Cocktailmixer

Erzeugt von Doxygen 1.8.15

Inhaltsverzeichnis

1 Verzeichnis der Namensbereiche	1
1.1 Liste aller Namensbereiche	1
2 Hierarchie-Verzeichnis	3
2.1 Klassenhierarchie	3
3 Klassen-Verzeichnis	5
3.1 Auflistung der Klassen	5
4 Datei-Verzeichnis	7
4.1 Auflistung der Dateien	7
5 Dokumentation der Namensbereiche	9
5.1 Cocktailmixer-Namensbereichsreferenz	9
5.1.1 Dokumentation der Funktionen	10
5.1.1.1 create_error_window()	10
5.1.1.2 decr_size()	10
5.1.1.3 incr_size()	11
5.1.1.4 key_decr_size()	11
5.1.1.5 key_incr_size()	11
5.1.1.6 key_load()	11
5.1.1.7 key_open_secret_settings()	11
5.1.1.8 key_open_settings()	11
5.1.1.9 key_save()	12
5.1.1.10 key_save_to()	12

ii Inhaltsverzeichnis

5.1.1.11 load_file()	12
5.1.1.12 save_file()	12
5.1.1.13 save_to_file()	12
5.1.1.14 update_textsize()	12
5.1.2 Variablen-Dokumentation	13
5.1.2.1 arduino	13
5.1.2.2 bgcolor	13
5.1.2.3 boxbgcolor	13
5.1.2.4 boxfgcolor	13
5.1.2.5 fgcolor	13
5.1.2.6 filename	13
5.1.2.7 glas_size	14
5.1.2.8 hlcolor	14
5.1.2.9 ingredientlist	14
5.1.2.10 mainwindow	14
5.1.2.11 number_of_ingredients	14
5.1.2.12 recepies	14
5.1.2.13 schriftart	15
5.1.2.14 settings	15
5.1.2.15 textsize	15

INHALTSVERZEICHNIS iii

6 Klassen-Dokumentation	17
6.1 Cocktailmixer.Arduino Klassenreferenz	17
6.1.1 Ausführliche Beschreibung	17
6.1.2 Beschreibung der Konstruktoren und Destruktoren	17
6.1.2.1init()	18
6.1.3 Dokumentation der Elementfunktionen	18
6.1.3.1 close()	18
6.1.3.2 send_command()	18
6.1.4 Dokumentation der Datenelemente	18
6.1.4.1 socket	18
6.2 Cocktailmixer.Ingredientlist Klassenreferenz	19
6.2.1 Ausführliche Beschreibung	19
6.2.2 Beschreibung der Konstruktoren und Destruktoren	20
6.2.2.1init()	20
6.2.3 Dokumentation der Elementfunktionen	20
6.2.3.1 add_ingredient()	20
6.2.3.2 del_ingredient()	20
6.2.3.3 save()	20
6.2.3.4 show_in_main()	20
6.2.3.5 show_in_settings()	21
6.2.4 Dokumentation der Datenelemente	22
6.2.4.1 addbutton	22
6.2.4.2 delbutton	22
6.2.4.3 editframe	22
6.2.4.4 ientrys	22
6.2.4.5 iframe_m	22
6.2.4.6 iframe_s	22
6.2.4.7 ilabel	23
6.2.4.8 ilist	23
6.2.4.9 inames	23

iv INHALTSVERZEICHNIS

6.3 Cocktailmixer.IP_Window Klassenreferenz	23
6.3.1 Ausführliche Beschreibung	24
6.3.2 Beschreibung der Konstruktoren und Destruktoren	24
6.3.2.1init()	24
6.3.3 Dokumentation der Elementfunktionen	24
6.3.3.1 done()	24
6.3.4 Dokumentation der Datenelemente	24
6.3.4.1 button	24
6.3.4.2 ip_entry	24
6.3.4.3 ip_text	25
6.3.4.4 port_entry	25
6.3.4.5 port_text	25
6.3.4.6 window	25
6.4 Cocktailmixer.Main_window Klassenreferenz	25
6.4.1 Ausführliche Beschreibung	26
6.4.2 Beschreibung der Konstruktoren und Destruktoren	27
6.4.2.1init()	27
6.4.3 Dokumentation der Elementfunktionen	27
6.4.3.1 create_bodyframe()	27
6.4.3.2 create_botframe()	27
6.4.3.3 create_hlframe()	27
6.4.3.4 create_menubar()	28
6.4.3.5 key_combinations()	28
6.4.3.6 mix_userrecepie()	28
6.4.3.7 open_IP_Window()	28
6.4.3.8 open_secret_settings()	28
6.4.3.9 open_settings()	29
6.4.4 Dokumentation der Datenelemente	29
6.4.4.1 bodyframe	29
6.4.4.2 botbody	29

