Why Does My Computer Do That? Intro to Coding with Python– Documentation

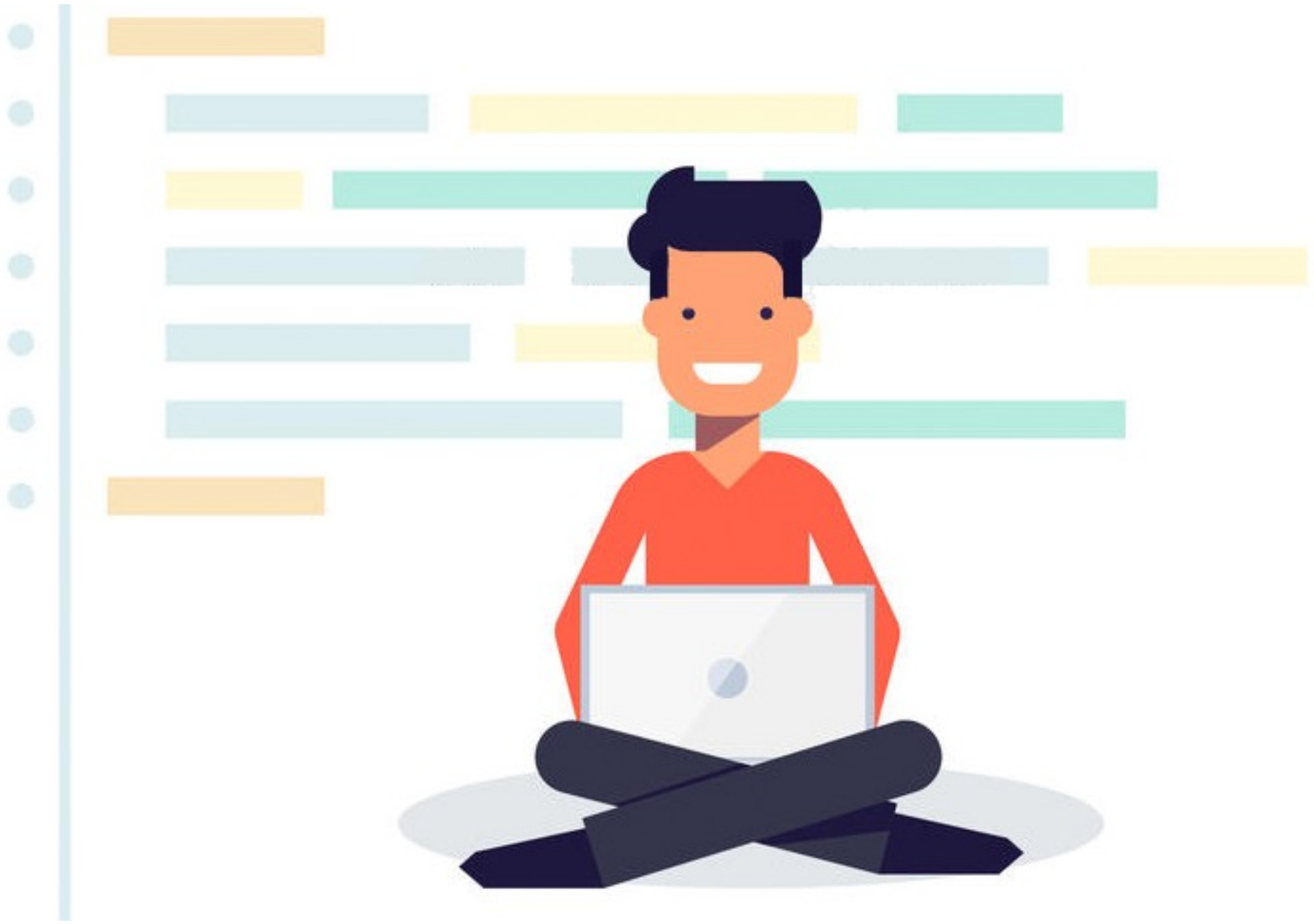
Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

