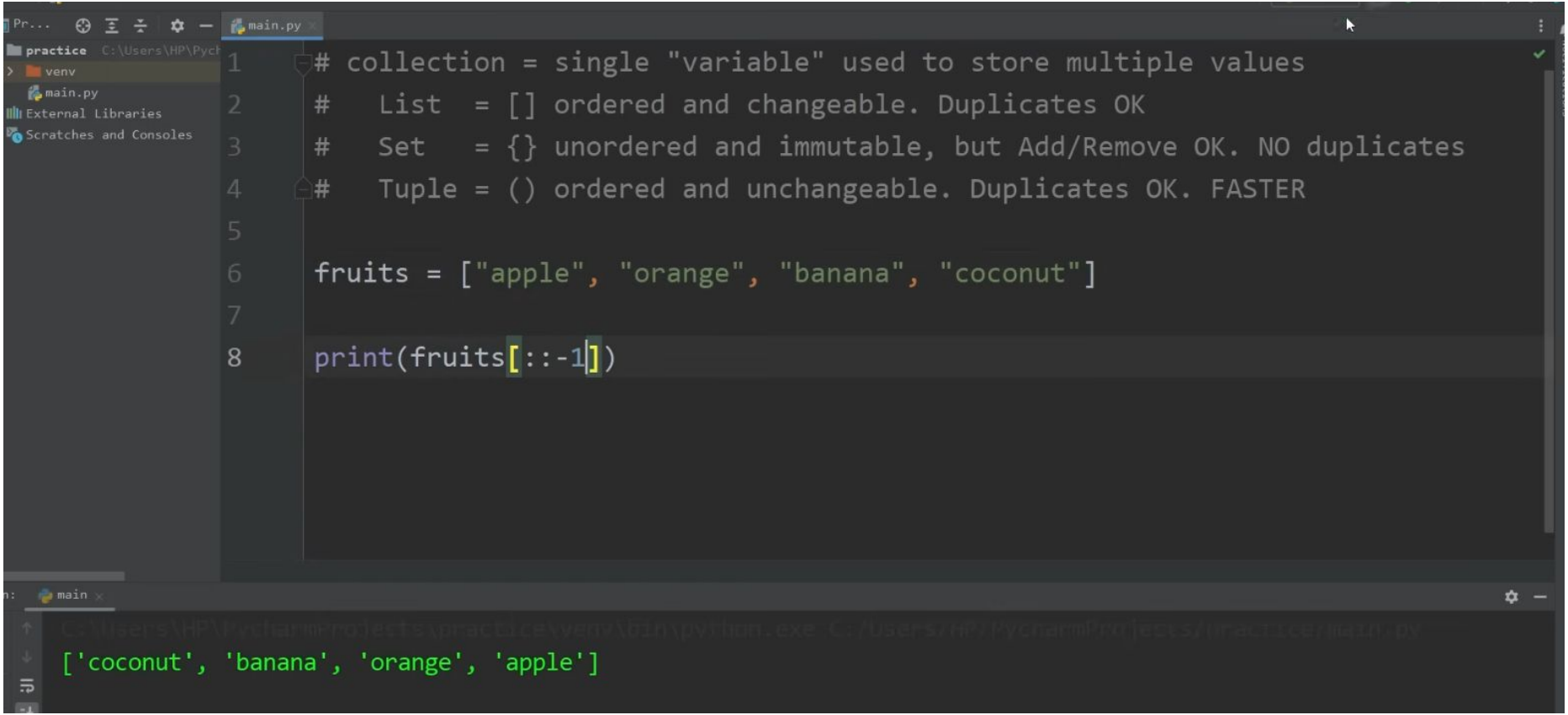
```
# collection = single "variable" used to store multiple values
#   List  = [] ordered and changeable. Duplicates OK
#   Set   = {} unordered and immutable, but Add/Remove OK. NO duplicates
#   Tuple = () ordered and unchangeable. Duplicates OK. FASTER
```

main.py

```
# 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[1])
```

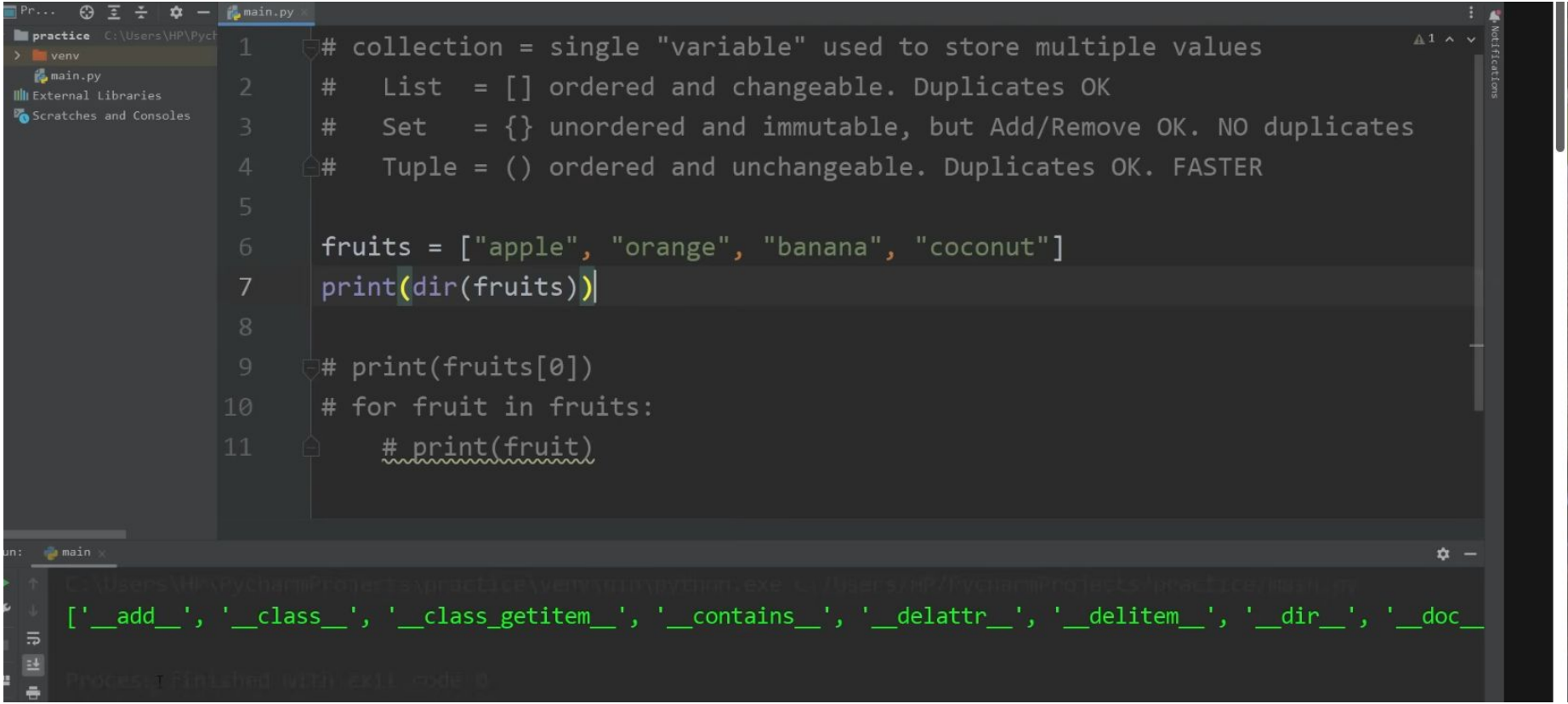

practice C:\Users\HP\Pych
venv
main.py
External Libraries
Scratches and Consoles
main.py
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and unchangeable. Duplicates OK. FASTER
5
