python

Programming Essentials

Day 6

python

Cyrel P. Nicolas

TUPLE

A tuple is a collection of objects which ordered, immutable and allows duplicate values.

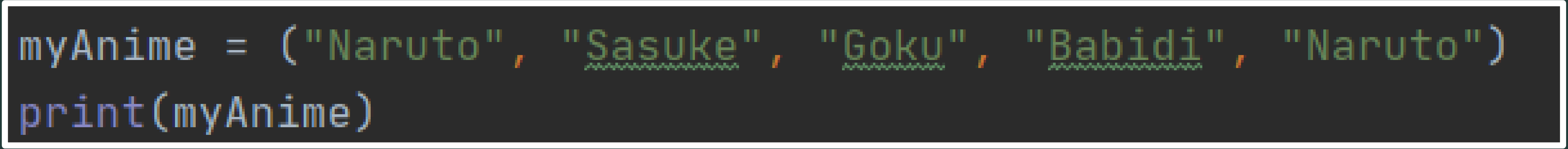
Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

```
tup1 = ('physics', 'chemistry', 1997, 2000)
tup2 = (1, 2, 3, 4, 5 )
tup3 = ("a", "b", "c", "d")
```

Tuples are ordered, if items have a defined order, and that order will not change.

Tuples are unchangeable, it means we cannot update, add or delete items after the tuple has been created.

Since tuple are indexed, tuples can have items with the same value, it means it allows duplicate items

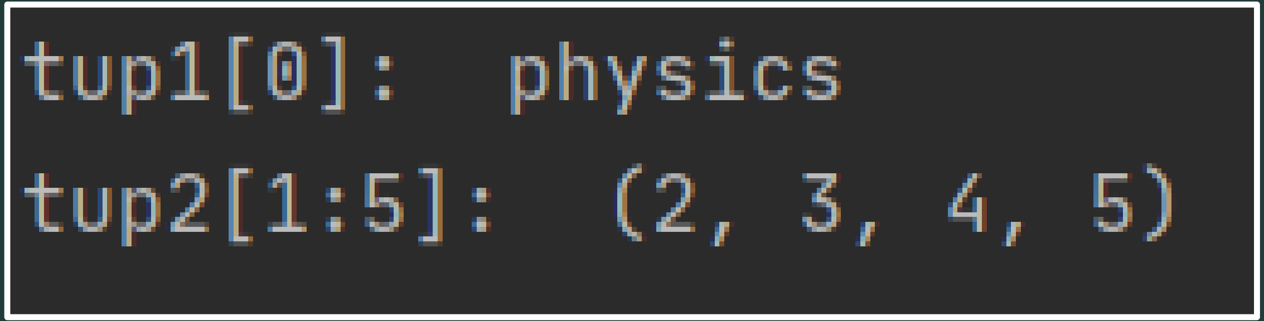


```
myAnime = ("Naruto", "Sasuke", "Goku", "Babidi", "Naruto")
print(myAnime)
```

Like string indices, tuple indices start at 0, and they can be sliced, concatenated, and so on.

ACCESSING TUPLE

```
tup1 = ('physics', 'chemistry', 1997, 2000)
tup2 = (1, 2, 3, 4, 5, 6, 7)

print ("tup1[0]: ", tup1[0])
print ("tup2[1:5]: ", tup2[1:5])
```



```
tup1[0]:  physics
tup2[1:5]:  (2, 3, 4, 5)
```

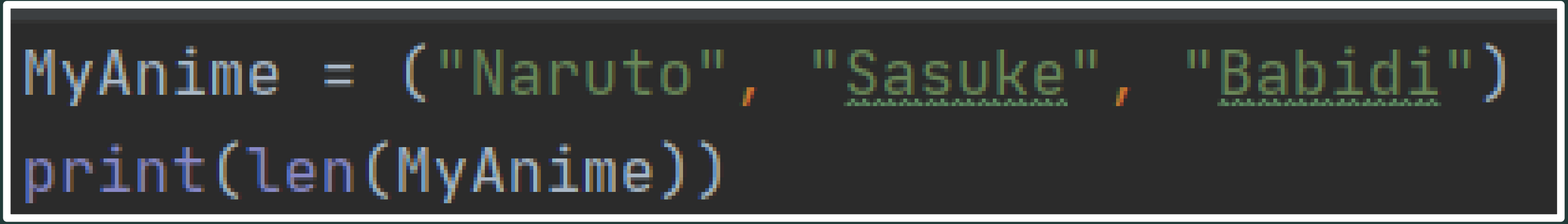
USEFUL TUPLE FUNCTIONS

python Programming Essentials Day 6

Function Name	Description
len(t)	Gives the total length of the tuple.
max(t)	Returns item from the tuple with max value.
min(t)	Returns item from the tuple with min value.
tuple(l)	Converts a list into tuple.
del t	Delete tuple
+	Concatenate Tuple
in	Checks if value exist in tuple

