



Direktorat Jenderal Pendidikan Tinggi, Riset, dan, Teknologi
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Republik Indonesia

DIKTI
SIGAP
MELAYANI

Kampus
Merdeka
INDONESIA JAYA



MICROCREDENTIAL: ASSOCIATE DATA SCIENTIST

01 November – 10 Desember 2021

Pertemuan ke-4

Struktur Data Dalam Python



[ditjen.dikti](#)



[@ditjendikti](#)



[ditjen.dikti](#)



Ditjen Diktiristek



<https://dikti.kemdikbud.go.id/>



Profil Pengajar: Nama Lengkap dan Gelar Akademik



Poto
Pengajar

Contak Pengajar:

Ponsel:

xxxxxx

Email:

xxxxxxx

Jabatan Akademik:

Latar Belakang Pendidikan:

- S1:
- S2:
- S3:

Riwayat/Pengalaman Pekerjaan:

- Dosen
- XXXX
- XXXX
- XXXX



Materi tambahan



Bagian 1

Tuples





Tuples

Tuples

Merupakan struktur data yang digunakan untuk menyimpan sekumpulan data.

Tuples ditulis secara berurutan, memiliki anggota/element yang dituliskan diantara tanda kurung buka dan kurung tutup, serta antar anggota/element dituliskan dengan dipisahkan oleh tanda koma (,)

Ratings = (10, 9, 6, 5, 10, 8, 9, 6, 2)



Tuples

Pada Python terdapat tipe data: string, integer, dan float. Ketiga tipe data tersebut dapat berada pada satu tuples.



```
tuple1 =('disco', 10, 1.2)
```

```
type (tuple1)=tuple
```



Tuples

Anggota/element pada tuples dapat diberikan indeks positif maupun indeks negatif, sehingga anggota/element tuples dapat diakses dengan menggunakan indeks.

Tuple1 =("disco", 10, 1 . 2)

-3
-2
-1

0	"disco"
1	10
2	1 . 2

Tuple1[-3]: "disco"

Tuple1[-2]: 10

Tuple1[-1]: 1 . 2



Tuples

Anggota/element pada tuples dapat ditambahkan anggota/elementnya dengan menggunakan tanda (+)

(“disco”, 10, 1.2)



tuple2 = tuple1 + (“hard rock”, 10)

(“disco”, 10, 1.2, “hard rock”, 10)



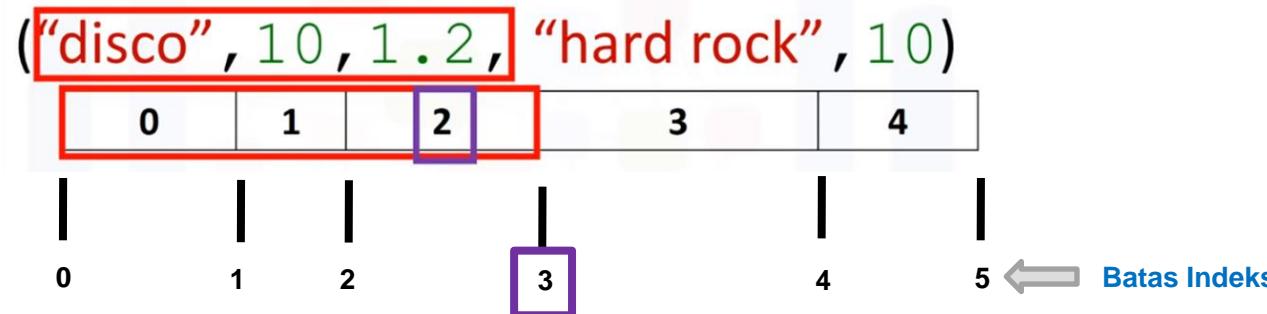


Tuples

Pemotongan Tuples

Pemotongan tuples dilakukan untuk mengambil beberapa anggota/element yang ada di tuples dengan menggunakan indeks tuples.

Contoh ke-1:



`tuple2[0:3] : ('disco', 10, 1.2)`

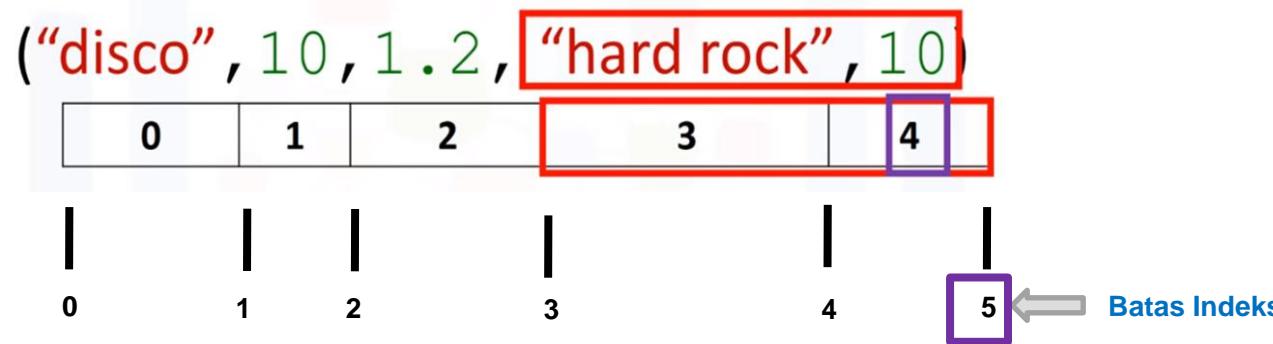


Tuples

Pemotongan Tuples

Pemotongan tuples dilakukan untuk mengambil beberapa anggota/element yang ada di tuples dengan menggunakan indeks tuples.

