In [1]:
```
print('Asaad\'s Book')
print("Asaad's Book")
print('''
Hello World
Hello World''')
```

```
Asaad's Book
Asaad's Book

Hello World
Hello World
```

In [2]:
```
print('3+5')
x : int = 3+5
print('x=', x)
```

```
3+5
x= 8
```

In [9]:
```
print(type(22))
print(type(2.2))
print(type('Asaad'))
print(type(True))
myList=[1,2,3,4,5]
myTupe=(1,2,3,4,5)
mySet={1,2,3,4.5}
myDictionary={1:'x',2:'y',3:'z'}
print(type(myList))
print(type(myTupe))
print(type(mySet))
print(type(myDictionary))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'list'>
<class 'tuple'>
<class 'set'>
<class 'dict'>
```

In [12]:
```
fname="Asaad"
lname="Elsaadany"
fullname=fname+' '+lname
print(fullname)
print(fname,lname,sep='****')
```

```
Asaad Elsaadany
Asaad****Elsaadany
```

In [15]:
```
name='Asaad' 'Elsaadany'
print(name)
print(type(name))
name1='Asaad','Elsaadany'
print(name1)
print(type(name1))
```

```
AsaadElsaadany
<class 'str'>
('Asaad', 'Elsaadany')
<class 'tuple'>
```

In [19]:
```
x=10
y=6
print(f'the x is equal to {x} and the y equal to {y}')
```

```
the x is equal to 10 and the y equal to 6
```

In [24]:
```
string1="this is the way to the world ends"
print(string1[12])   # طباعة الحرف رقم 13
print(string1[0:4])  #4 طباعة الحروف من 0 الي
print(string1[5:])   #طباعة الحروف بداية من الحرف الخامس الي الاخر
print(string1[:4])   #
print(string1[-1])   #طباعة الحرف الاخير
print(string1[-4:-1])# طباعة اخر 4 حروف ماعدا الحرف الاخير
print(string1[-4:])  #طباعة اخر 4 حروف
```

```
w
this
is the way to the world ends
this
s
end
ends
```

In [32]:
```
print(3+4)
print(3-4)
print(3*4.0)
print(9/4)
print(9//4)
print(9%4)
print(3**2)
print((3+2)/2)
print(3+2/2)
```

```
7
-1
12.0
2.25
2
1
9
2.5
4.0
```

In [33]:
```
print(5>2)
print(5<2)
print(5<=2)
print(5>=2)
```

```
True
False
False
True
```

In [50]:
```
print(5 > 2)
print(5 < 2)
print(5 >= 2)
print(5 <= 2)

print('--------------')

print(5<2 and 3>0)
print(5<2 or 3>0)
```

```
True
False
True
False
--------------
False
True
```

In [49]:

```
x = 5
y = 2
result = (x == y)
print(f'Result = {result}')
result = (x != y)
print(f'Result = {result}')
result = (x is y)
print(f'Result = {result}')
result = (x is not y)
print(f'Result = {result}')
```

```
Result = False
Result = True
Result = False
Result = True
```

In [47]:

```
print(True and 66)
print(True and "aa")
print(True and ())

print(False and "aa")
print(True or {})
```

```
66
aa
()
False
True
```

Collection

List

In [114…

```
# إنشاء القائمة
myList = [1, 2, 3, 4, 'five', True]

# الوصول إلى العناصر بواسطة الفهرس (indexing)
print(myList[0])  # الناتج: 1
print(myList[4])  # الناتج: 'five'

# التعديل على العناصر
myList[2] = 'three'
print(myList)  # الناتج: [2 ,1, 'three', 4, 'five', True]

# إضافة عنصر جديد في نهاية القائمة باستخدام append()
myList.append(6)
print(myList)  # الناتج: [2 ,1, 'three', 4, 'five', True, 6]

# إزالة عنصر من القائمة باستخدام remove()
myList.remove('five')
print(myList)  # الناتج: [2 ,1, 'three', 4, True, 6]
```

```
# الطول (عدد العناصر) في القائمة
print(len(myList))  # الناتج: 6
```

```
1
five
[1, 2, 'three', 4, 'five', True]
[1, 2, 'three', 4, 'five', True, 6]
[1, 2, 'three', 4, True, 6]
6
```

In [59]:

```
a = [1, 2, 3]
b = [4, 5, 6]
c = a+b
print(a)
print(b)
print(c)
```

```
[1, 2, 3]
[4, 5, 6]
[1, 2, 3, 4, 5, 6]
```

In [68]:

```
a = []
a.append(2) # اضافة عنصر الي القائمة
a.append(3)
print(a)

print( len(a) )
a.extend([4,5])# اضافة قائمة اخري الي القائمة


b = a.copy()
print(b)
```

```
[2, 3]
2
[2, 3, 4, 5]
```