Tuple Length

To determine how many items a tuple has, use the len() function:



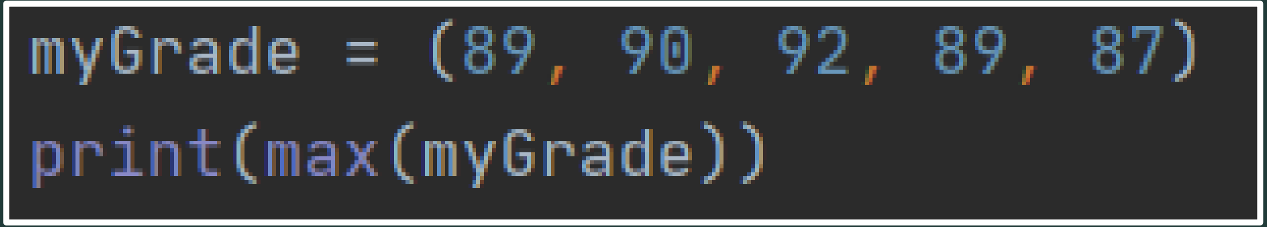
```python
MyAnime = ("Naruto", "Sasuke", "Babidi")
print(len(MyAnime))
```

Output

```
3
```

Tuple Max

To determine the highest value, use max() function:

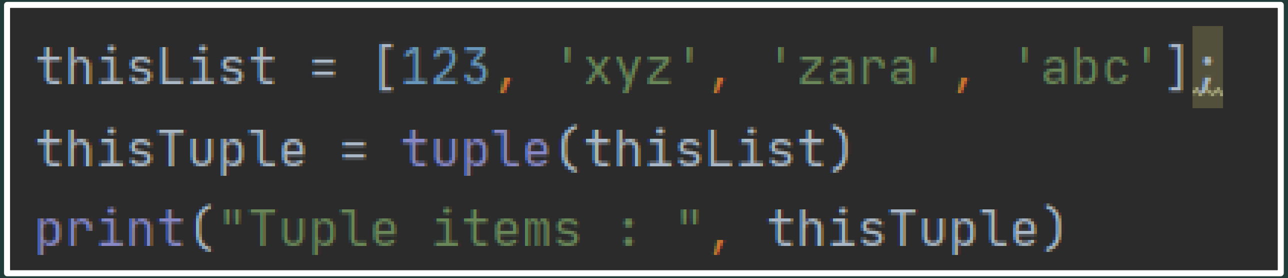


```python
myGrade = (89, 90, 92, 89, 87)
print(max(myGrade))
```

Output

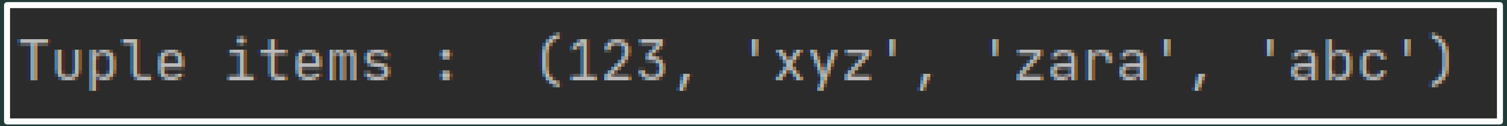
```
92
```

Convert List to Tuple



```python
thisList = [123, 'xyz', 'zara', 'abc'];
thisTuple = tuple(thisList)
print("Tuple items : ", thisTuple)
```

Output

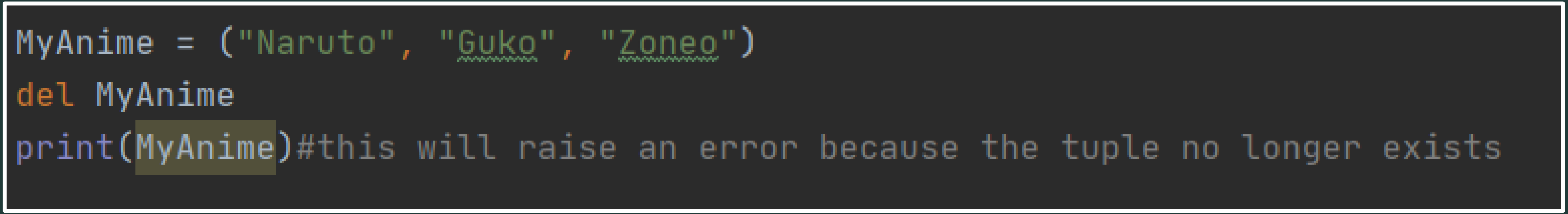


```
Tuple items :  (123, 'xyz', 'zara', 'abc')
```

Delete Tuple

Note: You cannot remove items in a tuple.