Contoh ke-2:



`tuple2[3:5] => ("hard rock", 10)`



Tuples

Panjang/banyaknya anggota/element dapat dihitung dengan menggunakan fungsi “LEN”

`len(("disco", 10, 1.2, "hard rock", 10))=5`

indeks	→	0	1	2	3	4
LEN	→	1	2	3	4	5



Tuples

Tuples bersifat immutable

Artinya bahwa isi dari tuples tersebut tetap, tidak dapat diubah.

Ratings = (10, 9, 6, 5, 10, 8, 9, 6, 2)

Ratings1=Ratings





Tuples

Tuples bersifat immutable

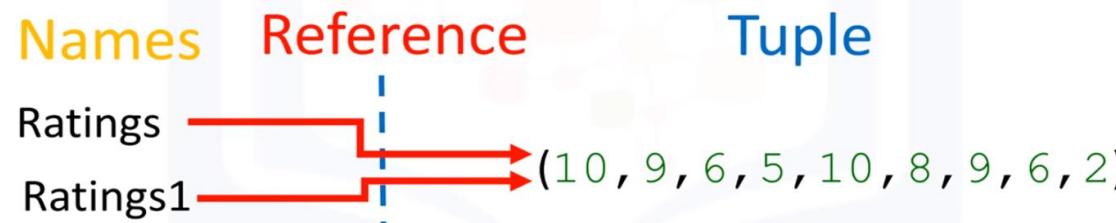
Artinya bahwa isi dari tuples tersebut tetap, tidak dapat diubah.

Contoh:

Ratings [2] = 4, hal ini TIDAK DAPAT dilakukan, karena nilai sebenarnya dari

Ratings [2] = 6

~~Ratings [2] = 4~~



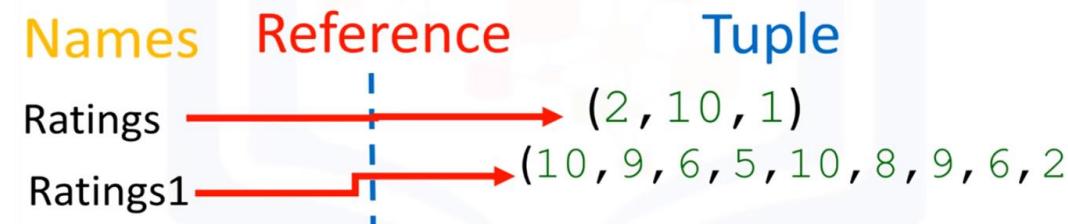


Tuples

Tuples bersifat immutable

Jika ada 2 tuples, dapat berisi anggota/element yang berbeda.

Ratings = (2, 10, 1)





Tuples

Jika ingin melakukan manipulasi pada isi tuples, maka buatlah variabel baru untuk menyimpan hasil manipulasinya.

Contoh:

Akan melakukan pengurutan (sort) dari isi tuple. Maka digunakan variabel baru dengan nama: RatingsSorted

Ratings = (10, 9, 6, 5, 10, 8, 9, 6, 2)

RatingsSorted = sorted(Ratings)

(10, 9, 6, 5, 10, 8, 9, 6, 2)

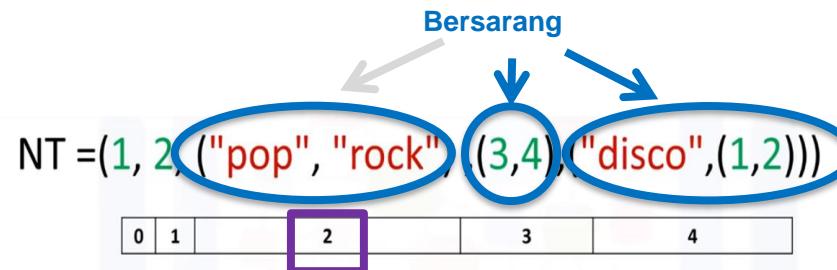


Tuples

Pada tuples dapat dibuat tuples bersarang (tuples nesting)

Pada contoh:

NT [2] berisi 2 anggota yaitu: “pop”, dan “rock” , hal ini mengakibatkan tuples menjadi bersarang.



