespeichert, sodass z.B.
- string Cocktailmixer.bgcolor = 'grey12'
- string Cocktailmixer.fgcolor = 'grey70'
- string Cocktailmixer.boxbgcolor = 'grey30'
- string Cocktailmixer.boxfgcolor = 'grey75'
- int Cocktailmixer.textsize = 4

  Starteinstellungen.
- string Cocktailmixer.schriftart = 'Arial'
- string Cocktailmixer.filename = ''
- int Cocktailmixer.number_of_ingredients = 3
- int Cocktailmixer.glas_size = 25
- list Cocktailmixer.recepies = [ ]
- int Cocktailmixer.settings = 0
- int Cocktailmixer.arduino = 0
- Cocktailmixer.ingredientlist = Ingredientlist()

  Objekt wird erstellt.
- Cocktailmixer.mainwindow = Main_window('Cocktails')

  Objekt mit dem Namen mainwindow der Klasse Main_window und mit dem Titel Cocktails wird erstellt.