

TerminalTalk

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Packages	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	client Namespace Reference	7
4.1.1	Detailed Description	7
4.2	font Namespace Reference	7
4.2.1	Detailed Description	7
4.3	server Namespace Reference	7
4.3.1	Detailed Description	7
5	Class Documentation	9
5.1	font.constants.Colors Class Reference	9
5.1.1	Detailed Description	9
5.2	server.Orator.Orator Class Reference	9
5.2.1	Detailed Description	10
5.2.2	Constructor & Destructor Documentation	10
5.2.2.1	__init__(self, socket)	10
5.3	font.constants.Styles Class Reference	10
5.3.1	Detailed Description	11

6	File Documentation	13
6.1	/home/bryan/TerminalTalk/client/eavesdrop.py File Reference	13
6.2	/home/bryan/TerminalTalk/font/constants.py File Reference	13
6.2.1	Detailed Description	13
6.3	/home/bryan/TerminalTalk/font/functions.py File Reference	13
6.3.1	Detailed Description	14
6.3.2	Function Documentation	14
6.3.2.1	blue(txt)	14
6.3.2.2	bold(txt)	14
6.3.2.3	cyan(txt)	15
6.3.2.4	default(txt)	15
6.3.2.5	green(txt)	15
6.3.2.6	highlight(txt)	15
6.3.2.7	italic(txt)	16
6.3.2.8	magenta(txt)	16
6.3.2.9	red(txt)	16
6.3.2.10	underline(txt)	16
6.3.2.11	yellow(txt)	17
6.4	/home/bryan/TerminalTalk/server/Orator.py File Reference	17
6.5	/home/bryan/TerminalTalk/server/pontification.py File Reference	17
6.5.1	Function Documentation	17
6.5.1.1	pontificate(orator, verbiage, connections, server, client, orators)	17
6.5.1.2	server_pontificate(verbiage, connections, server, orators)	18
6.6	/home/bryan/TerminalTalk/TerminalTalk.py File Reference	18
6.6.1	Detailed Description	18
6.7	/home/bryan/TerminalTalk/TerminalTalk_server.py File Reference	18
6.7.1	Detailed Description	19

Chapter 1

Namespace Index

1.1 Packages

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

client	Functions and definitions specific to TerminalTalk client	7
font	A collection of functions that style fonts in terminal	7
server	Functions and definitions specific to TerminalTalk server	7

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

font.constants.Colors	
Contains constants for ANSI color codes	9
server.Orator.Orator	
Stores information about a connected client	9
font.constants.Styles	
Contains constants for ANSI text style codes	10

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

/home/bryan/TerminalTalk/ TerminalTalk.py	
Main code for TerminalTalk client	18
/home/bryan/TerminalTalk/ TerminalTalk_server.py	
Main code for TerminalTalk server	18
/home/bryan/TerminalTalk/client/ eavesdrop.py	
Main code for TerminalTalk client	13
/home/bryan/TerminalTalk/font/ constants.py	
Defines classes contains ANSI constants for formating text	13
/home/bryan/TerminalTalk/font/ functions.py	
Defines functions for surrounding strings with ANSI codes for formating text	13
/home/bryan/TerminalTalk/server/ Orator.py	
Main code for TerminalTalk server	17
/home/bryan/TerminalTalk/server/ pontification.py	
Main code for TerminalTalk server	17

Chapter 4

Namespace Documentation

4.1 client Namespace Reference

Functions and definitions specific to TerminalTalk client.

4.1.1 Detailed Description

Functions and definitions specific to TerminalTalk client.

4.2 font Namespace Reference

A collection of functions that style fonts in terminal.

4.2.1 Detailed Description

A collection of functions that style fonts in terminal.

Each function flanks the inputted text with ANSI codes.

4.3 server Namespace Reference

Functions and definitions specific to TerminalTalk server.

4.3.1 Detailed Description

Functions and definitions specific to TerminalTalk server.

Chapter 5

Class Documentation

5.1 font.constants.Colors Class Reference

Contains constants for ANSI color codes.

Static Public Attributes

- string **BLACK** = '\033[30m'
- string **RED** = '\033[31m'
- string **GREEN** = '\033[32m'
- string **YELLOW** = '\033[33m'
- string **BLUE** = '\033[34m'
- string **MAGENTA** = '\033[35m'
- string **CYAN** = '\033[36m'
- string **WHITE** = '\033[37m'

5.1.1 Detailed Description

Contains constants for ANSI color codes.

Prefix a string with color constant to change subsequent text to the color. Example: `print(font.Colors.RED + "This text will be red.")`

The documentation for this class was generated from the following file:

- [/home/bryan/TerminalTalk/font/constants.py](#)

5.2 server.Orator.Orator Class Reference

Stores information about a connected client.

Public Member Functions

- `def __init__ (self, socket)`
Constructor for [Orator](#) class.

Public Attributes

- **socket**
- **username**
- **color**

5.2.1 Detailed Description

Stores information about a connected client.

