

[220 / 319] Iterators

Meena Syamkumar
Andy Kuemmel

Iterators

Outline

- review problems
 - recursion
 - function object
 - map + lambda
- the scary vocabulary of iteration
- the open function

```
def get_one_digit_nums():  
    print("START")  
    nums = []  
    i = 0  
    while i < 10:  
        nums.append(i)  
        i += 1  
    print("END")  
    return nums  
  
for x in get_one_digit_nums() [0,1,2,3,4,5,6,7,8,9]:  
    print(x)
```

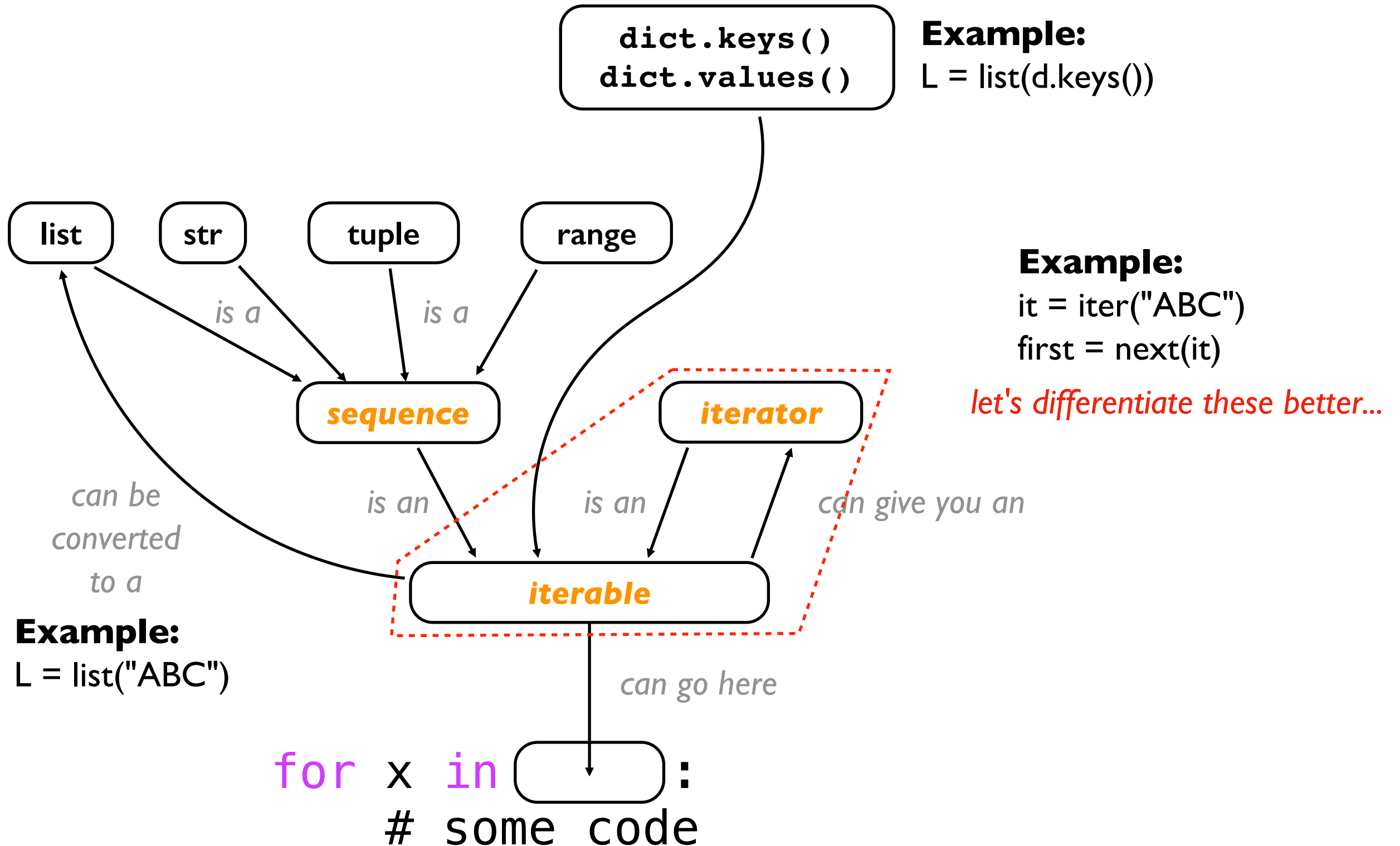
how many times is the word "START" printed?

Iterators

Outline

- review problems
 - recursion
 - function object
 - map + lambda
- the scary vocabulary of iteration
- the open function
- demos

The Vocabulary of Iteration



is `x` **iterable**?

if this works, then yes:

`y = iter(x)` **returns an iterator over x**

is `y` an **iterator**?

if this works, then yes:

`next(y)` **returns next value from y**



Can you classify x, y, and z?

`x = [1, 2, 3]`

`y = enumerate(['A', 'B', 'C'])`

`z = 3`

Things to try:

`iter(x)`

`next(x)`

etc.

Iterators

Outline

- review problems
 - recursion
 - function object
 - map + lambda
- the scary vocabulary of iteration
- the open function
- demos

Reading Files

```
path = "file.txt"  
f = open(path)
```



open(...) function is built in

Reading Files

```
path = "file.txt"  
f = open(path)
```



it takes a string argument,
which contains path to a file

file.txt

```
This is a test!  
3  
2  
1  
Go!
```

c:\users\meena\my-doc.txt

/var/log/events.log

../data/input.csv

Reading Files

```
path = "file.txt"  
f = open(path)
```



it returns a file object

file objects are iterators!

file.txt

This is a test!

3

2

1

Go!

Reading Files

```
path = "file.txt"
f = open(path)

for line in f:
    print(line)
```



Output

This is a test!

3

2

1

Go!

file.txt

This is a test!

3

2

1

Go!