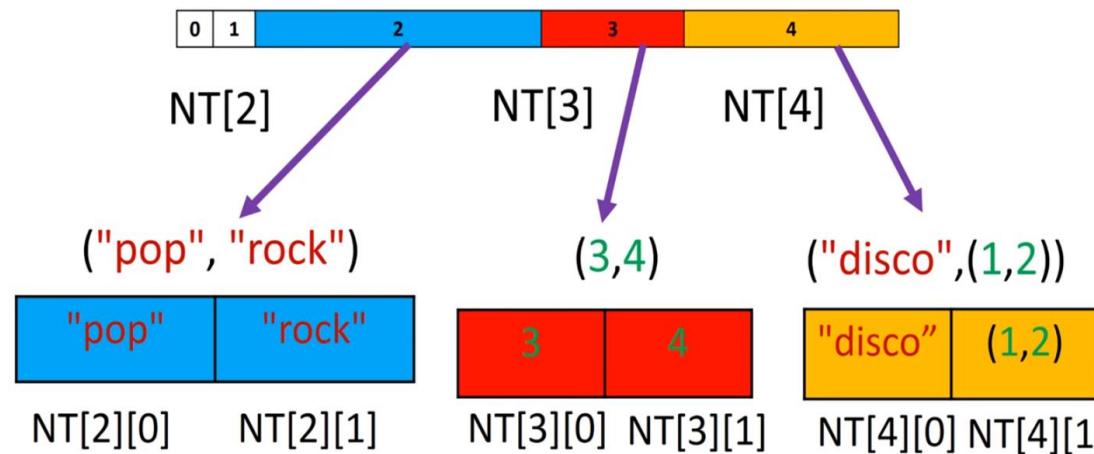
NT[2]: ("pop", "rock") [1] = "rock" → NT[2] [1] = "rock"

0	1
---	---

Tuples

Tuples bersarang

$NT = (1, 2, ("pop", "rock"), (3,4), ("disco", (1,2)))$





Tuples

Tuples bersarang

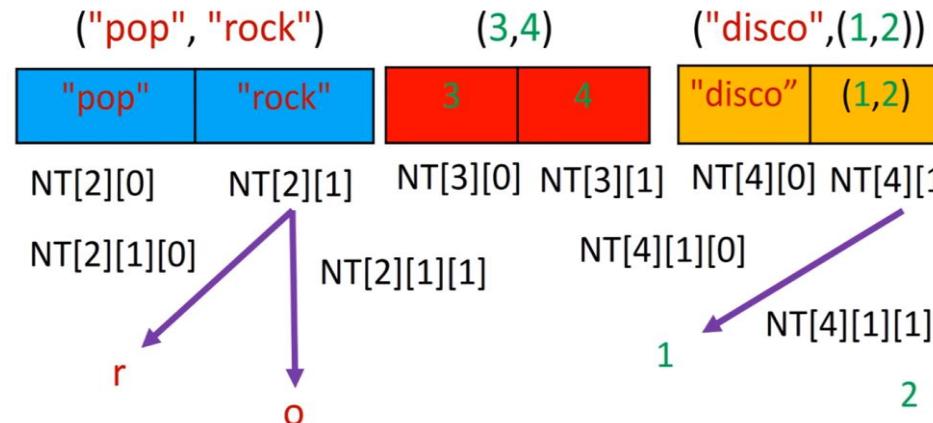
Dapat dibuat lebih dari 2 level bersarang.

Tampak pada Contoh ke-1:

NT [2][1] memiliki level sarang ke-3 dengan anggota/elemennya adalah r dan o

Maka:

NT [2][1][0] = r dan NT [2][1][1] = o





Bagian 2

List



List

List merupakan daftar yang memiliki anggota/element yang berurutan.

Anggota/element dari list dituliskan diantara tanda kurung siku [...] dan masing-masing anggota/element dipisahkan oleh tanda koma (,).

List bersifat **mutable** (tidak tetap).

```
L = [ "Michael Jackson", 10.1, 1982 ]
```

```
[ "Michael Jackson", 10.1, 1982, [ 1, 2 ], ( 'A', 1 ) ]
```

List dapat berisikan anggota/element dengan tipe data string, integer, float, bersarang.



List

Anggota/elemen pada list dapat diakses dengan menggunakan indeks.
Indeks pada list dapat dibuat dalam bentuk indeks positif maupun indeks negatif.

L = ["Michael Jackson", 10.1, 1982]

0	"Michael Jackson"
1	10.1
2	1982

L[0]: "Michael Jackson"

L[1]: 10

L[2]: 1982



Indeks Positif



List

Anggota/elemen pada list dapat diakses dengan menggunakan indeks.
Indeks pada list dapat dibuat dalam bentuk indeks positif maupun indeks negatif.

