Csci 1523
1523 Study Guide Module 08 Dictionaries and Sets

Student (Print): Amethyst O'Connell

This study guide contains 6 pages (including this cover page) and 28 problems. Check to see if any pages are missing. Enter all requested information on the top of this page, and put your initials on the top of every page, in case the pages become separated.

You may use your books, notes, calculator or internet sources while completing this study guide.

Please try to answer the sections clearly and PRINT your answers legibly.

Special Note: The completion of study guides in this course is to supplement your learning of the materials they are not required as a part of normal grading. However they may enhance the extra credit portion of the course if attendance has been regular and laboratories completed satisfactorily.

Chapter 9 Dierbach Study Guide

Data structures are an important part of program development. A properly constructed data structure conserves memory, improves execution time and often simplifies logic.

In this section we look at two such data structures *dictionaries* and *sets*.

A *dictionary* is an indexed data store. That is part of the data stored in such a data structure serves as a *key* or index by which the data to which it relates is stored. These types of data structure are used where we have needs to quickly recall and store data based on the index to that data.

A *set* has more function as a logical operator since the set data structure has behaviors which are nearly identical to mathematical sets. Their use often simplifies logic as the programmer can leverage their behavior when developing a program.

Below you will find a series of questions concerning dictionaries and sets. The materials required to answer them are in Chapter 9 of Dierbach, course notes and movies.

1. Elements in an indexed linear data structure are ordered in contrast the elements of an associative data structure are unordered.
2. In Python an associative data structure using a key-value pair is called a dictionary type.
3. F A Python dictionary is fixed length and therefore *immutable*.
4. T The values of an associative data structure are unordered.
5. F The location that an element is stored in and retrieved from in an associative data structure depends on both its key value and its index value.
6. Dictionaries in Python are

   [✓ A.] mutable

   B. immutable

   C. neither mutable nor immutable

   D. mutable or immutable, depending on the situation

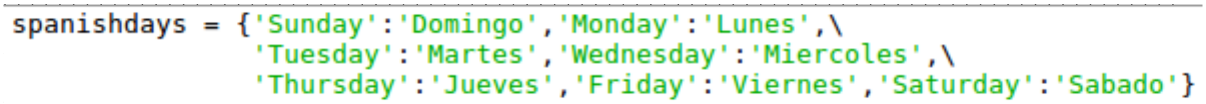
   E. none of the above
7. The values stored in a dictionary are assigned/retrieved by use of a corresponding key value. Therefore, dictionaries in Python are an associative data structure.
8. The location that a given value is stored in a dictionary in Python is determined by its key value. The process of determining the storage location of a value based on the value itself is called hashing.

9. F In order maintain maximum flexibility Python dictionaries must use mutable data types as keys.

10. The dict() creates a new empty dictionary.

11. Three examples of the types of values that can be used as key values of dictionaries in Python are strings, tuples and integers.

12. Define a dictionary named *spanishdays* that contains as key values the days of the week (Sunday, Monday, etc), in which each key value has the corresponding name in Spanish (Domingo, Lunes, Martes, Miercoles, Jueves, Viernes, Sabado).

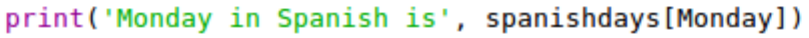


```
spanishdays = {'Sunday':'Domingo','Monday':'Lunes',\
               'Tuesday':'Martes','Wednesday':'Miercoles',\
               'Thursday':'Jueves','Friday':'Viernes','Saturday':'Sabado'}
```

13. The len function returns the number of key-value pairs in a Python dictionary.

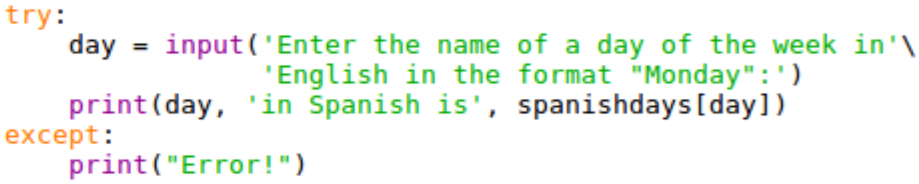
14. Assume a key, *myKey* is used to access a particular Python dictionary, *jDict*. Write a logical statement which returns True if the *myKey* and its data are in the dictionary and False otherwise: myKey in jDict.

