

## Mycroft Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Namespace Documentation</b>	<b>5</b>
3.1	MycroftApp Namespace Reference . . . . .	5
3.1.1	Function Documentation . . . . .	6
3.1.1.1	clear_all() . . . . .	6
3.1.1.2	help_button() . . . . .	6
3.1.1.3	new_project_button() . . . . .	6
3.1.1.4	select_file_button() . . . . .	7
3.1.1.5	t1_button1_code() . . . . .	7
3.1.1.6	t2_button1_code() . . . . .	7
3.1.1.7	t3_button1_code() . . . . .	7
3.1.2	Variable Documentation . . . . .	7
3.1.2.1	canvas . . . . .	7
3.1.2.2	column . . . . .	7
3.1.2.3	command . . . . .	8
3.1.2.4	label . . . . .	8
3.1.2.5	menu . . . . .	8
3.1.2.6	notebook . . . . .	8
3.1.2.7	root . . . . .	8

---

3.1.2.8	style	8
3.1.2.9	subMenu	8
3.1.2.10	t1	9
3.1.2.11	t1_canvas	9
3.1.2.12	t1_points1	9
3.1.2.13	t1_shape1	9
3.1.2.14	t2	9
3.1.2.15	t2_canvas	9
3.1.2.16	t2_points1	10
3.1.2.17	t2_shape1	10
3.1.2.18	t3	10
3.1.2.19	t3_canvas	10
3.1.2.20	t3_points1	10
3.1.2.21	t3_shape1	10
3.1.2.22	tabposition	10
3.1.2.23	text	10
<b>4</b>	<b>File Documentation</b>	<b>11</b>
4.1	MycroftApp.py File Reference	11
<b>Index</b>		<b>13</b>

# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">MycroftApp</a> . . . . .	5
--------------------------------------	---



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

[MycroftApp.py](#) . . . . . 11





## Chapter 3

# Namespace Documentation

### 3.1 MycroftApp Namespace Reference

#### Functions

- def `select_file_button ()`  
*Function to open a window allowing the user to select a file when called.*
- def `new_project_button ()`  
*Function to open a command prompt with a loaded Mycroft questionnaire when called.*
- def `help_button ()`  
*Function to open the README in the application when called.*
- def `clear_all ()`  
*Function to clear the canvas when called.*
- def `t1_button1_code (self)`  
*Function with code assigned for the first button.*
- def `t2_button1_code (self)`
- def `t3_button1_code (self)`

#### Variables

- `root = tk.Tk()`
- `canvas = Canvas(root, width=800, height=500, bg="white")`  
*Sets up the main canvas size and colour.*
- `menu = Menu(root)`  
*Sets up the TKinter menu.*
- `subMenu = Menu(menu)`  
*Creates a submenu for multiple buttons to be hidden in it.*
- `label`  
*Label which gives the option a name in menu.*
- `command`
- `style = ttk.Style(root)`
- `tabposition`  
*Configures the notebook style and set the tab (North, South, East, West)*
- `notebook = ttk.Notebook(root, style='lefttab.TNotebook')`  
*Sets the theme and side of the tab buttons.*