$L = ["Michael Jackson", 10.1, 1982]$

-3	0	"Michael Jackson"
-2	1	10.1
-1	2	1982

$L[-3]: "Michael Jackson"$

$L[-2]: 10$

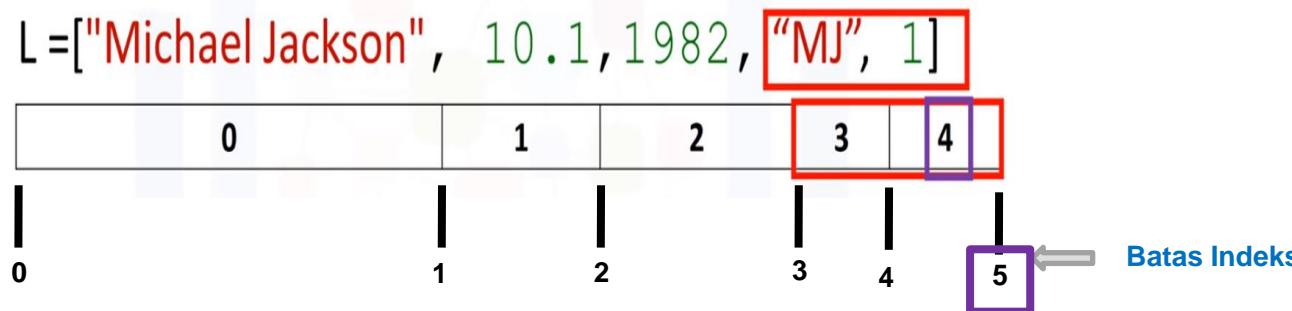
$L[-1]: 1982$

Indeks Negatif



List

Pada list dapat dilakukan pemotongan (slicing) dengan menggunakan indeks.



L[3:5]:["MJ", 1]



List

Anggota/element pada list dapat ditambahkan dengan perintah penggabungan menggunakan tanda (+)

```
L=[“Michael Jackson”, 10.1, 1982]
```

```
L1 = L+[“pop”, 10]
```

```
L1=[“Michael Jackson”, 10.1, 1982, “pop”, 10]
```

0	1	2	3	4
---	---	---	---	---



List

List bersifat mutable (tidak tetap), dan untuk melakukan pengubahan dapat digunakan metode.

Penggunaan metode pada list didahului oleh tanda titik (.) setelah itu nama metodenya, dan perubahan ditulis diantara tanda buka dan tutup kurung (....)

Contoh:

L.Extend (.....)

```
L=[“Michael Jackson”, 10.1, 1982]
```

```
L.extend([“pop”, 10])
```

```
L=[“Michael Jackson”, 10.1, 1982, “pop”, 10]
```

0	1	2	3	4
---	---	---	---	---



List

Penambahan anggota/element juga dapat dilakukan dengan menggunakan tanda (+), tampak pada contoh dibawah ini.

```
L=[“Michael Jackson”, 10.1, 1982]
```

```
L1 = L+[“pop”, 10]
```

```
L1=[“Michael Jackson”, 10.1, 1982, “pop”, 10]
```

0	1	2	3	4
---	---	---	---	---

List

Penggunaan metode “append” digunakan untuk menambahkan anggota/elemen ke dalam list, hanya saja data yang ditambahkan berada hanya pada satu indeks saja.

```
L=[“Michael Jackson”, 10.1, 1982]
```

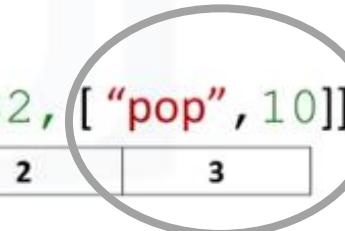
```
L.append([“pop”, 10])
```

Ditambahkan 2 anggota pada list

```
L=[“Michael Jackson”, 10.1, 1982, [“pop”, 10]]
```

0	1	2	3
---	---	---	---

Data yang ditambahkan berada hanya pada satu indeks saja





List

Isi dalam list dapat diubah dengan menggunakan indeks.

```
A=[“disco”, 10, 1.2]
```

```
A[0]=“hard rock”
```

```
A=[“hard rock”, 10, 1.2]
```



List

Isi dari list dapat dihapus dengan menggunakan fungsi “**DEL**”

Contoh ke-1:

```
A=["hard rock", 10, 1.2]
```



```
del(A[0])
```



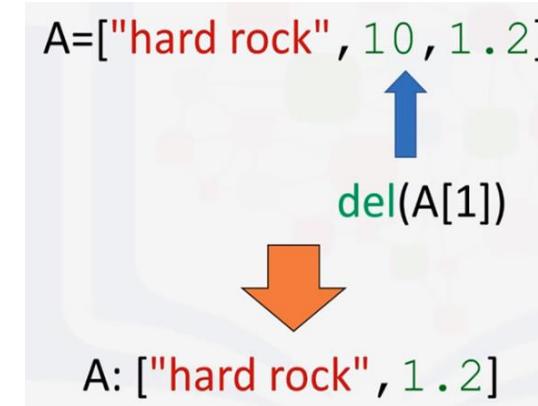
```
A:[10, 1.2]
```



List

Isi dari list dapat dihapus dengan menggunakan fungsi “**DEL**”

Contoh ke-2:





List

Suatu string dapat diubah ke bentuk list dengan menggunakan metode “**Split**”. Split yang dilakukan berdasarkan pada spasi antar kata.

```
"hard rock".split()
```

```
["hard", "rock"]
```



List

Suatu string dapat diubah ke bentuk list dengan menggunakan metode “**Split**” dan bisa diberikan kondisi khusus, misalkan akan mengubah string menjadi list dengan string yang dipisahkan oleh tanda koma (,)

```
"A,B,C,D".split(",")
```

```
["A", "B", "C", "D"]
```