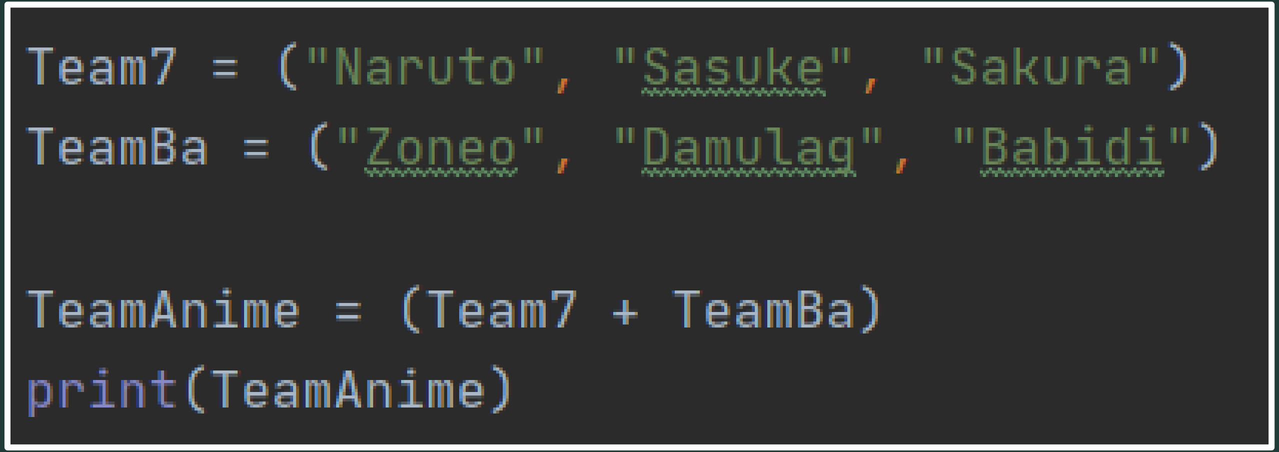
Tuples are unchangeable, so you cannot remove items from it, but you can delete the tuple completely:



```
MyAnime = ("Naruto", "Guko", "Zoneo")
del MyAnime
print(MyAnime)#this will raise an error because the tuple no longer exists
```

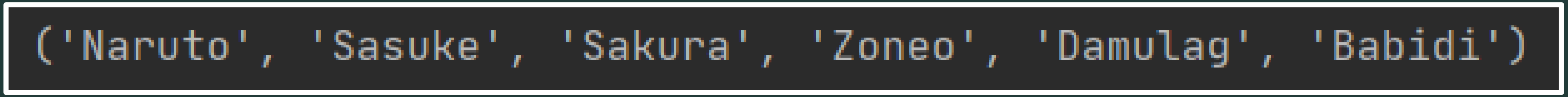
Concatenating

The + operator can be used to concatenate two or more tuples together.



```
Team7 = ("Naruto", "Sasuke", "Sakura")
TeamBa = ("Zoneo", "Damulag", "Babidi")

TeamAnime = (Team7 + TeamBa)
print(TeamAnime)
```

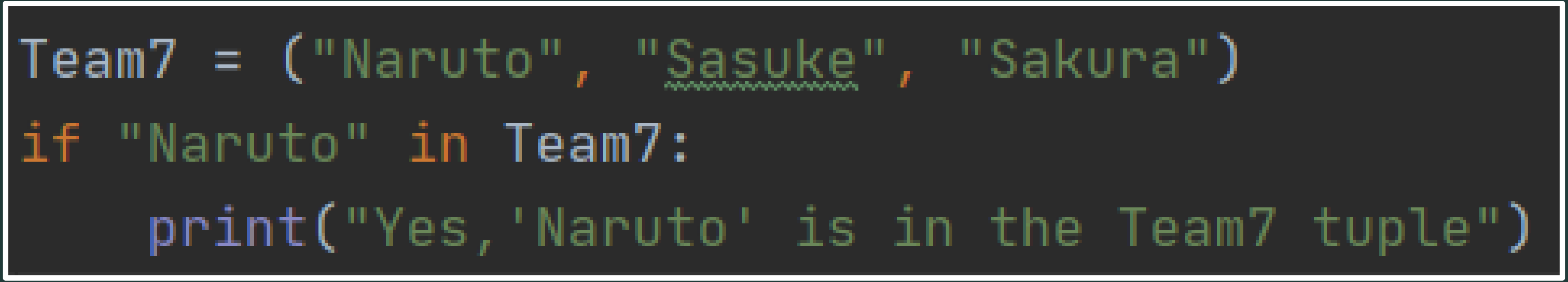
Output



```
('Naruto', 'Sasuke', 'Sakura', 'Zoneo', 'Damulag', 'Babidi')
```

