



PYTHON

Dictionaries

Dictionaries

- Dictionaries are denoted with curly Brackets {}
- The keys have to be immutable and unique
- The values can be can immutable, mutable and duplicates

```
{ "key1":1, "key2 ":"2","key3 ":[3,3,3], "key4":(4,4,4), ('key5'):5}
```

Key

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"

DICT["The Bodyguard"]:"1992"

Value

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"
'Graduation'	"2007"

DICT['Graduation']='2007'

"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"

`del(DICT['Thriller'])`

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"

'The Bodyguard' in DICT

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
"Saturday Night Fever"	"1977"
"Rumors"	"1977"

```
DICT.keys()=[ "Thriller", "Back in Black", "The Dark Side of the Moon", "The Bodyguard",
    "Bat Out of Hell", "Their Greatest...","Saturday Night Fever", "Rumors" ]
```

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"
"Bat Out of Hell"	"1977"
"Their Greatest..."	"1976"
Saturday Night Fever	"1977"
"Rumors"	"1977"

DICT.values() =["1982", "1980", "1973", "1992", "1977", "1976", "1977", "1977"]





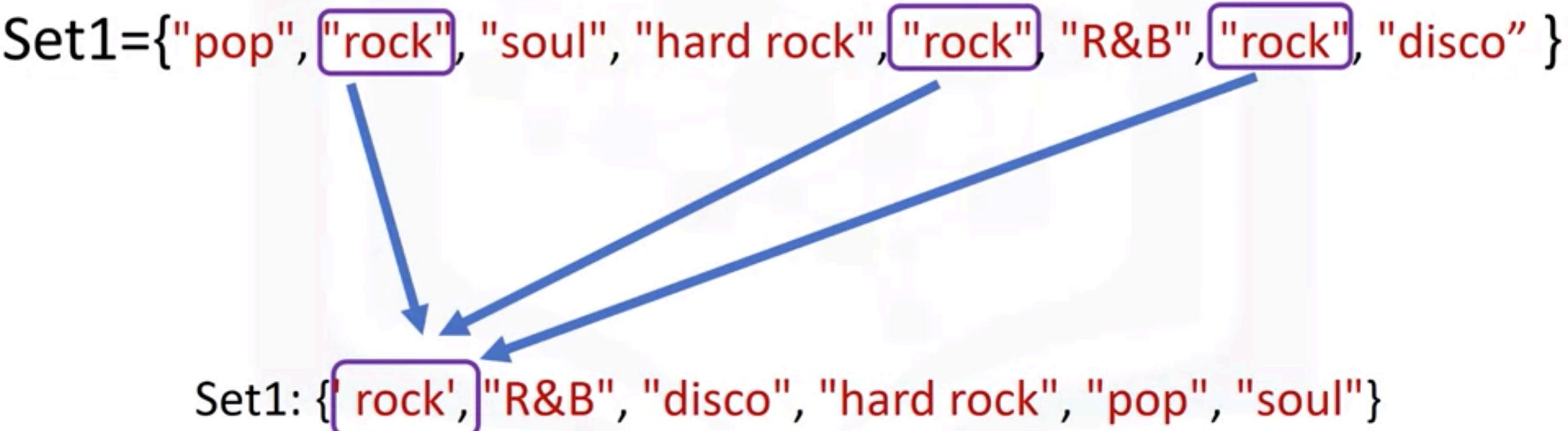
PYTHON

Sets

Sets

- Sets are a type of collection
 - This means that like lists and tuples you can input different Python types
- Unlike lists and tuples they are unordered
 - This means sets do not record element position
- Sets only have unique elements
 - This means there is only one of a particular element in a set

Sets: Creating a Set



Sets: Creating a Set

```
album_list = ["Michael Jackson", "Thriller", "Thriller", 1982]
```

```
album_set = set(album_list)
```

```
album_set : {'Michael Jackson', 'Thriller', 1982}
```

set()