6 fruits = ["apple", "orange", "banana", "coconut"]
7
8 print(fruits[::-1])
main
['coconut', 'banana', 'orange', 'apple']

main.py

```python
# 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[::2])
```

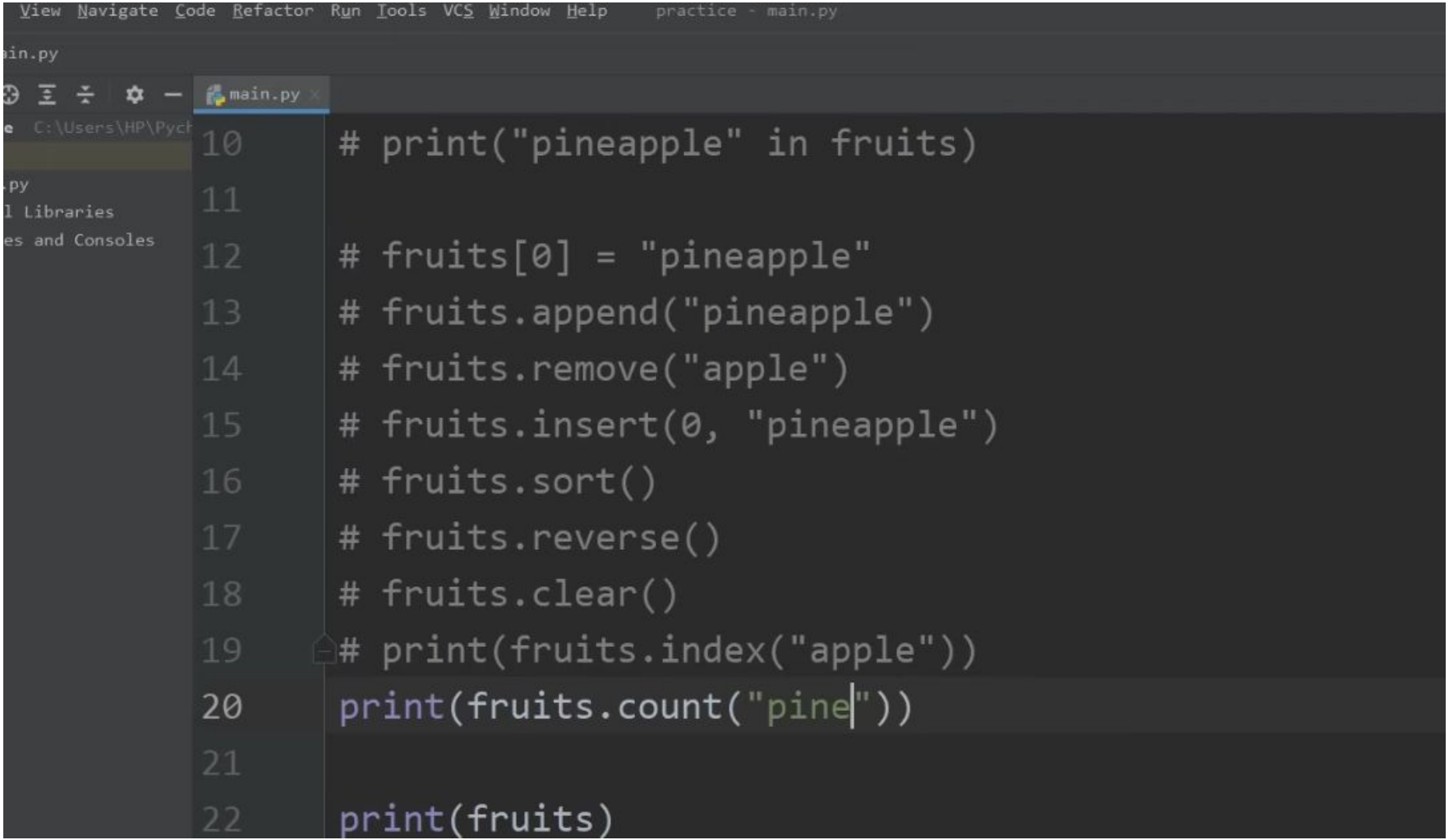
ain.py

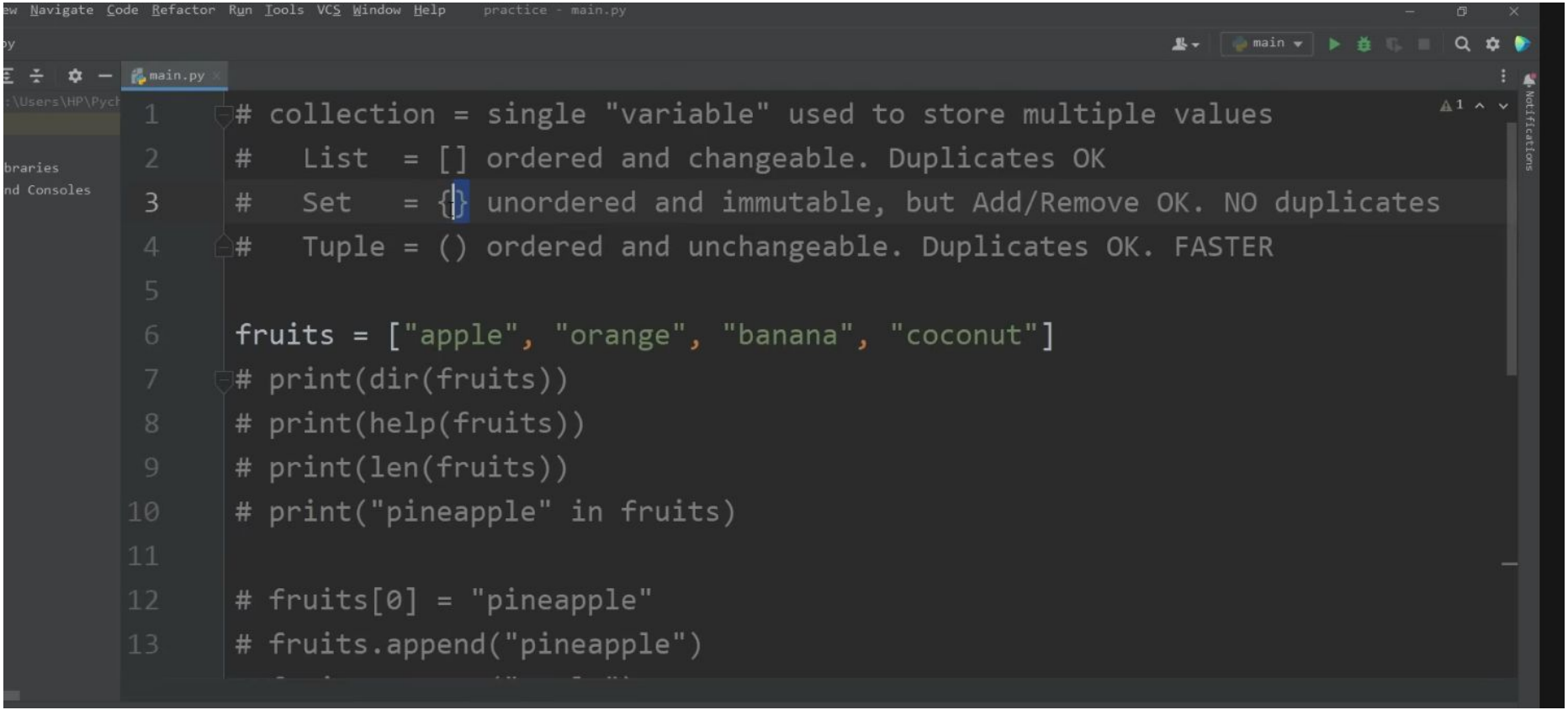
```
# collection = single "variable" used to store multiple values
#   List  = [] ordered and changeable. Duplicates OK
#   Set   = {} unordered and immutable, but Add/Remove OK. NO duplicates
#   Tuple = () ordered and unchangeable. Duplicates OK. FASTER

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

# print(fruits[0])

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