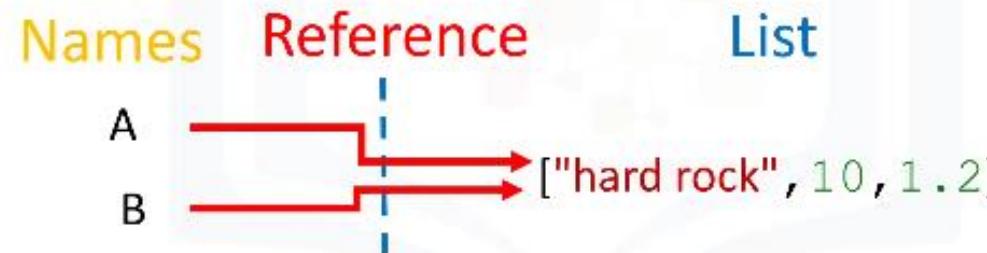


List

Suatu variabel akan memiliki isi list yang sama jika menggunakan tanda sama dengan seperti pada contoh: B= A

```
A=["hard rock", 10, 1.2]
```

```
B=A
```



List

Suatu variabel akan memiliki isi list yang sama jika menggunakan tanda sama dengan (=)

Misalkan:

List B sama dengan list A

Maka:

Jika ada perubahan pada anggota/element di list A akan mempengaruhi anggota/element yang ada di list B (anggota/element yang berada pada indeks yang sama)

B=A

B[0] = "hard rock"  B[0]: " banana "
A[0] = " banana "





List

Isi list dapat di cloning/duplikasi pada variabel list yang lainnya.

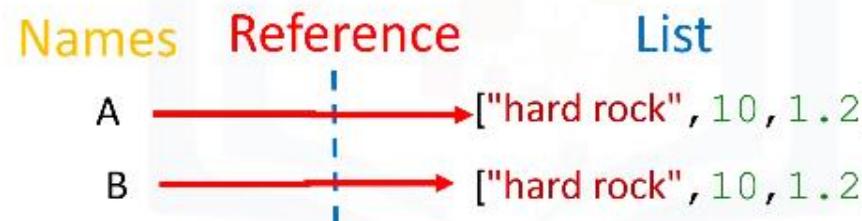
Cloning/duplikasi dilakukan dengan menggunakan tanda double titik dua.

Contoh:

B = A[::]

A=["hard rock", 10, 1.2]

B=A[:]

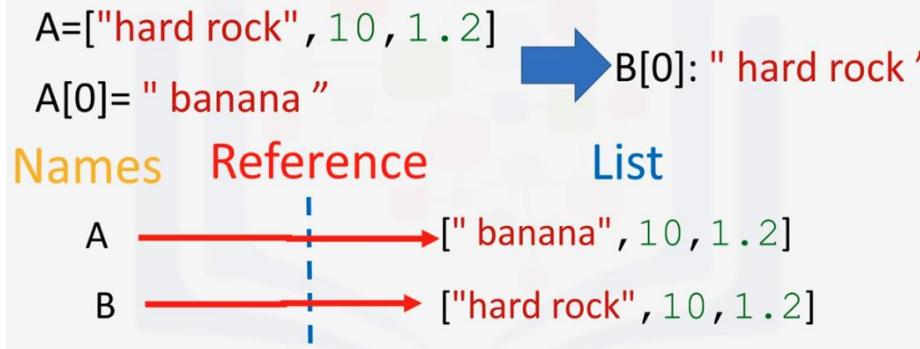




List

Jika isi list di cloning/duplikasi pada suatu variabel yang lain, maka perubahan isi list yang terjadi pada salah satu list tidak akan mempengaruhi list yang lain.

B=A[:]





List

Fungsi “Help” dapat digunakan untuk mendapatkan informasi tentang list.

```
A=["hard rock", 10, 1.2]
```

```
help(A)
```



Bagian 3

Sets





Set

- Set adalah sebuah tipe data koleksi data himpunan (set).
 - Selain list dan tuple dapat diinput tipe data yang berbeda.
- Tidak seperti list dan tupel, set tidak diurutkan.
 - Set tidak memperhatikan posisi elemen
- Set hanya memiliki elemen unik
 - Hanya ada satu elemen tertentu dalam satu set.



Membuat sebuah Set

```
Set1={"pop", "rock", "soul", "hard rock", "rock", "R&B", "rock", "disco" }
```

Untuk membuat Set menggunakan tanda Kurung Kurawal { }, dan elemen-elemen set ditempatkan diantara tanda kurung kurawal.



Sets: Creating a Set

Set1={"pop", "rock", "soul", "hard rock", "rock", "R&B", "rock", "disco" }

Set1: { 'rock', "R&B", "disco", "hard rock", "pop", "soul"}

Anda perhatikan ada item duplikat. Ketika set sebenarnya dibuat, item duplikat tidak akan ada.



Konversi List ke Set

LIST

set()



Konversi List ke Set

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

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

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

Konversi dari List
ke Set

set()

album_set

Untuk mengkonversi list ke set dengan menggunakan fungsi set yang disebut tipe-casting. Cukup menggunakan list sebagai input ke set fungsi.