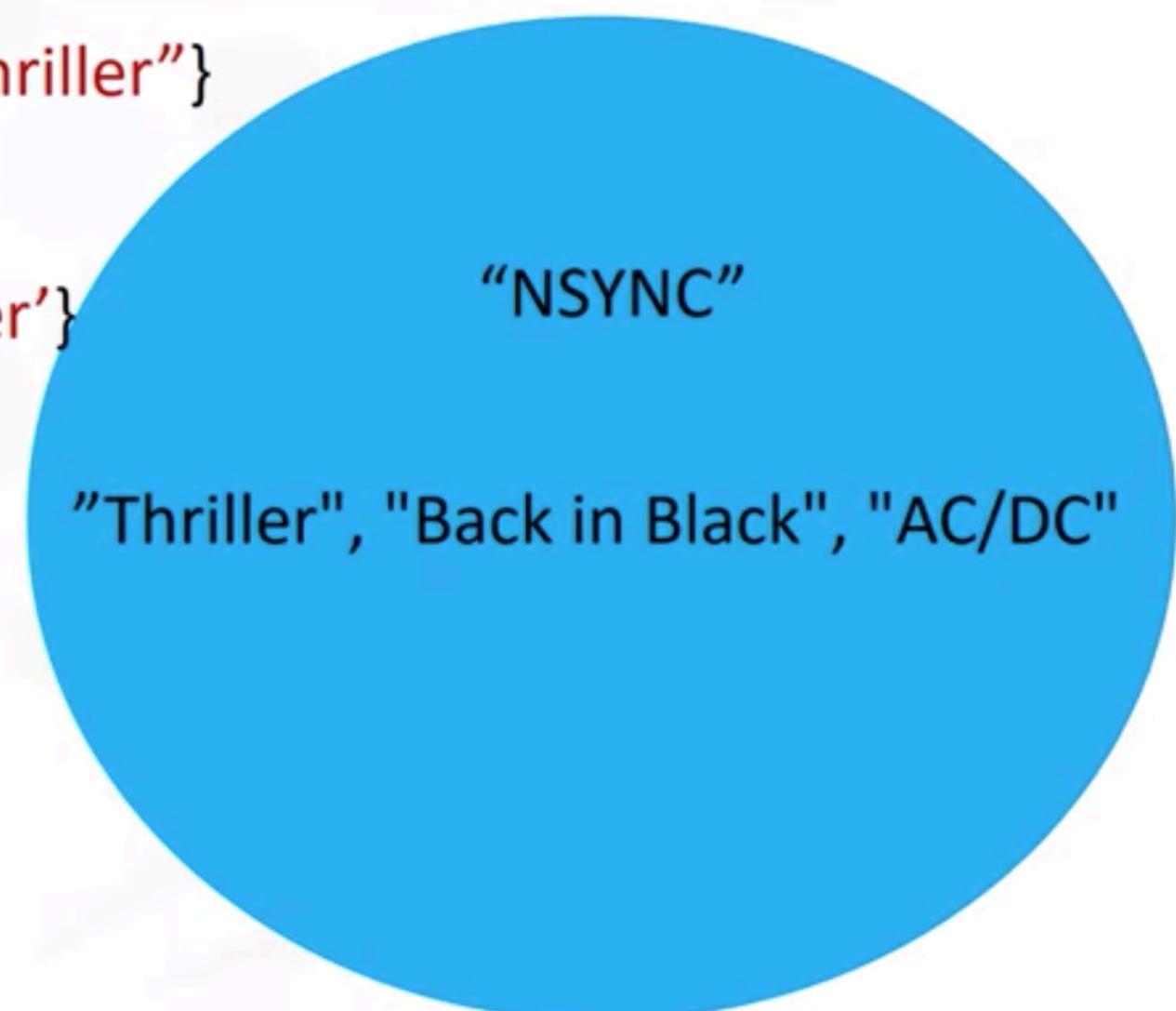
album_set

Set Operations

A : {"AC/DC", "Back in Black", "NSYNC", "Thriller"}

A.add("NSYNC")

