

# learning python3 - framelt.py - command line tool to frame text on the console

These are my first steps in python programming language. The result is a little cli tool/function/library (whatever you call it ) that will encapsulate any given text within a ASCII Character frame. Either as command line arguments with dynamic resizing of the frame, or within python as a function with more freedom to define width and look of the frame.

See these Demos:

```
python frameIt.py "Hello World!" "How are you?"
```

```
#####
# Hello World! #
# How are you? #
#####
```

of when called without any CLI Arguments it runs the internal demo function call:

```
python frameIt.py
```

```
DEMO of the frameIt() function:
```

```
#####
#
# Hello World!
# Isn't this Cool,eh?
# python is nice sometimes.
# And easy to learn too for beginners.
#
# This was just a little training for my
# first steps in python3. free for
# personal, non-commercial use only!
#
# in 2015 by Axel Werner
# axel.werner.1973@gmail.com
#
#####
```

## The code

[framelt.py](#)

```
...
```

```
This was just a little python3 experiment and training
```

for my first steps in learning python3.

Code is free for educational, personal use as long it is used for non-commercial purposes.

written in python3 by by Axel Werner in 2015  
eMail: axel.werner.1973@gmail.com

...

```
import sys
```

```
def hr(fillCharacter='#', width=79):  
    return fillCharacter * width
```

```
def frameIt(textList,width=79,fillCharacter='*'):  
    '''  
    frameIt() takes three arguments, a list of strings (lines of text),  
    an integer which determines the width of your  
terminal/console/frame,  
    and a string of a single character used to "frame" the  
text into.  
    '''
```

```
    frame= hr(fillCharacter, width) + "\n" + \  
            enclose(textList, fillCharacter, width) + \  
            hr(fillCharacter, width)+ "\n"  
    return frame
```

```
def frameItAuto(textList,fillCharacter='*'):  
    '''  
    frameIt() takes two arguments, a list of strings (lines of text),  
    and a string of a single character used to "frame" the  
text into.  
    '''
```

*#peak detector - determine max length all lines given*

```
maxLineLen=0
```

```
for line in textList:  
    lenOfLine=len(line)  
    if lenOfLine >= maxLineLen:  
        maxLineLen=lenOfLine
```

*width=maxLineLen+4 # add 4 framing charakters to maxLineLen*

```
frame= hr(fillCharacter, width) + "\n" + \  
        enclose(textList, fillCharacter, width) + \  
        hr(fillCharacter, width)+ "\n"  
return frame
```

```
def enclose(textList, frameCharacter, width):  
    newTextBlock=""  
    for line in textList:
```

```
        lineLength=len(line)
        fillerCharakter=" "
        filler=fillerCharakter*(width-
lineLength-2-2*len(frameCharacter))
        newTextBlock=newTextBlock+ frameCharacter + " " + line + filler
+ " " + frameCharacter + "\n"
        return newTextBlock

def main(argv):

    if not argv:
        print("DEMO of the frameIt() function: \n\n")

        textList=[
            "",
            "Hello World!",
            "Isn't this Cool,eh?",
            "python is nice sometimes.",
            "And easy to learn too for beginners.",
            "",
            "This was just a little training for my ",
            "first steps in python3. free for",
            "personal, non-commercial use only!",
            "",
            "in 2015 by Axel Werner ",
            "axel.werner.1973@gmail.com",
            ""
        ]

        print ( frameIt(textList, 50, '#') )
        exit(0)
    else:
        print ( frameItAuto(argv, '#') )
        exit(0)

if __name__ == "__main__":
    main(sys.argv[1:])
```

— Axel Werner 2015-08-03 16:56

python3, python, programming, computer, science, learning, software, development

From:  
<https://awerner.myhome-server.de/> - Axel Werner's OPEN SOURCE Knowledge Base

Permanent link:  
<https://awerner.myhome-server.de/doku.php?id=it-artikel:python:learning-python3-frameit.py-command-line-tool-to-frame-text-on-the-console>

Last update: 2022-08-31 12:30