Operasi Set

```
A = {"Thriller", "Back in Black", "AC/DC"}  
A.add("NSYNC")
```

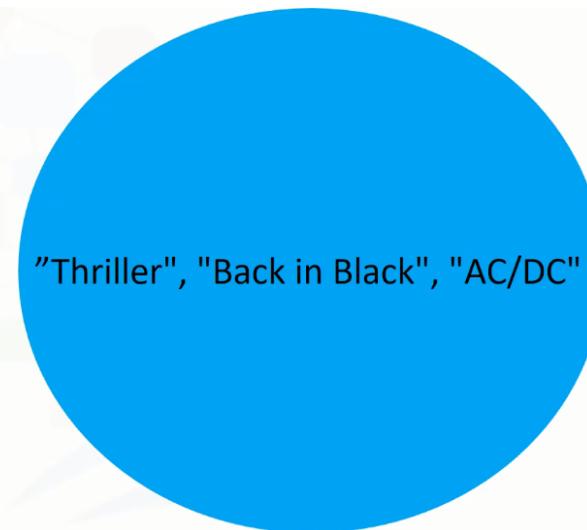


Diagram Venn Set A



Menambahkan elemen set

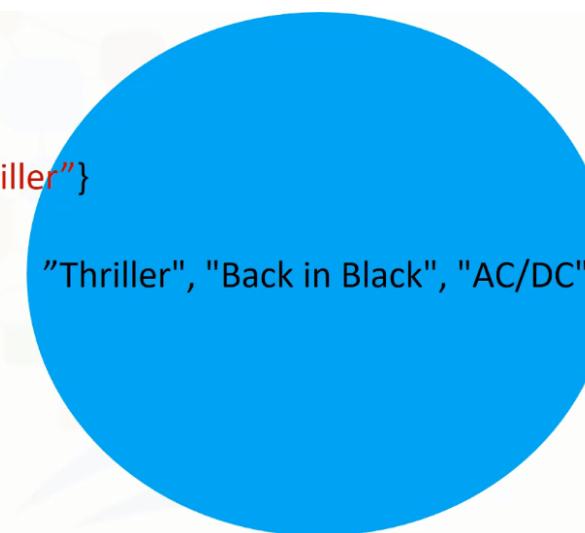
Untuk menambahkan elemen set menggunakan perintah **Add**.

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

```
A.add("NSYNC")
```

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

Menambahkan elemen set
'NSYNC' pada set A





Menghapus elemen Set

Untuk menghapus elemen set menggunakan perintah `remove`.

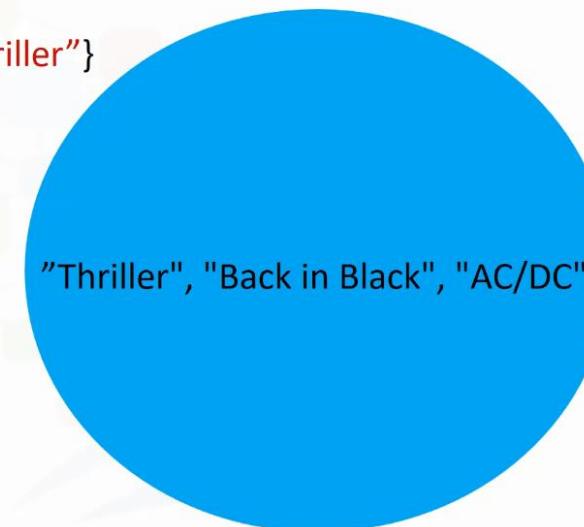
Set Operations

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

A.remove(**“NSYNC”**)

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

Menghapus elemen set ‘NSYNC’
pada set A





Mengecek elemen dalam Set

Untuk mengecek elemen dalam suatu set menggunakan perintah `in`.

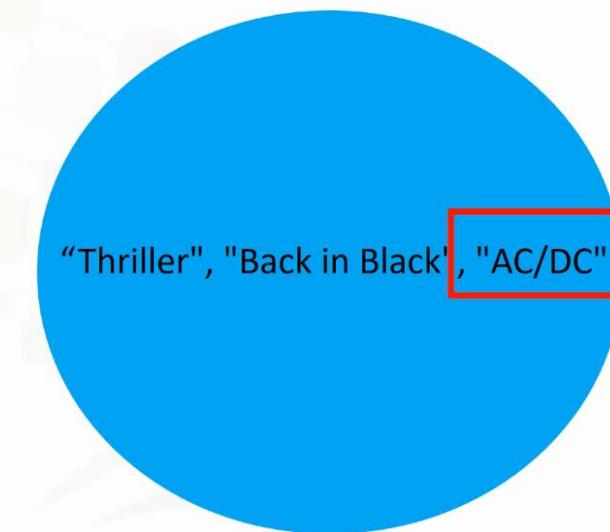
Set Operations

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

“AC/DC” `in` A

True

Mengecek elemen set ‘AC/DC’
pada set A





Set Operations

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

“Who” in A

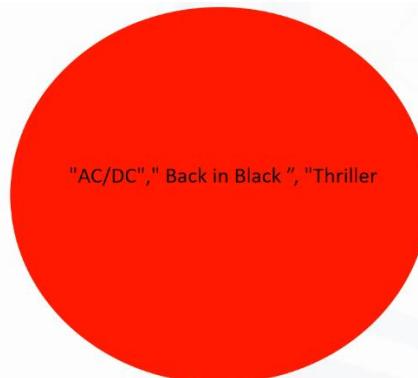
False

“Thriller”, "Back in Black", "AC/DC"

