Exam 2: Review Lecture

(The list is not complete)

Sequences: String, List, Tuple

A. Strings: have seven operations

1. Indexing

   Var[idx], idx = 0, 1, 2, 3, .. or … -3, -2, -1

   IndexError: if idx does not exist

2. Slicing: var[start:end], colon part is necessary,
   a. does not through IndexError
   b. creates a brand new object (regardless mutable or immutable)
   c. creates same data structure type (list of lists – list of lists)
3. for loop

   for item in sequence: when sequence is a string then item is a char

4. len:

   for idx in range(len(sequence)):

   You can get value from idx, but converse is not always possible. For example, sequence contains duplicate items

5. in operator:

   substring match

   list or item exists

6. + operator: concatenation

   “Hello” + “ World” = “Hello World”

   [1, 2, 3] + [4, 5, 6] = [1, 2, 3, 4, 5, 6]

   Creates a brand new object instance

   Write on your cheat-sheet: when a new object is created-copying lecture

7. * some integer

   “Hello”*2 = “HelloHello”

   [1, 2, 3]*2 = [1, 2, 3, 1, 2, 3]

8. Comparison

   Lower case vs upper case

Digits: “11” < “2”

Prefixes: “bat” < “batman”

9. Methods (produce new object)

   upper/lower

   startswith/endswith

   strip/lstrip/rstrip

   Replace

   find → -1 (if not found)

   format → replaces { }

   split—> list of strings

   join→ “,”.join(L)

   Invocation of methods: some_var.method():

   For example, “hello”.upper() → “HELLO”

   str.upper(some_var)

B. List/csv

1. L = [1, 2, 3], emptyLIST [ ] or list()
2. Methods
   a. append()
   b. extend()
   c. pop-last index
   d. index: IndextError (different from find in string)
   e. sort
      i. increasing (default)
      ii. Reverse (Ture→ decreasing)
      iii. Key (function object reference)
   f. sorted function (creates a brand new object instance)
3. Process_csv: list of lists

C. Dictionary

1. immutable value as key
2. value can be anything
3. {}, dict()
4. Method:

   Pop

keys-> a kind of list

values->a kind of list

items() creates a list of tuples

D. json: useful for the data structures other than list of lists

know the difference between json and python

E. Objects

assignment operation & argument passing => reference copy

stack (reference) vs Heap (object)

F. namedtuple (custom -> immutable)

know how constructor function to create named tuple

G. Copying:

Import copy

Shallow (copy.copy→ depth level 1)

Deep (copy.deepcopy→ all depth levels only for immutable objects)

H. Recursion:

Base case

Recursive case

RecursionError (Infinite recursion or wrong arguments)

I: Function Object references

Sorting [(fname, lname),]

def extract(item)

item[1]

lambda item: item[1]:

J: Compressions

[<expression> <for> <if>]

[<expression1> if condition else <expression2> <for>]

[key: value <for> <if>]

K: Error Handling

-assert

-try except

-raise

L: File and Directories

f= open()

Read:

1. f.read()
2. list(f)
3. for line in f

Write: “w”

f.write(“\n”)

f..close