LECTURE 12

DOHYUNG KIM

WHAT IS DISCUSSED IN THE LAST CLASS

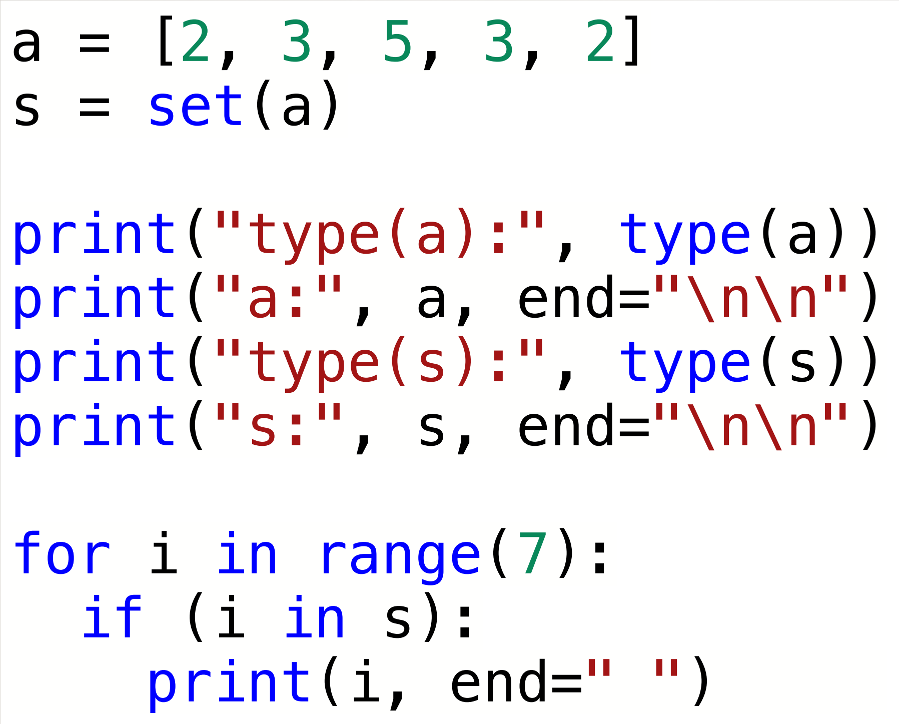
- Animation
- https://repl.it/languages/tkinter

TODAY, WE WILL LEARN ABOUT

- Set

SET

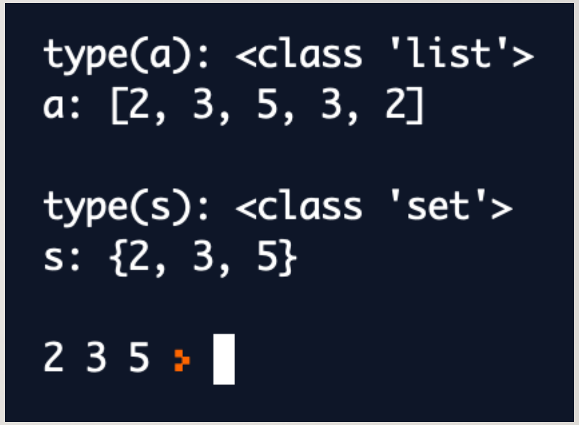
- An unordered collection of distinct item



```
a = [2, 3, 5, 3, 2]
s = set(a)

print("type(a):", type(a))
print("a:", a, end="\n\n")
print("type(s):", type(s))
print("s:", s, end="\n\n")

for i in range(7):
  if (i in s):
    print(i, end=" ")
```



```
type(a): <class 'list'>
a: [2, 3, 5, 3, 2]

type(s): <class 'set'>
s: {2, 3, 5}

2 3 5 ›
```

CREATING SETS

- Creating an empty set

```
s1 = set()
print(s1)
```

- Creating a set from a list

```
s1 = set([1, 3, 2, 2, 5])
print(s1)

a = ["python", "programming"]
s2 = set(a)
print(s2)
```

