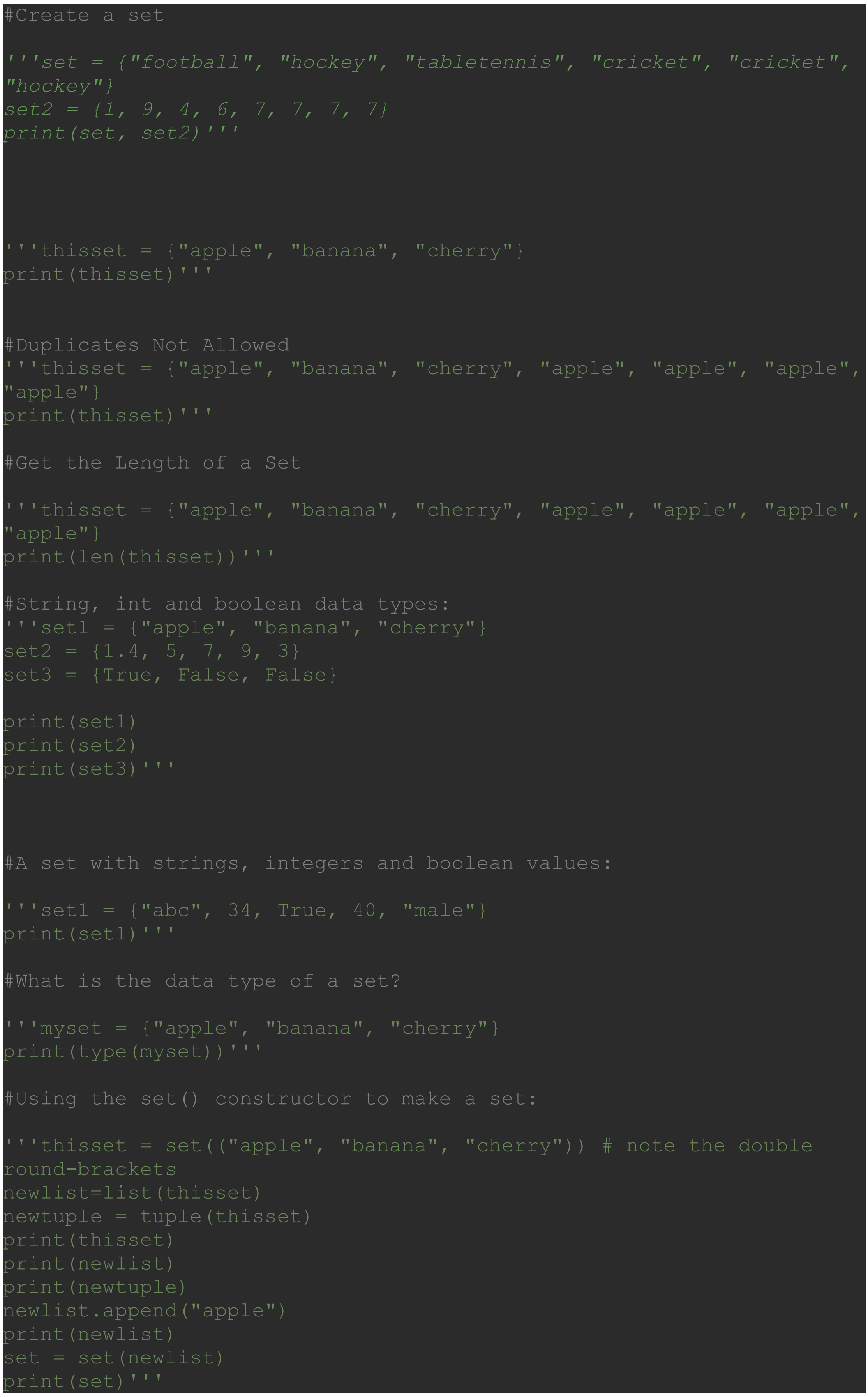
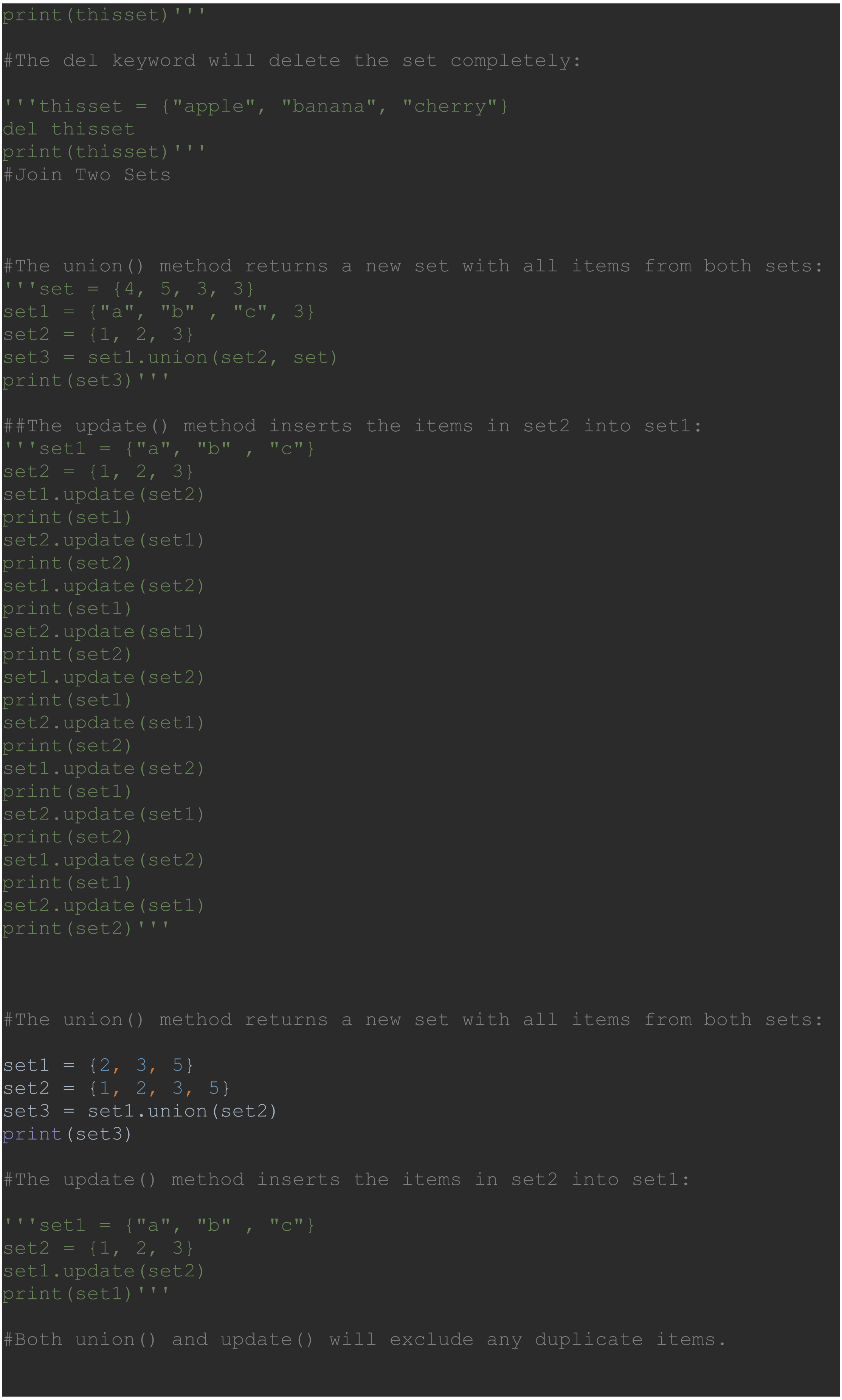
Lab 4 python



```
#Create a set

'''set = {"football", "hockey", "tabletennis", "cricket", "cricket",
"hockey"}
set2 = {1, 9, 4, 6, 7, 7, 7, 7}
print(set, set2)'''




'''thisset = {"apple", "banana", "cherry"}
print(thisset)'''


#Duplicates Not Allowed
'''thisset = {"apple", "banana", "cherry", "apple", "apple", "apple",
"apple"}
print(thisset)'''

#Get the Length of a Set

'''thisset = {"apple", "banana", "cherry", "apple", "apple", "apple",
"apple"}
print(len(thisset))'''

#String, int and boolean data types:
'''set1 = {"apple", "banana", "cherry"}
set2 = {1.4, 5, 7, 9, 3}
set3 = {True, False, False}

print(set1)
print(set2)
print(set3)'''



#A set with strings, integers and boolean values:

'''set1 = {"abc", 34, True, 40, "male"}
print(set1)'''

#What is the data type of a set?

