CSE 8A: Intro to Programming in Python

Spring 2021

Lecture 6 - strings, lists

UC San Diego

Announcement

- Get started with PA2 early when it is released!
    - It is more involved than PA1
- Seek help at autograder.ucsd.edu. If you don't have an account, you can request to enroll in CSE 8A in Fall 2021 there!
- Clicker frequency: AC

Topics for Today

- more strings processing
- lists

Review

- Boolean expressions (comparison and relational)
- if…else
  - else has no condition
  - else is optional
- if…elif…else
  - difference between this and a sequence of if statements

indexes for string characters

```
name = “cse-8a”
```

+index	0	1	2	3	4	5
letter	c	s	e	-	8	a
-index	-6	-5	-4	-3	-2	-1

[a:b] all the characters from index a to index b (not including index b)
[a:] all the characters from index a to the end of the string (end is the right end of the string)
[:b] all the characters from index 0 to index b (not including index b)

https://docs.python.org/2/library/stdtypes.html#sequence-types-str-unicode-list-tuple-bytearray-buffer-xrange

+index	0	1	2	3	4	5
letter	c	s	e	-	8	a
-index	-6	-5	-4	-3	-2	-1

[a:b] all the characters from index a to index b (not including index b)

[a:] all the characters from index a to the end of the string (end is the right end of the string)

[:b] all the characters from index 0 to index b (not including index b)

```
name = “cse-8a”
print(name[2:])
print(name[-2:])
print(name[1:-1])
```

A	B	C	D	E
e-8a	e-8	e-8a	e-8	error
8a	8	8a	8a	
(empty str)	se-8	se-8	(empty str)	

List Basics

A list is a collection of elements.

- Why do we need a list?
- How to create a list in python

```
name_of_list = [data separated by ,]
```

Note: not all data have to be of the same type!

List Basics

```
name_of_list = [data separated by ,]
```

```
paulGrades = [20, 33, 90]
christineGrades = [100, 101, 90]
#I created two lists and each of them has 3 elements

#indexing is very similar to strings
print(paulGrades[2])# 90
print(christineGrade[-2]) #101
```

Exercise: Lists

What is the length of the list new_students?

```
students = ["Maggie", "Tasnim", "Diego"]
new_students = students + ['Alessa', 'Wang']
len(new_students)
```

A) 3   B) 4   C) 5   D) 6

Exercise: List Indexing

What will be printed by the program below?

```
constants = [3.14, 2.71, 9.8, 1.414]
print(constants[len(constants) - 2])
```

A) 3.14
B) 2.71
C) 9.8
D) 1.414
E) Error: list index out of range

Exercise: List Indexing

What will be printed by the program below?

```
constants = [3.14, 2.71, 9.8, 1.414]
print(constants[len(constants)])
```

A) 3.14
B) 2.71
C) 9.8
D) 1.414
E) Error: list index out of range

Exercise: List Slicing

What will be printed by the program below?

```
constants = [3.14, 2.71, 9.8, 1.414]
constants[1:]
```

A) [3.14]
B) [3.14, 2.71, 9.8]
C) [3.14, 2.71, 9.8, 1.414]
D) [2.71, 9.8, 1.414]
E) Error

Exercise: List Slicing

What will be printed by the program below?

```
constants = [3.14, 2.71, 9.8, 1.414]
constants[:5]
```

A) [3.14]
B) [3.14, 2.71, 9.8]
C) [3.14, 2.71, 9.8, 1.414]
D) [2.71, 9.8, 1.414]
E) Error

Exercise: List Slicing

What will be printed by the program below?

```
constants = [3.14, 2.71, 9.8, 1.414]
constants[55:5555]
```

A) [3.14]
B) [3.14, 2.71, 9.8]
C) [3.14, 2.71, 9.8, 1.414]
D) [2.71, 9.8, 1.414]
E) []

Exercise: List Slicing

What is the value of the expression below?

```
tas = [‘Vanessa', ‘Yifeng', 'Hannah', 'Tim', 'Carmen', 'Claudia', ‘Sean']
tas[0:2] + tas[3:4]
```

A) [‘Vanessa', ‘Yifeng', 'Tim']

B) [‘Vanessa', ‘Yifeng', 'Hannah', 'Tim', 'Carmen']

C) ['Hannah', 'Tim', 'Carmen', 'Claudia', ‘Sean']

D) [‘Yifeng']

E) Error

Exercise: List Membership

What is the value of the expression below?

```
primes = [2, 3, 5, 7, 11]
(2 in primes) and (13 not in primes)
```

A) True

B) False

C) None

D) Syntax Error

Exercise: Functions over Lists

Write a function that takes a student_id, a list of ids, and a list of grades and returns the grade for the student_id

```
def lookup_grade(student_id, ids, grades):
        # TODO: Write function here
```

Sample Runs:

```
>>> ids = [10, 20, 30, 40, 50]
>>> grades = ['A', 'A-', 'B-', 'B+', 'A+']
>>> lookup_grade(40, ids, grades)
'B+'
>>> lookup_grade(20, ids, grades)
'A-'
```