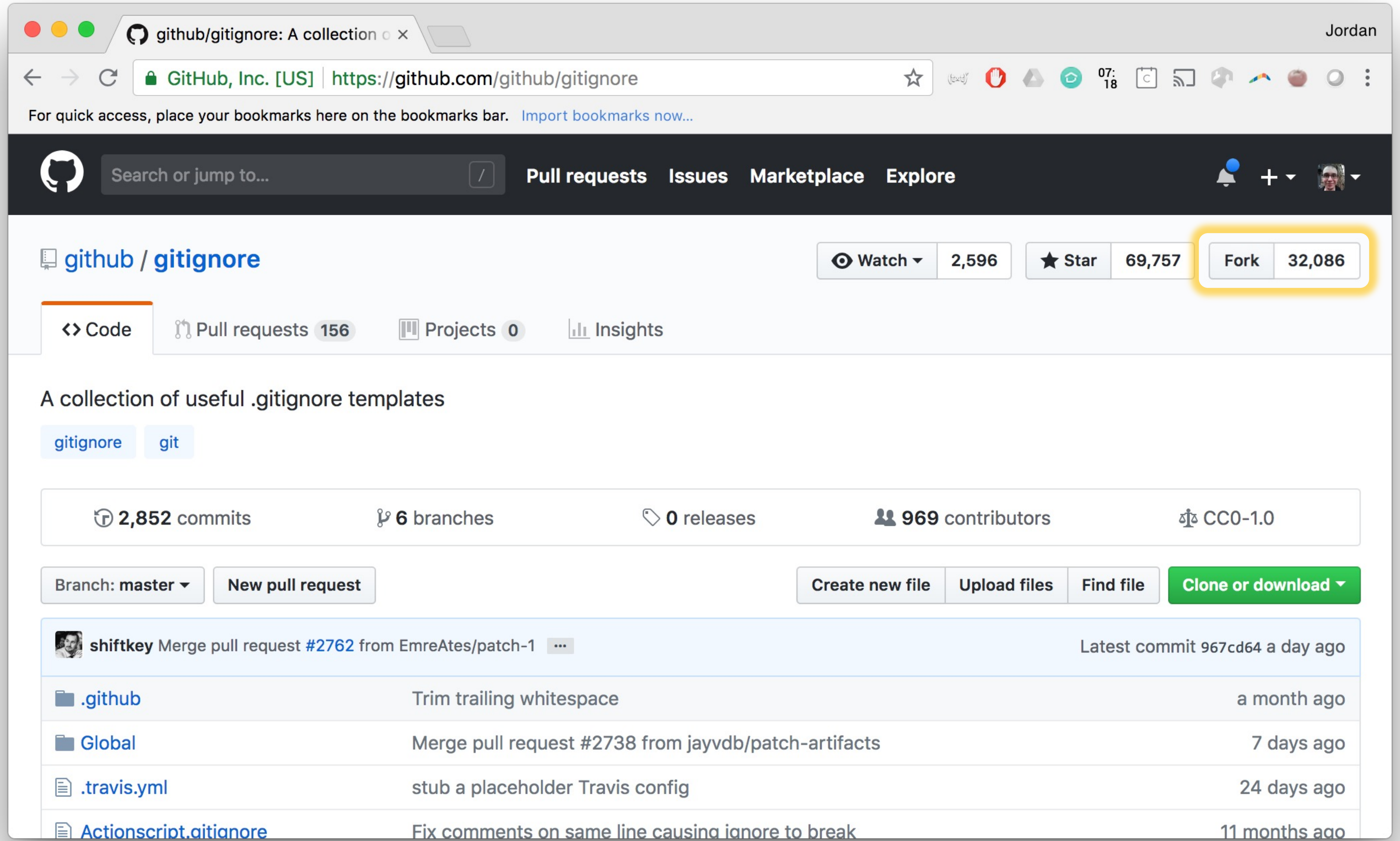
- Tracing code
- Documenting code

In the beginning…



Now…

Eventually…


github/gitignore: A collection o ×
Jordan
GitHub, Inc. [US] https://github.com/github/gitignore
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now…
Search or jump to…
Pull requests
Issues
Marketplace
Explore

github / gitignore

Watch 2,596 | Star 69,757 | Fork 32,086

Code | Pull requests 156 | Projects 0 | Insights

A collection of useful .gitignore templates

gitignore git

2,852 commits | 6 branches | 0 releases | 969 contributors | CC0-1.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

shiftkey Merge pull request #2762 from EmreAtes/patch-1 … Latest commit 967cd64 a day ago

.github	Trim trailing whitespace	a month ago
Global	Merge pull request #2738 from jayvdb/patch-artifacts	7 days ago
.travis.yml	stub a placeholder Travis config	24 days ago
Actionscript.gitignore	Fix comments on same line causing ignore to break	11 months ago

The point

- **Other people** need to be able to understand your code
- **Future you** needs to be able to understand your code

The point

- **Other people** need to be able to understand your code
- **Future you** needs to be able to understand your code

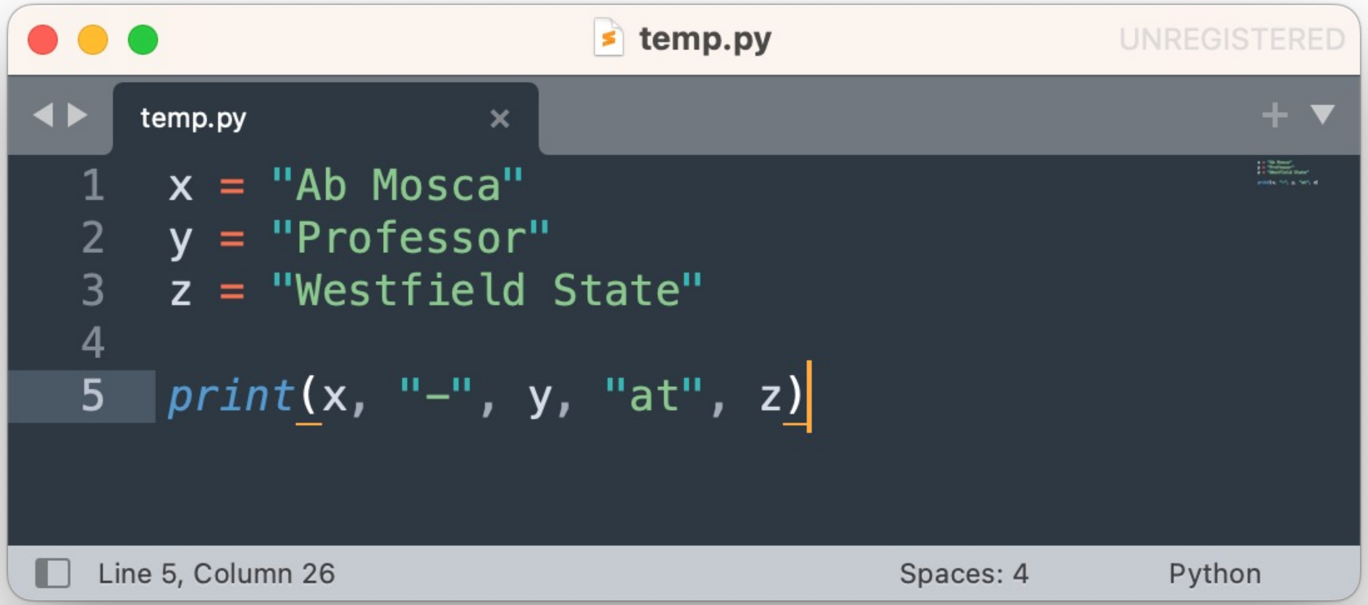
… but how?