A:{'AC/DC', 'Back in Black', 'NSYNC', 'Thriller'}

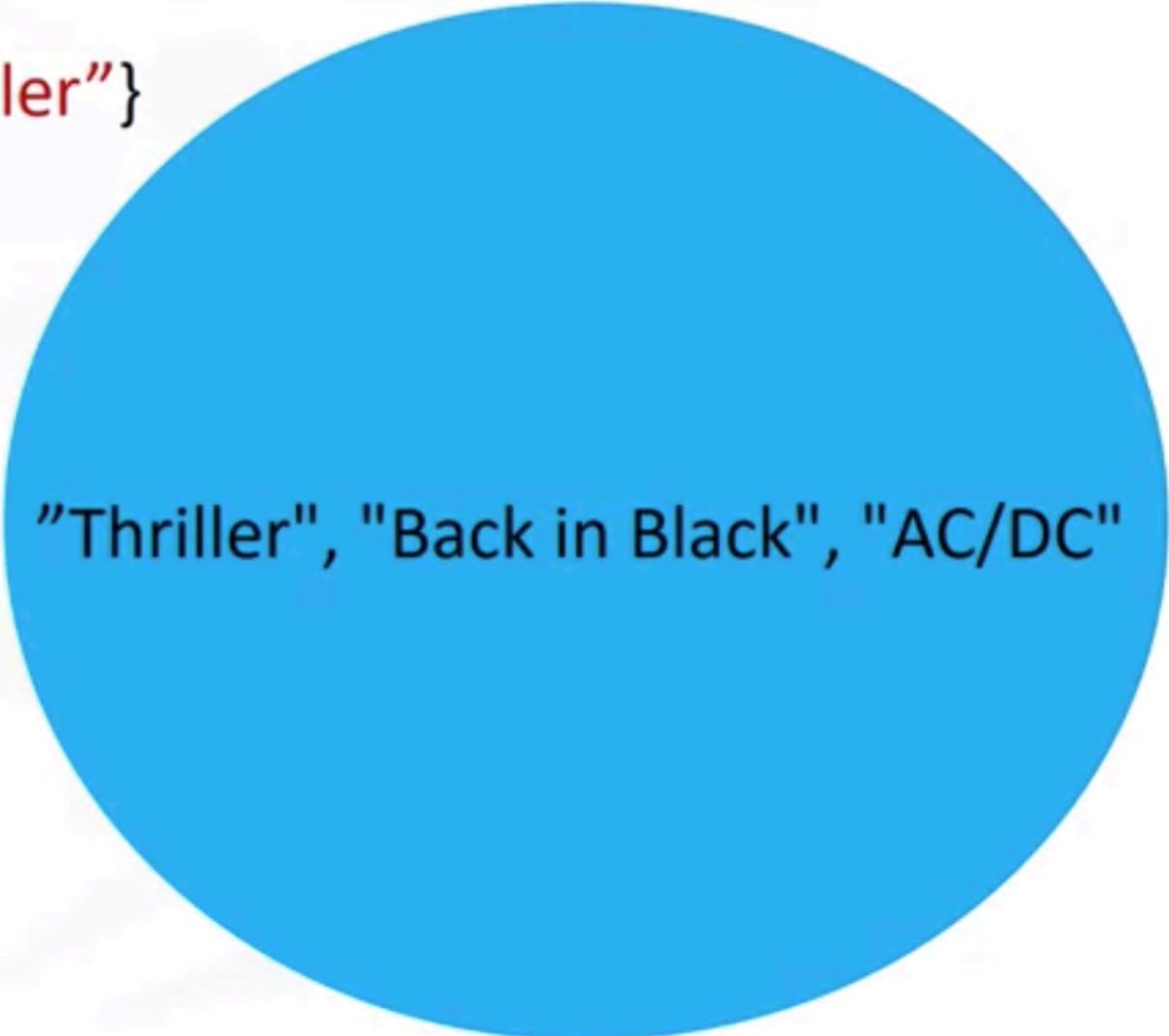


Set Operations

```
A :{“AC/DC”, “Back in Black”, “NSYNC”, “Thriller”}
```

```
A.remove("NSYNC")
```

```
A:{“AC/DC”, “Back in Black”, “Thriller”}
```



"Thriller", "Back in Black", "AC/DC"

Set Operations

A:{“AC/DC”, “Back in Black”, “Thriller”}

“AC/DC” **in** A

True

“Thriller”, “Back in Black”, **“AC/DC”**

Sets: Mathematical set operations

"AC/DC", "Back in Black ", "Thriller

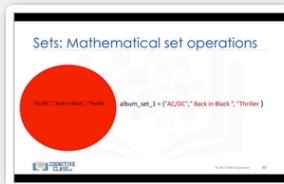
```
album_set_1 = {"AC/DC", "Back in Black ", "Thriller }
```