- `t1 = tk.Frame(notebook)`  
*Tab frame created and assigned a variable name.*
- `t2 = tk.Frame(notebook)`
- `t3 = tk.Frame(notebook)`
- `text`
- `column`  
*Sets the specified UI position in the Y axis.*
- `t1_canvas = Canvas(t1, width=200, height=500, bg='red')`  
*The side tabs are changed into canvases allowing shapes to be created on them.*
- `t2_canvas = Canvas(t2, width=200, height=500, bg='blue')`
- `t3_canvas = Canvas(t3, width=200, height=500, bg='green')`
- `list t1_points1 = [20, 20, 20, 100, 100, 100, 100, 80, 120, 80, 120, 40, 100, 40, 100, 20]`  
*The points represent the x and y axis of the canvas and each 2 points a point where the line stops.*
- `t1_shape1 = t1_canvas.create_polygon(t1_points1, outline='#000', fill='#7e2530', width=2)`  
*shape is created by taking the pre-established points and specifying the shape's border and inside colour as well as border width.*
- `list t2_points1 = [20, 20, 20, 40, 40, 40, 40, 80, 20, 80, 20, 100, 100, 100, 100, 80, 120, 60, 100, 40, 100, 20]`
- `t2_shape1 = t2_canvas.create_polygon(t2_points1, outline='#000', fill='#003153', width=2)`
- `list t3_points1 = [20, 20, 20, 40, 40, 60, 20, 80, 20, 100, 100, 100, 100, 20]`
- `t3_shape1 = t3_canvas.create_polygon(t3_points1, outline='#000', fill='#00630d', width=2)`

### 3.1.1 Function Documentation

#### 3.1.1.1 `clear_all()`

```
def MycroftApp.clear_all ( )
```

Function to clear the canvas when called.

#### 3.1.1.2 `help_button()`

```
def MycroftApp.help_button ( )
```

Function to open the README in the application when called.

#### 3.1.1.3 `new_project_button()`

```
def MycroftApp.new_project_button ( )
```

Function to open a command prompt with a loaded Mycroft questionnaire when called.

#### 3.1.1.4 select\_file\_button()

```
def MycroftApp.select_file_button ( )
```

Function to open a window allowing the user to select a file when called.

#### 3.1.1.5 t1\_button1\_code()

```
def MycroftApp.t1_button1_code (
    self )
```

Function with code assigned for the first button.

#### 3.1.1.6 t2\_button1\_code()

```
def MycroftApp.t2_button1_code (
    self )
```

#### 3.1.1.7 t3\_button1\_code()

```
def MycroftApp.t3_button1_code (
    self )
```

### 3.1.2 Variable Documentation

#### 3.1.2.1 canvas

```
MycroftApp.canvas = Canvas(root, width=800, height=500, bg="white")
```

Sets up the main canvas size and colour.

#### 3.1.2.2 column

```
MycroftApp.column
```

Sets the specified UI position in the Y axis.

### 3.1.2.3 command

```
MycroftApp.command
```

### 3.1.2.4 label

```
MycroftApp.label
```

Label which gives the option a name in menu.

Function the button executes when clicked.

### 3.1.2.5 menu

```
MycroftApp.menu = Menu(root)
```

Sets up the TKinter menu.

### 3.1.2.6 notebook

```
MycroftApp.notebook = ttk.Notebook(root, style='lefttab.TNotebook' )
```

Sets the theme and side of the tab buttons.

### 3.1.2.7 root

```
MycroftApp.root = tk.Tk()
```

### 3.1.2.8 style

```
MycroftApp.style = ttk.Style(root)
```

### 3.1.2.9 subMenu

```
MycroftApp.subMenu = Menu(menu)
```

Creates a submenu for multiple buttons to be hidden in it.

### 3.1.2.10 t1

```
MycroftApp.t1 = tk.Frame(notebook)
```

Tab frame created and assigned a variable name.

### 3.1.2.11 t1\_canvas

```
MycroftApp.t1_canvas = Canvas(t1, width=200, height=500, bg='red')
```

The side tabs are changed into canvases allowing shapes to be created on them.

The canvas size and colour are set and tabs are assigned.

### 3.1.2.12 t1\_points1

```
list MycroftApp.t1_points1 = [20, 20, 20, 100, 100, 100, 100, 80, 120, 80, 120, 40, 100, 40, 100, 20 ]
```

The points represent the x and y axis of the canvas and each 2 points a point where the line stops.