- Creating a set from any iterable object

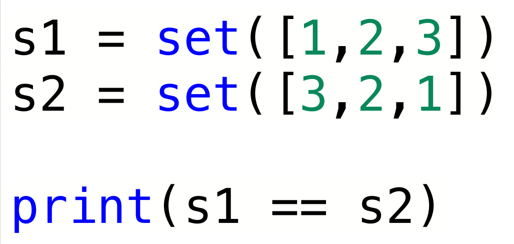
```
s = set("Good")
print(s)
```

* Note : { } is not an empty set

```
s = { }
print(type(s))
```

PROPERTIES OF SETS

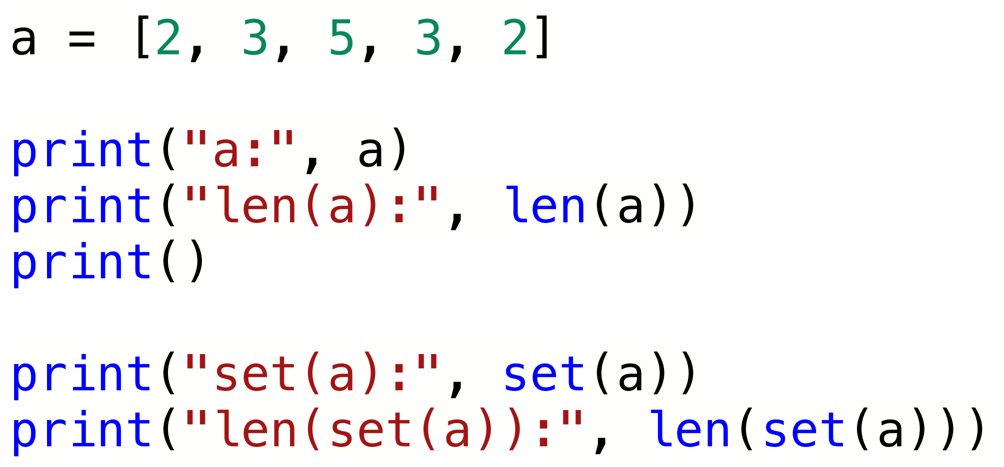
- Sets are unordered



```
s1 = set([1,2,3])
s2 = set([3,2,1])

print(s1 == s2)
```

- Elements are unique

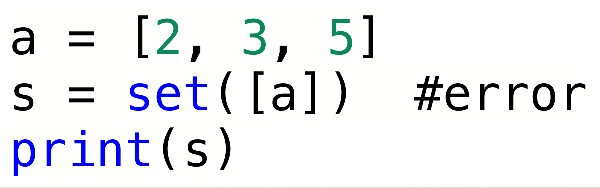


```
a = [2, 3, 5, 3, 2]

print("a:", a)
print("len(a):", len(a))
print()

print("set(a):", set(a))
print("len(set(a)):", len(set(a)))
```

- Element must be immutable



```
a = [2, 3, 5]
s = set([a])  #error
print(s)
```

SET OPERATIONS

- len()

```
s = { 1, 2, 3, 2, 1}
print(len(s))
```

- s.copy() - shallow copy

```
s1 = {2, 3, 5}
s2 = s1.copy()
s2.add(7)
print(s1)
print(s2)
```

- s.clear()

```
s = {2, 3, 5}
s.clear()
print(s, len(s))
```

- in / not in

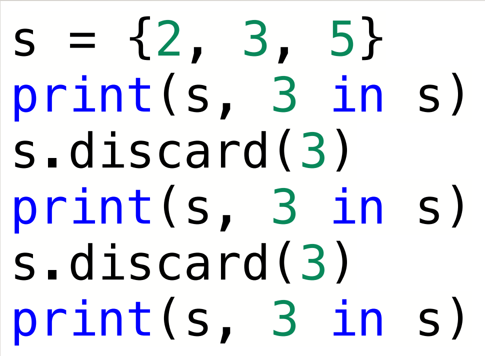
```
s = {2, 3, 5}
print(0 in s)
print(3 in s)
print(0 not in s)
```

