Python & C++ Program Design -- Procedural Programming

Tuples, Dictionaries, Strings

Junjie Cao @ DLUT

Summer 2022

https://github.com/jjcao-school/c

Review

- Return values of Python functions

```
def ret1():
    return 2
def ret2():
    return 2, ‘ok’
if __name__ == "__main__":
    print(type(ret1()))
    print(type(ret2()))
    print(ret2())
```

<class 'int'>
<class 'tuple’> 元组
(2, 'ok')

Tuples 元组 vs Lists 列表

```
tup1 = ('physics', 'chemistry', 1997, 2000)
tup2 = (1, 2, 3, 4, 5 )
tup3 = "a", "b", "c", "d"
tup4 =()
lis1 = ['physics', 'chemistry', 1997, 2000]
lis2 = [1, 2, 3, 4, 5 ]
lis4 =[]
print("tuple {}, list {}".format(tup1[0],lis1[0]))
```

Updating Tuples

- Tuples Are Immutable 不可改变 Sequences
- Lists Are Mutable Sequences

```
tup1 = (12, 34.56)
tup2 = ('abc', 'xyz')
lis1 = [12, 34.56]

# Following action is not valid for tuples
# tup1[0] = 100
lis1[0] = 100

# So let's create a new tuple as follows
tup3 = tup1 + tup2
print(tup3) #(12, 34.56, 'abc', 'xyz')
```

Basic Tuples Operations

Python Expression	Results	Description
len((1, 2, 3))	3	Length
(1, 2, 3) + (4, 5, 6)	(1, 2, 3, 4, 5, 6)	Concatenation
('Hi!',) * 4	('Hi!', 'Hi!', 'Hi!', 'Hi!')	Repetition
3 in (1, 2, 3)	True	Membership
for x in (1, 2, 3): print x,	1 2 3	Iteration

Indexing, Slicing, and Matrixes

- L = ('spam', 'Spam', 'SPAM!')

Python Expression	Results	Description
L[2]	'SPAM!'	Offsets start at zero
L[-2]	'Spam'	Negative: count from the right
L[1:]	['Spam', 'SPAM!']	Slicing fetches sections

9.4 Challenge: List of lists of《Python Basics: A Practical Introduction to Python 3》

- Do a challenge of list, since tuples are immutable list.
- Write a program that contains the following lists of lists:
  - (a) name, (b) enrolled students, (c) annual tuition fees 学费.

```
universities = [
['California Institute of Technology', 2175, 37704],
['Harvard', 19627, 39849],
['Massachusetts Institute of Technology', 10566, 40732],
['Princeton', 7802, 37000],
['Rice', 5879, 35551],
['Stanford', 19535, 40569],
['Yale', 11701, 40500]
]
```

Define the following functions

- enrollment_stats() should return two lists: the first containing all of the student enrollment values and the second containing all of the tuition fees.
- Next, define a mean() and a median() function. Both functions should take a single list as an argument and return the mean and median of the values in each list.
- Using universities, enrollment_stats(), mean(), and median(), calculate the total number of students, the total tuition, the mean and median of the number of students, and the mean and median tuition values.

- Finally, output all values, and format the output so that it looks like this:

```
******************************
Total students: 77,285
Total tuition: $ 271,905

Student mean: 11,040.71
Student median: 10,566

Tuition mean: $ 38,843.57
Tuition median: $ 39,849
******************************
```

More challenges

- 9.5. Challenge: Wax Poetic of 《Python Basics: A Practical Introduction to Python 3》

Dictionaries 字典

- A dictionary is a data type similar to arrays, but works with keys 键 and values 值 instead of indexes 索引.

```
phonebook = {}
phonebook["John"] = 938477566
phonebook["Jack"] = 938377264
phonebook["Jill"] = 947662781
print(phonebook)
```

```
phonebook = {
    "John" : 938477566,
    "Jack" : 938377264,
    "Jill" : 947662781
}
print(phonebook)
```

```
{'Jill': 947662781, 'Jack': 938377264, 'John': 938477566}
```

Using get() to Access Values

```
alien = {'color': 'green', 'speed': 'slow'}
print(alien['points'])
```

```
[Running] python -u
"/Users/jjcao/work/courses/c/lab/test.py"
Traceback (most recent call last):
File "/Users/jjcao/work/courses/c/lab/test.py", line 2, in
<module>
print(alien['points'])
KeyError: 'points'
```

```
point_value = alien.get('points', 'No point value
assigned.')
print(point_value)
```

Iterating over dictionaries 遍历字典

```
phonebook = {"John" : 938477566,"Jack" :
938377264,"Jill" : 947662781}

for name, number in phonebook.items():
    print("Phone number of %s is %d" % (name,
number))
```

```
phonebook = {"John" : 938477566,"Jack" :
938377264,"Jill" : 947662781}

for name in phonebook.keys():
    print(type(name), name.title())
```

```
<class 'str'> John
<class 'str'> Jack
<class 'str'> Jill
```

Looping Through a Dictionary's Keys in a Particular Order

```
favorite_languages = {'jen': 'python','sarah': 'c',
'edward': 'ruby','phil': 'python',}
for name in sorted(favorite_languages.keys()):
    print(f"{name.title()}, thank you for taking
the poll.")
```

```
Edward, thank you for taking the poll.
Jen, thank you for taking the poll.
Phil, thank you for taking the poll.
Sarah, thank you for taking the poll.
```

```
print("The following languages have been
mentioned:")
for language in favorite_languages.values():
    print(language.title())
```

Removing a value

```
phonebook = {
"John" : 938477566,
"Jack" : 938377264,
"Jill" : 947662781
}
# del phonebook["Jack"]
phonebook.pop("Jack")
print(phonebook)
```

Exercise: Adding New Key-Value Pairs

- Add "Jake" to the phonebook with the phone number 938273443, and remove Jill from the phonebook.

```
phonebook = {
"John" : 938477566, "Jack" : 938377264,
"Jill" : 947662781}
# your code goes here

# testing code
if "Jake" in phonebook:
    print("Jake is listed in the phonebook.")
if "Jill" not in phonebook:
    print("Jill is not listed in the phonebook.")
```

Notes

- Different types of key-value objects in one dictionary
  - alien = {'color': 'green', 'points': 5}
  - print(alien)
- A List of Dictionaries

```
alien_0 = {'color': 'green', 'points': 5}
alien_1 = {'color': 'yellow', 'points': 10}
alien_2 = {'color': 'red', 'points': 15}
aliens = [alien_0, alien_1, alien_2]
for alien in aliens:
    print(alien)
```

Basic String Operations

```
astring = "Hello world!"
print(len(astring))#12
print(astring.index("o"))#4
print(astring.count("l"))#3
print(astring[3:7]) #lo w
print(astring[3:7:2])# len of it is 2
print(astring[::-1])
print(astring.lower())
print(astring.startswith("Hello"))
print(astring.endswith("asdfasdfasdf"))
afewwords = astring.split(" ")
```

```
12
4
3
lo w
l
!dlrow olleH
hello world!
True
False
```