PYTHON: TUPLES, DICTIONARIES AND SETS II

Objectives

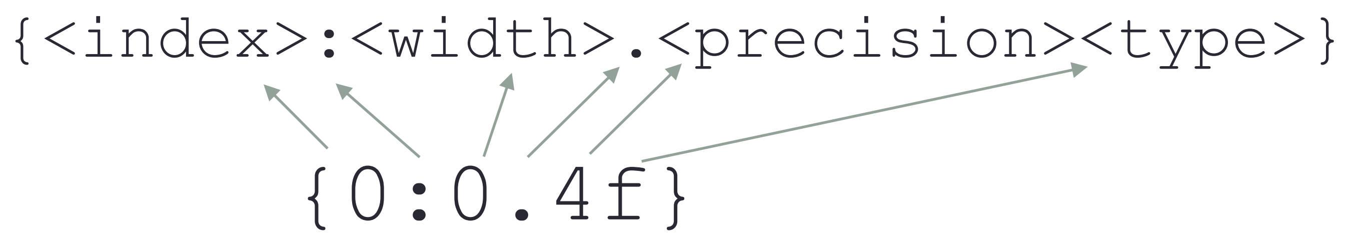
- To know how to properly format and display data on the console
- To understand the Python data model – Set
- To apply combinations of the data models introduced so far to solve practical problems

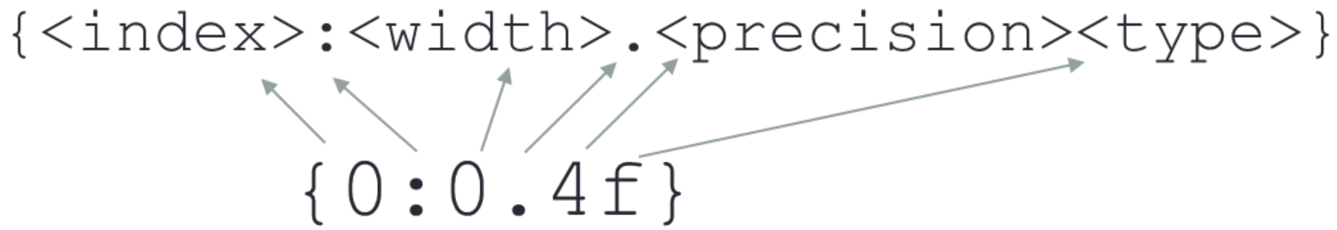
String Formatting

- We have so far used some simple string operations to format the output.
- But it is difficult to format finer control with them.
- String object has a `format` method to perform more sophisticated formatting.

The string `format()` method

- String formatting syntax:
  - `<template-string>.format(<values>)`
  - E.g., `"The value of pi is {0:0.4f}".format(math.pi)`
- Curly braces(`{}`) inside the `template-string` marks "slots" into which the provided values are inserted.
  - `{<index>:<format-specifier>},` e.g., `{0:0.4f}`
  - `index`: indicate which parameter to insert here
  - One form of specifiers is `<width>.<precision><type>`.
    - `width`: the number of spaces for printing the parameter
    - `precision`: the number of decimal places
    - `f`: will print out 0 to fill up the number of decimal places.





EXERCISE 8.1

Try

```
hongva = 100
dennis = 200
print("hongva has ${0:0.2f} and Dennis has
${1:0.2f}".format(hongva, dennis))
print("hongva has ${1:0.2f} and Dennis has
${0:0.2f}".format(hongva, dennis))
```

EXERCISE 8.2

Try

```
dennis = 100
print("Dennis has ${0:0.2f}".format(dennis))
print("Dennis has ${0:1.2f}".format(dennis))
print("Dennis has ${0:5.2f}".format(dennis))
print("Dennis has ${0:10.2f}".format(dennis))
```

EXERCISE 8.3

Try

```
import math
print("The value of pi is {0:0.4f}".format(math.pi))
print("The value of pi is {0:0.20f}".format(math.pi))
print("The value of pi is {0:0.20}".format(math.pi))
```