INHALTSVERZEICHNIS

6.4.4.3 botbodyleft	. 29
6.4.4.4 botbodyright	. 29
6.4.4.5 botframe	. 29
6.4.4.6 bothl	. 29
6.4.4.7 filemenu	. 30
6.4.4.8 headline	. 30
6.4.4.9 hlframe	. 30
6.4.4.10 ip_window	. 30
6.4.4.11 mainmenu	. 30
6.4.4.12 menubar	. 30
6.4.4.13 mix_button	. 30
6.4.4.14 secret_settings	. 30
6.4.4.15 send_text	. 31
6.4.4.16 total_amount	. 31
6.4.4.17 userentrys	. 31
6.4.4.18 userrecepi	. 31
6.4.4.19 viewmenu	. 31
6.4.4.20 window	. 31
6.5 Cocktailmixer.Recipe Klassenreferenz	. 31
6.5.1 Ausführliche Beschreibung	. 32
6.5.2 Beschreibung der Konstruktoren und Destruktoren	. 32
6.5.2.1 <u>init</u> ()	. 32
6.5.3 Dokumentation der Elementfunktionen	. 33
6.5.3.1 delete()	. 33
6.5.3.2 give_total_amount()	
6.5.3.3 save()	
6.5.3.4 send()	
6.5.3.5 show_in_main()	
6.5.3.6 show_in_settings()	
6.5.4 Dokumentation der Datenelemente	
0.0.4 Donamentation del Datenetiente	. 54

vi INHALTSVERZEICHNIS

6.5.4.1 amounts	 . 34
6.5.4.2 delete_button	 . 34
6.5.4.3 innerframe	 . 34
6.5.4.4 name	 . 34
6.5.4.5 ramount	 . 34
6.5.4.6 rbutton	 . 34
6.5.4.7 rframe	 . 35
6.5.4.8 rname	 . 35
6.5.4.9 rnamelabel	 . 35
6.5.4.10 rspinbox	 . 35
6.5.4.11 send_text	 . 35
6.5.4.12 total_amount	 . 35
6.6 Cocktailmixer.Secret_settings_window Klassenreferenz	 . 35
6.6.1 Ausführliche Beschreibung	 . 36
6.6.2 Beschreibung der Konstruktoren und Destruktoren	 . 36
6.6.2.1init()	 . 36
6.6.3 Dokumentation der Elementfunktionen	 . 36
6.6.3.1 done()	 . 36
6.6.4 Dokumentation der Datenelemente	 . 37
6.6.4.1 button	 . 37
6.6.4.2 headline	 . 37
6.6.4.3 size_spinbox	 . 37
6.6.4.4 text	 . 37
6.6.4.5 window	 . 37
6.7 Cocktailmixer.Settings_window Klassenreferenz	 . 37
6.7.1 Ausführliche Beschreibung	 . 38
6.7.2 Beschreibung der Konstruktoren und Destruktoren	 . 38
6.7.2.1init()	 . 38
6.7.3 Dokumentation der Elementfunktionen	 . 39
6.7.3.1 add()	 . 39

INHALTSVERZEICHNIS vii

6.7.3.2 create_bodyframe()	39
6.7.3.3 create_botframe()	39
6.7.3.4 create_hlframe()	39
6.7.3.5 done()	39
6.7.3.6 save_all()	40
6.7.3.7 update_main()	40
6.7.4 Dokumentation der Datenelemente	40
6.7.4.1 add_button	40
6.7.4.2 bodyframe	40
6.7.4.3 botframe	40
6.7.4.4 done_button	40
6.7.4.5 headline	41
6.7.4.6 hlframe	41
6.7.4.7 max_amount	41
6.7.4.8 window	41
Datei-Dokumentation	43
7.1 Cocktailmixer.py-Dateireferenz	43

Kapitel 1

Verzeichnis der Namensbereiche

4							1						
1.	7		.iste	വി	or I	N	am	0	hΔ	ra		h	Δ
т.		_	JOLE	all		ıw	alli	13	NE		16		•

