

UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Summer 2026

TA: Nahee Jeong



Discussion 9: Python Practice and Data Structures

Instructions:

- If you missed this discussion, fill out this entire worksheet, and upload it to the Gradescope assignment titled “Discussion 9” by next Discussion.
- Please open up snap.berkeley.edu/run on your computer.
- For the worksheet, you can either explain the process in words, show a screenshot, or draw the block/process.

Group Activity / Question of the Day

- Do you play any sports? What’s your favorite sports or outdoor activities?

Section I – Lists

Warm Up

1. In Python, lists are _____. *Select all that apply.*
 - a. **Mutable** and Immutable
 - b. **Ordered**
 - c. **Allows duplicates**
 - d. Fixed in size
 - e. **Indexed**
 - f. **Iterable**
2. **True** or False. Python lists can contain elements of different data types.
3. `lst = [[2], [4, 5, 'a'], ['c', 3], ["tree", [2]]]`
 - a. `len(lst)`? 4 items, so 4
 - b. `len(lst[3])`? 2

Quick Check: What would Python print?

1. `nums = [0, 1, 2, 3, 4, 5]`
`print(nums[:3] == list(range(3)))`
2. `letters = ["a", "b", "c", "d", "e"]`
`print(letters[::-1])`
3. `nums = [10, 20, 30, 40, 50]`
`print(nums[::-2])`
4. `nums = [1, 2, 3, 4, 5]`
`print(nums[1:1])`
5. `nums = [0, 1, 2, 3]`
`double_odds = [n * 2 for n in nums if n % 2 != 0]`
`print(double_odds)`

List Comprehension

Translate the following Snap! blocks in Python using List Comprehension.

1.



Keep → [word for word in my_list if (len(word) > 5)]

[word[0] for word in [word for word in my_list if (len(word) > 5)]]

[x for x in <list_name>]

[x for x in <list_name> if (some condition)]

[(do something) if (some condition) else (do something else) for x in <list_name>]



[sum([n for n in nums if n % 2 == 0])] → [30]

3. Use list comprehension to find the index of an item in a list. You may assume that the item appears only once in the list.

```
def find_index(item, lst):
```

```
    return _____
```

4. Use list comprehension to replace all odd numbers with [0] in a list, keeping even numbers the same. You can assume that this is a flat list.

```
def replace_odds(nums):
```

```
    return _____
```

Section II – Dictionaries

1. Match the method to its description.

Method	Description
1. <code>dict.keys()</code>	A. Returns a shallow copy of the dictionary
2. <code>dict.values()</code>	B. Returns all key-value pairs as tuples
3. <code>dict.items()</code>	C. Returns all keys in the dictionary
4. <code>dict.get(key, default)</code>	D. Safely retrieves a value, returns default if missing
5. <code>dict.copy()</code>	E. Returns all values in the dictionary

- 1 - C.
- 2 - E.
- 3 - B.
- 4 - D
- 5 - A

2. Fill out the table below with the appropriate Python syntax.

`class_dict = {'Math': '1A', 'English': 'R1A'}`

Task	Python Syntax
Add the key 'CS' with the value '10'	
Access the value of 'Math'	
Change the value of 'Math' to '1B'	
Check if 'MCB' is a key in class_dict	
Check if '10' is a value in class_dict	
Get a list of all keys in class_dict	
Get a list of all values in class_dict	

Task	Python Syntax
Remove the key 'English' from class_dict	
Loop through the dictionary and print key: value pairs	

3. Merge Dictionaries

Fill the function merge_dicts(dict1, dict2) that takes two dictionaries and returns a new dictionary containing all keys from both.

- If a key appears in both dictionaries, the value of the result should be the sum of the values
- You can assume keys are strings and values are numbers.

```

""" Below is an example of how merge_dicts works:
>>> d1 = {'Dan': 10, 'Oski': 15}
>>> d2 = {'Alonzo': 5, 'Oski': 20, 'Dan': -10}
>>> merge_dicts(d1, d2)
{'Dan': 0, 'Alonzo': 5, 'Oski': 35} """

```

```
def merge_dicts(dict1, dict2):
```

```
    result = {}
```

```
    for k in _____:
```

```
        result[k] = _____
```

```
    for k in dict2:
```

```
        if k in result:
```

```
            result[k] += dict2[k]
```

```
        else:
```

```
            _____
```

```
    return _____
```