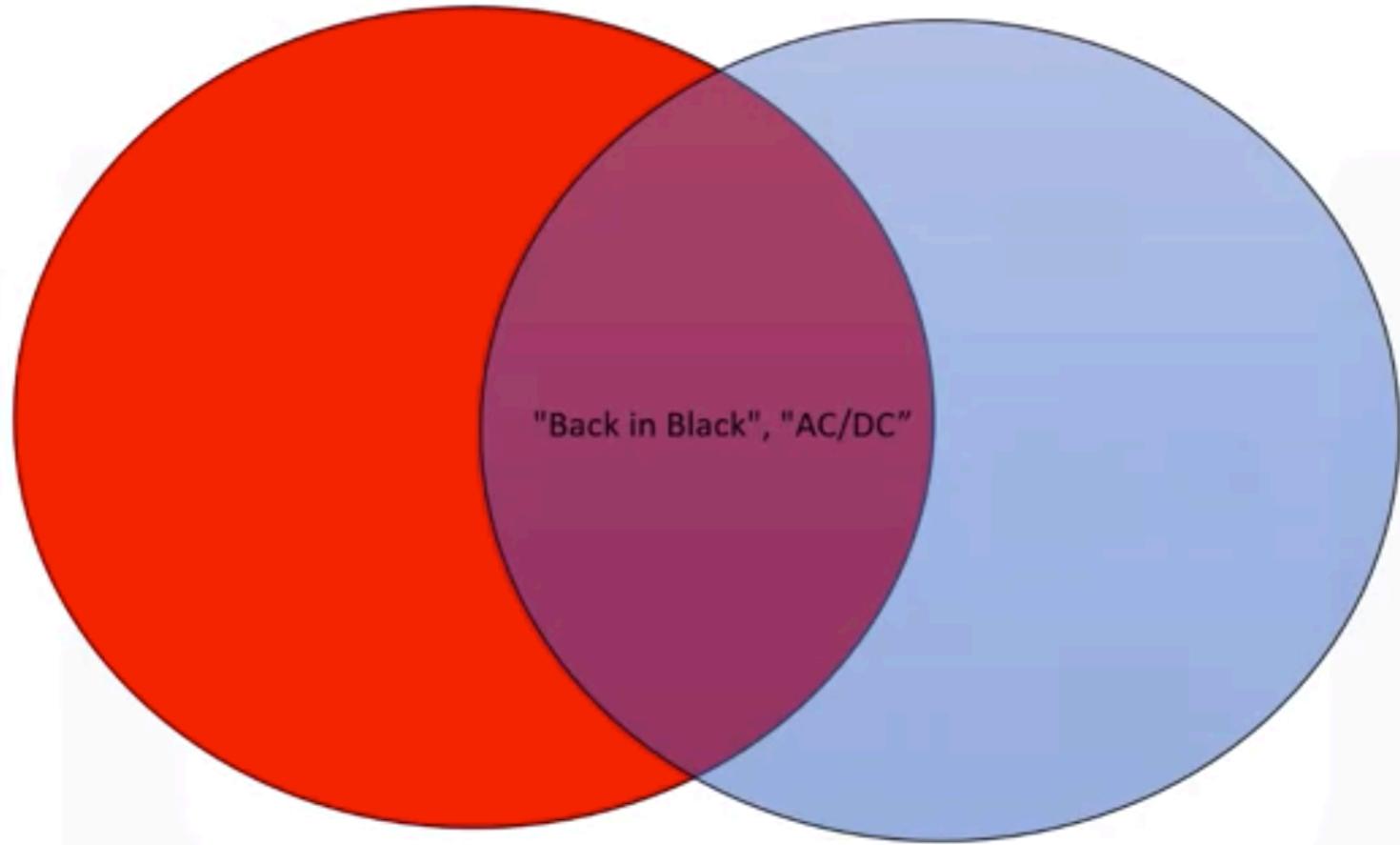
Sets: Mathematical set operations



"AC/DC", "Back in Black",
"The Dark Side of the Moon"

album_set_2 = {"AC/DC", "Back in Black", "The Dark Side of the Moon"}





album_set_1 & album_set_2

```
album_set_1 ={"AC/DC"," Back in Black ", "Thriller }
```