Liste aller Namensbereiche mit Kurzbeschreibung:	
Cocktailmixer	ç

Kapitel 2

Hierarchie-Verzeichnis

2.1 Klassenhierarchie

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

object

Cocktailmixer.Arduino	17
Cocktailmixer.Ingredientlist	19
Cocktailmixer.IP_Window	23
Cocktailmixer.Main_window	25
Cocktailmixer.Recipe	31
Cocktailmixer.Secret_settings_window	35
Cocktailmixer.Settings_window	37

4 Hierarchie-Verzeichnis

Kapitel 3

Klassen-Verzeichnis

3.1 Auflistung der Klassen

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

Cocktailmixer.Arduino	
Klasse zur Kommunikation mit dem Arduino	17
Cocktailmixer.Ingredientlist	
Eine Klasse, die die Liste der Zutaten enthält und mehrere Fenster zur Eingabe erstellen kann	19
Cocktailmixer.IP_Window	
Klasse in der das Fenster mit Eingabefeld für iP und Port erstellt wird	23
Cocktailmixer.Main_window	
Klasse des Hauptfensters	25
Cocktailmixer.Recipe	
Klasse zum Erstellen der Rezepte	31
Cocktailmixer.Secret_settings_window	
Klasse in der das Fenster für die cl-Eingabe des Glases eingestellt werden kann	35
Cocktailmixer.Settings_window	
Klasse des Einstellungsfensters	37

6 Klassen-Verzeichnis

Kapitel 4

Datei-Verzeichnis

4.1 Auflistung der Dateier	4.1	Aufli	stung	der	Dateier
----------------------------	-----	-------	-------	-----	----------------

lier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:	
Cocktailmixer.py	4;

8 Datei-Verzeichnis

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()
```

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 (
```

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

event)

5.1.1.9 key_save()

5.1.1.10 key save to()

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 Hauptüber		

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.

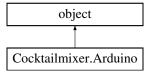
Kapitel 6

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

18 Klassen-Dokumentation

```
6.1.2.1 __init__()
```

Initiator der Klasse (Verbindungsaufbau)

Parameter

h	ost	iP-Adresse des Netzwerks
р	ort	Passwort zum Verbinden

6.1.3 Dokumentation der Elementfunktionen

6.1.3.1 close()

```
\label{eq:cocktailmixer.Arduino.close} \mbox{ (} \\ self \mbox{ )}
```

Verbindung wird wieder geschlossen.

6.1.3.2 send_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

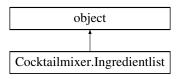
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· 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.

20 Klassen-Dokumentation

6.2.2 Beschreibung der Konstruktoren und Destruktoren

Liste der Zutaten wird initialisiert.

6.2.3 Dokumentation der Elementfunktionen

6.2.3.1 add_ingredient()

```
\begin{tabular}{ll} \tt def Cocktailmixer.Ingredientlist.add\_ingredient & ( \\ & self \end{tabular} \label{table}
```

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

6.2.3.2 del_ingredient()

```
\begin{tabular}{ll} \tt def Cocktailmixer.Ingredientlist.del\_ingredient ( \\ self ) \end{tabular}
```

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()

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

6.2.3.5 show_in_settings()

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.

22 Klassen-Dokumentation

Parameter

master	Zeigt Zutatenfenster mit Eingabefeldern im Einstellungsmenü an
mactor	Loigt Latatorionotto mit Lingaporotaom in Linetonangomena 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

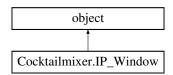
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· 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

24 Klassen-Dokumentation

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

Initialisierung des Fensters IP.

6.3.3 Dokumentation der Elementfunktionen

6.3.3.1 done()

```
\begin{tabular}{ll} \tt def Cocktailmixer.IP\_Window.done \ ( \\ self \ ) \end{tabular}
```

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

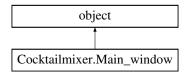
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· Cocktailmixer.py

6.4 Cocktailmixer.Main_window Klassenreferenz

Klasse des Hauptfensters.

Klassendiagramm für Cocktailmixer.Main_window:



26 Klassen-Dokumentation

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

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

6.4.3 Dokumentation der Elementfunktionen

6.4.3.1 create_bodyframe()

Die Rezepte aus Namen und Mengen werden im Hauptfenster erstellt.