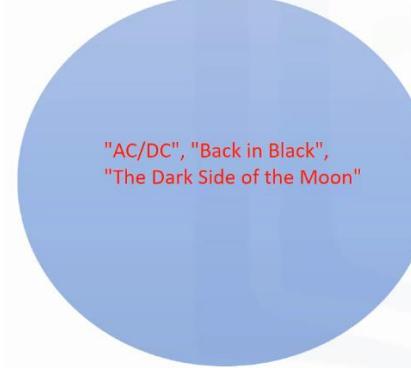




Operasi Matematika pada Set



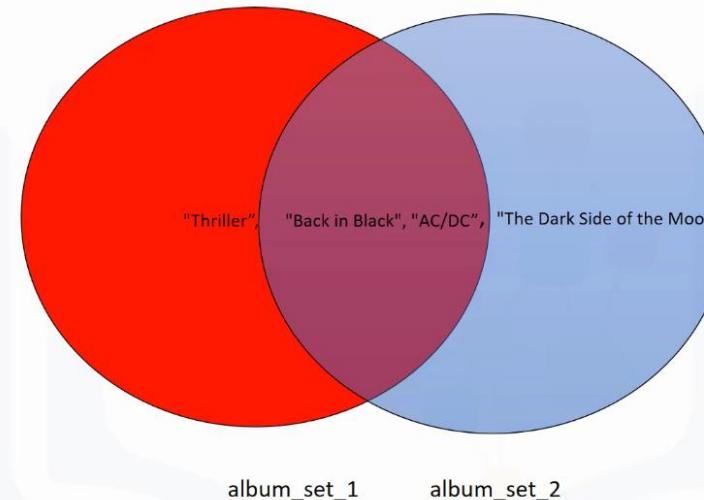
```
album_set_1 = {"AC/DC", "Back in Black", "Thriller }
```



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



Operasi Matematika pada Set



Operasi Matematika pada Set

Intersection adalah operasi untuk mencari elemen yang sama yang dimiliki oleh dua set. Operasi intersection menggunakan simbol ampersand (&).

```
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"}
```

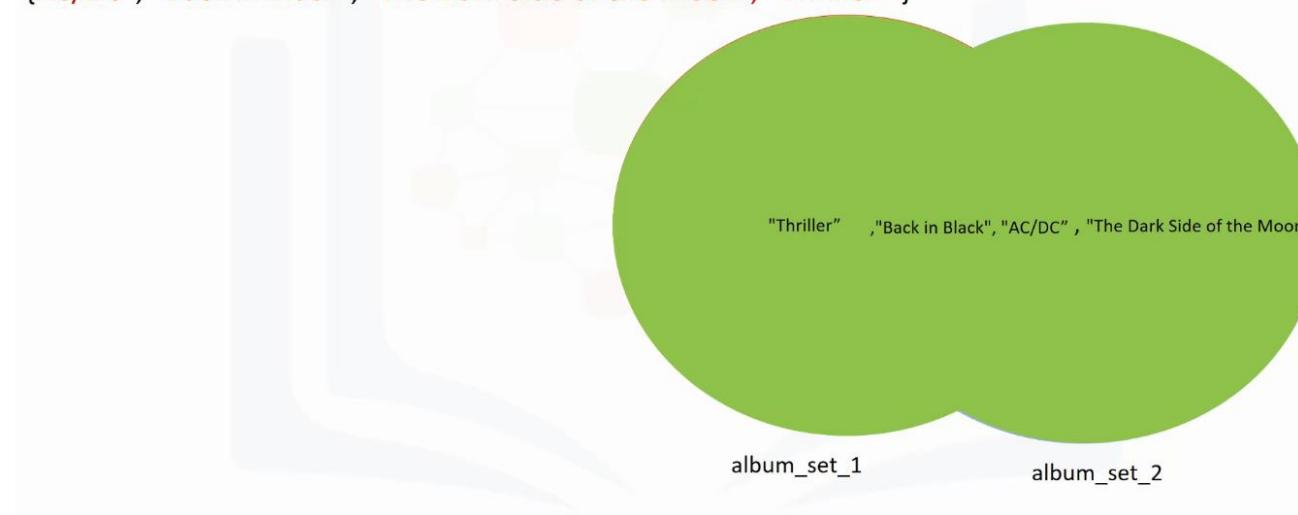
Operasi
Intersection



Operasi Matematika pada Set

Union adalah operasi menggabungkan set atau himpunan.
`album_set_1. union(album_set_2)`