Eg if you take the first 6 points [20, 20, 20, 100, 100, 100]. This means the line starts at 20, 20 and goes to 20, 100 and from that point it turns to 100, 100 to make another point there

### 3.1.2.13 t1\_shape1

```
MycroftApp.t1_shape1 = t1_canvas.create_polygon(t1_points1, outline='#000', fill='#7e2530', width=2)
```

shape is created by taking the pre-established points and specifying the shape's border and inside colour as well as border width.

### 3.1.2.14 t2

```
MycroftApp.t2 = tk.Frame(notebook)
```

### 3.1.2.15 t2\_canvas

```
MycroftApp.t2_canvas = Canvas(t2, width=200, height=500, bg='blue')
```

### 3.1.2.16 t2\_points1

```
list MycroftApp.t2_points1 = [20, 20, 20, 40, 40, 40, 40, 80, 20, 80, 20, 100, 100, 100, 100, 80, 120, 60, 100, 40, 100, 20]
```

### 3.1.2.17 t2\_shape1

```
MycroftApp.t2_shape1 = t2_canvas.create_polygon(t2_points1, outline='#000', fill='#003153', width=2)
```

### 3.1.2.18 t3

```
MycroftApp.t3 = tk.Frame(notebook)
```

### 3.1.2.19 t3\_canvas

```
MycroftApp.t3_canvas = Canvas(t3, width=200, height=500, bg='green')
```

### 3.1.2.20 t3\_points1

```
list MycroftApp.t3_points1 = [20, 20, 20, 40, 40, 60, 20, 80, 20, 100, 100, 100, 100, 20]
```

### 3.1.2.21 t3\_shape1

```
MycroftApp.t3_shape1 = t3_canvas.create_polygon(t3_points1, outline='#000', fill='#00630d', width=2)
```

### 3.1.2.22 tabposition

```
MycroftApp.tabposition
```

Configures the notebook style and set the tab (North, South, East, West)

### 3.1.2.23 text

```
MycroftApp.text
```

# Chapter 4

## File Documentation

### 4.1 MycroftApp.py File Reference

#### Namespaces

- [MycroftApp](#)

#### Functions

- def [MycroftApp.select\\_file\\_button](#) ()  
*Function to open a window allowing the user to select a file when called.*
- def [MycroftApp.new\\_project\\_button](#) ()  
*Function to open a command prompt with a loaded Mycroft questionnaire when called.*
- def [MycroftApp.help\\_button](#) ()  
*Function to open the README in the application when called.*
- def [MycroftApp.clear\\_all](#) ()  
*Function to clear the canvas when called.*
- def [MycroftApp.t1\\_button1\\_code](#) (self)  
*Function with code assigned for the first button.*
- def [MycroftApp.t2\\_button1\\_code](#) (self)
- def [MycroftApp.t3\\_button1\\_code](#) (self)

#### Variables

- [MycroftApp.root](#) = tk.Tk()
- [MycroftApp.canvas](#) = Canvas(root, width=800, height=500, bg="white")  
*Sets up the main canvas size and colour.*
- [MycroftApp.menu](#) = Menu(root)  
*Sets up the TKinter menu.*
- [MycroftApp.subMenu](#) = Menu(menu)  
*Creates a submenu for multiple buttons to be hidden in it.*
- [MycroftApp.label](#)  
*Label which gives the option a name in menu.*
- [MycroftApp.command](#)
- [MycroftApp.style](#) = ttk.Style(root)