In [115…

```
a = [2, 2, 5]
a.reverse() # مقلوب المصفوفة.
print(a)

a.sort()
print(a)

a.clear() # قائمة فارغة
print(len(a))
```

```
[5, 2, 2]
[2, 2, 5]
0
```

In [88]:

```
list1 = ["a", "b", "d", "e"]
list1.insert(2, "c")      #insert element "c" at position 2, increasing the length b
print(list1)
```

```
['a', 'b', 'c', 'd', 'e']
```

In [89]:

```
list1.pop()           # remove the last element of the list.
print( list1)
```

```
['a', 'b', 'c', 'd']
```

In [117…
```
list1 = ["d", "b", "b", "c", "d", "d", "a"]

print(list1.count("d"))           # count the number of times "d" occurs
print( list1.index("d"))          # return the index of the first occurrence of "d"

list1.sort()  # sort the list
print(list1)
```

```
3
0
['a', 'b', 'b', 'c', 'd', 'd', 'd']
```

In [78]:
```
del list1[2]            # remove the element at index 2
print( list1)
```

```
['a', 'b', 'd', 'd', 'd']
```

In [80]:
```
del list1[2:4]          # remove the elements from index 2 to 4.
print(list1)
```

```
['a', 'b']
```

Tuples

In [94]:
```
a=()
b =(2, 3)
c = 5, 6, 5, 5 ,5
print(len(c))
print(c.count(5))
print(c.index(6))
print(c[0])
```

```
5
4
1
5
```

In [96]:
```
# إنشاء الترتيب
myTuple = (1, 2, 'three', 4.0, True)

# طباعة الترتيب
print(myTuple)

# الوصول إلى عناصر الترتيب بواسطة الفهرس (indexing)
print(myTuple[0])  # الناتج: 1
print(myTuple[2])  # الناتج: 'three'

# لا يمكن تعديل عناصر الترتيب بمجرد إنشائه
# myTuple[0] = 5  # سيؤدي إلى خطأ TypeError

# استخدام الترتيب في التكوينات أو العمليات الأخرى
x, y, z, w, v = myTuple  # يقوم بتخزين القيم في المتغيرات x, y, z, w, v بالتتابع
print( w, v)  # الناتج: 4.0 True 'three' 2 1
```

```
(1, 2, 'three', 4.0, True)
1
three
4.0 True
```

In [97]:

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5, 6}
{1, 2, 3, 4, 5, 6, 7, 8, 9}
{1, 2, 3, 5, 6, 7, 8, 9}
True
False
8
```

In [98]:

```
# إنشاء مجموعة
mySet = {1, 2, 3, 4, 5, 5, 5}  # الملاحظة: يتم تجاهل العناصر المكررة
print(mySet)  # الناتج: {1, 2, 3, 4, 5}

# إضافة عنصر إلى المجموعة
mySet.add(6)
print(mySet)  # الناتج: {1, 2, 3, 4, 5, 6}

# إضافة عدة عناصر إلى المجموعة باستخدام update()
mySet.update({7, 8, 9})
print(mySet)  # الناتج: {1, 2, 3, 4, 5, 6, 7, 8, 9}

# إزالة عنصر من المجموعة
mySet.remove(4)
print(mySet)  # الناتج: {1, 2, 3, 5, 6, 7, 8, 9}

# التحقق من وجود عنصر في المجموعة
print(3 in mySet)  # الناتج: True
print(4 in mySet)  # الناتج: False

# حجم المجموعة
print(len(mySet))  # الناتج: 8
```

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5, 6}
{1, 2, 3, 4, 5, 6, 7, 8, 9}
{1, 2, 3, 5, 6, 7, 8, 9}
True
False
8
```

In [109…

```
a = {'a', 'b', 'c', 'd'}
b = set('hello world')
print(a)
print(b)
```

```
{'c', 'b', 'd', 'a'}
{'l', 'h', 'd', 'o', 'r', 'w', ' ', 'e'}
```

print(a|b): الاتحاد (Union)

هذه العملية تجمع بين عناصر كل من المجموعتين دون تكرار العناصر المكررة.

print(a&b): التقاطع (Intersection)

print(a-b): الفرق (Difference) هذه العملية تجد العناصر المشتركة بين المجموعتين

print(b-a): الفرق (Difference) هذه العملية تجد العناصر في مجموعه الاولي التي لا توجد في المجموعه الثانية

print(a^b): الفرق المتماثل (Symmetric Difference) هذه العملية تجد العناصر في التي لا توجد في

هذه العملية تجد العناصر التي توجد في الاولي أو في الثانية ولكن ليست في كلاهما.