The point

- **Other people** need to be able to understand your code
- **Future you** needs to be able to understand your code
- **Document it**

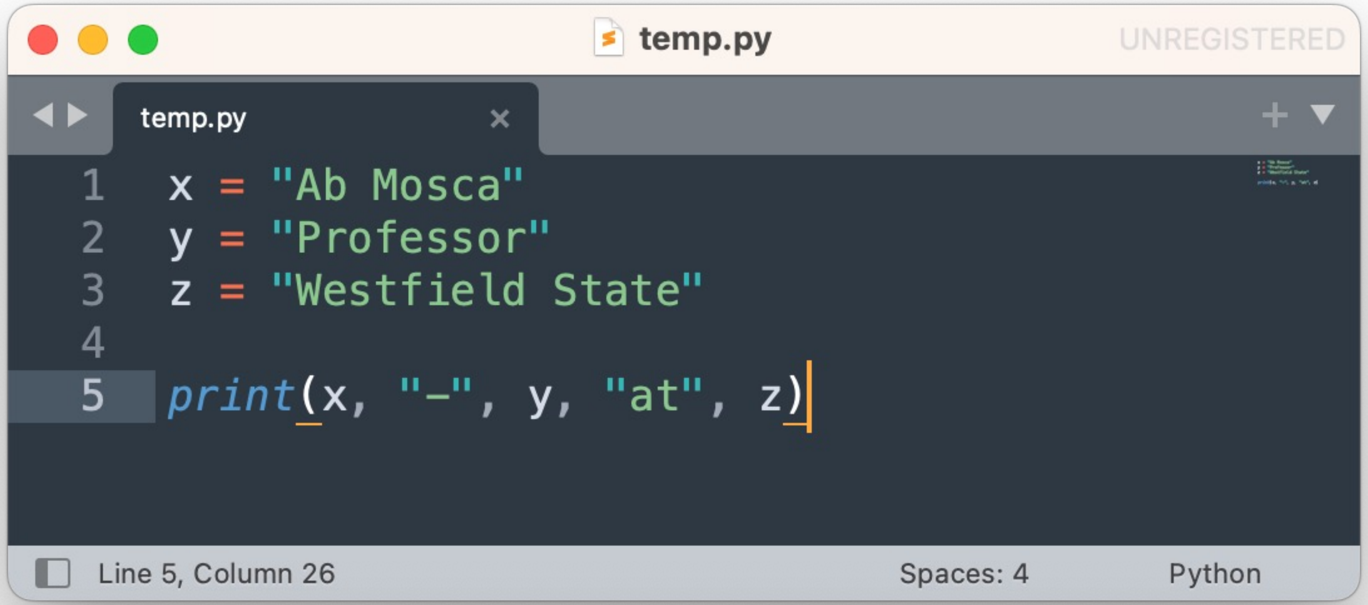
… but how?

Step 1: meaningful **nouns** for variables



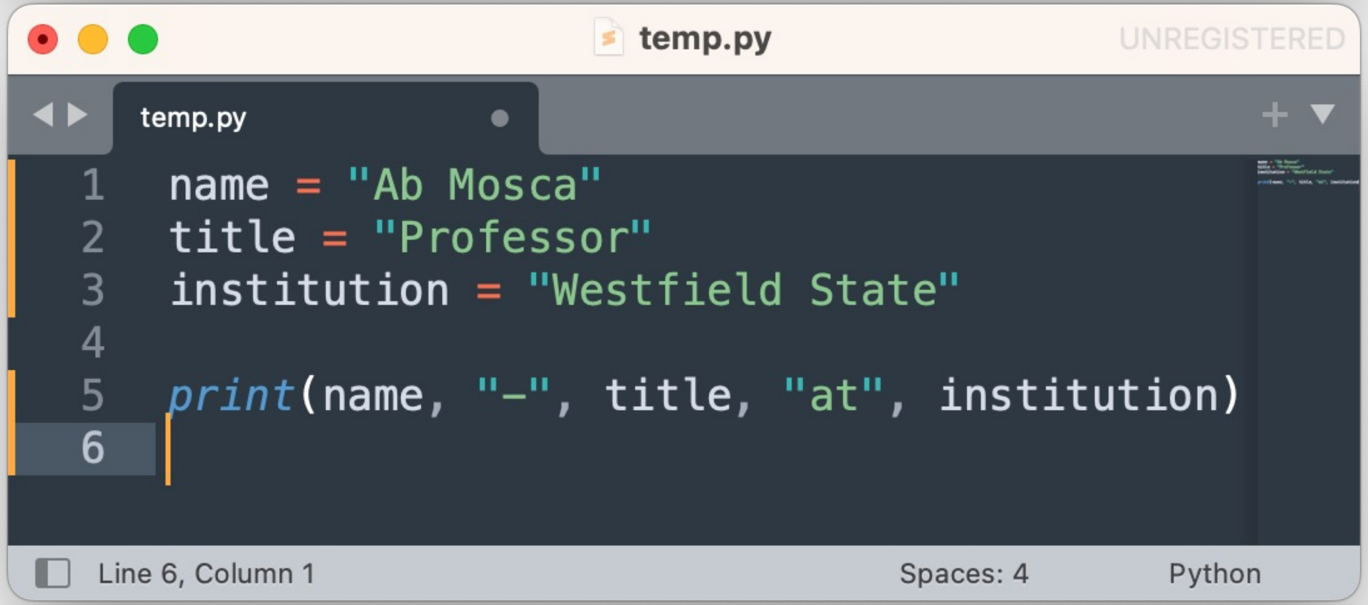
```python
x = "Ab Mosca"
y = "Professor"
z = "Westfield State"

print(x, "-", y, "at", z)
```

Step 1: meaningful **nouns** for variables


temp.py UNREGISTERED

```python
x = "Ab Mosca"
y = "Professor"
z = "Westfield State"

print(x, "-", y, "at", z)
```

