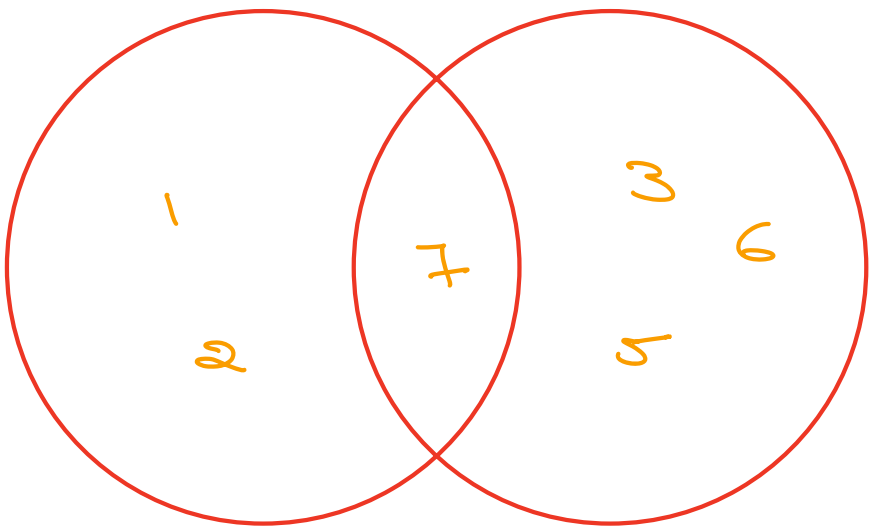
# Agenda

- Mutables vs Immutables
- Tuple
  - Definition and Use-Case
  - Properties of Tuple
- Sets
  - Set Methods
  - Set operations




Set1 → {1, 2, 7}

Set2 → {3, 5, 6, 7}

## Questions

1. Set1 ∪ Set2
2. Set1 ∩ Set2
3. Set1 − Set2 (difference)
4. Set2 − Set1
5. Set1 ^ Set2 (Symmetric-difference)

e.g.

