# TerminalTalk

Bryan Harper

# Contents

**1 Namespace Index** 1

**2 Packages** 1

**3 Class Index** 1

**4 Class List** 1

**5 File Index** 1

**6 File List** 1

**7 Namespace Documentation** 2

**8 client Namespace Reference** 2

8.1 Detailed Description . . . . . 2

**9 font Namespace Reference** 2

9.1 Detailed Description . . . . . 2

**10 server Namespace Reference** 2

10.1 Detailed Description . . . . . 2

**11 Class Documentation** 2

**12 font.constants.Colors Class Reference** 2

12.1 Detailed Description . . . . . 3

**13 server.Orator.Orator Class Reference** 3

13.1 Detailed Description . . . . . 3

13.2 Constructor & Destructor Documentation . . . . . 3

13.2.1 __init__(self, telegraph) . . . . . 3

**14 font.constants.Styles Class Reference** 4

14.1 Detailed Description . . . . . 4

**15 File Documentation** 4

**16 /home/bryan/TerminalTalk/client/eavesdrop.py File Reference** **4**

**17 /home/bryan/TerminalTalk/font/constants.py File Reference** **5**

17.1 Detailed Description . . . . . 5

**18 /home/bryan/TerminalTalk/font/functions.py File Reference** **5**

18.1 Detailed Description . . . . . 6

18.2 Function Documentation . . . . . 6

18.2.1 blue(txt) . . . . . 6

18.2.2 bold(txt) . . . . . 6

18.2.3 cyan(txt) . . . . . 6

18.2.4 default(txt) . . . . . 6

18.2.5 green(txt) . . . . . 7

18.2.6 highlight(txt) . . . . . 7

18.2.7 italic(txt) . . . . . 7

18.2.8 magenta(txt) . . . . . 7

18.2.9 red(txt) . . . . . 8

18.2.10 underline(txt) . . . . . 8

18.2.11 yellow(txt) . . . . . 8

**19 /home/bryan/TerminalTalk/server/Orator.py File Reference** **8**

**20 /home/bryan/TerminalTalk/server/pontification.py File Reference** **9**

20.1 Function Documentation . . . . . 9

20.1.1 megaphone(verbiage, connections, ear_trumpet, orators) . . . . . 9

20.1.2 pontificate(orator, verbiage, connections, ear_trumpet, orators) . . . . . 9

**21 /home/bryan/TerminalTalk/TerminalTalk.py File Reference** **10**

21.1 Detailed Description . . . . . 10

**22 /home/bryan/TerminalTalk/TerminalTalk_server.py File Reference** **10**

22.1 Detailed Description . . . . . 10

# 1 Namespace Index

# 2 Packages

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

**client**
**Functions and definitions specific to TerminalTalk client** 2

**font**
**A collection of functions that style fonts in terminal** 2

**server**
**Functions and definitions specific to TerminalTalk server** 2

# 3 Class Index

# 4 Class List

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

**font.constants.Colors**
**Contains constants for ANSI color codes** 2

**server.Orator.Orator**
**Stores information about a connected client** 3

**font.constants.Styles**
**Contains constants for ANSI text style codes** 4

# 5 File Index

# 6 File List

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

**/home/bryan/TerminalTalk/TerminalTalk.py**
**Main code for TerminalTalk client** 10

**/home/bryan/TerminalTalk/TerminalTalk_server.py**
**Main code for TerminalTalk server** 10

**/home/bryan/TerminalTalk/client/eavesdrop.py** 4

**/home/bryan/TerminalTalk/font/constants.py**
**Defines classes contains ANSI constants for formating text** 5

**/home/bryan/TerminalTalk/font/functions.py**
**Defines functions for surronding strings with ANSI codes for formating text** 5

/home/bryan/TerminalTalk/server/Orator.py 8

/home/bryan/TerminalTalk/server/pontification.py 9

# 7 Namespace Documentation

# 8 client Namespace Reference

Functions and definitions specific to TerminalTalk client.

## 8.1 Detailed Description

Functions and definitions specific to TerminalTalk client.

# 9 font Namespace Reference

A collection of functions that style fonts in terminal.

## 9.1 Detailed Description

A collection of functions that style fonts in terminal.

Each function flanks the inputed text with ANSI codes.

# 10 server Namespace Reference

Functions and definitions specific to TerminalTalk server.

## 10.1 Detailed Description

Functions and definitions specific to TerminalTalk server.

# 11 Class Documentation

# 12 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'

## 12.1 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

# 13 server.Orator.Orator Class Reference

Stores information about a connected client.

### Public Member Functions

- def __init__ (self, telegraph)
  *Constructor for Orator class.*

### Public Attributes

- **telegraph**
- **moniker**
- **color**

## 13.1 Detailed Description

Stores information about a connected client.