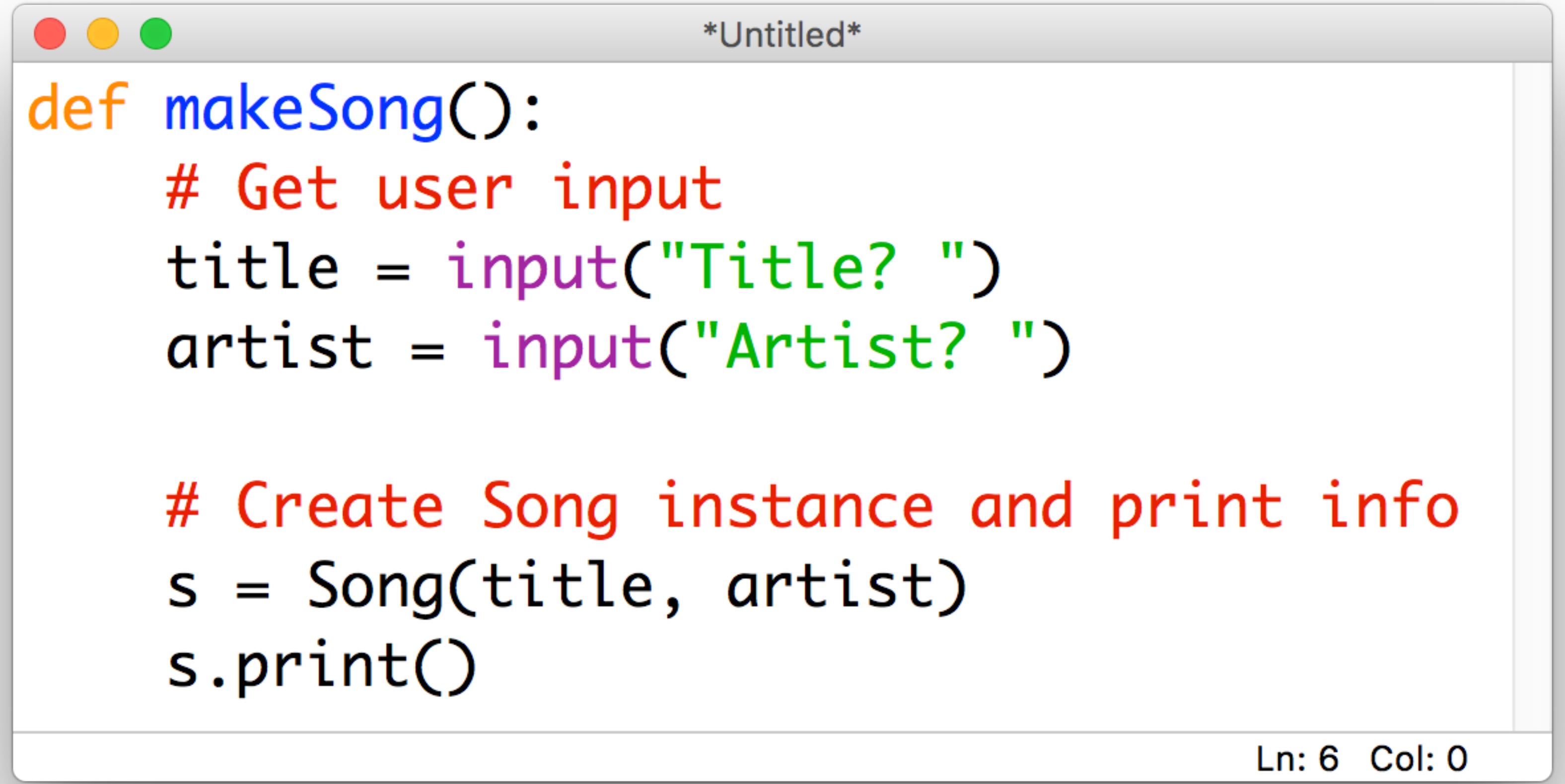
Line 5, Column 26 Spaces: 4 Python


temp.py UNREGISTERED

```python
name = "Ab Mosca"
title = "Professor"
institution = "Westfield State"

print(name, "-", title, "at", institution)

```

Line 6, Column 1 Spaces: 4 Python

Step 2: lots of comments

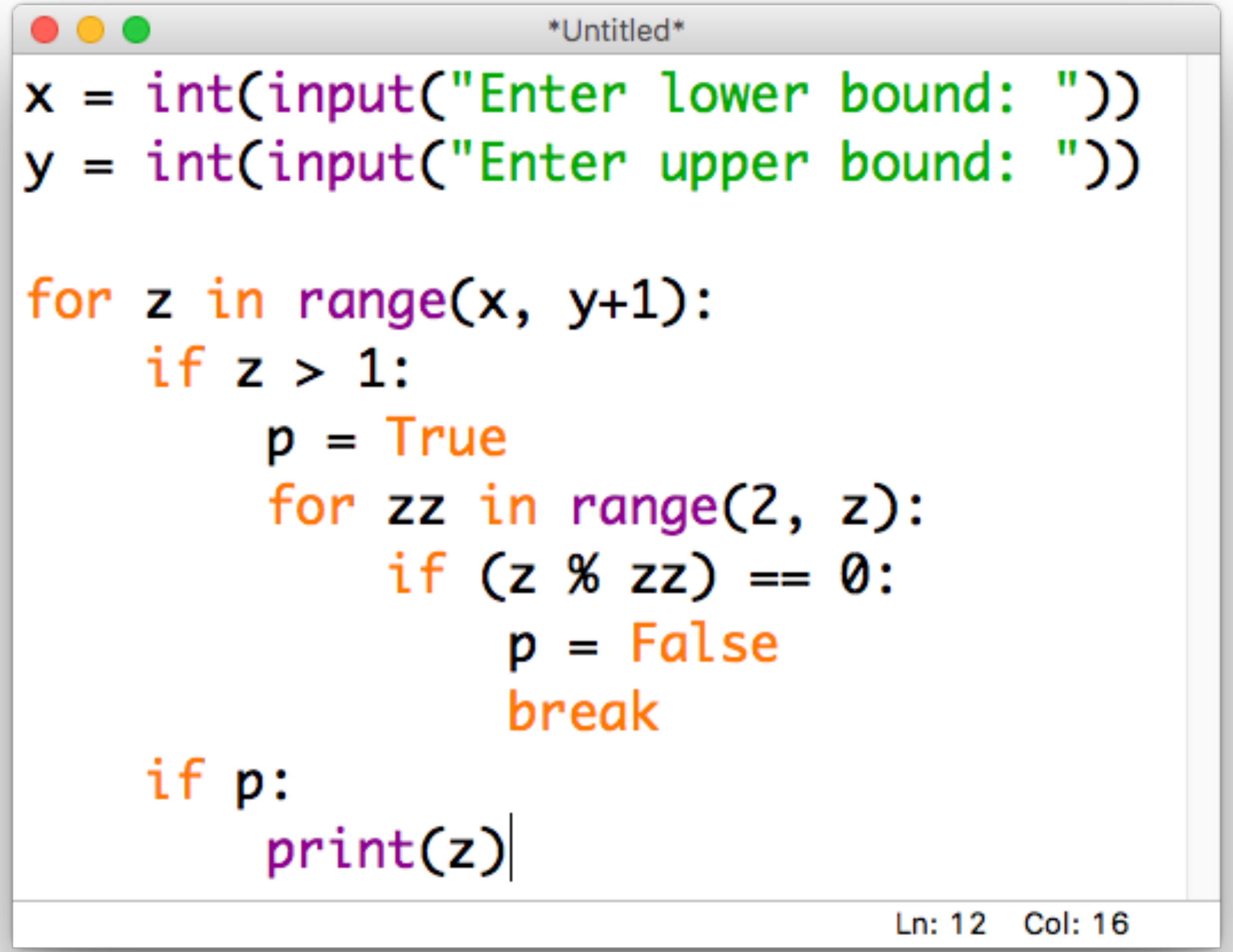


```
def makeSong():
    # Get user input
    title = input("Title? ")
    artist = input("Artist? ")

    # Create Song instance and print info
    s = Song(title, artist)
    s.print()
```