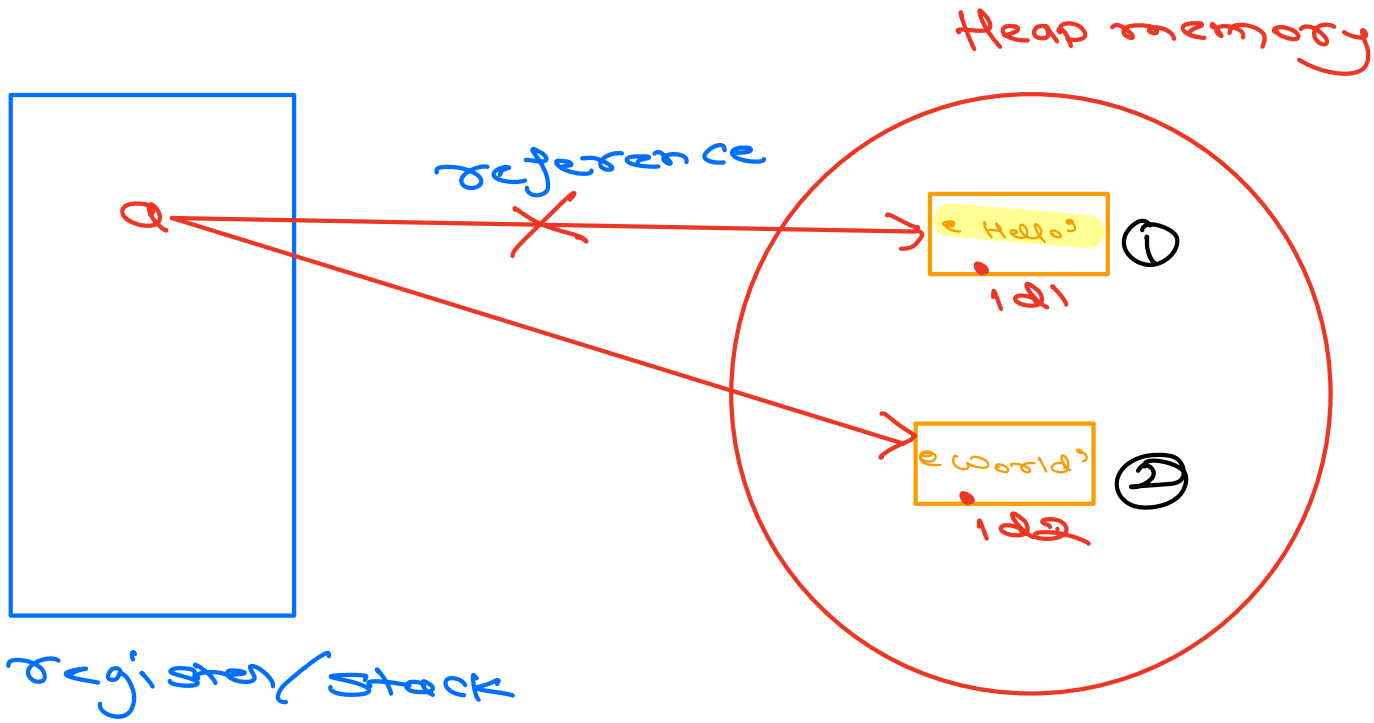
A ∪ B − A ∩ B

# Mutables vs Immutables

| Mutables | Immutables |
| --- | --- |
| list | String |
| set | Tuple |
| dictionary | int |
| | bool and Everything Else |

* Mutables: Something that can be changed/updated
* Immutables: Cannot changed/updated