15. Assuming that dictionary *spanishdays* from the question above has been defined, give an instruction that uses *spanishdays* to display: **The word for Monday in Spanish is Lunes**.



```
print('Monday in Spanish is', spanishdays[Monday])
```

16. Assuming the dictionary *spanishdays* from the question above has been defined, and that variable *day* contains one of Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday. Give an instructions that displays: **The word for *English day name* in Spanish is *Spanish day name***.



```
try:
    day = input('Enter the name of a day of the week in'\
                'English in the format "Monday":')
    print(day, 'in Spanish is', spanishdays[day])
except:
    print("Error!")
```

17. A Python set is a mutable data type with nonduplicate and unordered values.

18. T A set in Python is a mutable data type, and a frozenset is an immutable type.

19. T As with sequence types, the in operator is used to determine if a particular value is in a given set.

20. Which of the following is NOT valid in Python for set s1 and s2?

    A. $apples \quad in \quad s1$

    B. $s1 \quad | \quad s2$

    [✓ C.] $s1 + s2$

    D. $s1 - s2$

    E. $s1 \quad \& \quad s2$

21. The set type in Python is mutable, while the frozenset type is immutable.

22. The set type in Python provides the usual mathematical operations of union, intersection and symmetric difference.

23. In order to create an empty set, set() is used, not a pair of empty braces, { }.

24. The proper syntax for creating a set containing the values apple, pear, and banana is:

    set(['apple', 'pear', 'banana'])

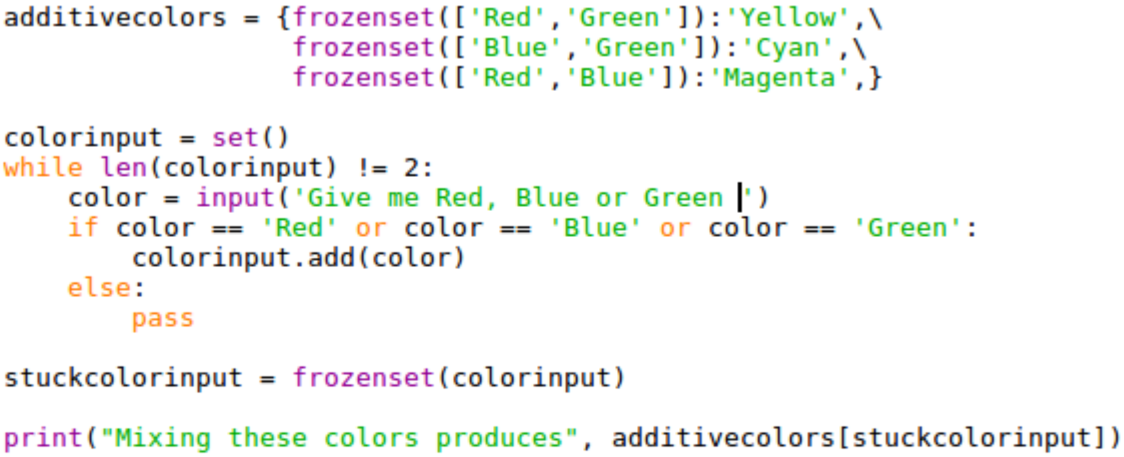
25. Define a set named colors that contains the following colors: red, blue, green, yellow.

```
colors = set(['red', 'blue', 'green', 'yellow'])
```

26. Declare a set named RGBprimarycolors containing the values red, green, and blue, such that the set is immutable.

```
RGBprimarycolors = frozenset(['red','green','blue'])
```

27. In the additive (RGB) color model, red and green produce yellow, green and blue produce cyan, and blue and red produce magenta. Define a dictionary named additivecolormixing able to be given any two of these primary colors, and retrieve the resulting color from the combining of the two.



```
additivecolors = {frozenset(['Red','Green']):'Yellow',\
                  frozenset(['Blue','Green']):'Cyan',\
                  frozenset(['Red','Blue']):'Magenta',}

colorinput = set()
while len(colorinput) != 2:
    color = input('Give me Red, Blue or Green ')
    if color == 'Red' or color == 'Blue' or color == 'Green':
        colorinput.add(color)
    else:
        pass

stuckcolorinput = frozenset(colorinput)

print("Mixing these colors produces", additivecolors[stuckcolorinput])
```

28. Given the following sets: $A = \{4, 5, 6, 8, 9\}$ *and* $B = \{4, 10, 11, 8, 6, 7\}$ using the Python operations on either or both of these sets complete the questions below.

   (a) $A \mid B$: {4, 5, 6, 7, 8, 9, 10, 11}

   (b) $A \;\&\; B$: {8, 4, 6}

   (c) $1 \;\; in \;\; B$: False

   (d) **A.add(10)** : None

   (e) $A - B$: {9, 5}

   (f) **len(B)** : 6