6.4.3.2 create_botframe()

```
\label{lem:cocktailmixer.Main_window.create_botframe (} self \ )
```

Erstellung des Bereichs 'Dein Cocktail' im Hauptfenster.

6.4.3.3 create_hlframe()

Überschrift im Hauptfenster wird erstellt.

6.4.3.4 create_menubar()

```
\begin{tabular}{ll} \tt def Cocktailmixer.Main\_window.create\_menubar ( \\ self ) \end{tabular}
```

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()

```
\label{lem:cocktailmixer.Main_window.key_combinations} \mbox{ (} \\ self \mbox{ )}
```

Funktion der Definition der Tastenkombinationen für Tastenkombinationen.

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

6.4.3.6 mix_userrecepie()

```
\label{lem:cocktailmixer.Main_window.mix\_userrecepie} \mbox{ (} \\ self \mbox{ )}
```

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()

```
\label{lem:cocktailmixer.Main_window.open_IP_Window} \mbox{ (} \\ self \mbox{ )}
```

Funktion zum erstellen des IP-Fensters.

6.4.3.8 open_secret_settings()

```
\label{lem:cocktailmixer.Main_window.open_secret_settings (} self \ )
```

Funktion zum erstellen des Einstellungsfensters für die Glasgröße.

6.4.3.9 open_settings()

```
\label{lem:cocktailmixer.Main_window.open_settings} \mbox{ (} \\ self \mbox{ )}
```

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

 ${\tt Cocktailmixer.Main_window.botbodyright}$

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 ${\tt Cocktailmixer.Main_window.mix_button}$ 6.4.4.14 secret_settings

Cocktailmixer.Main_window.secret_settings

Erzeugt von Doxygen

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

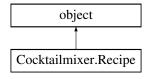
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· 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
- ramountrbutton
- send_text
- total_amount

6.5.1 Ausführliche Beschreibung

Klasse zum Erstellen der Rezepte.

6.5.2 Beschreibung der Konstruktoren und Destruktoren

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

6.5.3 Dokumentation der Elementfunktionen

6.5.3.1 delete()

```
\begin{tabular}{ll} \tt def \ Cocktailmixer.Recipe.delete \ ( \\ self \ ) \end{tabular}
```

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

6.5.3.2 give_total_amount()

```
\label{lem: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 (
```

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()

Vohandene Namen und Mengen werden im Hauptfenster angezeigt.

6.5.3.6 show_in_settings()

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

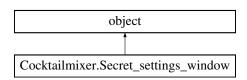
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· 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

Initialisierung des Fensters Glasgröße.

6.6.3 Dokumentation der Elementfunktionen

6.6.3.1 done()

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

 ${\tt Cocktailmixer.Secret_settings_window.button}$

6.6.4.2 headline

Cocktailmixer.Secret_settings_window.headline

6.6.4.3 size_spinbox

 ${\tt Cocktailmixer.Secret_settings_window.size_spinbox}$

6.6.4.4 text

Cocktailmixer.Secret_settings_window.text

6.6.4.5 window

 ${\tt Cocktailmixer.Secret_settings_window.window}$

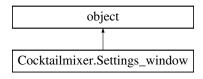
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

· 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

Fenster wird als Nebenfenster initialisiert.

6.7.3 Dokumentation der Elementfunktionen

6.7.3.1 add()

```
\begin{tabular}{ll} \tt def Cocktailmixer.Settings\_window.add ( \\ & self ) \end{tabular}
```

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

6.7.3.2 create_bodyframe()

```
\label{lem: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()

```
\label{lem: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()

```
\label{lem:cocktailmixer.Settings_window.done} \mbox{ (} \\ self \mbox{ )}
```

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()

```
\label{lem:cocktailmixer.Settings_window.save_all (} self \ )
```

Die eingegebenen Zutaten und Rezepte werden gespeichert.

6.7.3.7 update_main()

```
\begin{tabular}{ll} \tt def Cocktailmixer.Settings\_window.update\_main ( \\ self ) \end{tabular}
```

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

 ${\tt Cocktailmixer.Settings_window.add_button}$

6.7.4.2 bodyframe

 ${\tt Cocktailmixer.Settings_window.bodyframe}$

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

 ${\tt Cocktailmixer.Settings_window.headline}$

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.

44 Datei-Dokumentation

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.