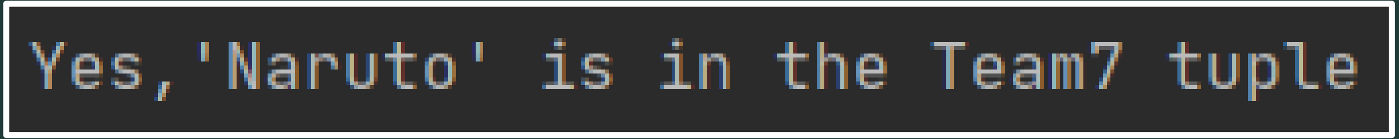
Checks if value exists in tuple

To determine if a specified item is present in a tuple use the **in** keyword:



```python
Team7 = ("Naruto", "Sasuke", "Sakura")
if "Naruto" in Team7:
    print("Yes,'Naruto' is in the Team7 tuple")
```

Output



```
Yes,'Naruto' is in the Team7 tuple
```

The + operator can be used to concatenate two or more tuples together.

DICTIONARY

Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Keys are unique within a dictionary while values may not be.

A dictionary is a collection which is unordered, changeable and does not allow duplicates.

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
```

The + operator can be used to concatenate two or more tuples together.

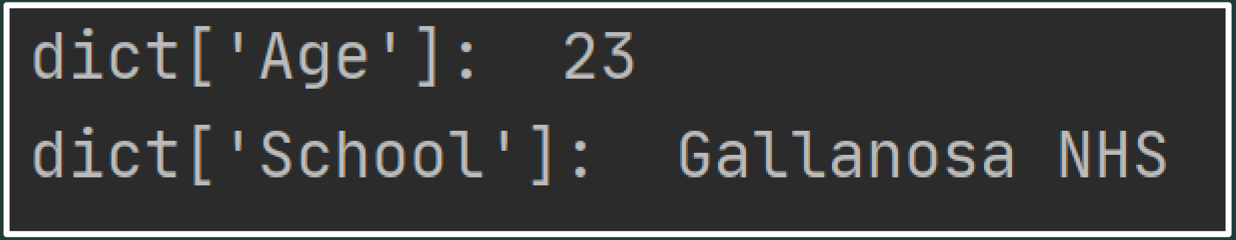
ACCESSING DICTIONARY

```
dict = {"Name":"Cyrel", "Age": 19, "Class":"1st"}
print("dict['Name']: ", dict['Name'])
print("dict['Age']: ", dict['Age'])
print(dict["Class"])
```

UPDATING DICTIONARY

Programming Essentials
Day 6

```
dict = {"Name":"Cyrel", "Age": 19, "Class":"1st"}

dict['Age'] = 23  # update existing entry
dict['School'] = "Gallanosa NHS" # Add new entry