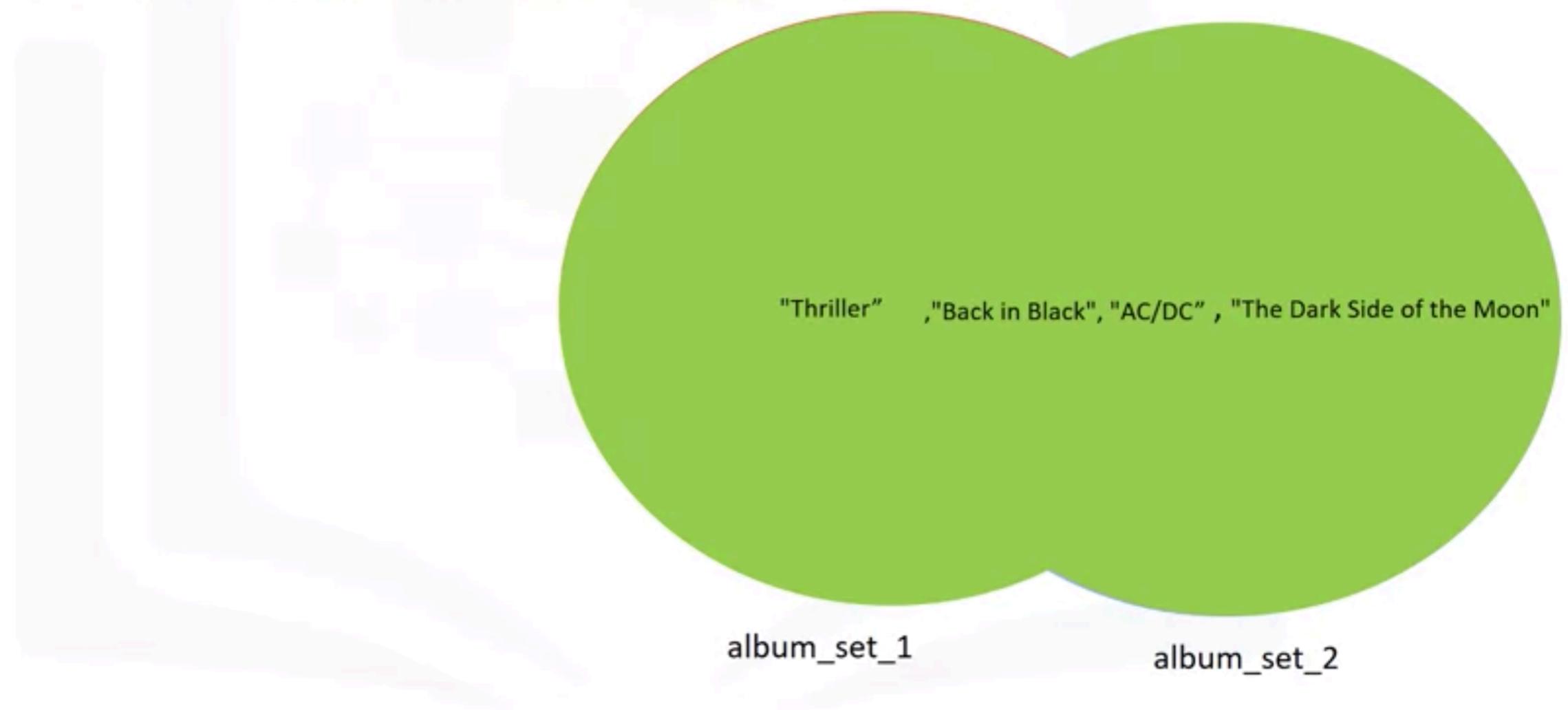
```
album_set_2={"AC/DC", "Back in Black", "The Dark Side of the Moon"}
```