'''myset = {"apple", "banana", "cherry"}
print(type(myset))'''

#Using the set() constructor to make a set:

'''thisset = set(("apple", "banana", "cherry")) # note the double
round-brackets
newlist=list(thisset)
newtuple = tuple(thisset)
print(thisset)
print(newlist)
print(newtuple)
newlist.append("apple")
print(newlist)
set = set(newlist)
print(set)'''
```

```
#set2 = {1.4, 5, 7, 9, 3}

#You cannot access items in a set by referring to an index or a key.
#Loop through the set, and print the values:

'''thisset = {"apple", "banana", "cherry", "apple", "apple", "apple",
"apple"}

for x in thisset:
  print(x)'''

# Check if "banana" is present in the set:

'''thisset = {"apple", "banana", "cherry"}

print("year" in thisset)'''

#Add an item to a set, using the add() method:

'''thisset = {"apple", "banana", "cherry"}
thisset.add("orange")
print(thisset)
thisset.add("mango")
print(thisset)
thisset.add("kiwi")
print(thisset)
thisset.add("kiwi")
print(thisset)
thisset.add("kiwi")
print(thisset)
thisset.add("kiwi")
print(thisset)
thisset.add("kiwi")
print(thisset)
thisset.add("kiwi")
print(thisset)

thisset.remove("kiwi")
print(thisset)

thisset.remove("apple")
print(thisset)

thisset.discard("kiwi")
print(thisset)'''

#Add Sets:To add items from another set into the current set, use the
update() method.

'''seta = {2, 8, 3, 5, 6}

seta.remove(6)
print(seta)'''




'''thisset = {"apple", "banana", "cherry"}
tropical = {"pineapple", "mango", "papaya"}
thisset.update(tropical)
```

```
print(thisset)
tropical.update(thisset)
print(tropical)
thisset.update(tropical)
print(thisset)
tropical.update(thisset)
print(tropical)
thisset.update(tropical)
print(thisset)
tropical.update(thisset)
print(tropical)
thisset.update(tropical)
print(thisset)
tropical.update(thisset)
print(tropical)
thisset.update(tropical)
print(thisset)
tropical.update(thisset)
print(tropical)
print(tropical)

tropical.remove("apple")
print(tropical)
tropical.pop()'''



#Add elements of a list to at set:

'''thisset = {"apple", "banana", "cherry"}
mylist = ["kiwi", "orange"]
thisset.update(mylist)
print(thisset)'''