In [110…
```
print(a|b)
print(a&b)
print(a-b)
print(b-a)
print(a^b)
```

```
{'l', 'h', 'c', 'd', 'o', 'r', 'b', 'w', ' ', 'a', 'e'}
{'d'}
{'c', 'b', 'a'}
{'l', 'o', 'r', 'w', ' ', 'h', 'e'}
{'l', 'h', 'c', 'o', 'r', 'b', 'w', 'a', ' ', 'e'}
```

In [111…
```
a.intersection(b)
```

Out[111… {'d'}

In [112…
```
a.union(b)
```

Out[112… {' ', 'a', 'b', 'c', 'd', 'e', 'h', 'l', 'o', 'r', 'w'}

Dictionary

In [113…
```
# إنشاء القاموس
myDictionary = {1: 'x', 2: 'y', 3: 'z'}

# الوصول إلى القيم باستخدام المفاتيح
print(myDictionary[1])  # الناتج: 'x'
print(myDictionary[2])  # الناتج: 'y'
print(myDictionary[3])  # الناتج: 'z'

# إضافة زوج مفتاح-قيمة جديد
myDictionary[4] = 'w'
print(myDictionary)  # الناتج: {1: 'x', 2: 'y', 3: 'z', 4: 'w'}

# تحديث القيمة المرتبطة بمفتاح موجود
myDictionary[1] = 'a'
print(myDictionary)  # الناتج: {1: 'a', 2: 'y', 3: 'z', 4: 'w'}

# إزالة عنصر من القاموس باستخدام المفتاح
del myDictionary[3]
print(myDictionary)  # الناتج: {1: 'a', 2: 'y', 4: 'w'}

# التحقق من وجود مفتاح في القاموس
print(2 in myDictionary)  # الناتج: True
print(3 in myDictionary)  # الناتج: False

# حجم القاموس
print(len(myDictionary))  # الناتج: 3
```

```
x
y
z
{1: 'x', 2: 'y', 3: 'z', 4: 'w'}
{1: 'a', 2: 'y', 3: 'z', 4: 'w'}
{1: 'a', 2: 'y', 4: 'w'}
True
```

```
False
3
```

In [120…

```
x = {'fName':'Asaad', 'lName':'Elsaadany'}

addr = {}                                   # an empty dictionary
addr['number'] = 1234
addr['street'] = "Elm St"
addr['city'] = "Athens"
addr['state'] = "GA"


x['address'] = addr
print(x)
```

```
{'fName': 'Asaad', 'lName': 'Elsaadany', 'address': {'number': 1234, 'street': 'Elm S
t', 'city': 'Athens', 'state': 'GA'}}
```

Conversion Between Types

In [123…

```
# قائمة
myList = [1, 2, 3, 4, 5]

# تحويل القائمة إلى ترتيب
myTuple = tuple(myList)
print(myTuple)  # الناتج: (1, 2, 3, 4, 5)
```

```
(1, 2, 3, 4, 5)
```

In [124…

```
# ترتيب
myTuple = (1, 2, 3, 4, 5)

# تحويل الترتيب إلى قائمة
myList = list(myTuple)
print(myList)  # الناتج: [1, 2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5]
```

In [128…

```
# قاموس
myDictionary = {'a': 1, 'b': 2, 'c': 3}

# تحويل القيم إلى قائمة
List1 = list(myDictionary.values())
print(List1)
```

```
[1, 2, 3]
```

In [131…

```
q = (('one',1), ('two',2), ('three',3))
w = dict(q)
print(w)

print(tuple(w.keys()))
print(tuple(w.values()))
```

```
{'one': 1, 'two': 2, 'three': 3}
('one', 'two', 'three')
(1, 2, 3)
```

Functions

In [133…
```
#تعريف الدالة باستخدام def
def greet(name):
    print(f"Hello {name}")

# تعيين متغير name واستدعاء الدالة
name = "Alice"
greet(name)
```

```
Hello Alice
```

In [135…
```
def multiply(x, y):
    result = x * y
    return result

result = multiply(3, 4)
print(result)
```

```
12
```

In [136…
```
def add(x, y):
    return x + y

result = add(2, 3)
print(result)  # 5 :الناتج
```

```
5
```

In [139…
```
def add_subtract(x, y):
    s1 = x + y
    s2 = x - y
    return s1, s2

w1, w2 = add_subtract(8, 3)
print(w1)                    # 11 :الناتج
print(w2)                    # 5 :الناتج
```

```
11
5
```

In [142…
```
def greet(name="Asaad"):
    print(f"Hello {name}")

greet()
greet("Ahmed")
```

```
Hello Asaad
Hello Ahmed
```

In [148…
```
#يمكننا أيضاً تعريف الدوال بدون اسم باستخدام lambda.

y = lambda x: x * 4

result = y(3)
print(result)
```

```
12
```

In [149…
```
def f1(a):
  print(a)
```