EXERCISE 8.4

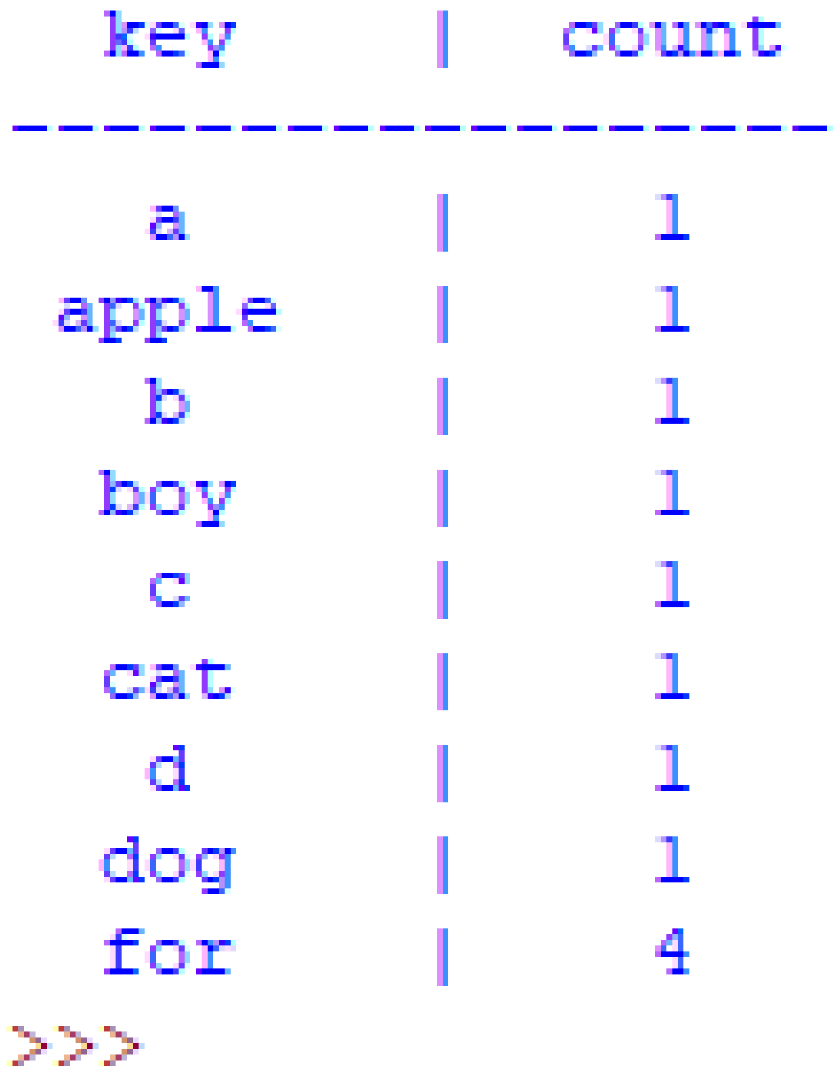
Try

```
print("{0:>10}".format("Dennis"), "|", "Liu", sep="")
print("{0:10}".format("Dennis"), "|", "Liu", sep="")
print("{0:^10}".format("Dennis"), "|", "Liu", sep="")
```

EXERCISE 8.5

Modify Exercise 7.8.

Format the output so that the statistics are displayed properly.



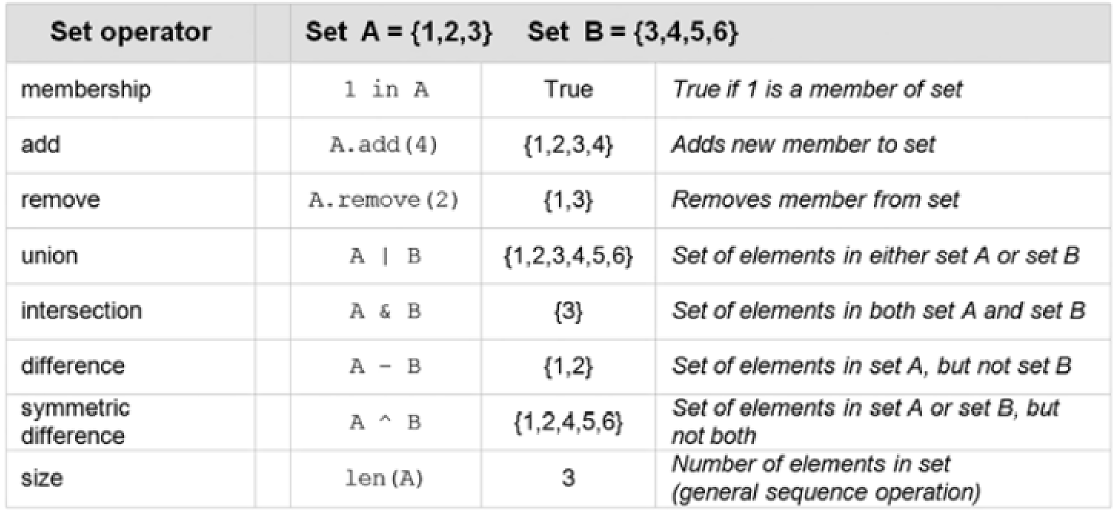
```
   key      |    count
---------------------
    a       |      1
  apple     |      1
    b       |      1
   boy      |      1
    c       |      1
   cat      |      1
    d       |      1
   dog      |      1
   for      |      4
>>>
```