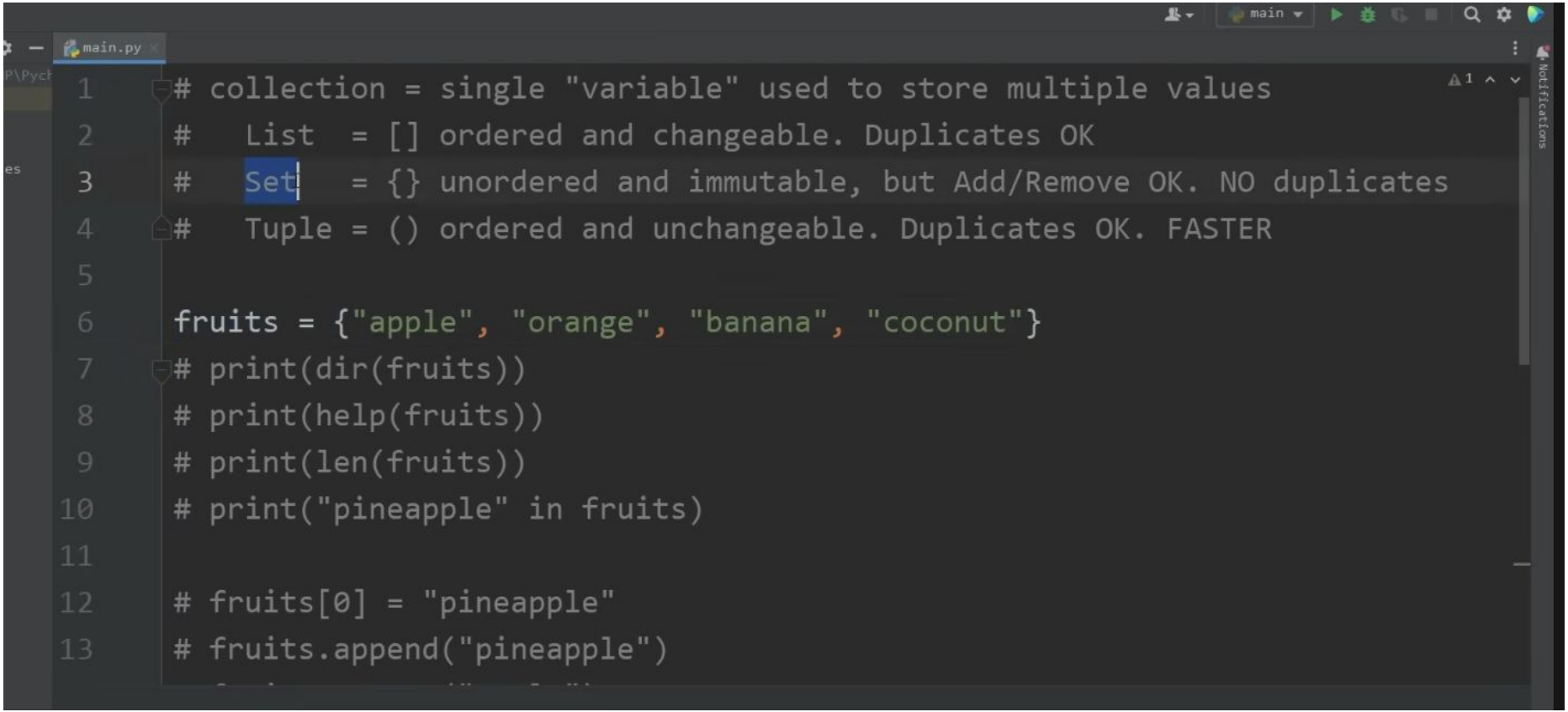

practice C:\Users\HP\Pycl
venv
main.py
External Libraries
Scratches and Consoles
main.py
1 # collection = single "variable" used to store multiple values
2 # List = [] ordered and changeable. Duplicates OK
3 # Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
4 # Tuple = () ordered and 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)
Notifications
main
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__
Process finished with exit code 0

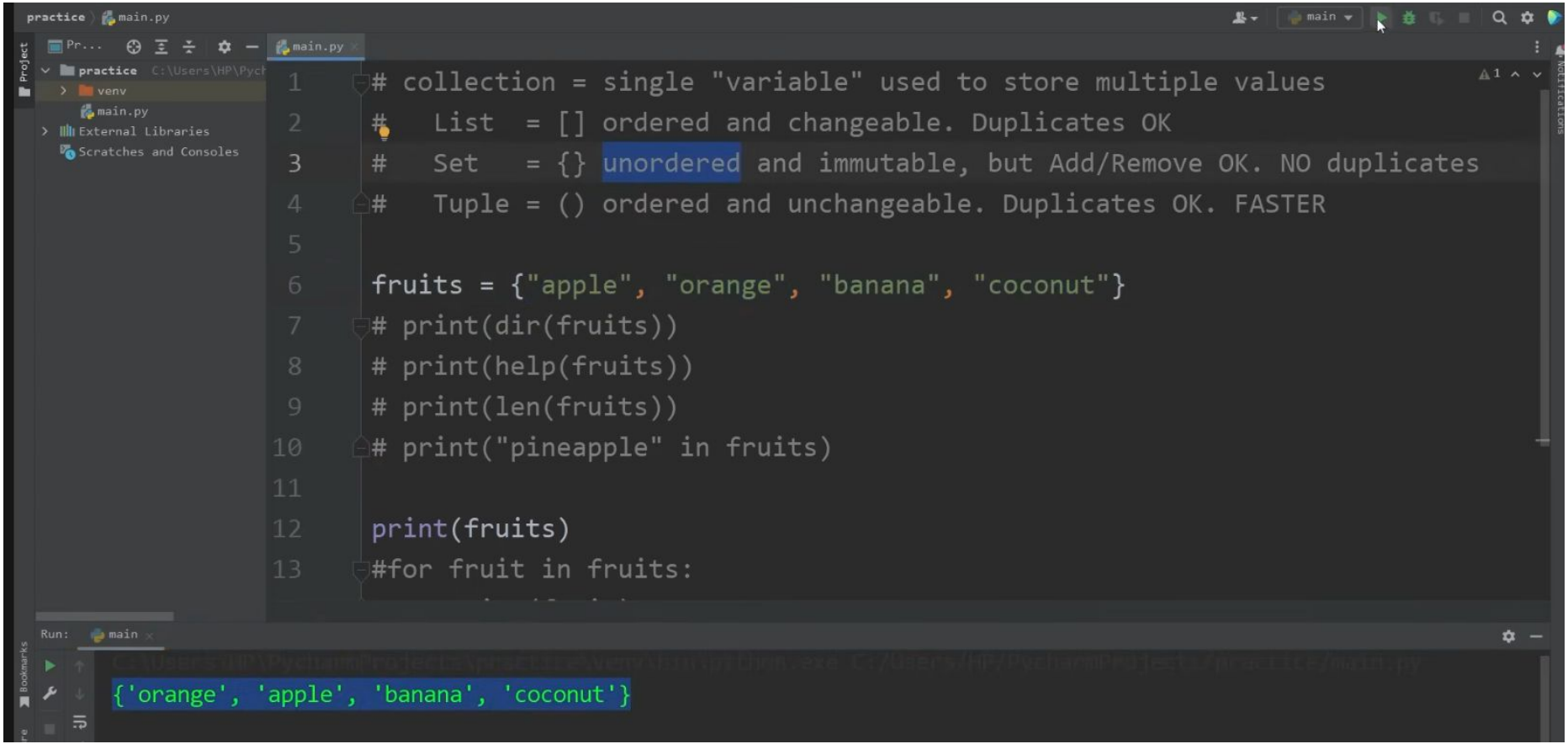
main.py

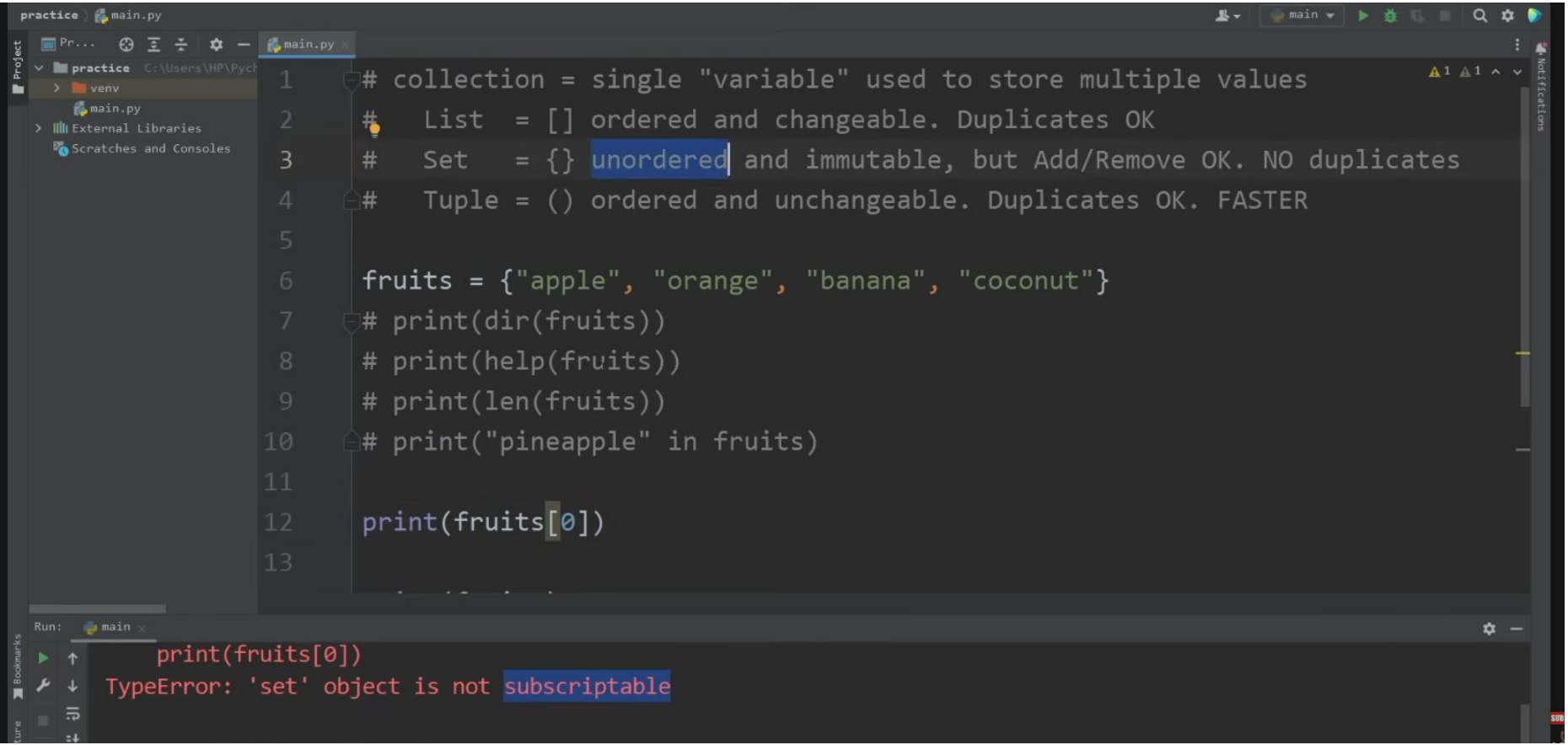
```python
