String and Loop

Python Open Lab

String

- Last week we talked about basic operations like element fetch and slice in string.

- There are other aspects of string.

String operations

- len(str) — find the length of present string
- str.find(“ab”) — search a string in present string
- str.rstrip() — remove whitespace
- str.replace(“red”,”green”) — replacement
- str.split(“,”) — split
- str.isdigit() — decide whether string is all digit
- lower(), upper() — change string to uppercase or lowercase
- str.endswith(“hello”) — end test

String conversion

```
# Python 3.X

>>> "42" + 1
TypeError: Can't convert 'int' object to str implicitly

# Python 2.X

>>> "42" + 1
TypeError: cannot concatenate 'str' and 'int' objects
```

.

String conversion

- num = int(“42”)
- type(num)
- num = str(42)
- type(num)

String conversion

- Not only int…
  - str(3.1415)   float(“1.5”)
  - text = “1.234E-10”
  - float(text)

Loop

- It is about repeat steps/statements.

- Each step/statement is very similar, but could differ in details.

```
n = 5
while n > 0:
    print(n)
    n = n - 1
print(“Lift off!!”)
```

Loop

- When you are in the context of loop, make sure you use ‘tab’ to do the indentation

```
1: n = 5
2: while n > 0:
3:      print(n)
4:      n = n - 1
5: print(“Lift off!!”)
```

line3/4 is the content of loop, so line3/4 are in the context of loop

Loops: Iteration Variable

- Changes through each ‘iteration’ of loop
- Unique value at each iteration
- May act as iteration identifier

```
n = 5
while n > 0:
    print(n)
    n = n - 1
print(“Lift off!!”)
```

Finite loop and infinite loop

- Finite loop stops, infinite loop never stops
- Key is the change of iteration variable

```
n =10
while n >= 0:
    print(“I am executing”)
```

Finite loop and infinite loop

- Termination condition needs to be satisfied
- For ‘while n > 0’ loop, the termination condition is n > 0 is not true

```
n = 5
while n > 0:
    print(n)
    n = n - 1
print(“Lift off!!”)
```

Finite loop and infinite loop

- If clause behind the ‘while’ is true, the next iteration will be executed

```
while True:
    print(“hello”)
```

```
n = 5
while n >= 10:
    print(n)
    n = n - 1
print(“Lift off!!”)
```

Exercise

- print number from 2 to 100

- print number from 2 to 100, and print sum of printed numbers during each iteration

For loop

- A ‘for’ loop is used for iterating over a sequence (that is either a list, a dictionary, a set, or a string).
- Each iteration means operation on an element of sequence.
- Format of for loop: “for xx in xx:”

```
list = [‘red’,’black’,’yellow’]
for element in list:
    print(element)
```

For loop

- range in the for loop
    - range(6) means 0 - 5, not 0 - 6
    - range(1,6) means 1 - 5, not 1 - 6

```
for i in range(0,6):
    print(i)
```

For loop in list

- print all elements in a list

```
list = [‘red’,’black’,’yellow’]
for element in list:
    print( element )

for i in range(len(list)):
    print( list[i] )
```

For loop in dictionary

- If we want to use for loop in dictionary, we need to iterate over the keys of the dictionary

```
dict = {“mike”: 90, “jeff”: 80}
for key in dict:
    print(key)
    print(dict[key])
```

- ‘for key in dict’ is same as ‘for key in dict.keys()’

Exercise

- dict = {“class1”: { “Mike” : 90, “Jeff” : 85}, “class2”: {“Danny” : 87, “Jack” : 92} }

- print name and score of students in class1 first, then print “class1 done”, then print name and score of students in class2

For loop in string

- The basic unit of a string is character
- “_abc_   ”
- First character in this string, last character in this string?

```
str = “abcd”
for unit in str:
    prin(unit)
```

Next week

- If-else statement
- More about loop
- IO
- Everything combined, if-else, loop, IO

Reference

- Learning Python(Fifth Edition, Mark Lutz)
    - Chapter 7(189-237) and Chapter 13(387-395)