An [Orator](#) objects is used to represent a client. Each client has a username and color associated with them. The username is provided by the client. The color is assigned randomly each time the client connects.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `def server.Orator.Orator.__init__ (self, socket)`

Constructor for [Orator](#) class.

Parameters

<code>socket</code>	The socket (file descriptor) used to connect with client.
---------------------	---

The documentation for this class was generated from the following file:

- `/home/bryan/TerminalTalk/server/Orator.py`

5.3 font.constants.Styles Class Reference

Contains constants for ANSI text style codes.

Static Public Attributes

- string **RESET** = '\033[0m'
- string **BOLD** = '\033[1m'
- string **BOLD_OFF** = '\033[22m'
- string **ITALIC** = '\033[3m'
- string **ITALIC_OFF** = '\033[23m'
- string **UNDERLINE** = '\033[4m'
- string **UNDERLINE_OFF** = '\033[24m'
- string **STRIKETHROUGH** = '\033[9m'
- string **STRIKETHROUHG_OFF** = '\033[29m'
- string **INVERSE** = '\033[7m'
- string **INVERSE_OFF** = '\033[27m'

5.3.1 Detailed Description

Contains constants for ANSI text style codes.

Prefix a string with a style constant to change subsequent text style. Example: `print(font.Styles.BOLD + "This text will be bold.")`

The documentation for this class was generated from the following file:

- </home/bryan/TerminalTalk/font/constants.py>

Chapter 6

File Documentation

6.1 /home/bryan/TerminalTalk/client/eavesdrop.py File Reference

Functions

- def [client.eavesdrop.eavesdrop](#) ()
Monitors for user input from terminal.

6.2 /home/bryan/TerminalTalk/font/constants.py File Reference

Defines classes contains ANSI constants for formatting text.

Classes

- class [font.constants.Styles](#)
Contains constants for ANSI text style codes.
- class [font.constants.Colors](#)
Contains constants for ANSI color codes.

6.2.1 Detailed Description

Defines classes contains ANSI constants for formatting text.

6.3 /home/bryan/TerminalTalk/font/functions.py File Reference

Defines functions for surrounding strings with ANSI codes for formatting text.

Functions

- def `font.functions.blue` (txt)
Makes inputed string blue.
- def `font.functions.cyan` (txt)
Makes inputed string cyan.
- def `font.functions.green` (txt)
Makes inputed string green.
- def `font.functions.magenta` (txt)
Makes inputed string magenta.
- def `font.functions.red` (txt)
Makes inputed string red.
- def `font.functions.yellow` (txt)
Makes inputed string red.
- def `font.functions.underline` (txt)
Makes inputed string underline.
- def `font.functions.bold` (txt)
Makes inputed string bold.
- def `font.functions.italic` (txt)
Makes inputed string italic.
- def `font.functions.default` (txt)
Gives inputed string default formatting.
- def `font.functions.highlight` (txt)
Highlights inputed string.

6.3.1 Detailed Description

Defines functions for surrounding strings with ANSI codes for formating text.

6.3.2 Function Documentation

6.3.2.1 `def font.functions.blue (txt)`

Makes inputed string blue.

Returns with inputed string with the prefix '\033[34m' and suffix '\033[0m'. The prefix is the ANSI code for blue foreground. The suffix is the ANSI code for reset (white forground, black background).

Parameters

<code>txt</code>	A string. Will be displayed as blue.
------------------	--------------------------------------

6.3.2.2 `def font.functions.bold (txt)`

Makes inputed string bold.

Returns with inputed string with the prefix '\033[1m' and suffix '\033[22m'. The prefix is the ANSI code for bold font on. The suffix is the ANSI code for bold font off.

Parameters

<i>txt</i>	A string. Will be made bold.
------------	------------------------------

6.3.2.3 def font.functions.cyan (*txt*)

Makes inputed string cyan.

Returns with inputed string with the prefix '\033[36m' and suffix '\033[0m'. The prefix is the ANSI code for cyan foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be displayed as cyan.
------------	--------------------------------------

6.3.2.4 def font.functions.default (*txt*)

Gives inputed string default formatting.

Returns with inputed string with the prefix '\033[0m'. The prefix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be formatted as default.
------------	---

6.3.2.5 def font.functions.green (*txt*)

Makes inputed string green.

Returns with inputed string with the prefix '\033[32m' and suffix '\033[0m'. The prefix is the ANSI code for green foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be displayed as green.
------------	---------------------------------------

6.3.2.6 def font.functions.highlight (*txt*)

Highlights inputed string.

Returns with inputed string with the prefixes '\033[47m', '\033[30m', and '\033[3m', and with the suffix '\033[0m'. The prefix is the ANSI code for italic, white background, and black foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be highlighted.
------------	--------------------------------

6.3.2.7 def font.functions.italic (*txt*)

Makes inputed string italic.

Returns with inputed string with the prefix '\033[3m' and suffix '\033[23m'. The prefix is the ANSI code for italic on. The suffix is the ANSI code for italic off.