A useful technique: code tracing

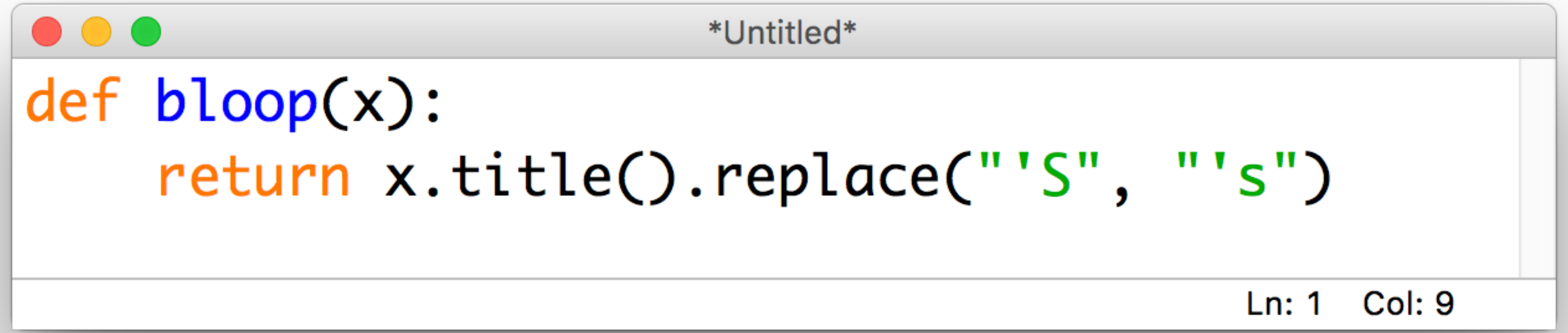
- **Given**: a very short, poorly-documented program
- **Your goal**: try to figure out what it's doing
- **Recommendations**:
  - walk through the program step-by-step ("**trace its execution**") using the **whiteboard or paper** instead of running lines
  - once you understand what's happening, then rewrite it using **informative variable names** and **comments**

Example



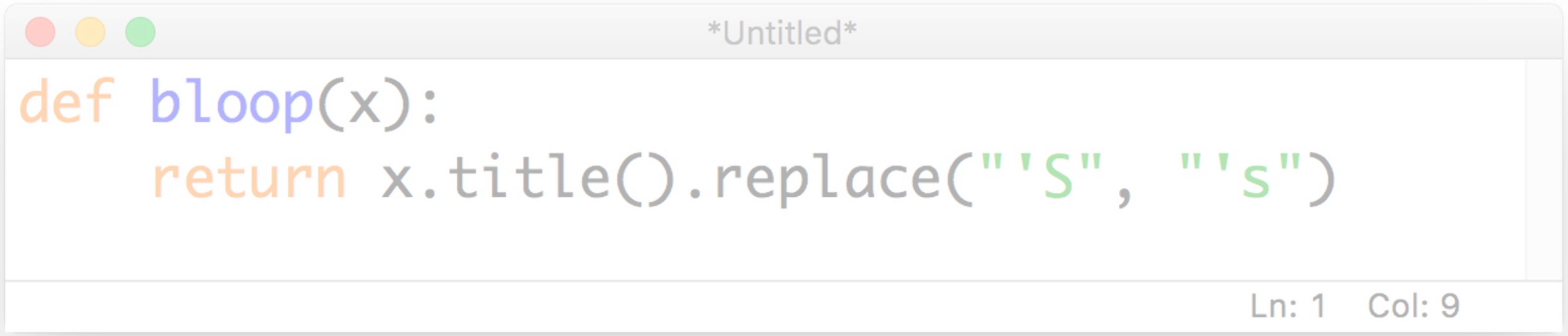
```
x = int(input("Enter lower bound: "))
y = int(input("Enter upper bound: "))

for z in range(x, y+1):
    if z > 1:
        p = True
        for zz in range(2, z):
            if (z % zz) == 0:
                p = False
                break
    if p:
        print(z)
```