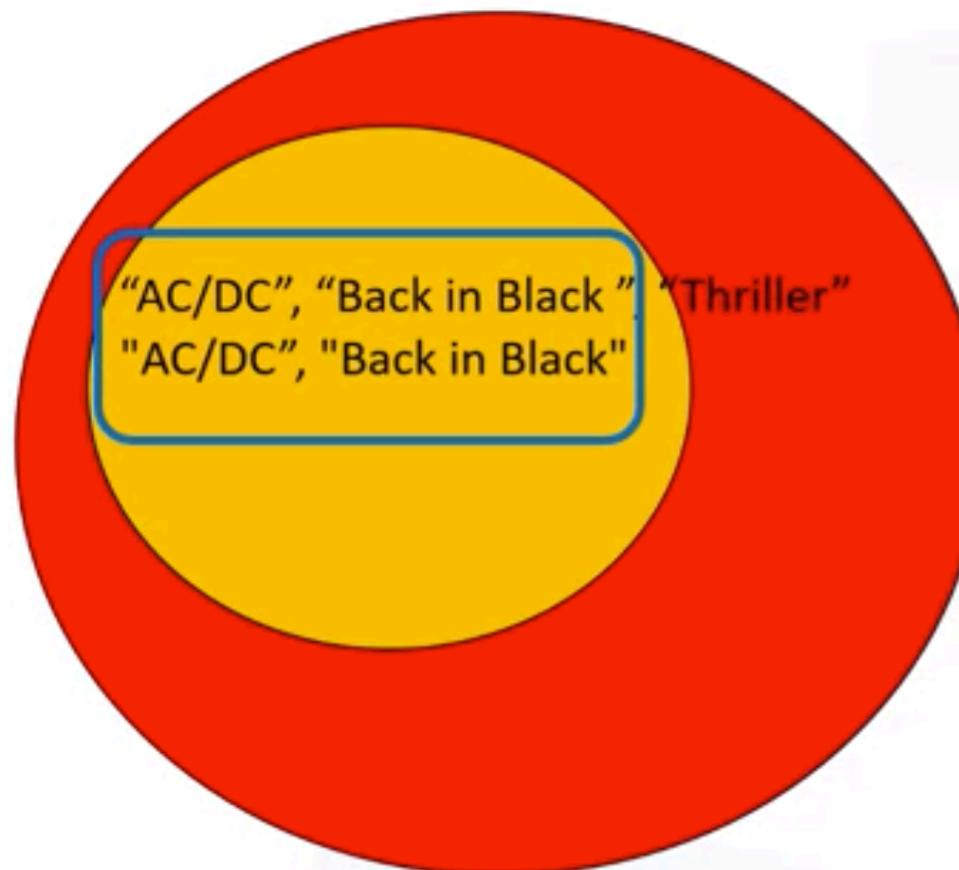
```
album_set_3=album_set_1 & album_set_2
```

```
album_set_3: {"AC/DC", " Back in Black "}
```

```
album_set_1.union(album_set_2)
```

```
{AC/DC", "Back in Black", "The Dark Side of the Moon", "Thriller" }
```





album_set_1

album_set_3

```
album_set_1 = {"AC/DC", "Back in Black", "Thriller"}  
album_set_3 = {"AC/DC", "Back in Black"}  
album_set_3.issubset(album_set1)  
True
```



PYTHON

Conditions and Branching

Comparison Operators

```
6==6
```

True

```
a=6
```

```
a==6
```

i=2

i<6

True



“AC/DC”==“Michael Jackson”

False

“AC/DC”!=“Michael Jackson”

True

```
age=17
```

```
if (age>18):
```

```
    print("you can enter" )
```

```
    print("move on")
```

False

ACDC

move on
17

```
age=17
```

```
if (age>18):
```

False

```
    print("you can enter" )
```

```
else:
```

```
    print("go see Meat Loaf" )
```