print("dict['Age']: ", dict['Age'])
print("dict['School']: ", dict['School'])
```



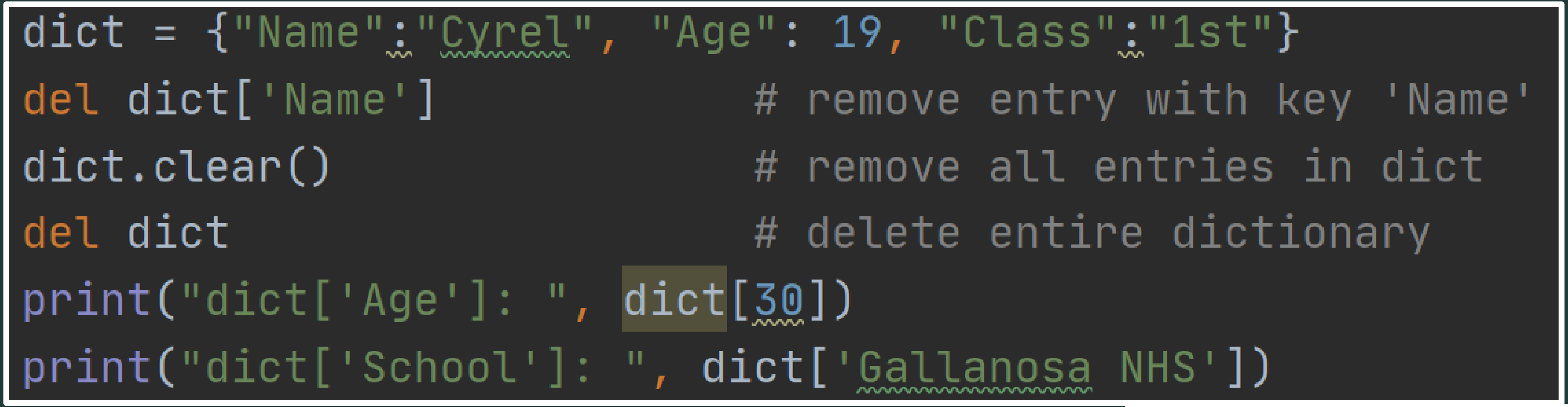
```
dict['Age']:  23
dict['School']:  Gallanosa NHS
```

DELETING DICTIONARY ELEMENTS

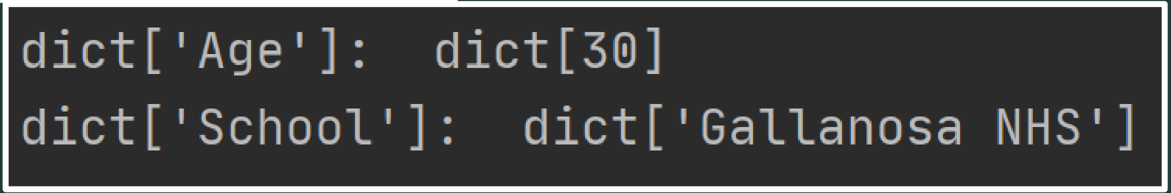
Programming Essentials
Day 6

You can either remove individual dictionary elements or clear the entire contents of a dictionary.

You can also delete entire dictionary in a single operation. To explicitly remove an entire dictionary, just use the del statement.



```
dict = {"Name":"Cyrel", "Age": 19, "Class":"1st"}
del dict['Name']                # remove entry with key 'Name'
dict.clear()                    # remove all entries in dict
del dict                        # delete entire dictionary
print("dict['Age']: ", dict[30])
print("dict['School']: ", dict['Gallanosa NHS'])
```



```
dict['Age']:   dict[30]
dict['School']:   dict['Gallanosa NHS']
```

The + operator can be used to concatenate two or more tuples together.

SETS

A set is a collection which is unordered and unindexed. In Python, sets are written with curly brackets.

```
fruits= {"apple", "banana", "cherry"}
print(fruits)
```

Set items are unordered, unchangeable, and do not allow duplicate values.

The + operator can be used to concatenate two or more tuples together.

Unordered SETS

Unordered means that the 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.

Unchangeable SETS

Sets are unchangeable, meaning that we cannot change the items after the set has been created. Once a set is created, you cannot change its items, but you can add new items.

ADD ITEMS SETS

```
fruits = {"Apple", "Banana", "Santol"}
fruits.add("Guyabano")
print(fruits)
```

UPDATE ITEMS SETS

```
prog = {"C", "C#", "Java"}
prog.update(["PHP", "JS"])
print(prog)
```

The + operator can be used to concatenate two or more tuples together.

REMOVE ITEMS SETS

python
Programming Essentials
Day 6

To remove an item in a set, use the remove(), or the discard() method.

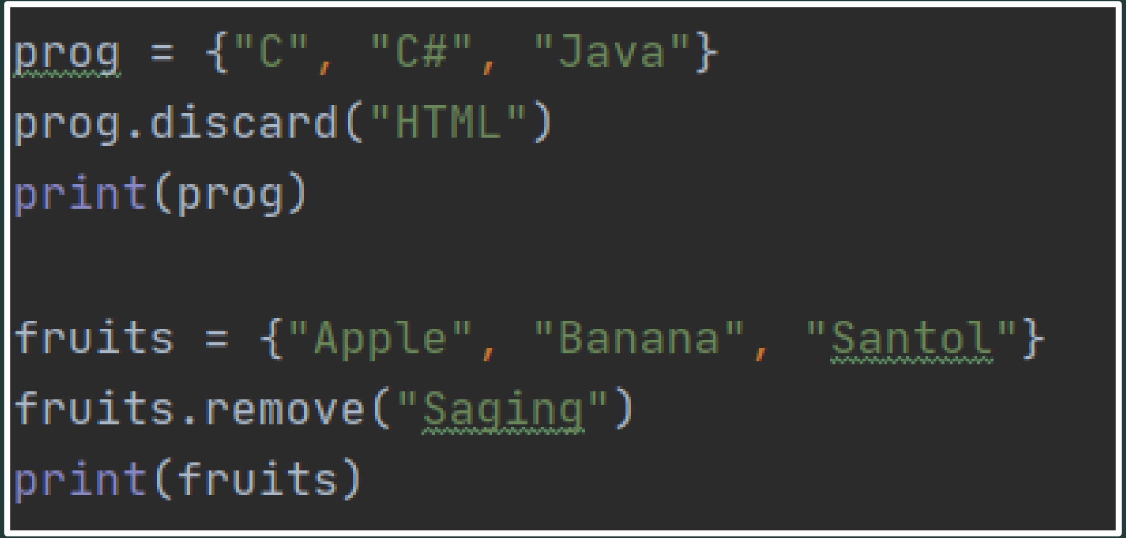
```
fruits = {"Apple", "Banana", "Santol"}
fruits.remove("Banana")
print(fruits)


