

```
# collection = single "variable" used to store multiple values
# List = [] ordered and changeable. Duplicates OK
# Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
# Tuple = () ordered and unchangeable. Duplicates OK. FASTER
```

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7
8 print(fruits[1])
```

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7
8 print(fruits[::-1])
```


```
['coconut', 'banana', 'orange', 'apple']
```

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7
8 print(fruits[:2])
```

```
# collection = single "variable" used to store multiple values
# List = [] ordered and changeable. Duplicates OK
# Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
# Tuple = () ordered and unchangeable. Duplicates OK. FASTER
```

```
fruits = ["apple", "orange", "banana", "coconut"]
```

```
# print(fruits[0])
```

```
R  for fruit in fruits:
    print(x)
```

```
Pr...  main.py
practice C:\Users\HP\PyCharmProjects\practice
> venv
main.py
External Libraries
Scratches and Consoles

1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7 print(dir(fruits))
8
9 # print(fruits[0])
10 # for fruit in fruits:
11     # print(fruit)

Run: main
C:\Users\HP\PyCharmProjects\practice\venv\python.exe C:\Users\HP\PyCharmProjects\practice\main.py
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__']
Process finished with exit code 0
```

```
main.py x
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 print("pineapple" in fruits)
11
12 # print(fruits[0])
```

main.py

main.py x

C:\Users\HP\PycF

.py

Libraries

es and Consoles

```
10 # print("pineapple" in fruits)
11
12 # fruits[0] = "pineapple"
13 # fruits.append("pineapple")
14 # fruits.remove("apple")
15 # fruits.insert(0, "pineapple")
16 # fruits.sort()
17 # fruits.reverse()
18 # fruits.clear()
19 # print(fruits.index("apple"))
20 print(fruits.count("pine|"))
21
22 print(fruits)
```




main



main.py

C:\Users\HP\PyCh

Libraries
and Consoles

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 # fruits[0] = "pineapple"
13 # fruits.append("pineapple")
```

Notifications

```
main.py x
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = {"apple", "orange", "banana", "coconut"}
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 # fruits[0] = "pineapple"
13 # fruits.append("pineapple")
```

The image shows a PyCharm IDE window with a project named 'practice'. The file explorer on the left shows the project structure: 'practice' (C:\Users\HP\PycharmProjects\practice) containing a 'venv' folder and a 'main.py' file. The 'main.py' file is open in the editor, showing the following code:

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = {"apple", "orange", "banana", "coconut"}
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 print(fruits)
13 #for fruit in fruits:
```

The output of the script is displayed in the 'Run' console at the bottom, showing the set of fruits: {'orange', 'apple', 'banana', 'coconut'}.

The image shows a Python IDE with a project named 'practice' and a file named 'main.py'. The code in 'main.py' defines a set 'fruits' containing 'apple', 'orange', 'banana', and 'coconut'. It includes comments explaining that a set is unordered and immutable. The code attempts to print the first element of the set using indexing. The output window shows a 'TypeError: 'set' object is not subscriptable' error.

```
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = {"apple", "orange", "banana", "coconut"}
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 print(fruits[0])
13
```

Run: main x

print(fruits[0])
TypeError: 'set' object is not subscriptable

You cannot use indexing on a set because it is unordered.

practice C:\Users\HP\PyCh

venv

main.py

External Libraries

atches and Consoles

5

6

7

8

9

10

11

12

13

14

15

16

```
fruits = {"apple", "orange", "banana", "coconut"}
```

```
# print(dir(fruits))
```

```
# print(help(fruits))
```

```
# print(len(fruits))
```

```
# print("pineapple" in fruits)
```

```
# fruits.add("pineapple")
```

```
# fruits.remove("apple")
```

```
# fruits.pop()
```

```
fruits.clear()
```

```
main.py x
HP\Pyct
les
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ("apple", "orange", "banana", "coconut", "coconut")
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 # fruits.add("pineapple")
13 # fruits.remove("apple")
```

```
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ("apple", "orange", "banana", "coconut", "coconut")
7 # print(dir(fruits))
8 # print(help(fruits))
9 # print(len(fruits))
10 # print("pineapple" in fruits)
11
12 # print(fruits.index("apple"))
13 print(fruits.count("coconut"))
14
15 print(fruits)
16 #for fruit in fruits:
17     #print(fruit)
```