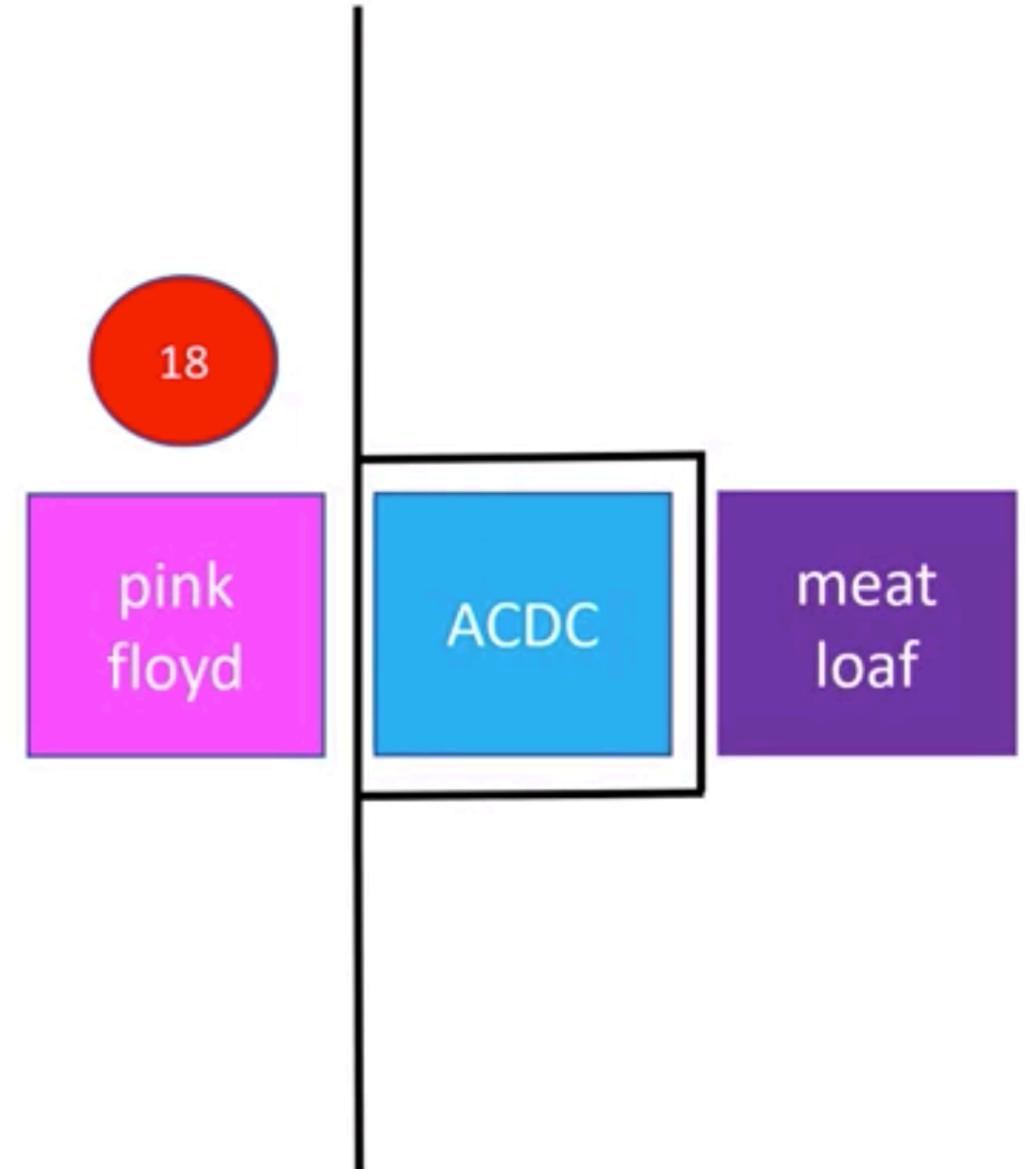
```
    print("move on")
```

go see Meat Loaf

ACDC

meat loaf

```
age=18  
if (age>18):  
    print("you can enter")  
  
elif(age==18):  
    print("go see Pink Floyd")  
  
else:  
    print("go see Meat Loaf")  
  
print("move on")
```



Press **esc** to exit full screen

Logic Operators: OR

A	B	A or B
False	False	False
False	True	True
True	False	True
True	True	True

```
album_year = 1990
```

```
If (album_year < 1980) or (album_year > 1989):
```

```
    print ("The Album was made in the 70 's or 90's")
```

```
else:
```

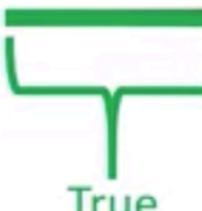
```
    print("The Album was made in the 1980's ")
```

The Album was made in the 70's or 90's

album_year < 1980



or



Logic Operators: AND

A	B	A & B (AND)
False	False	False
False	True	False
True	False	False
True	True	True

```
album_year = 1983
```

```
if(album_year > 1979) and (album_year < 1990):
```

```
    print ("This album was made in the 80's ")
```

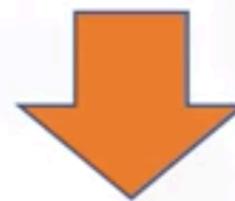




PYTHON Loops



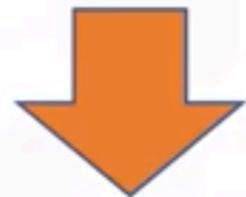
range(3)



[0,1,2]

Press **esc** to exit full screen

range(10,15)



[10, 11, 12, 13, 14]

for loops



© 2017 IBM Corporation

A screenshot of a Jupyter Notebook cell. The code is:

```
range(10,15)
```

The output is:

```
[10, 11, 12, 13, 14]
```

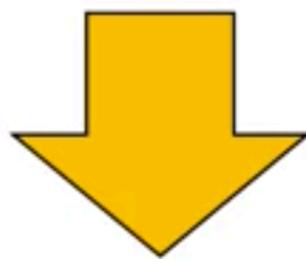
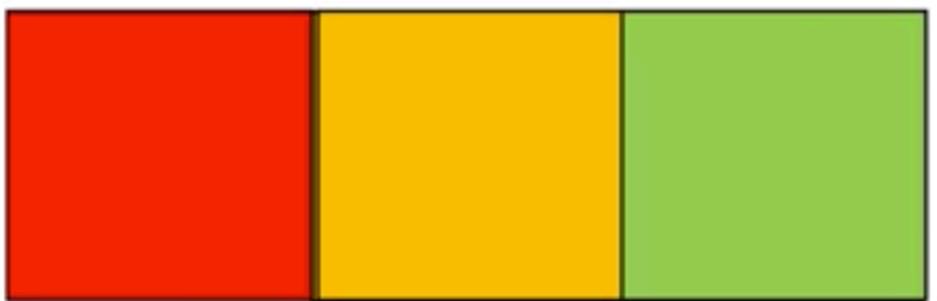