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

'''thisset = {"apple", "banana", "cherry"}
thisset.remove("mango")
print(thisset)'''

'''thisset = {"apple", "banana", "cherry"}
thisset.discard("mango")
print(thisset)'''

#Pop() method remove items randomely
'''thisset = {"apple", "banana", "cherry"}
thisset.pop()
print(thisset)'''

#The clear() method empties the set:
'''dict = {"a":"c", "b":"f", "c":"b"}
list = ["a", "b", "c"]
thisset = {"apple", "banana", "cherry"}

dict.clear()
print(dict)
list.clear()
print(list)
thisset.clear()
```



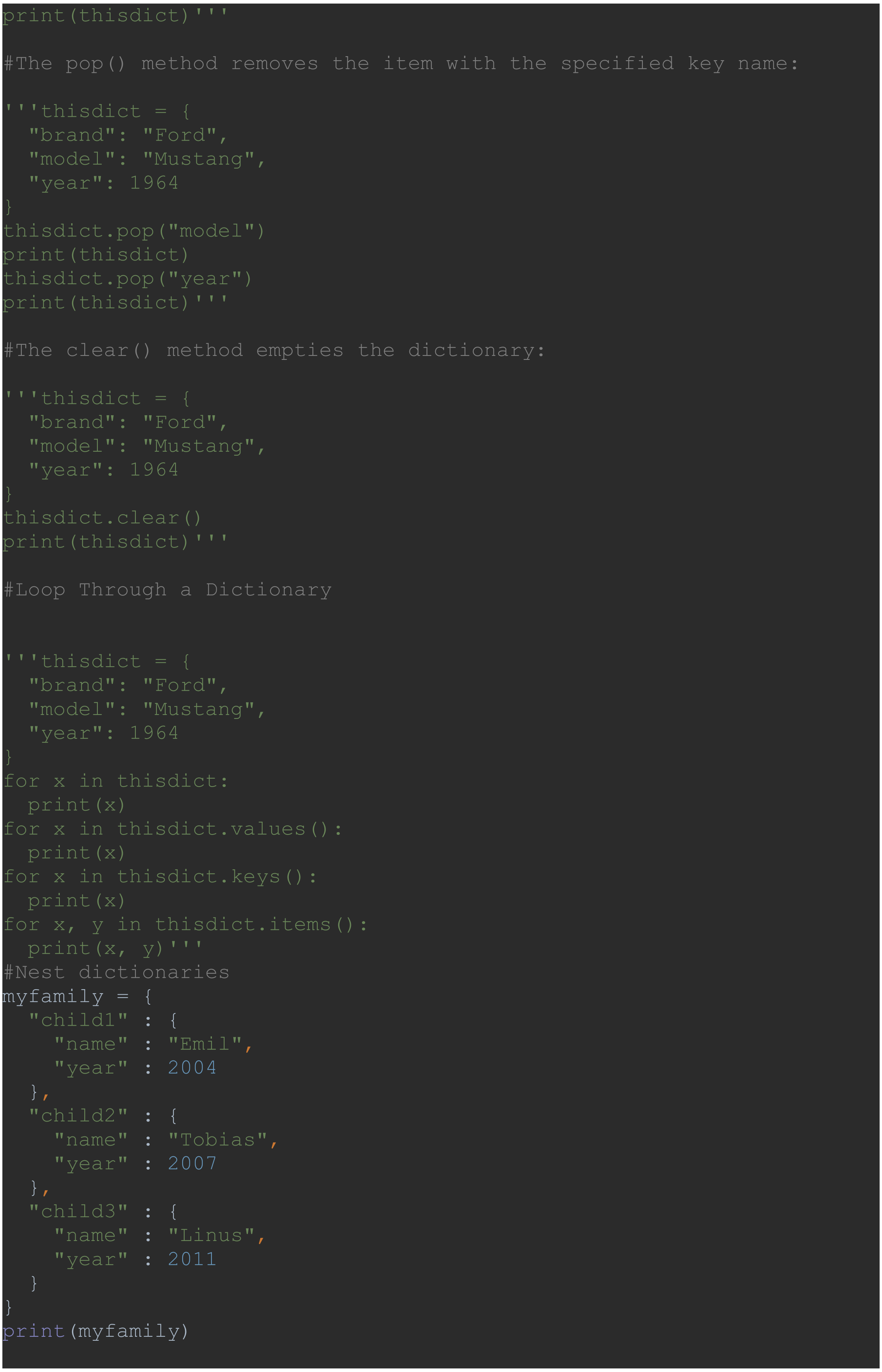
```
print(thisset)'''

#The del keyword will delete the set completely:

'''thisset = {"apple", "banana", "cherry"}
del thisset
print(thisset)'''
#Join Two Sets



#The union() method returns a new set with all items from both sets:
'''set = {4, 5, 3, 3}
set1 = {"a", "b" , "c", 3}
set2 = {1, 2, 3}
set3 = set1.union(set2, set)
print(set3)'''

##The update() method inserts the items in set2 into set1:
'''set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}
set1.update(set2)
print(set1)
set2.update(set1)
print(set2)
set1.update(set2)
print(set1)
set2.update(set1)
print(set2)
set1.update(set2)
print(set1)
set2.update(set1)
print(set2)
set1.update(set2)
print(set1)
set2.update(set1)
print(set2)
set1.update(set2)
print(set1)
set2.update(set1)
print(set2)'''



#The union() method returns a new set with all items from both sets:

set1 = {2, 3, 5}
set2 = {1, 2, 3, 5}
set3 = set1.union(set2)
print(set3)

#The update() method inserts the items in set2 into set1:

'''set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}
set1.update(set2)
print(set1)'''

#Both union() and update() will exclude any duplicate items.
```

```
#Intersection:Keep ONLY the Duplicates
#The intersection_update() method will keep only the items that are
present in both sets.
'''a = {"apple", "mango", "cherry","apple"}
x = {"apple", "banana", "cherry", "apple"}
y = {"google", "microsoft", "apple", "apple"}
x.intersection_update(a,y)
print(x)'''

#The intersection() method will return a new set, that only contains
the items that are present in both sets.