Sets in Python

- A set is a collection of non-mutable objects which are elements of the set.
- Set is mutable.
- The elements could be of any non-mutable types.
  - Numbers, strings, tuples are allowed, but not lists.
- There is no order in a set. (items in a set do not have a defined order. Set items can appear in a different order every time you use them, and cannot be referred to by index or key.)
- There are no duplicate elements in a set.
- Note that in Python, `{}` is not the empty set, which should be **`set`**`()` instead!
  - So, what is `{}` in Python?

EXERCISE 8.6

- Create a set using `set()` and fill in elements `{. . .}` with duplicate elements.
- Check that the duplicates are gone.
- Create a set containing a tuple as one of the elements.
- Create a set containing a list as one of the elements.
- Use **`len`**`()` to get the size of a set.
- Use the keyword "`in`" to test whether an object is an element of a set.
- Use a for loop to print all the elements of a set.



Set operator	Set A = {1,2,3} Set B = {3,4,5,6}		
membership	`1 in A`	True	*True if 1 is a member of set*
add	`A.add(4)`	{1,2,3,4}	*Adds new member to set*
remove	`A.remove(2)`	{1,3}	*Removes member from set*
union	`A | B`	{1,2,3,4,5,6}	*Set of elements in either set A or set B*
intersection	`A & B`	{3}	*Set of elements in both set A and set B*
difference	`A - B`	{1,2}	*Set of elements in set A, but not set B*
symmetric difference	`A ^ B`	{1,2,4,5,6}	*Set of elements in set A or set B, but not both*
size	`len(A)`	3	*Number of elements in set (general sequence operation)*

^ : Caret

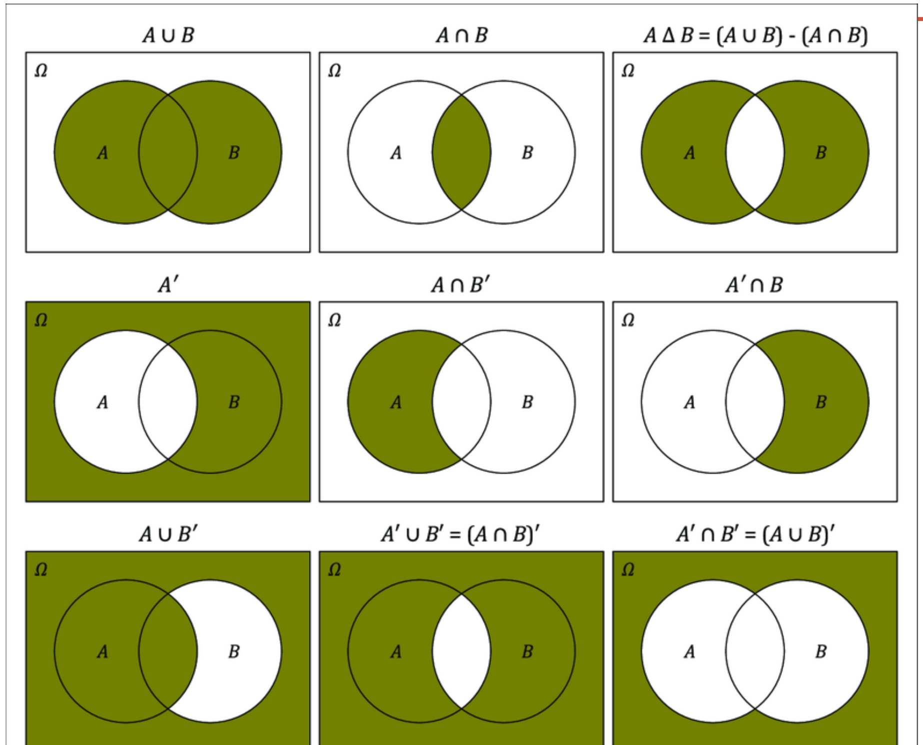
and

Source: Charles Dierbach. 2013. Introduction to Computer Science Using Python. Wiley.

EXERCISE 8.7

Try all four set operators: `|`, `&`, `-`, `^`.

You can also use four methods for sets: **`intersection`**, **`union`**, **`difference`**, **`symmetric_difference`**. For example, **`set_A.union(set_B)`**.