squares	red	yellow	green	purple	blue
	0	1	2	3	4

```
squares=[“red”, “yellow ”, “green”, “purple”, “blue ”]
```

```
for i in range(0,5):
```

```
    squares[i]=“white”
```



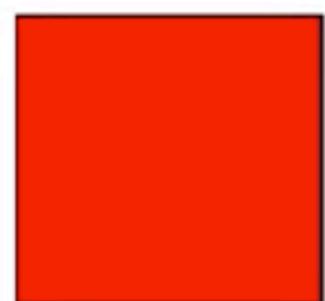
```
squares=[“red”, “yellow”, “green”]
```

```
for square in squares:
```

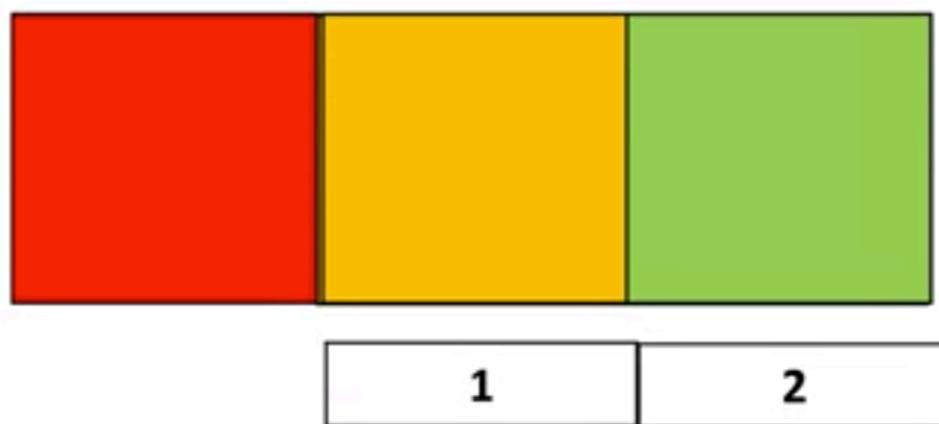
square



square



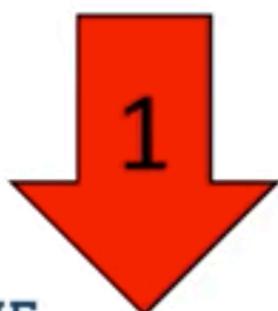
red

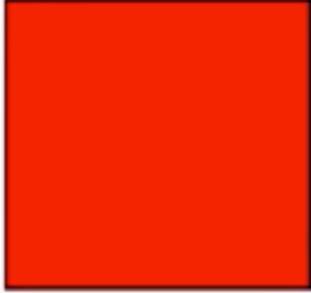


squares=[“red”, “yellow”, “green”]

for i,square in enumerate(squares):

square
i



square
i

red

while loops

```
squares=['orange','orange','purple','blue']
```

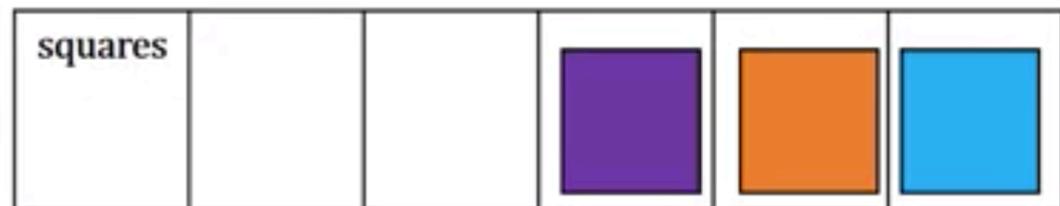
```
Newsquares=[]
```

```
i=0
```

```
while(squares[i]=='orange'):
```

```
    Newsquares.append(squares[i])
```

```
i=i+1
```



```
while( squares[i] =='orange'):
```