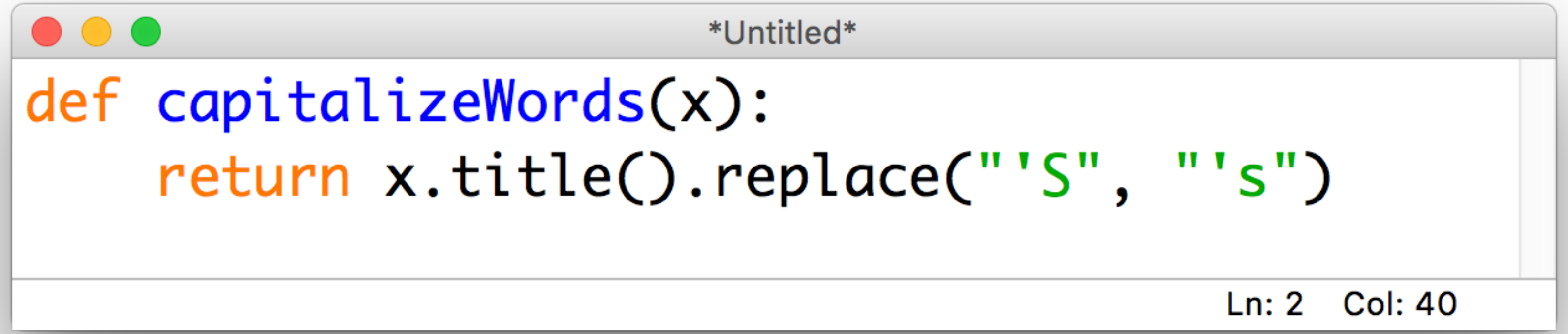
Step 3*: describe the **action** for functions



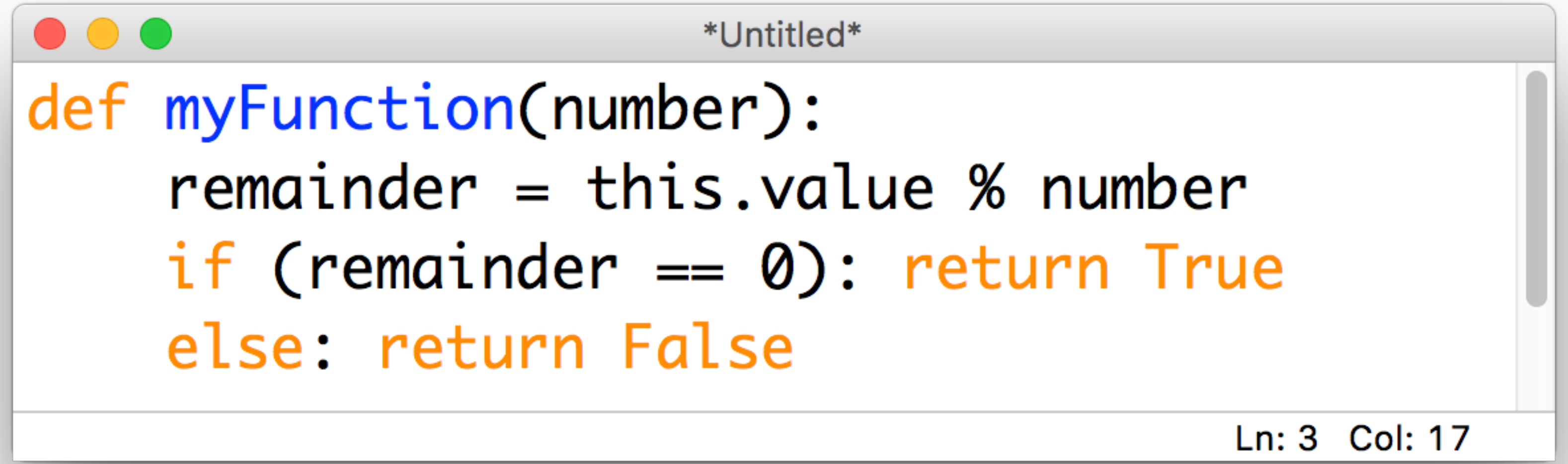
```python
def bloop(x):
    return x.title().replace("'S", "'s")
```

Step 3*: describe the **action** for functions


```
def bloop(x):
    return x.title().replace("'S", "'s")
```
Ln: 1 Col: 9

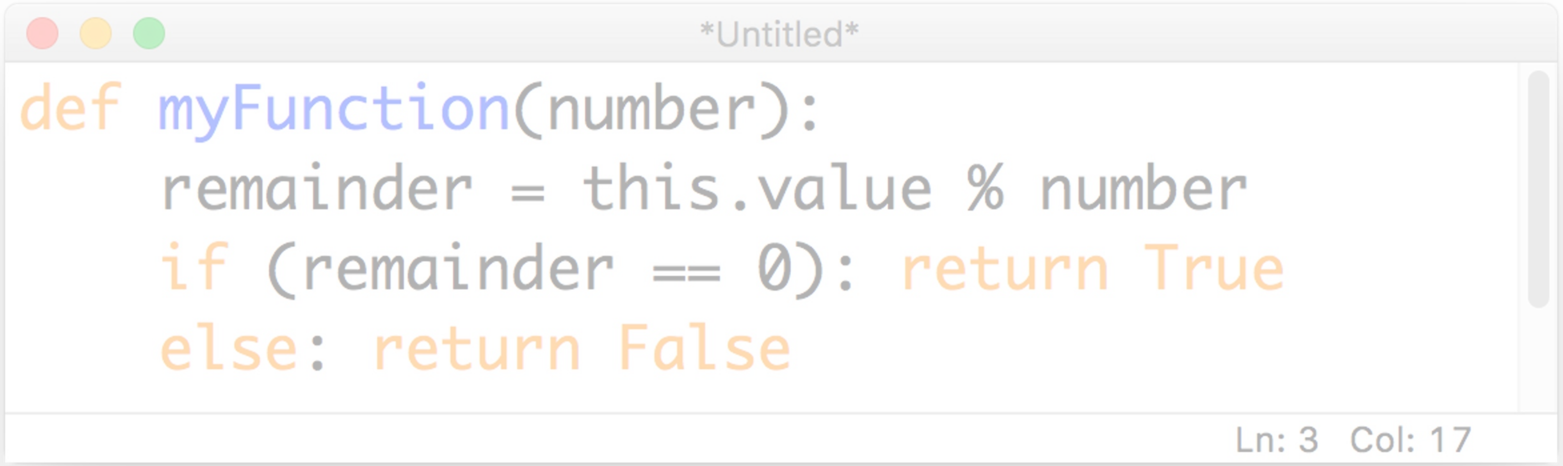

```
def capitalizeWords(x):
    return x.title().replace("'S", "'s")
```
Ln: 2 Col: 40