---

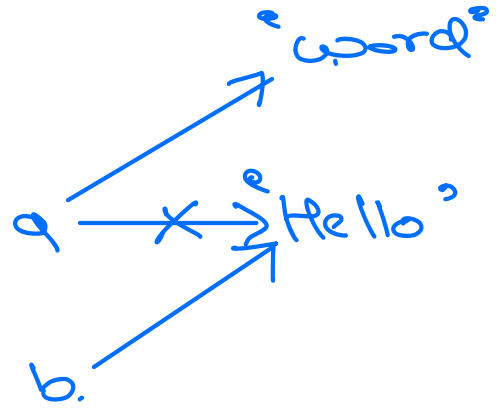


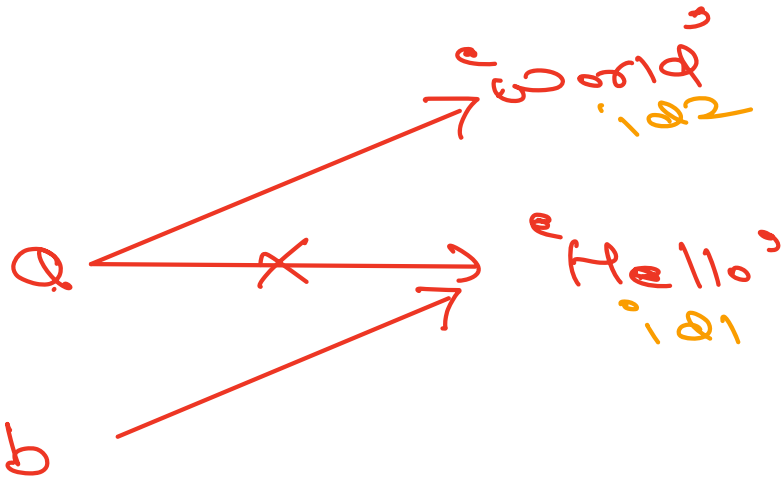

immutables

a = 'Hello' ①

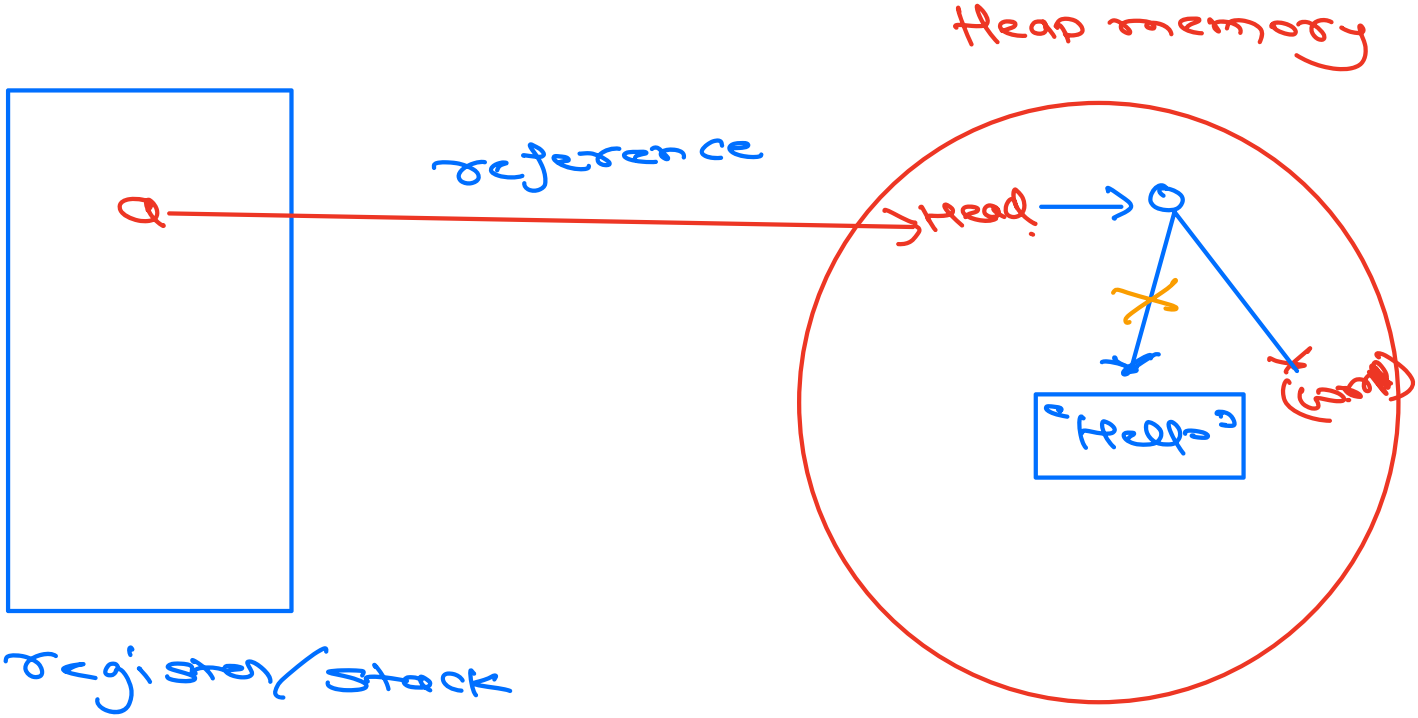
a = 'World' ②

# Reference Variable



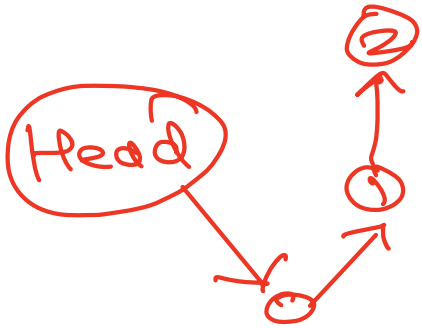




a = "Hello"
b = a
a = "World"




a = ["Hello"]

a[0] = "World"



# Tuple

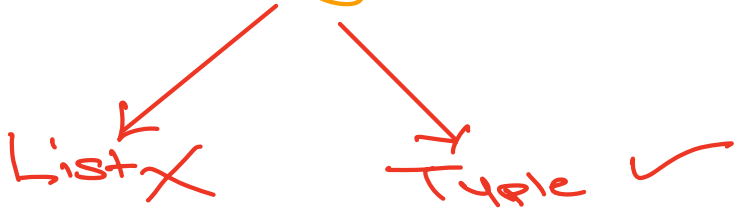
(immutable List)

```
a = tuple()
a = (1,)
a = ()
```

8 planets in Solar System

Tuples are faster due to immutable

Use-case: Name of all 8 planets



# Set

- It store Unique Elements (No duplicates)
- Insertion Ordered Pyth > 3.7

Set ⟶ 1, 2, 2, 3, 2

↓

{1, 2, 3}

S1 = set()

S1 = {1, 2}

S1 = {} ✗ ⟶ dictionary

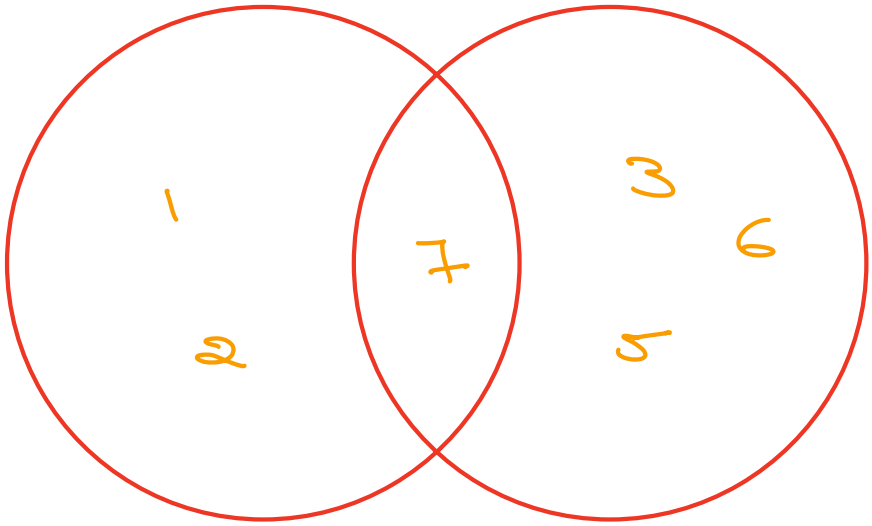
Note :

