# CS 114 Tutorial 5

Oct 10 2025

# Goals for this Week:

- Submit assignment 02 (due tonight 5:30)
- Look at assignment 03 (releases tonight)
- Be familiar with mutation vs non-mutation
- Know how to use tuples
- Be familiar with sorted vs list.sort
- Understand how to use dictionaries

# A Quick Overview

**Tuples:**

- An immutable list
- Represented using ()
- my_tuple: tuple[int, str, …]
- Annotate each value in the tuple

**Soting:**

- sorted() creates a new list
- .sort() mutates the list
- Key: sorts by a given criteria
- Reverse: highest to lowest

**Dictionary:**

- X = {key: value}
- Keys are like the indices
- Dictionaries are references
- Adding: X[“new key”] = “value”
- Access values: X[“existing key”]
- Remove with .pop(“key name”)

Write a function min_max(nums) that returns a tuple contain the smallest and largest integers from the non-empty list nums. Do not use the built-ins min() or max().

Try this with & without sorting!

Given a dictionary of the type dict[str, int], create a function sort_dictionary(d) that returns a new dictionary sorted by the length of its keys. If two keys are the same length, the keys are in alphabetical order.

{“wow”: 3, “any”: 76, “hello”: 2, “c”:8} → {“hello”: 2, “any”: 76, “wow”: 3, “c”:8}

Go to vevox.com

Sign in using the session ID: 144-133-796

# Annotations

```python
from typing import Any

x = ("hello", 100, [3, 5.0, "hi"])

def use_tuple(x:...) -> None:
    print(x[1])
```

How would you annotate the tuple x?

A. x: tuple(str, float, list[Any])

B. x: tuple[str, int, list[Any]]

C. x: tuple([str, int, list[Any]])

D. x: tuple[str, float, list[Any]]

# Annotations

```
d = {10400: "hi", 758: "yay", 8926: "abc"}
```

How would you annotate d if it were used as a function parameter?

A. dict(str, int)  B. dict(int, str)  C. dict[int, str]  D. dict[str, int]

# Sorting

```
lst = [10, 9, 34, 60, 5]
print(lst.sort(reverse=True))
```

What will be printed?

A. [5, 9, 10, 34, 60]  B. [60, 34, 10, 9, 5]  C. None

# Sorting

```
lst = [10, 9, 34, 60, 5]
print(sorted(lst, reverse=True))
```

What will be printed?

A. [5, 9, 10, 34, 60]  B. [60, 34, 10, 9, 5]  C. None

# Sorting

```
dictionary = {10400: "hi", 758: "yay", 8926: "abc"}

print(sorted(dictionary))
```

What will the above code print?

A. A list of the sorted keys

B. A list of the sorted values

C. A tuple with the sorted keys and values

D. Nothing, sorted doesn't work on dictionaries

# Dictionaries

```
A = {"hello", 6, "why", 8, "say", 90, "hi", 4}
B = {65: "why", 6: "are", 8: "hey", 90: "care", 8: "no"}
C = {6.0: "name", 7.8: "hi", 0.7: "hello", 0.3: "hi"}
```

Which of the following is a valid dictionary?

A. B. C. D. None E. All

Given the following list: [student_A, student_B, student_C] where each student is a dictionary containing the keys “name”: str, “number”: int, “average”: float.

Create a function that finds the student with the lowest grade and changes their average to 95.0. This should mutate the dictionary and return None. If two students have the same lowest average, change the first instance of it.

*def changeGrade(lst: list[dict[str, Any]]) → None:*

Ex. Student_A = {“name”: “Elena” , “number”: 123456, “average”: 100.0}