- s.add() / s.remove()

```
s = {2, 3, 5}
print(s, 4 in s)
s.add(7)
print(s, 4 in s)
print()
print("removing 3")
s.remove(3)
print(s)
print("removing 10")
s.remove(10)
print(s)
```

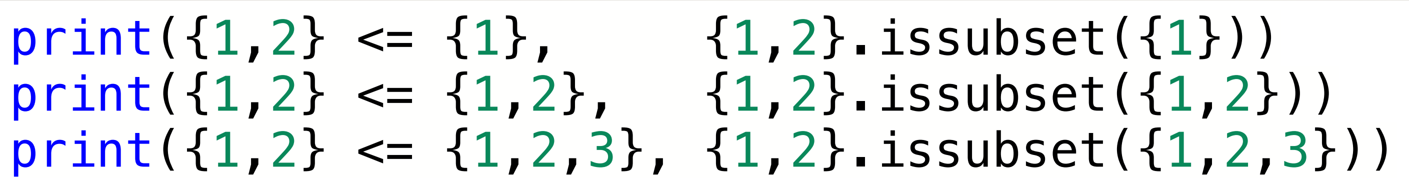
SET OPERATIONS

- s.discard()



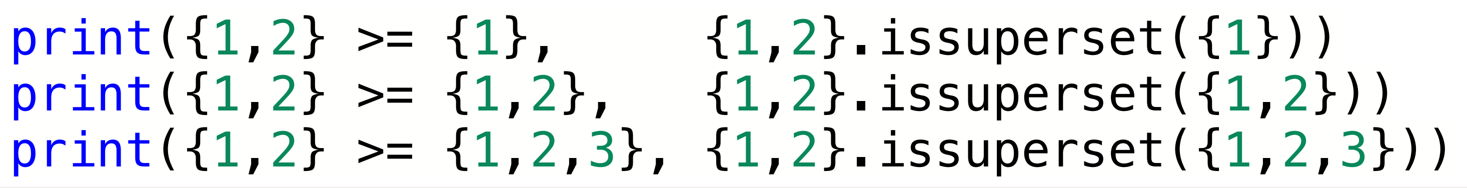
```
s = {2, 3, 5}
print(s, 3 in s)
s.discard(3)
print(s, 3 in s)
s.discard(3)
print(s, 3 in s)
```

- s.issubset() and operator **<=**



```
print({1,2} <= {1},     {1,2}.issubset({1}))
print({1,2} <= {1,2},   {1,2}.issubset({1,2}))
print({1,2} <= {1,2,3}, {1,2}.issubset({1,2,3}))
```

- s.issuperset() and operator **>=**



```
print({1,2} >= {1},     {1,2}.issuperset({1}))
print({1,2} >= {1,2},   {1,2}.issuperset({1,2}))
print({1,2} >= {1,2,3}, {1,2}.issuperset({1,2,3}))
```

SET OPERATIONS

- s.union() and operator **|**

```
print({1,2} | {1},     {1,2}.union({1}))
print({1,2} | {1,3},   {1,2}.union({1,3}))
s1 = {1,2}
s2 = s1 | {1,3}
print(s1, s2)
```

- s.intersection() and operator **&**

```
print({1,2} & {1},     {1,2}.intersection({1}))
print({1,2} & {1,3},   {1,2}.intersection({1,3}))
s1 = {1,2}
s2 = s1 & {1,3}
print(s1, s2)
```

SET OPERATIONS

- s.difference() and operator **-**

```
print({1,2} - {1},     {1,2}.difference({1}))
print({1,2} - {1,3},   {1,2}.difference({1,3}))
s1 = {1,2}
s2 = s1 - {1,3}
print(s1, s2)
```

- s.update() and operator **|=**

```
s = {1, 2, 3}
t = {1, 2, 4}
u = {1, 3, 5}
s.update(u)
t |= u
print(s, t, u)
```

QUESTION?