prog = {"C", "C#", "Java"}
prog.discard("C")
print(prog)
```

The + operator can be used to concatenate two or more tuples together.

remove() vs discard()

The only difference between the two is that the discard() function leaves a set unchanged if the element is not present in the set.

On the other hand, the remove() function will raise an error in such a condition (if element is not present in the set).



```python
prog = {"C", "C#", "Java"}
prog.discard("HTML")
print(prog)

fruits = {"Apple", "Banana", "Santol"}
fruits.remove("Saging")
print(fruits)
```

The + operator can be used to concatenate two or more tuples together.

DAY 6 ACTIVITY 1

python Programming Essentials Day 6

1. Create a Record Keeping App

2. The application will ask the user to choose between:
    a. Add Data
    b. Delete Data
    c. End

3. If Add data, the application will ask the user to input key and its value
    a. Enter Key: Lastname
    b. Enter Value: Doe

4. Store the information in a dictionary

5. Display the Result

6. If Delete Data, the application will ask for the key
a. Enter Key: Lastname

7. Remove the item from the dictionary

8. Display the result

9. If End, display THANK YOU

MODULE

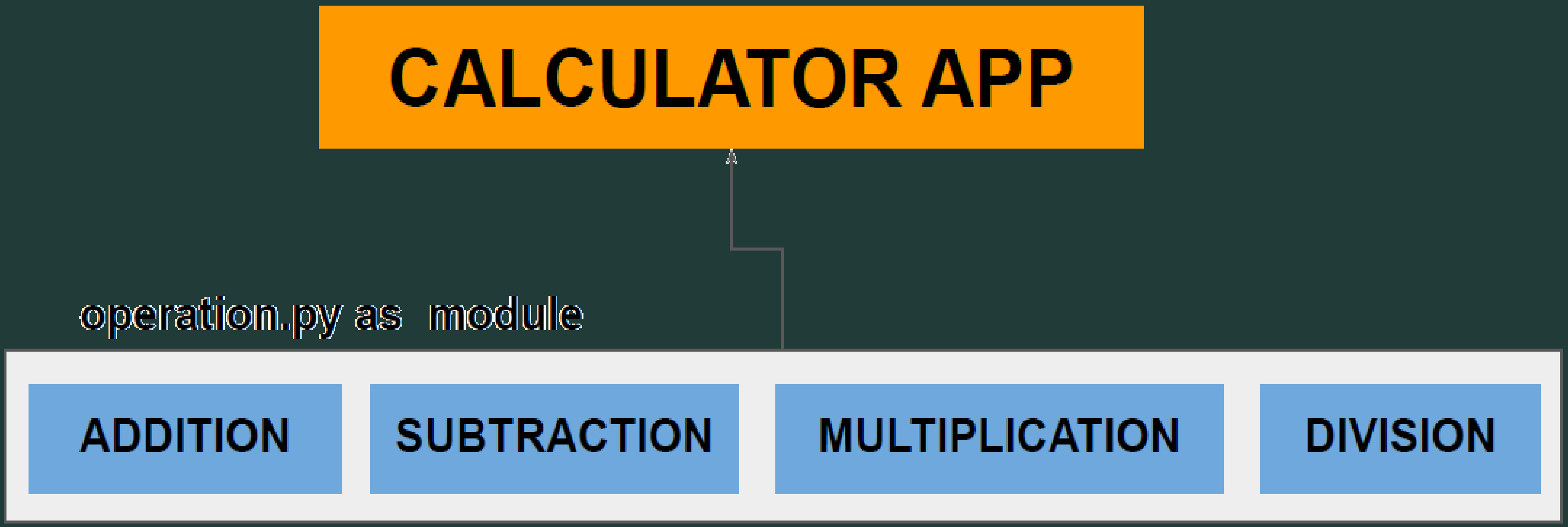
A module is simply a files containing python code, it may contain functions, classes etc.

Advantages of using a module

- Logically organize your Python code
- Code is easier to understand and use

MODULE

12345
7
8
9
÷
4
5
6
x
1
2
3
−
0
.
+
=

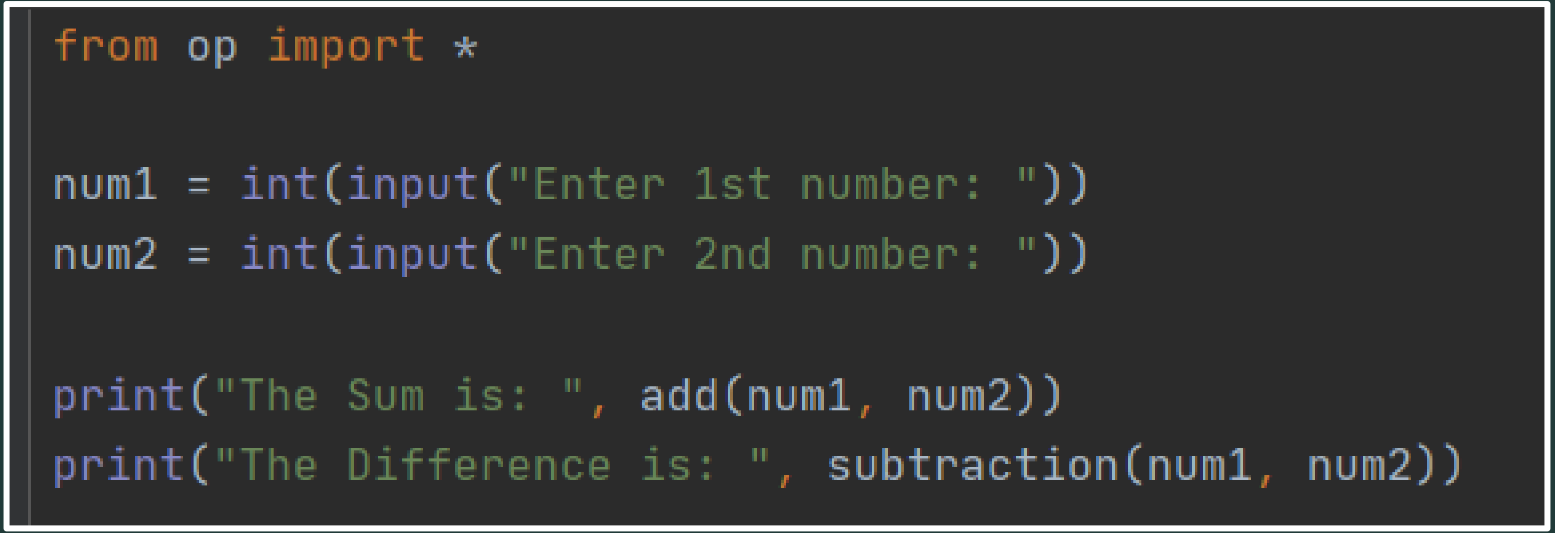

CALCULATOR APP
operation.py as module
ADDITION
SUBTRACTION
MULTIPLICATION
DIVISION

The + operator can be used to concatenate two or more tuples together.

CALL MODULE