'''x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
z = x.intersection(y)
print(z)'''


#Difference:Keep All, But NOT the Duplicates
#The symmetric_difference_update() method will keep only the elements
that are NOT present in both sets.
'''a = {"apple", "mango", "cherry","apple"}
x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
x.symmetric_difference_update(y)
print(x)'''

'''seta = {1, 2, 3, 4}
setb = {3, 4, 5, 6}


setc = seta.symmetric_difference(setb)

print(setc)

seta.symmetric_difference_update(setb)

print(seta)'''



#The symmetric_difference() method will return a new set, that
contains only the elements that are NOT present in both sets.

'''x = {"apple", "banana", "cherry"}
y = {"google", "microsoft", "apple"}
z = x.symmetric_difference(y)
print(z)
z = y - x
print(z)'''

#Loop throu sets

'''thisset = {"apple", "banana", "cherry"}
for x in thisset:
  print(x)'''
```

Dictionaries

```
#Create Dictionary with the combination of key and value we create an item of a dictionary

'''newdiction = {2:4, 4:2, 5:6, 7:8}
print(newdiction)


thisdict = { "brand": "Ford", "model": "Mustang", "year": 1964}
print(thisdict)

#Print the "brand" value of the dictionary:

thisdict = {
    "brand": {"Ford":"yellow"},
  "model": ["Mustang", "oldmodel", "firstmodel", "latesmodel"],
  "year": (1964, 1966, 1972)
}
print(thisdict)'''

#Duplicate values will overwrite existing values:

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 1950
}
print(thisdict)'''

#Print the number of items in the dictionary:key is unique and value can be duplicated

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year1": 1964
}
print(len(thisdict))'''

#String, int, boolean, and list data types:

'''thisdict = {
  "brand": "Ford",
  "electric": False,
  "year": {1964, 192},
  "colors": {"red":"hello"}
}
print(thisdict)'''

#type

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(type(thisdict))'''



#Get the value of the "model" key
```

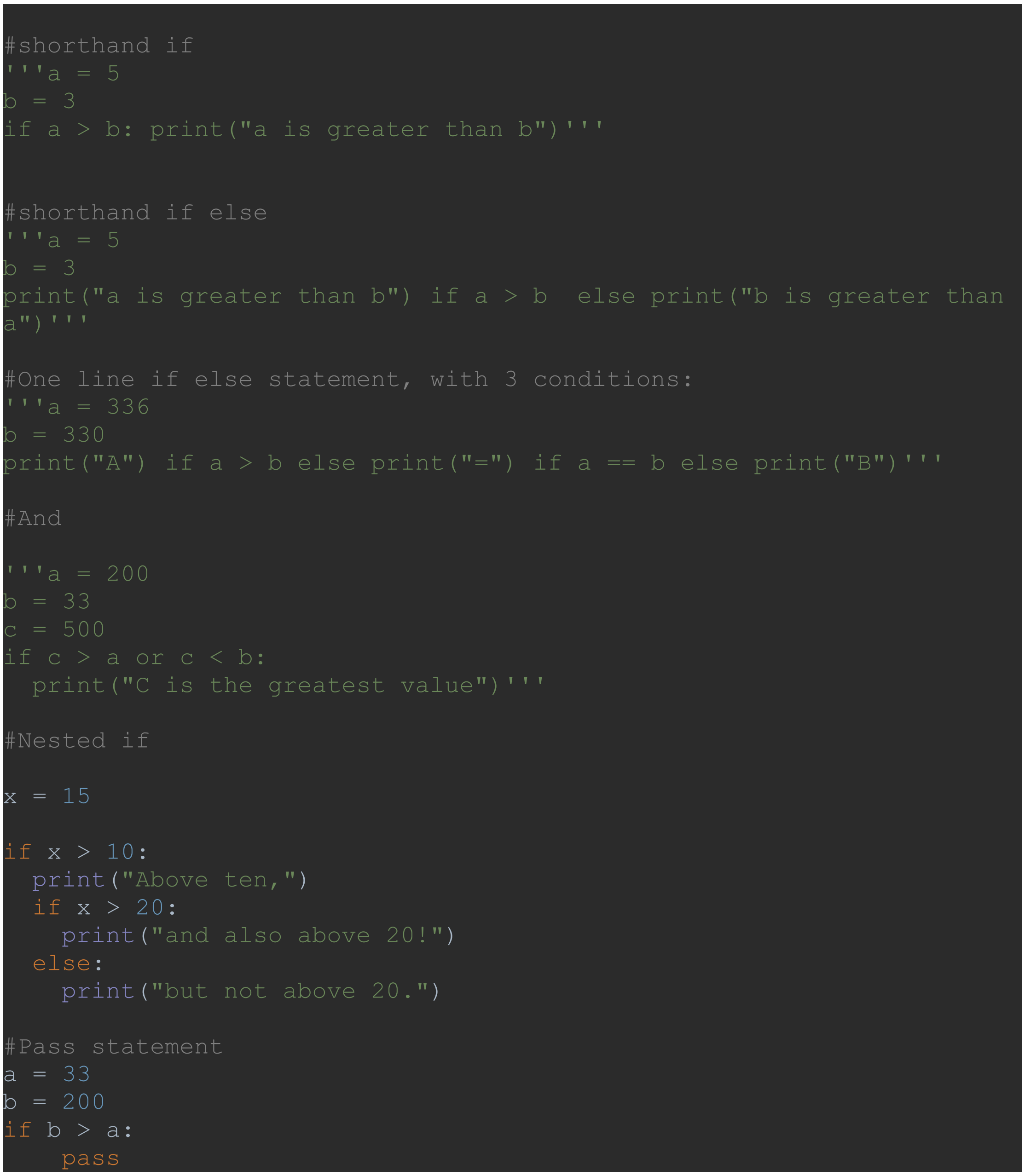
```
'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["model"])

x = thisdict.get("model")
print(x)

y= thisdict.keys()

print(y)

z= thisdict.values()
print(z)

a = thisdict.items()

print(a)'''


#Changeable dictionary

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["year"] = 2021

print(thisdict)

#Adding Items

thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict["color"] = "red"
print(thisdict)