Step 3*: describe the **action** for functions



```
def myFunction(number):
    remainder = this.value % number
    if (remainder == 0): return True
    else: return False
```

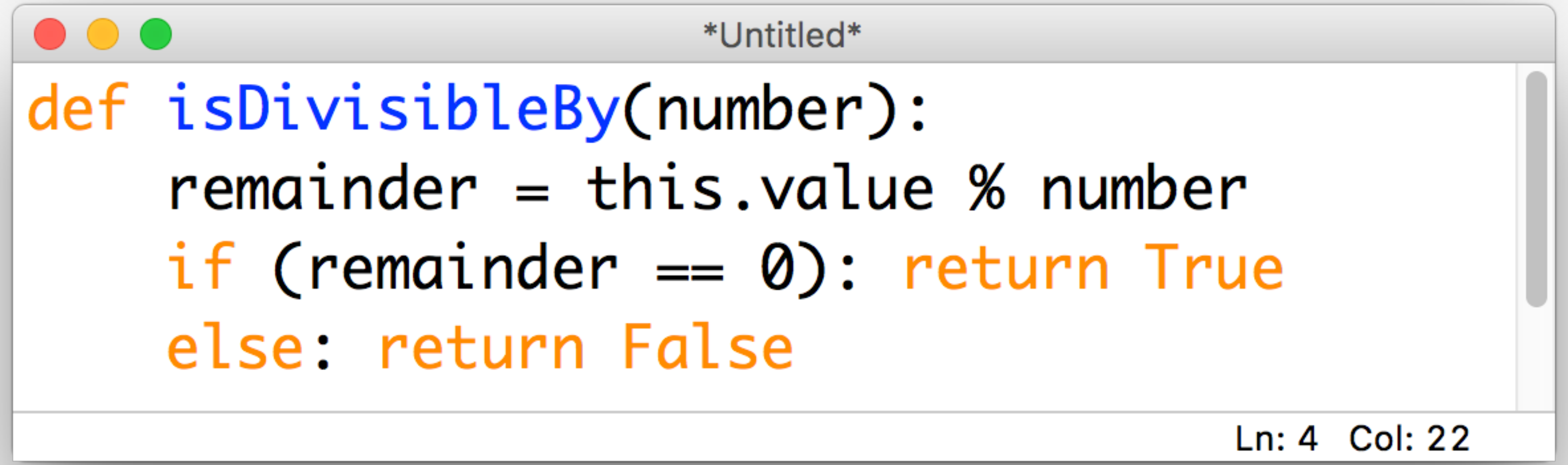
Step 3*: describe the **action** for functions



Untitled

```
def myFunction(number):
    remainder = this.value % number
    if (remainder == 0): return True
    else: return False
```

