

Name: _____

Partner: _____

Python Activity 27: Dictionaries

Lists are useful data structures, but what if the relationship between data isn't sequential?

Learning Objectives

Students will be able to:

Content:

- Define a **dictionary**.
- Identify the **key** and **value** pair of a dictionary.
- Explain why a dictionary is a good data structure for organizing data.

Process:

- Write code that accesses the keys, values, and length of a dictionary.
- Write code to create and modify dictionaries.
- Write code that iterates over a dictionary's keys.

Prior Knowledge

- Python concepts: lists of lists, indexing, mutability, len(..), for.. loops

Critical Thinking Questions:

1. Examine the sample code defining a list of lists, below:

Sample Code

```
dog2owner = [['pixel', 'iris'], ['sally', 'jeannie'], ['jerry', 'lida']]  
print(dog2owner[0][0]) # prints: 'pixel'
```

- a. What is stored at `dog2owner[1][1]`? _____
- c. Write a line of code to print the name of Sally's owner using `dog2owner`:


- d. Write a line of code to access and print the name of Lida's dog via `dog2owner`:

- e. As `dog2owner` gets bigger and bigger (the CS department is growing!), will a list of a lists be an accessible way to continue storing this information?

2. The following code occurs in interactive Python and introduces a new data structure:

```
0 >>> dt = {'pixel':'iris', 'sally':'jeannie', 'jerry':'lida'}  
1 >>> dt['sally']  
2 'jeannie'
```

- a. What does `dt['sally']` do?

-  b. How might python know that Sally (the dog) is mapped to Jeannie (the owner)?
Where is that relationship defined?

c. In the line, `dt['sally']`, what does the value in the square brackets represent?

FYI: A *dictionary* is an unordered data structure that instead of storing *values* at numerical indices, *values* are mapped to *keys*. Keys must be an immutable data type.

d. Write a line of code to print the name of your CS134 instructor's name, accessed via the dictionary, `dt`: _____

 e. Why might a dictionary be a better data structure for this data than a list of lists?

f. How would you describe the *keys* and *values* for this dictionary, `dt`?

keys: _____ values: _____

g. What type of data is stored in the keys and the values for `dt`?


keys: _____ values: _____

3. The following code occurs in interactive Python and introduces a new data structure:

```
0 >>> dt = {'pixel':'iris','sally':'jeannie','jerry':'lida'}
1 >>> dt['jeannie']
2 KeyError: 'jeannie'
```

a. What is the programmer trying to do with the `dt['jeannie']` on line 1?


b. Why might this code be throwing the error on line 2?



 c. What does this error tell you about what can go in the dictionary square brackets?

4. Examine the following code from interactive Python:

```
0 >>> dt = {'pixel':'iris','sally':'jeannie','jerry':'lida'}
1 >>> dt['wally'] = 'steve'
2 >>> dt
3 {'pixel':'iris','sally':'jeannie','jerry':'lida','wally':'steve'}
```


a. What does the line `dt['wally'] = 'steve'` do?

 b. What might this imply about the *mutability* of dictionaries?

-  c. What does the object in square brackets on the left hand side of the assignment operator in line 1 represent? (*Circle one*) key or value
-  d. What does the object on the right hand side of the assignment operator in line 1 represent? (*Circle one*) key or value
- e. Write a line of code to add Bill and his dog, Artie, to our dictionary.

5. Examine the following code from interactive Python:


```
0 >>> cs_pets = {'dogs':9, 'cats':4, 'bees':20000}
1 >>> len(cs_pets)
2 3
```

- a. What type of data is stored in the keys and the values for `cs_pets`?
keys: _____ values: _____
- b. How many keys does `cs_pets` have? _____
- c. What is the length `cs_pets`? _____
-  d. How does python determine the length of a dictionary object?
- e. If we added a line 3 of code, `cs_pets['others'] = ['hamster', 'ferret']`, what might `len(cs_pets)` return? _____

7. Examine the following example code:

```
>>> coll = dict()                    # can also do: coll = {}
>>> coll['colleges'] = 'williams'
>>> coll['colleges'] = 'amherst'
```

- a. If we wrote a fourth line of code, `print(coll)`, what might be the output?

-  b. At the end of this code execution, `coll` only has: `{'colleges': 'amherst'}`
Why might this be?

FYI: Dictionaries can only have one key of its value, any replicated key:value mappings added will simply overwrite the previous one!


8. Examine the following example code from interactive python:

```
0 >>> date = {'month':'dec', 'day':9, 'year':1906}
1 >>> for mykey in date:
2 ...     print("The", mykey, "is", date[mykey])
```

- a. What data does the dictionary, `date`, appear to hold?

- b. If you had to guess, what might the programmer want to be output by line 2?

- c. For the first defined item of `date` what might `mykey` and `date[mykey]` refer to on lines 1 & 2?
`mykey`: _____ `date[mykey]`: _____
- d. The first time through the loop defined on line 1, line 2 might print 'The month is dec' What might be printed the second time through the loop?

-  e. What does line 1, `for mykey in date:, do?`

- f. Write some code that will iterate over the items in `date` and print *only* the values:

8. Examine the following example code from interactive python:

```
0 >>> date = {'month': 'dec', 'day': 9, 'year': 1906}
1 >>> sorted(date)
```

- a. What do you think will be the output of line 1?

- b. Line 1 actually outputs `['day', 'month', 'year']`. How do these values relate to the data dictionary?

- c. Write some lines of code that will construct a list of the *values* in `date`, sorted by the *keys* in `date`:

Application Questions available in the digital file on Glow.