thisdict["price"] = "1mUSD"
print(thisdict)

thisdict["place"] = "New York"

print(thisdict)'''
#Add a color item to the dictionary by using the update() method:

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.update({"newmodel": "Autodrive"})
thisdict.update({"color": "red"})
thisdict.update({"price": "1mUSD"})
thisdict.update({"place": "NewYork"})
```



```
print(thisdict)'''

#The pop() method removes the item with the specified key name:

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.pop("model")
print(thisdict)
thisdict.pop("year")
print(thisdict)'''

#The clear() method empties the dictionary:

'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
thisdict.clear()
print(thisdict)'''

#Loop Through a Dictionary


'''thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
for x in thisdict:
  print(x)
for x in thisdict.values():
  print(x)
for x in thisdict.keys():
  print(x)
for x, y in thisdict.items():
  print(x, y)'''
#Nest dictionaries
myfamily = {
  "child1" : {
    "name" : "Emil",
    "year" : 2004
  },
  "child2" : {
    "name" : "Tobias",
    "year" : 2007
  },
  "child3" : {
    "name" : "Linus",
    "year" : 2011
  }
}
print(myfamily)
```

If..else

```
#if statement

'''Equals:a == b
Not Equals:a != b
Less than:a < b
Less than or equal to:a <= b
Greater than:a > b
Greater than or equal to:a >= b'''


'''a = 100
b = 200
if b > a:
  print("b is greater than a")'''



#Indentation

'''a = 33
b = 200
if b > a:
    print("b is greater than a") # you will get an error'''

#Elif

'''a = 100
b = 90
if a > b:
  print("a is greater than b")
elif a < b:
  print("a is less than b")'''

#Anything that isn't covered by the preceding conditions is caught by
the else keyword.

#Else

'''a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")'''


#else without elif

'''a = 200
b = 33
if b > a:
 print("b is greater than a")
else:
 print("b is less than a")'''
```



```
#shorthand if
'''a = 5
b = 3
if a > b: print("a is greater than b")'''


#shorthand if else
'''a = 5
b = 3
print("a is greater than b") if a > b  else print("b is greater than
a")'''

#One line if else statement, with 3 conditions:
'''a = 336
b = 330
print("A") if a > b else print("=") if a == b else print("B")'''

#And

'''a = 200
b = 33
c = 500
if c > a or c < b:
  print("C is the greatest value")'''

#Nested if

x = 15

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")

#Pass statement
a = 33
b = 200
if b > a:
    pass
```