

## Intro to Python 2018-10-16 99 Bottles of Beer

## The Song Lyrics

It starts with this:

99 bottles of beer on the wall,  
99 bottles of beer.  
Take one down and pass it around,  
98 bottles of beer on the wall.

98 bottles of beer on the wall,  
98 bottles of beer.  
Take one down and pass it around,  
97 bottles of beer on the wall.

97 bottles of beer on the wall,  
97 bottles of beer.  
Take one down and pass it around,  
96 bottles of beer on the wall.  
...

## How Does the Song End?

3 bottles of beer on the wall,  
3 bottles of beer.  
Take one down and pass it around,  
2 bottles of beer on the wall.

2 bottles of beer on the wall,  
2 bottles of beer.  
Take one down and pass it around,  
1 bottle of beer on the wall.

1 bottle of beer on the wall,  
1 bottle of beer.  
Take one down and pass it around,  
No more bottles of beer on the wall.

No more bottles of beer on the wall,  
No more bottles of beer.  
Go to the store and buy some more,  
99 bottles of beer on the wall.

Some Questions!

How many total verses are there?  
100

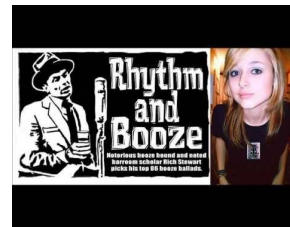
How many verses are similar?  
97 (verses 99 bottles through 3  
bottles)

How many verses are different?

3 (2 bottles, 1 bottle, 0 bottles)

What is the general structure of the  
problem?

## All 100 Verses!



<https://youtu.be/Z7bmyjxluVY>

## [www.99-bottles-of-beer.net](http://www.99-bottles-of-beer.net)

- This website has programs in over 1500 languages that produce the lyrics to 99 Bottles of Beer
- <http://www.99-bottles-of-beer.net/>
- And for each language there may be 5-10 versions

## Why 99 Bottles of Beer?

- Why would somebody collect the 99 bottles song in almost every known programming language?
- It is an elegant way to demonstrate all three structures that make a language Turing-complete
  - Sequential Execution
  - Iteration (looping)
  - Selection (branching)
- Multiple versions show different styles and/or features for any given language

## Using Stepwise Refinement

- Model the solution in English, breaking down larger problems into smaller problems that will be addressed by a single function
- Translate the English into pseudo-code or code stubs
- Translate to Python

```
def StandardVerse(n):
    print ("This produces the verse for",n,"bottles of beer on the wall.")
    print ("")

def TwoBottles():
    print ("This produces the verse for 2 bottles.")
    print ("")
def OneBottle():
    print ("This produces the verse for 1 bottle.")
    print ("")

def NoBottles():
    print ("This produces the verse for no bottles.")
    print ("")

for n in range(99,2,-1):
    StandardVerse(n)
TwoBottles()
OneBottle()
NoBottles()
```

```
def StandardVerse(n):
    print (n,"bottles of beer on the wall,")
    print (n,"bottles of beer.")
    print ("Take one down and pass it around,")
    print (n-1,"bottles of beer on the wall.")
    print ("")

def TwoBottles():
    print ("2 bottles of beer on the wall,")
    print ("2 bottles of beer.")
    print ("Take one down and pass it around,")
    print ("1 bottle of beer on the wall.")
    print ("")

def OneBottle():
    print ("1 bottle of beer on the wall,")
    print ("1 bottle of beer.")
    print ("Take one down and pass it around,")
    print ("no more bottles of beer on the wall.")
    print ("")
```

```
def NoBottles():
    print ("No more bottles of beer on the wall,")
    print ("no more bottles of beer.")
    print ("Go to the store and buy some more,")
    print ("99 bottles of beer on the wall.")
    print ("")

for n in range(99,2,-1):
    StandardVerse(n)
TwoBottles()
OneBottle()
NoBottles()
```

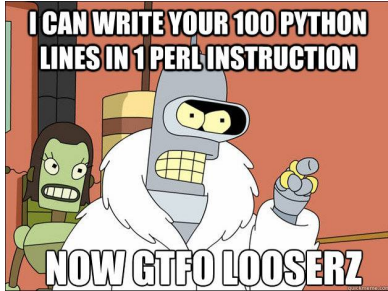
## Top-Rated at 99 Bottles

```
for quant in range(99, 0, -1):
    if quant > 1:
        print (quant, "bottles of beer on the wall,", quant,
              "bottles of beer.")
        if quant > 2:
            suffix = str(quant - 1) + " bottles of beer on the
            wall."
        else:
            suffix = "1 bottle of beer on the wall."
        elif quant == 1:
            print ("1 bottle of beer on the wall, 1 bottle of
            beer.")
            suffix = "no more beer on the wall!"
        print ("Take one down, pass it around,", suffix)
        print ("--")
```

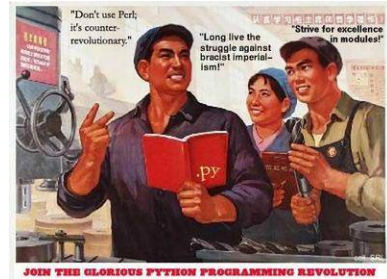
## Sidebar – The Language Wars

- Some people get hung up on the idea that language A is "better" than Language B
- A bit silly because the question is somewhat meaningless unless accompanied by an answer to the question "For what?"
- But it does at least lead to some humour at times ...

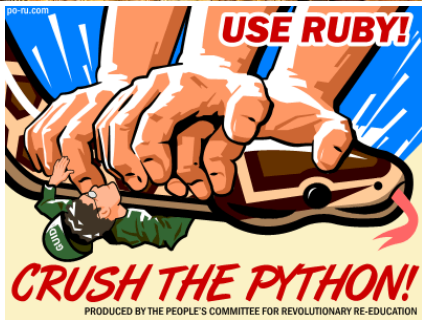
### Perl is Better Than Python



### Python is Better than Perl



### Ruby is Better than Python



### Try this in IDLE

```
import this
```