- [MycroftApp.tabposition](#)  
*Configures the notebook style and set the tab (North, South, East, West)*
- [MycroftApp.notebook](#) = `ttk.Notebook(root, style='lefttab.TNotebook' )`  
*Sets the theme and side of the tab buttons.*
- [MycroftApp.t1](#) = `tk.Frame(notebook)`  
*Tab frame created and assigned a variable name.*
- [MycroftApp.t2](#) = `tk.Frame(notebook)`
- [MycroftApp.t3](#) = `tk.Frame(notebook)`
- [MycroftApp.text](#)
- [MycroftApp.column](#)  
*Sets the specified UI position in the Y axis.*
- [MycroftApp.t1\\_canvas](#) = `Canvas(t1, width=200, height=500, bg='red')`  
*The side tabs are chaned into canvases allowing shapes to be created on them.*
- [MycroftApp.t2\\_canvas](#) = `Canvas(t2, width=200, height=500, bg='blue')`
- [MycroftApp.t3\\_canvas](#) = `Canvas(t3, width=200, height=500, bg='green')`
- list [MycroftApp.t1\\_points1](#) = `[20, 20, 20, 100, 100, 100, 100, 80, 120, 80, 120, 40, 100, 40, 100, 20 ]`  
*The points represent the x and y axis of the canvas and each 2 points a point where the line stops.*
- [MycroftApp.t1\\_shape1](#) = `t1_canvas.create_polygon(t1_points1, outline='#000', fill='#7e2530', width=2)`  
*shape is created by taking the pre-established points and specifying the shape's border and inside colour as well as border width.*
- list [MycroftApp.t2\\_points1](#) = `[20, 20, 20, 40, 40, 40, 40, 80, 20, 80, 20, 100, 100, 100, 100, 80, 120, 60, 100, 40, 100, 20]`
- [MycroftApp.t2\\_shape1](#) = `t2_canvas.create_polygon(t2_points1, outline='#000', fill='#003153', width=2)`
- list [MycroftApp.t3\\_points1](#) = `[20, 20, 20, 40, 40, 60, 20, 80, 20, 100, 100, 100, 100, 20]`
- [MycroftApp.t3\\_shape1](#) = `t3_canvas.create_polygon(t3_points1, outline='#000', fill='#00630d', width=2)`



# Index

- canvas
  - MycroftApp, 7
- clear\_all
  - MycroftApp, 6
- column
  - MycroftApp, 7
- command
  - MycroftApp, 7
- help\_button
  - MycroftApp, 6
- label
  - MycroftApp, 8
- menu
  - MycroftApp, 8
- MycroftApp, 5
  - canvas, 7
  - clear\_all, 6
  - column, 7
  - command, 7
  - help\_button, 6
  - label, 8
  - menu, 8
  - new\_project\_button, 6
  - notebook, 8
  - root, 8
  - select\_file\_button, 6
  - style, 8
  - subMenu, 8
  - t1, 8
  - t1\_button1\_code, 7
  - t1\_canvas, 9
  - t1\_points1, 9
  - t1\_shape1, 9
  - t2, 9
  - t2\_button1\_code, 7
  - t2\_canvas, 9
  - t2\_points1, 9
  - t2\_shape1, 10
  - t3, 10
  - t3\_button1\_code, 7
  - t3\_canvas, 10
  - t3\_points1, 10
  - t3\_shape1, 10
  - tabposition, 10
  - text, 10
- MycroftApp.py, 11
- new\_project\_button
  - MycroftApp, 6
- notebook
  - MycroftApp, 8
- root
  - MycroftApp, 8
- select\_file\_button
  - MycroftApp, 6
- style
  - MycroftApp, 8
- subMenu
  - MycroftApp, 8
- t1
  - MycroftApp, 8
- t1\_button1\_code
  - MycroftApp, 7
- t1\_canvas
  - MycroftApp, 9
- t1\_points1
  - MycroftApp, 9
- t1\_shape1
  - MycroftApp, 9
- t2
  - MycroftApp, 9
- t2\_button1\_code
  - MycroftApp, 7
- t2\_canvas
  - MycroftApp, 9
- t2\_points1
  - MycroftApp, 9
- t2\_shape1
  - MycroftApp, 10
- t3
  - MycroftApp, 10
- t3\_button1\_code
  - MycroftApp, 7
- t3\_canvas
  - MycroftApp, 10
- t3\_points1
  - MycroftApp, 10
- t3\_shape1
  - MycroftApp, 10
- tabposition
  - MycroftApp, 10
- text
  - MycroftApp, 10