Parameters

<i>txt</i>	A string. Will be displayed as italic.
------------	--

6.3.2.8 def font.functions.magenta (*txt*)

Makes inputed string magenta.

Returns with inputed string with the prefix '\033[35m' and suffix '\033[0m'. The prefix is the ANSI code for magenta foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be displayed as magenta.
------------	---

6.3.2.9 def font.functions.red (*txt*)

Makes inputed string red.

Returns with inputed string with the prefix '\033[31m' and suffix '\033[0m'. The prefix is the ANSI code for red foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be displayed as red.
------------	-------------------------------------

6.3.2.10 def font.functions.underline (*txt*)

Makes inputed string underline.

Returns with inputed string with the prefix '\033[4m' and suffix '\033[24m'. The prefix is the ANSI code for underline on. The suffix is the ANSI code for underline off.

Parameters

<i>txt</i>	A string. Will be underlined.
------------	-------------------------------

6.3.2.11 `def font.functions.yellow (txt)`

Makes inputted string red.

Returns with inputted string with the prefix '\033[33m' and suffix '\033[0m'. The prefix is the ANSI code for red foreground. The suffix is the ANSI code for reset (white foreground, black background).

Parameters

<i>txt</i>	A string. Will be displayed as red.
------------	-------------------------------------

6.4 /home/bryan/TerminalTalk/server/Orator.py File Reference

Classes

- class `server.Orator.Orator`
Stores information about a connected client.

6.5 /home/bryan/TerminalTalk/server/pontification.py File Reference

Functions

- def `server.pontification.server_pontificate` (*verbiage, connections, server, orators*)
Make announcements from server to everyone connected.
- def `server.pontification.pontificate` (*orator, verbiage, connections, server, client, orators*)
Transmit a clients message to all other connected users.

6.5.1 Function Documentation

6.5.1.1 `def server.pontification.pontificate (orator, verbiage, connections, server, client, orators)`

Transmit a clients message to all other connected users.

Parameters

<i>orator</i>	The sending client's Orator object.
<i>verbiage</i>	The message displayed.
<i>connections</i>	A list of all current connections (sockets).
<i>server</i>	The server's socket. Used to ensure message isn't sent to server itself (this would break pipe).
<i>client</i>	The sending client's socket. Used to avoid sending message back to client.
<i>orators</i>	A list of Orator objects for currently connected clients. If a disconnection is discovered while function is being run, it is necessary to remove the object from the list.

6.5.1.2 `def server.pontification.server_pontificate (verbiage, connections, server, orators)`

Make announcements from server to everyone connected.

The message will be formatted (see `font.highlight`) so as to stick out from all other text. Every connected user will see the message.

Parameters

<i>verbiage</i>	The message displayed.
<i>connections</i>	A list of all current connections (sockets).
<i>server</i>	The server's socket. Used to ensure message isn't sent to server itself (this would break pipe).
<i>orators</i>	A list of Orator objects for currently connected clients. If a disconnection is discovered while function is being run, it is necessary to remove the object from the list.

6.6 `/home/bryan/TerminalTalk/TerminalTalk.py` File Reference

Main code for TerminalTalk client.

Variables

- string **TerminalTalk.username** = "Anonymous"
- int **TerminalTalk.buffer_size** = 2
- int **TerminalTalk.server_port** = 7777
- string **TerminalTalk.server_ip** = "192.168.1.77"
- tuple **TerminalTalk.server_address** = (server_ip, server_port)
- **TerminalTalk.megaphone** = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- **TerminalTalk.request** = megaphone.recv(buffer_size)
- list **TerminalTalk.connections** = [megaphone, sys.stdin]
- **TerminalTalk.readables**
- **TerminalTalk.writables**
- **TerminalTalk.errors**
- **TerminalTalk.missive** = socket_i.recv(buffer_size)
- **TerminalTalk.verbiage** = sys.stdin.readline()

6.6.1 Detailed Description

Main code for TerminalTalk client.

Run this file to use as client.

6.7 `/home/bryan/TerminalTalk/TerminalTalk_server.py` File Reference

Main code for TerminalTalk server.

Variables

- int **TerminalTalk_server.port** = 7777
- int **TerminalTalk_server.buffer_size** = 2
- tuple **TerminalTalk_server.server_address** = ("", port)
- list **TerminalTalk_server.connections** = []
- list **TerminalTalk_server.orators** = []
- **TerminalTalk_server.server** = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
- **TerminalTalk_server.readables**
- **TerminalTalk_server.writables**
- **TerminalTalk_server.errors**
- **TerminalTalk_server.file_descriptor**
- **TerminalTalk_server.address**
- **TerminalTalk_server.username** = file_descriptor.recv(buffer_size)
- string **TerminalTalk_server.entrance_message** = username+" has entered."
- int **TerminalTalk_server.orator_index** = 0
- **TerminalTalk_server.verbiage** = socket_i.recv(buffer_size)
- string **TerminalTalk_server.missive** = orators[orator_index].username+" has exited."

6.7.1 Detailed Description

Main code for TerminalTalk server.

Run this file to start server.