# collection = single "variable" used to store multiple values
#   List  = [] ordered and changeable. Duplicates OK
#   Set   = {} unordered and immutable, but Add/Remove OK. NO duplicates
#   Tuple = () ordered and unchangeable. Duplicates OK. FASTER

fruits = ["apple", "orange", "banana", "coconut"]
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
print("pineapple" in fruits)

# print(fruits[0])
```


View Navigate Code Refactor Run Tools VCS Window Help
practice - main.py
main.py
# print("pineapple" in fruits)
# fruits[0] = "pineapple"
# fruits.append("pineapple")
# fruits.remove("apple")
# fruits.insert(0, "pineapple")
# fruits.sort()
# fruits.reverse()
# fruits.clear()
# print(fruits.index("apple"))
print(fruits.count("pine"))
print(fruits)

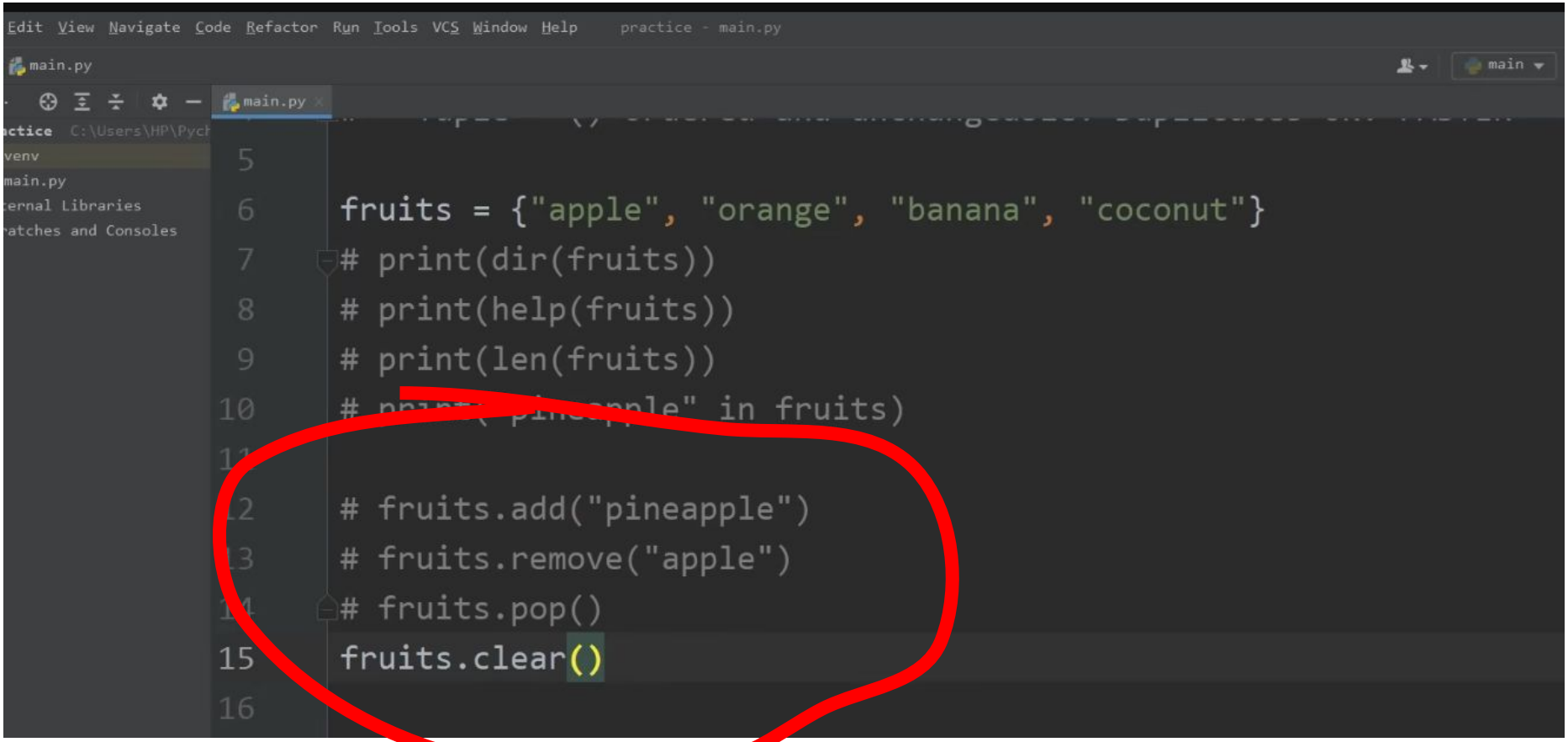

Navigate Code Refactor Run Tools VCS Window Help
practice - main.py
main
main.py
# collection = single "variable" used to store multiple values
#   List = [] ordered and changeable. Duplicates OK
#   Set   = {} unordered and immutable, but Add/Remove OK. NO duplicates
#   Tuple = () ordered and unchangeable. Duplicates OK. FASTER
fruits = ["apple", "orange", "banana", "coconut"]
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