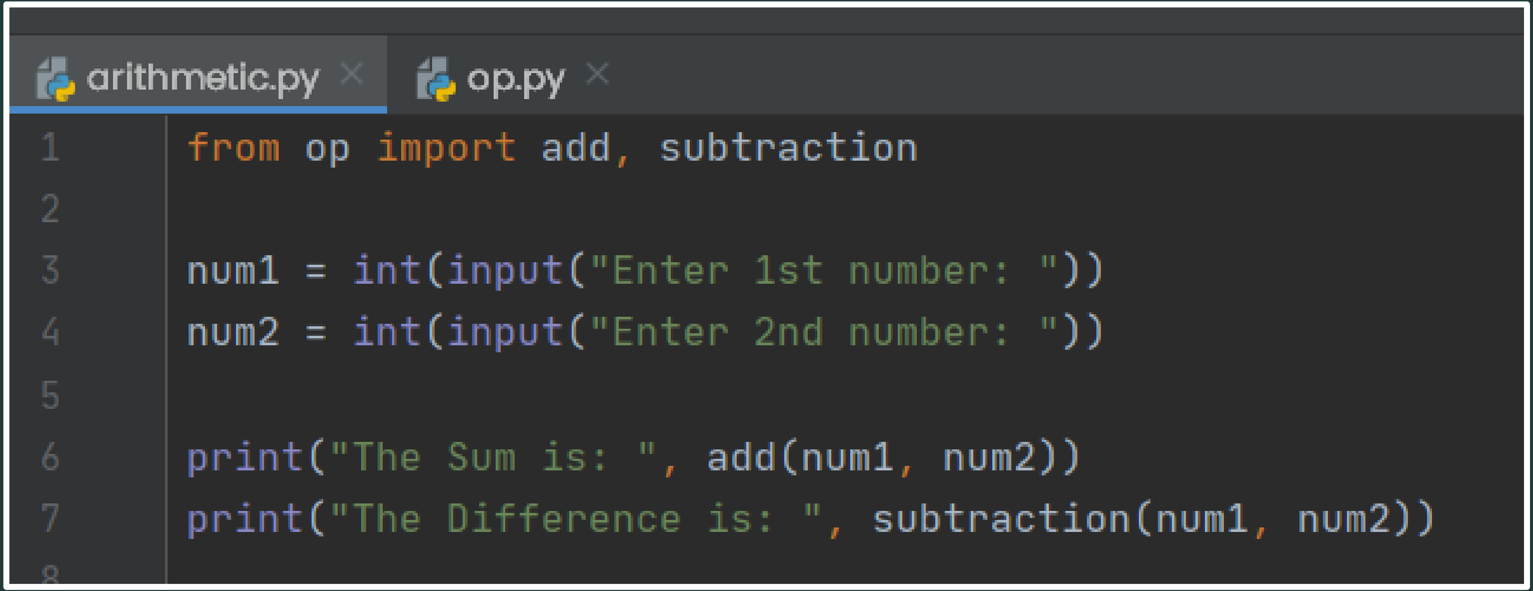
You can use any Python source file as a module by executing an import statement in some other Python source file.

- import moduleName as moduleAlias
- from moduleName import functioName
- from moduleName import *

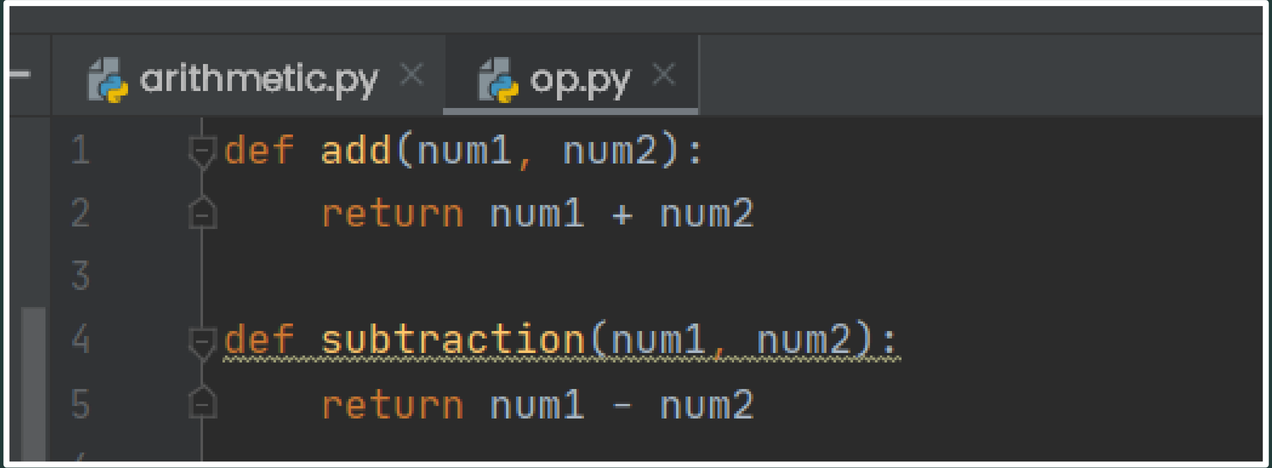


```
from op import *

num1 = int(input("Enter 1st number: "))
num2 = int(input("Enter 2nd number: "))

print("The Sum is: ", add(num1, num2))
print("The Difference is: ", subtraction(num1, num2))
```

MODULE



```python
from op import add, subtraction

num1 = int(input("Enter 1st number: "))
num2 = int(input("Enter 2nd number: "))

print("The Sum is: ", add(num1, num2))
print("The Difference is: ", subtraction(num1, num2))
```



```python
def add(num1, num2):
    return num1 + num2

def subtraction(num1, num2):
    return num1 - num2
```