Ln: 3 Col: 17

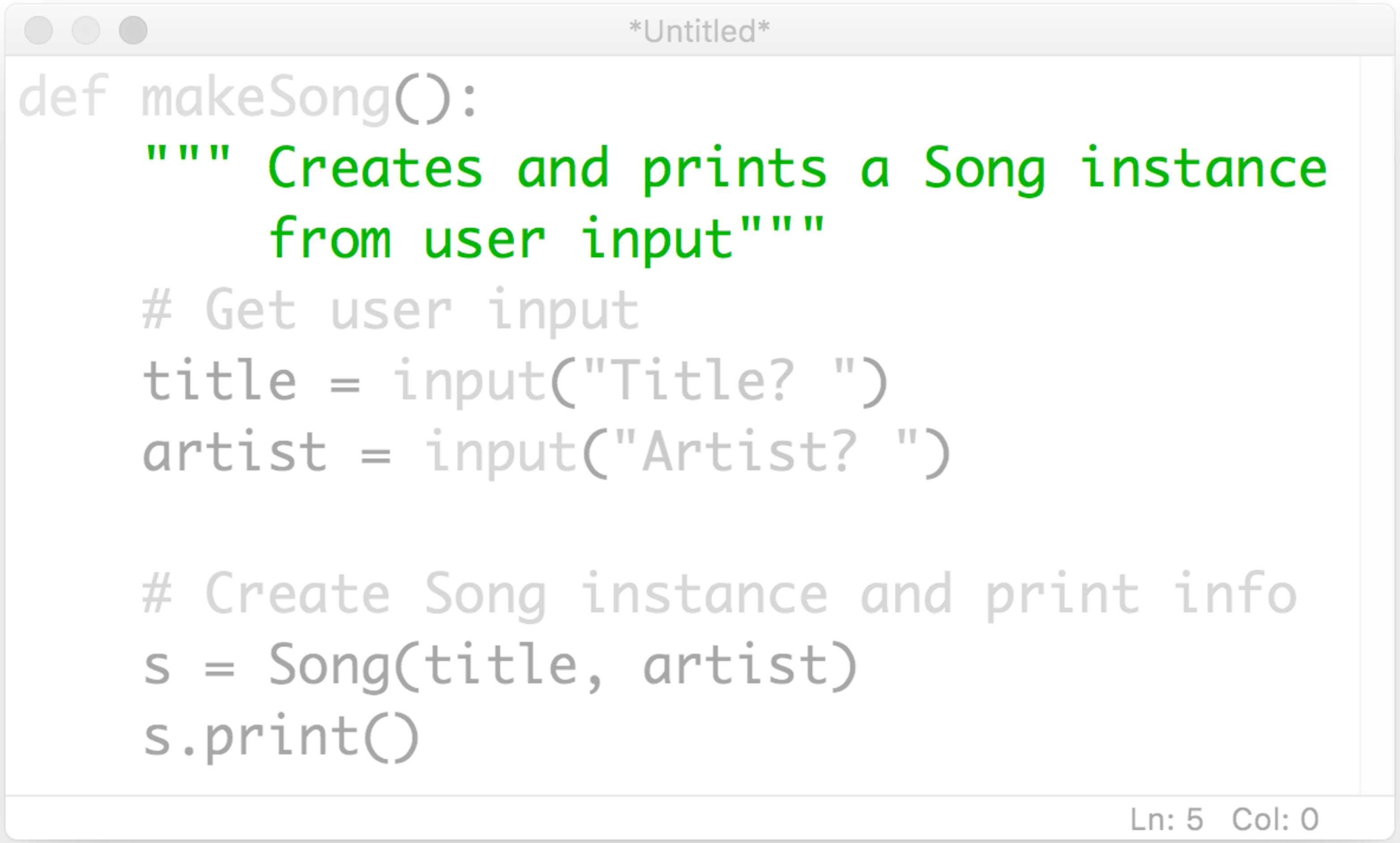


Untitled

```
def isDivisibleBy(number):
    remainder = this.value % number
    if (remainder == 0): return True
    else: return False
```

Ln: 4 Col: 22

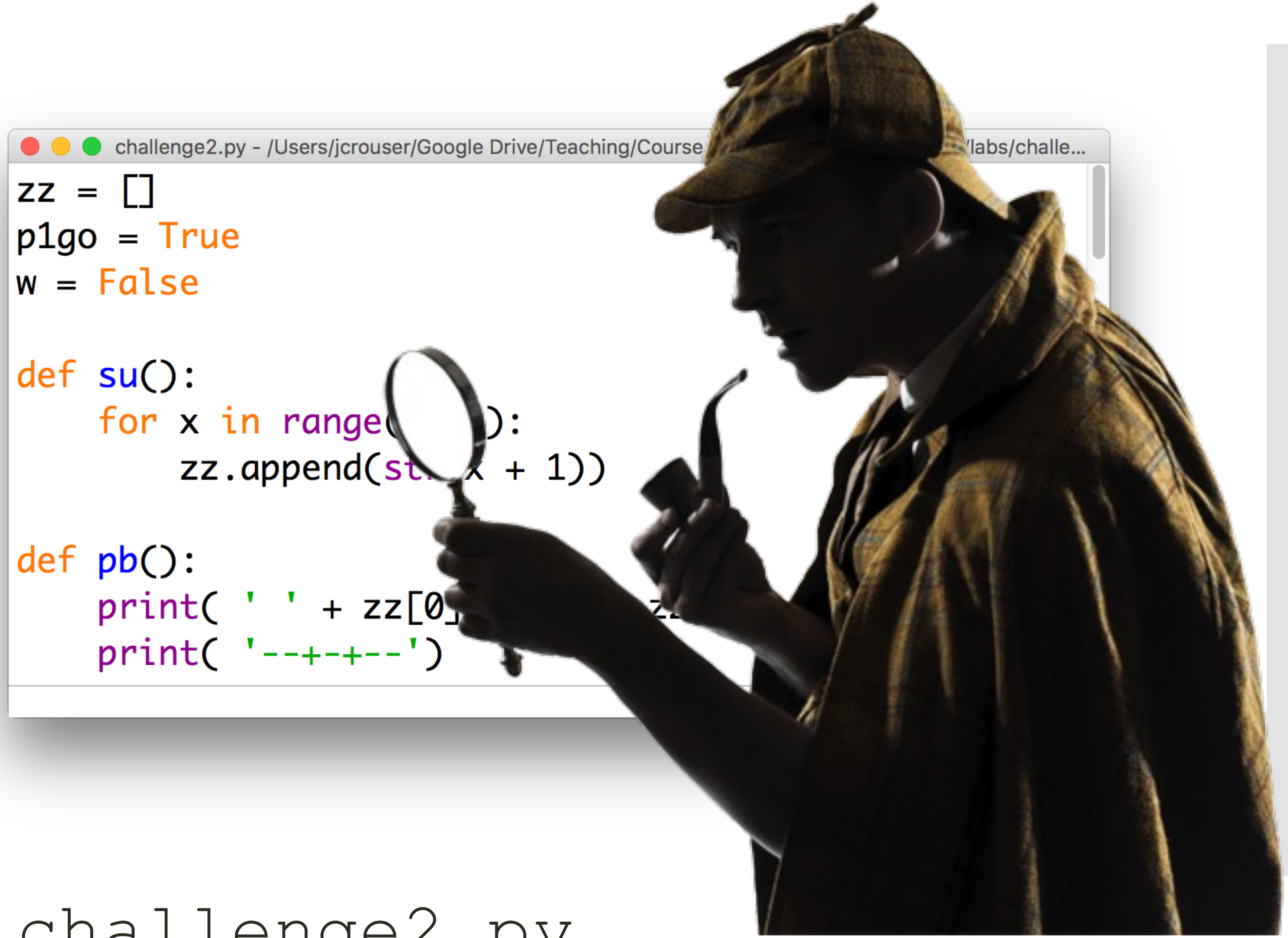
Step 4*: docstrings



```
def makeSong():
    """ Creates and prints a Song instance
        from user input"""
    # Get user input
    title = input("Title? ")
    artist = input("Artist? ")

    # Create Song instance and print info
    s = Song(title, artist)
    s.print()
```

DEMO TIME

Activity: “code detective”


challenge2.py - /Users/jcrouser/Google Drive/Teaching/Course /labs/challe...

```
zz = []
p1go = True
w = False

def su():
    for x in range(    ):
        zz.append(st   x + 1))

def pb():
    print( '  ' + zz[0]         z
    print( '--+--+--')
```

challenge2.py