# print("pineapple" in fruits)
# fruits[0] = "pineapple"
# fruits.append("pineapple")

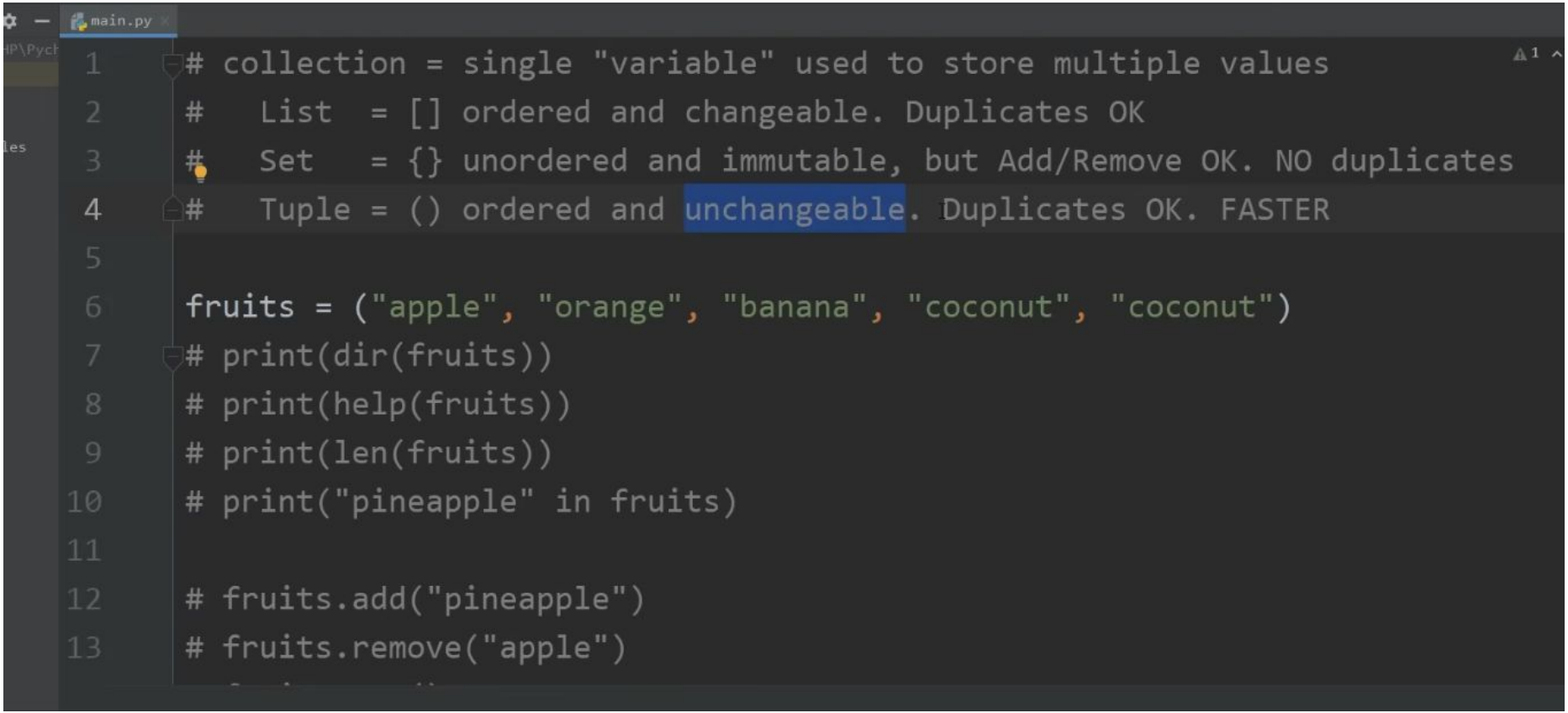

main.py
# collection = single "variable" used to store multiple values
#   List = [] ordered and changeable. Duplicates OK
#   Set   = {} unordered and immutable, but Add/Remove OK. NO duplicates
#   Tuple = () ordered and unchangeable. Duplicates OK. FASTER
fruits = {"apple", "orange", "banana", "coconut"}
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
# print("pineapple" in fruits)
# fruits[0] = "pineapple"
# fruits.append("pineapple")

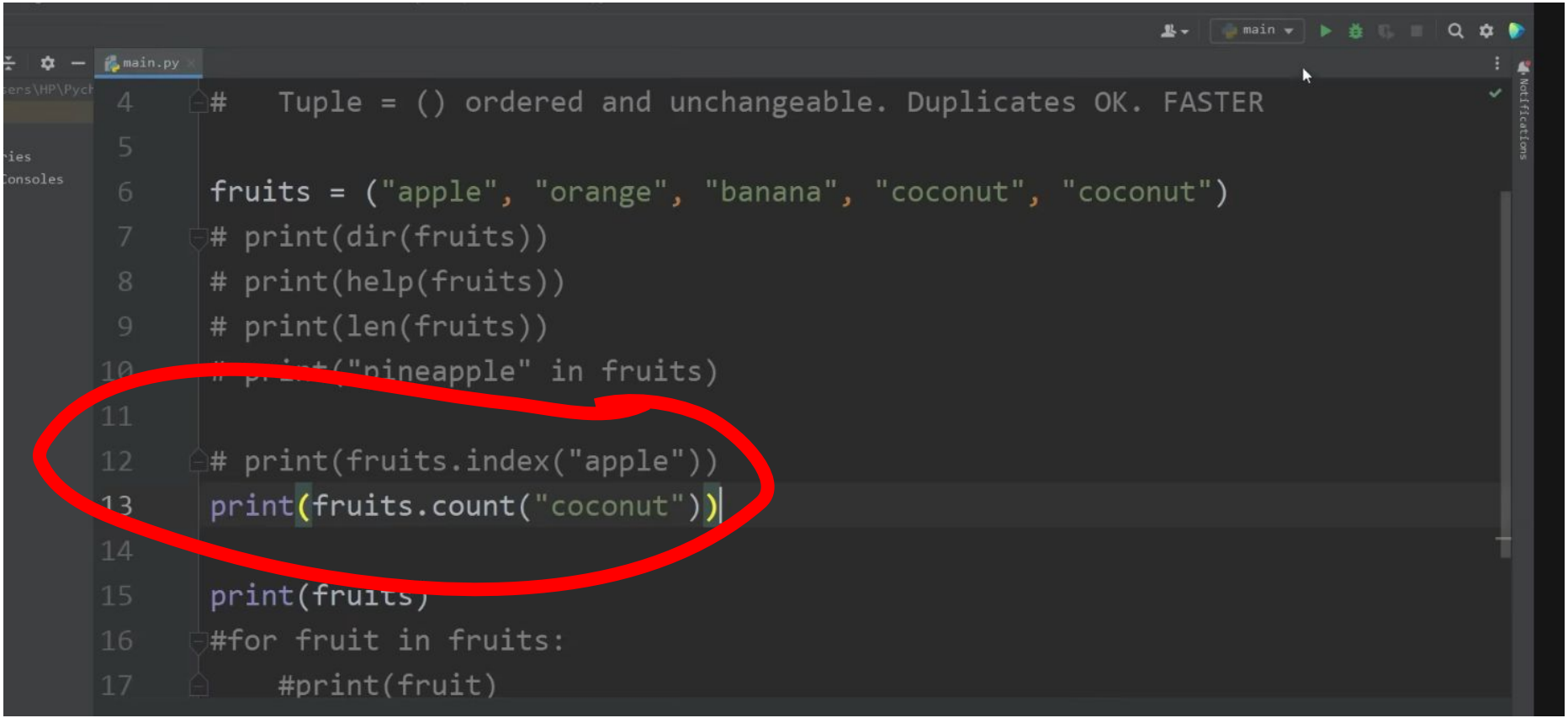

practice main.py
main
Project
practice C:\Users\HP\Pych
venv
main.py
External Libraries
Scratches and Consoles
main.py
# collection = single "variable" used to store multiple values
# List = [] ordered and changeable. Duplicates OK
# Set = {} unordered and immutable, but Add/Remove OK. NO duplicates
# Tuple = () ordered and unchangeable. Duplicates OK. FASTER
fruits = {"apple", "orange", "banana", "coconut"}
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
# print("pineapple" in fruits)
print(fruits)
#for fruit in fruits:
Run: main
{'orange', 'apple', 'banana', 'coconut'}



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


Edit View Navigate Code Refactor Run Tools VCS Window Help
practice - main.py
main.py
main
main.py
venv
main.py
ernal Libraries
atches and Consoles
fruits = {"apple", "orange", "banana", "coconut"}
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
in fruits)
# fruits.add("pineapple")
# fruits.remove("apple")
# fruits.pop()
fruits.clear()


main.py
# 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", "coconut")
# print(dir(fruits))
# print(help(fruits))
# print(len(fruits))
# print("pineapple" in fruits)
# fruits.add("pineapple")
# fruits.remove("apple")


main.py
main
Notifications
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)