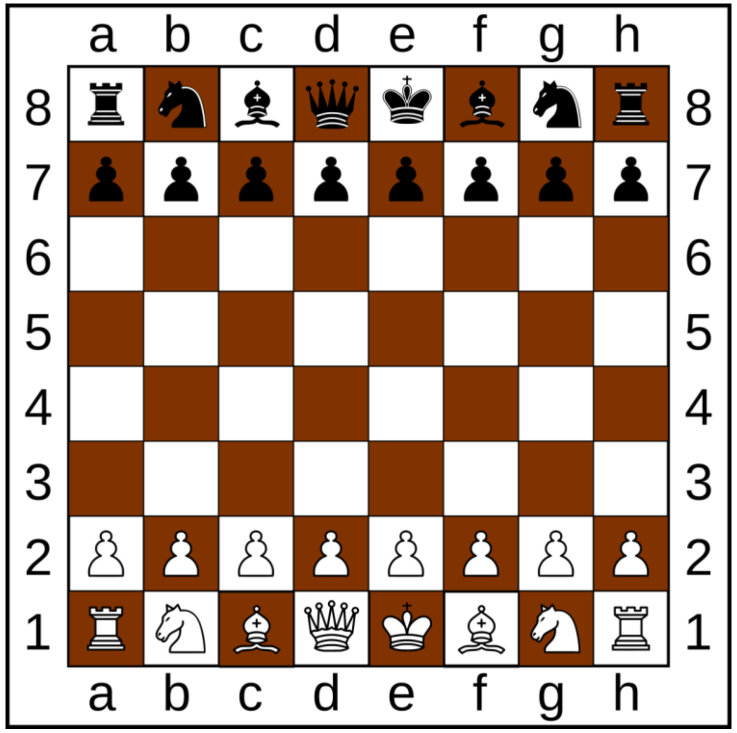

$A \cup B$
$A \cap B$
$A \Delta B = (A \cup B) - (A \cap B)$
A'
$A \cap B'$
$A' \cap B$
$A \cup B'$
$A' \cup B' = (A \cap B)'$
$A' \cap B' = (A \cup B)'$
Ω
A
B

More on Data Models

- Now you have studied
  - List
  - Tuple
  - Dictionary
  - Set
- Based on your application, it is possible to have more complex data model based on the fundamental ones:
  - A list of list?
  - A dictionary with tuples as keys and set as values?
  - A set of tuples?

Example 1

- How can a chess board be modelled in your program?



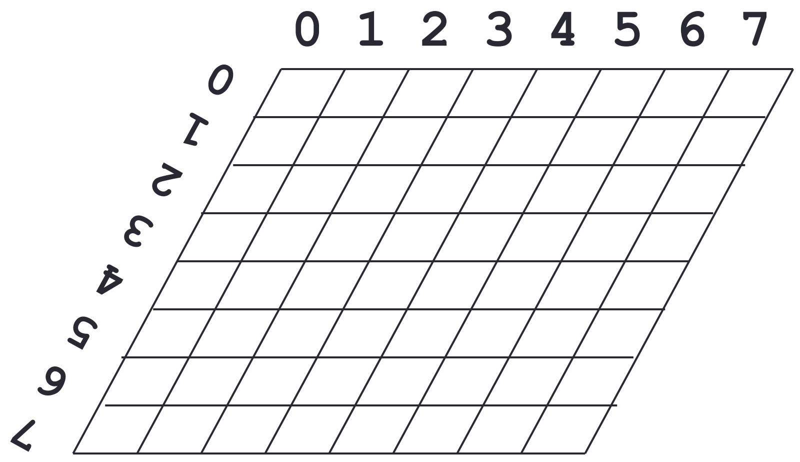


Image Source: https://commons.wikimedia.org/wiki/File:AAA_SVG_Chessboard_and_chess_pieces_02.svg

Example 2

- How can a student record table be created?

Student ID	Student Name	Major	Year of Cohort	Address	Phone Number
12345678D	Chan Tai Man	Computing	2018	Wan Chai	99912345
13579123D	Ng Siu Ching	Nursing	2019	Hung Hom	87654321
20123456D	Simon Lee	Chinese	2020	Kowloon City	22345123
56781234D	Wong Tai Sin	Financial Services	2017	Wong Tai Sin	45433453

Example 3

- As an extension of Example 2, if each student has a number of courses to take, how can the course code(s) be stored in the current data model, so that
  1. you can easily add and remove courses from the student's record; and
  2. you can know how many unique courses that a selected group of students take?

- Create the data model in your Python program.

END