cliflect Documentation

Release latest

Dec 28, 2021

CONTENTS

cliflect supports

- text coloring and decoration
- handle multiple arguments
- input decoration
- allows to create your own home
- etc…

CHAPTER ONE

CONTENTS

1.1 installing cliflect

use the following command to install cliflect

```
$ pip install cliflect
```

pypi page

1.2 text

1.2.1 hello world

a simple program to print hello world in terminal

```
import cliflect
console=cliflect.console()
@console.cmd('name')
def out(name):
        if name:
            console.echo(f"welcome, {name}")
        else:
            console.echo('please enter your name')
console.run()
```

i saved it as hello.py

at console

```
$ python hello.py

no options parsed
usage hello.py [option] args

options:
        name
```

```
$ python hello.py test

```

(continues on next page)

(continued from previous page)

```
invalid option --> test
usage hello.py [option] args

options:
        name
```

```
$ python hello.py name

please enter your name
```

```
$ python hello.py name guest

welcome, guest
```

1.2.2 text colouring

text colouring

usage

```
import cliflect
console=cliflect.console()
@console.cmd('out')
def out(nme):
         console.echo(nme, color="white", bright=True)#bright==bold
console.run()
```

active colors

- black
- red
- green
- yellow
- blue
- magenta
- cyan
- white

note: if bright is True, the text will be bold

1.2.3 text decoration

usage :

```
import cliflect
console=cliflect.console()
@console.cmd('out')
def out(nme):
         console.echo(nme, decor=None)#None means normal
console.run()
```

active decoration types

- normal, use normal as decor=None
- bold
- underline

1.3 input

1.3.1 hello to input

cliflect also supports simple input handling

usage:

```
import cliflect
console=cliflect.console()
@console.cmd('in')
def inp(arg):
    i=console.read('your name: ')
    console.echo(f'welcome, {i}')
console.run()
```

other than type, console.read attributes are same as console.echo

output: the input enterd in ‘your name: ‘ will be printed

1.3.2 input colouring

coloured input in cliflect

coloured input

```
import cliflect
console=cliflect.console()
@console.cmd('in')
def inp(text):
        console.read('hi: ', color='red')
console.run()
```

attributes are same of echo

output: the ‘hi: ‘ will be shown in color white or text color set for terminal and the text after ‘hi: ‘(input) will be in red, you can decorate it

1.3.3 input decoration

usage :

```
import cliflect
console=cliflect.console()
@console.cmd('in')
def inp(arg):
         i=console.read("name: ", decor=None)#None means normal
         console.echo(i)
console.run()
```

active decoration types

- normal, use normal as decor=None
- bold
- underline

1.3.4 password

usage :

```
import cliflect
console=cliflect.console()
@console.cmd('in')
def inp(arg):
         i=console.read("name: ", type="password")#default type is text
         console.echo(i)
console.run()
```

the password will only show blank in input whether you entered or not

1.4 multiple option

here is more options in one function

```
import cliflect
console=cliflect.console()
@console.cmd('-h', 'help', 'h')
def help_(arg):
        console.echo('you entered help option')
console.run()
```

just like ‘or’

in command line

```
$ python hello.py -h
you entered help option

$ python hello.py help
you entered help option

$python hello.py h
you entered help option
```

1.5 creating a home

how to create a home that will raise when an invalid option or nothing is entered

```
 import cliflect
 console=cliflect.console()
 @console.home
 def error(erropt):
         if erropt:
                 console.echo(f'{erropt} is an invalid option', color='red')
                 console.echo('''usage hello.py [option] arg...
options:
-h : help''')
         else:
                 console.echo('no option entered', color='red')
                 console.echo('use -h for help')
 @console.cmd('-h')
 def help_(arg):
         console.echo('you entered help option')
 console.run()
```

at console

```
$ python hello.py
no option entered
use -h for help
```

```
$ python hello.py test
test is an invalid option
usage hello.py [option] arg...
options:
        -h -- help
```

```
$ python hello.py -h
you entered help option
```