An Orator objects is used to represent a client. Each client has a moniker and color associated with them. The moniker is provided by the client. The color is assigned randomly each time the client connects.

## 13.2 Constructor & Destructor Documentation

### 13.2.1 def server.Orator.Orator.__init__ ( self, telegraph )

Constructor for Orator class.

Parameters

| | |
|---|---|
| *telegraph* | The socket used to connect with client. |

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

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

# 14 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'

## 14.1 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

# 15 File Documentation

# 16 /home/bryan/TerminalTalk/client/eavesdrop.py File Reference

## Functions

- def client.eavesdrop.eavesdrop ()

  *Monitors for user input from terminal.*

# 17 /home/bryan/TerminalTalk/font/constants.py File Reference

Defines classes contains ANSI constants for formating text.

## Classes

- class font.constants.Styles
  *Contains constants for ANSI text style codes.*
- class font.constants.Colors
  *Contains constants for ANSI color codes.*

## 17.1 Detailed Description

Defines classes contains ANSI constants for formating text.

# 18 /home/bryan/TerminalTalk/font/functions.py File Reference

Defines functions for surronding strings with ANSI codes for formating 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.*

## 18.1 Detailed Description

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

## 18.2 Function Documentation

### 18.2.1 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

| | |
|---|---|
| *txt* | A string. Will be displayed as blue. |

### 18.2.2 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

| | |
|---|---|
| *txt* | A string. Will be made bold. |

### 18.2.3 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be displayed as cyan. |

### 18.2.4 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

| | |
|---|---|
| *txt* | A string. Will be formatted as default. |

### 18.2.5 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be displayed as green. |

### 18.2.6 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be highlighted. |

### 18.2.7 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

| | |
|---|---|
| *txt* | A string. Will be displayed as italic. |

### 18.2.8 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be displayed as magenta. |

#### 18.2.9 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be displayed as red. |

#### 18.2.10 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

| | |
|---|---|
| *txt* | A string. Will be underlined. |

#### 18.2.11 def font.functions.yellow ( txt )

Makes inputed string red.

Returns with inputed 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 forground, black background).

Parameters

| | |
|---|---|
| *txt* | A string. Will be displayed as red. |

## 19 /home/bryan/TerminalTalk/server/Orator.py File Reference

### Classes

- class server.Orator.Orator

  *Stores information about a connected client.*

# 20 /home/bryan/TerminalTalk/server/pontification.py File Reference

## Functions

- def server.pontification.megaphone (verbiage, connections, ear_trumpet, orators)

  *Make announcements from server to everyone connected.*
- def server.pontification.pontificate (orator, verbiage, connections, ear_trumpet, orators)

  *Transmit a clients message to all other connected users.*

## 20.1 Function Documentation

### 20.1.1 def server.pontification.megaphone ( verbiage, connections, ear_trumpet, 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

| | |
|---|---|
| *verbiage* | The message displayed. |
| *connections* | A list of all current connections (sockets). |
| *ear_trumpet* | The server's socket. Used to ensure message isn't sent to server itself (this would break pipe). |
| *orators* | 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. |

### 20.1.2 def server.pontification.pontificate ( orator, verbiage, connections, ear_trumpet, orators )

Transmit a clients message to all other connected users.

Parameters

| | |
|---|---|
| *orator* | The sending client's Orator object. |
| *verbiage* | The message displayed. |
| *connections* | A list of all current connections (sockets). |
| *ear_trumpet* | The server's socket. Used to ensure message isn't sent to server itself (this would break pipe). |
| *orators* | 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. |

# 21 /home/bryan/TerminalTalk/TerminalTalk.py File Reference

Main code for TerminalTalk client.

## Variables

- string **TerminalTalk.moniker** = "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** = telegraph_i.recv(buffer_size)
- **TerminalTalk.verbiage** = sys.stdin.readline()

## 21.1 Detailed Description

Main code for TerminalTalk client.

Run this file to use as client.

# 22 /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.ear_trumpet** = socket.socket(socket.AF_INET, socket.SOCK_STR↩ EAM)
- **TerminalTalk_server.readables**
- **TerminalTalk_server.writables**
- **TerminalTalk_server.errors**
- **TerminalTalk_server.file_descriptor**
- **TerminalTalk_server.address**
- **TerminalTalk_server.moniker** = file_descriptor.recv(buffer_size)
- string **TerminalTalk_server.entrance_message** = moniker+" has entered."
- int **TerminalTalk_server.orator_index** = 0
- **TerminalTalk_server.verbiage** = telegraph_i.recv(buffer_size)
- string **TerminalTalk_server.missive** = orators[orator_index].moniker+" has exited."

## 22.1 Detailed Description

Main code for TerminalTalk server.

Run this file to start server.