Set can only contain Immutable (hashables)

↑

Searching O(1) (very fast)

# Set operations




## Questions

① Set1 ∪ Set2 ⇒ Set1 | Set2

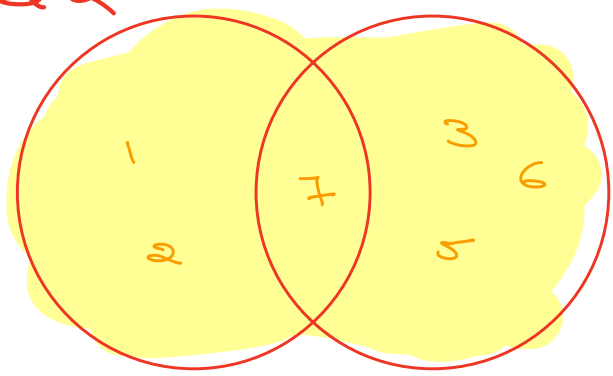
② Set1 ∩ Set2

③ Set1 − Set2 (difference)

④ Set2 − Set1 ,,

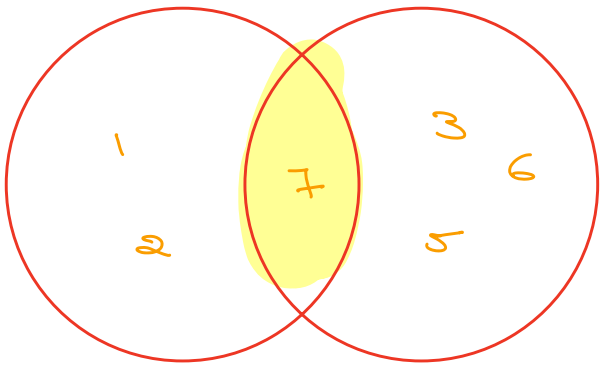
⑤ Set1 ^ Set2 (Symmetric − difference)

Set1 ∪ Set2



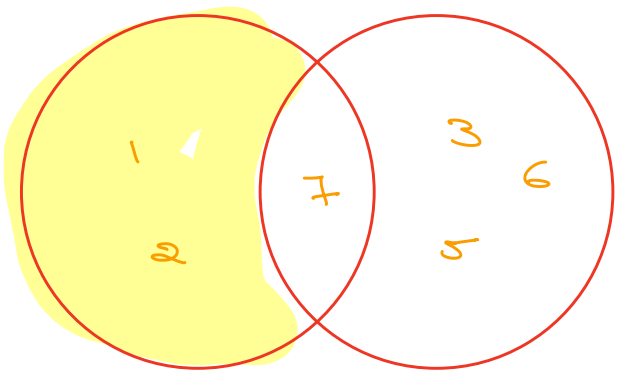

Set1 | Set2

Set1 ∩ Set2



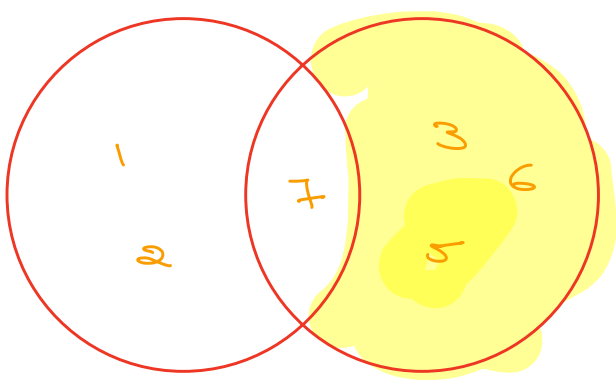

Set1 & Set2

Set1 − Set2




Set1 - Set2

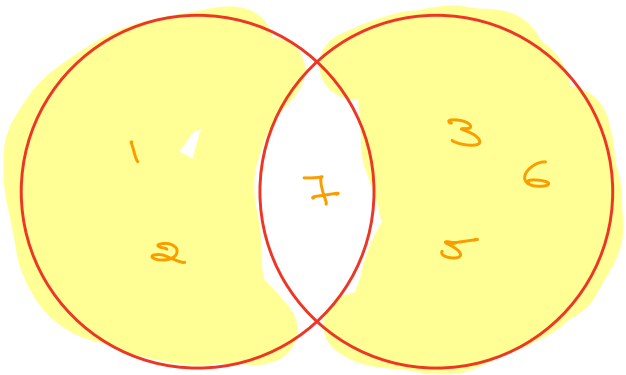
Set2 − Set1




Set2 - Set1

Set1 ^ Set2 (Symmetric – difference)

Set1 ∪ Set2 — Set1 ∩ Set2




↖ XOR

Set1 ^ Set2