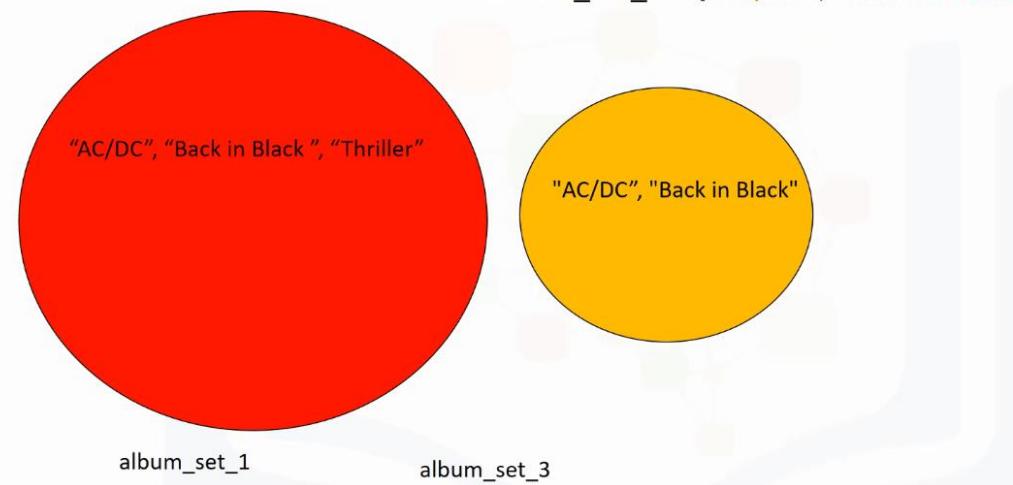
`{AC/DC", "Back in Black", "The Dark Side of the Moon", "Thriller" }`





Operasi Matematika pada Set

```
album_set_1 = {"AC/DC", "Back in Black", "Thriller"}  
album_set_3 = {"AC/DC", "Back in Black"}
```





Operasi Matematika pada Set

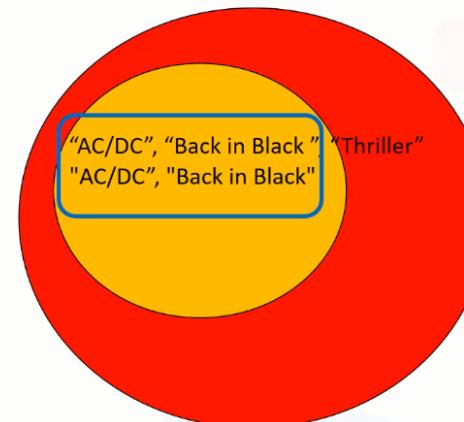
Subset : sebuah himpunan adalah bagian dari himpunan besar.

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

album_set_3 = {"AC/DC", "Back in Black"}

album_set_3.issubset(album_set1)

True



album_set_1

album_set_3



Bagian 4

Dictionary (Kamus)





Dictionary (Kamus)

- Dictionary (kamus) adalah jenis koleksi dalam Python yang memiliki Key dan Value.

List

Index	
0	Element 1
1	Element 2
2	Element 3
3	Element 4
.....

Element

Dictionary

Key: is a index by label

Key 1	Value 1
Key 1	Value 2
Key 2	Value 3
Key 3	Value 4
.....

Element/Values

Dictionary (Kamus)

- Untuk membuat Dictionary (kamus) dalam Python di gunakan kurung kurawal {}.
- Key adalah elemen pertama, bersifat tetap dan harus unik. Setiap key diikuti oleh value (nilai) yang dipisahkan oleh titik dua.
- Value (Nilai) dapat berniai tetap, berubah-ubah, dan boleh bernilai sama.
- Setiap { "key1":1, "key2 ":"2","key3": [3,3,3], "key4":(4,4,4), ('key5'):5 }



Dictionary (Kamus)

Berikut adalah contoh Dictionary (kamus) dari album, di mana key nya adalah judul album dan Value (nilai) nya adalah tahun rilis.

"Thriller":	1982,	"Back in Black":	1980,	"The Dark Side of the Moon":	1973,	"The Bodyguard":	1992,
-------------	-------	------------------	-------	------------------------------	-------	------------------	-------



Dictionary (Kamus)

Dictionary (kamus) dapat ditambahkan yaitu dengan menambahkan pasangan key dan value.

"Thriller"	"1982"
"Back in Black"	"1980"
"The Dark Side of the Moon"	"1973"
"The Bodyguard"	"1992"



"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"
"Rumours"	"1977"



Dictionary (Kamus)

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"
"Rumours"	"1977"



Dictionary (Kamus)

Key digunakan untuk mencari Value. Digunakan kurung siku []; argumen untuk melakukan pencarian berdasarkan key, dan selanjutnya menghasilkan nilai. Misalkan menggunakan kunci "Thriller," ini menghasilkan value 1982.

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"
"Rumours"	"1977"

DICTIONARY["Thriller"]:



Dictionary (Kamus)

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"
"Rumours"	"1977"

Value

DICT["Back in Black"]:"1980"



Dictionary (Kamus)

Untuk menghapus elemen dari Dictionary (kamus) menggunakan perintah **del**.

"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"

del(DICT['Thriller'])



Dictionary (Kamus)

- Untuk mengecek apakah terdapat key tertentu dalam Dictionary (kamus) digunakan
- perintah `in`. Jika key terdapat dalam kamus akan menghasilkan true, jika tidak terdapat key akan menghasilkan false.

"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

True



Dictionary (Kamus)

"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"

“Starboy” in DICT

False

Dictionary (Kamus)

"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"]

Untuk melihat semua key
dalam Dictionary (kamus).



Dictionary (Kamus)

"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”]

Untuk melihat semua value
dalam Dictionary (kamus).



Terima kasih