

# 3. Pemrograman Berorientasi Objek

## *String, List, Tuple, Dictionary*

### String Python

String adalah jenis tipe data pada Python yang berisi karakter-karakter atau simbol-simbol tertentu. String merupakan salah satu tipe data dasar dalam Python. Python mempunyai dua tipe string yaitu string literal dan string variabel.

Dibawah ini adalah contoh implementasi string:

print("Hello World")

### Manipulate String

String memiliki beberapa operasi manipulasi yang dapat dilakukan. Beberapa operasi tersebut antara lain:

- Menggabungkan String

- Mengambil Substring

- Mengubah String

- Operasi Format String

- Operasi String

### List Python

Daftar Python menggunakan Python struktur data yang paling dekat dengan struktur data list. Meskipun demikian strukturnya tetap sama dengan strukturnya list. Perbedaan utamanya antara tuple dan daftarnya adalah bahwa tuple tidak dapat dimodifikasi sebaliknya list dapat.

Python memiliki cara kerjanya sendiri dalam menjalankan operasi pada list. Misalkan ketika kita menulis list = [1, 2, 3] maka list akan dianggap sebagai sebuah objek list. Jadi ketika kita menulis list[0] maka list[0] akan dianggap sebagai objek list pertama dalam list. Jadi list[0] memiliki fungsi dan metode sendiri.

Daftar memiliki beberapa operasi yang dapat dilakukan. Beberapa operasi tersebut antara lain:

- Mengakses dan Mengubah Data

- Menghitung Panjang Daftar

- Mengambil Sublist

- Menggabungkan Daftar

- Mengubah Daftar

- Operasi Daftar

### Tuple Python

Sekumpulan tuple adalah urutan objek Python yang tidak berubah. Tuple adalah urutan, seperti daftar. Perbedaan utamanya antara tuple dan daftarnya adalah bahwa tuple tidak dapat dimodifikasi sebaliknya list dapat.

tup1 = ('Riska', 'Kimia', 1993, 2017)  
tup2 = (1, 2, 3, 4, 5)  
tup3 = "a", "b", "c", "d"

### Dictionary Python

Dictionary Python adalah dictionary atau kamus. Dictionary Python merupakan struktur data yang berisi pasangan kunci dan nilai. Dalam dictionary, setiap kunci yang diberikan memiliki nilai yang bersesuaian. Dictionary Python memiliki fungsi dan metode sendiri.

Company Logo

# 3. Pemrograman Berorientasi Objek

## *String, List, Tuple, Dictionary*

### String Python

String adalah jenis tipe data pada Python yang berisi karakter-karakter atau simbol-simbol tertentu. String merupakan salah satu tipe data dasar dalam Python. Python mempunyai dua tipe string yaitu string literal dan string variabel.

Dibawah ini adalah contoh implementasi string:

print("Hello World")

### Manipulate String

String memiliki beberapa operasi manipulasi yang dapat dilakukan. Beberapa operasi tersebut antara lain:

- Menggabungkan String

- Mengambil Substring

- Mengubah String

- Operasi Format String

- Operasi String

### List Python

Daftar Python menggunakan Python struktur data yang paling dekat dengan struktur data list. Meskipun demikian strukturnya tetap sama dengan strukturnya list. Untuk membuat list pada Python, kita dapat menulis list diantara tanda kurung siku [ ].

Python memiliki cara kerja khusus untuk mengakses elemen-elemen list. Misalnya, kita bisa menulis list sebagai [a, b, c, d, e, f]. Untuk mengakses element ke-3, kita bisa menulis list[3].

Untuk mengakses element ke-2, kita bisa menulis list[1].

Untuk mengakses element ke-4, kita bisa menulis list[4].

Untuk mengakses element ke-5, kita bisa menulis list[5].

Untuk mengakses element ke-6, kita bisa menulis list[6].

Untuk mengakses element ke-7, kita bisa menulis list[7].

Untuk mengakses element ke-8, kita bisa menulis list[8].

Untuk mengakses element ke-9, kita bisa menulis list[9].

Untuk mengakses element ke-10, kita bisa menulis list[10].

Untuk mengakses element ke-11, kita bisa menulis list[11].

Untuk mengakses element ke-12, kita bisa menulis list[12].

Untuk mengakses element ke-13, kita bisa menulis list[13].

Untuk mengakses element ke-14, kita bisa menulis list[14].

Untuk mengakses element ke-15, kita bisa menulis list[15].

Untuk mengakses element ke-16, kita bisa menulis list[16].

Untuk mengakses element ke-17, kita bisa menulis list[17].

Untuk mengakses element ke-18, kita bisa menulis list[18].

Untuk mengakses element ke-19, kita bisa menulis list[19].

Untuk mengakses element ke-20, kita bisa menulis list[20].

Untuk mengakses element ke-21, kita bisa menulis list[21].

Untuk mengakses element ke-22, kita bisa menulis list[22].

Untuk mengakses element ke-23, kita bisa menulis list[23].

Untuk mengakses element ke-24, kita bisa menulis list[24].

Untuk mengakses element ke-25, kita bisa menulis list[25].

Untuk mengakses element ke-26, kita bisa menulis list[26].

Untuk mengakses element ke-27, kita bisa menulis list[27].

Untuk mengakses element ke-28, kita bisa menulis list[28].

Untuk mengakses element ke-29, kita bisa menulis list[29].

Untuk mengakses element ke-30, kita bisa menulis list[30].

Untuk mengakses element ke-31, kita bisa menulis list[31].

Untuk mengakses element ke-32, kita bisa menulis list[32].

Untuk mengakses element ke-33, kita bisa menulis list[33].

Untuk mengakses element ke-34, kita bisa menulis list[34].

Untuk mengakses element ke-35, kita bisa menulis list[35].

Untuk mengakses element ke-36, kita bisa menulis list[36].

Untuk mengakses element ke-37, kita bisa menulis list[37].

Untuk mengakses element ke-38, kita bisa menulis list[38].

Untuk mengakses element ke-39, kita bisa menulis list[39].

Untuk mengakses element ke-40, kita bisa menulis list[40].

Untuk mengakses element ke-41, kita bisa menulis list[41].

Untuk mengakses element ke-42, kita bisa menulis list[42].

Untuk mengakses element ke-43, kita bisa menulis list[43].

Untuk mengakses element ke-44, kita bisa menulis list[44].

Untuk mengakses element ke-45, kita bisa menulis list[45].

Untuk mengakses element ke-46, kita bisa menulis list[46].

Untuk mengakses element ke-47, kita bisa menulis list[47].

Untuk mengakses element ke-48, kita bisa menulis list[48].

Untuk mengakses element ke-49, kita bisa menulis list[49].

Untuk mengakses element ke-50, kita bisa menulis list[50].

Untuk mengakses element ke-51, kita bisa menulis list[51].

Untuk mengakses element ke-52, kita bisa menulis list[52].

Untuk mengakses element ke-53, kita bisa menulis list[53].

Untuk mengakses element ke-54, kita bisa menulis list[54].

Untuk mengakses element ke-55, kita bisa menulis list[55].

Untuk mengakses element ke-56, kita bisa menulis list[56].

Untuk mengakses element ke-57, kita bisa menulis list[57].

Untuk mengakses element ke-58, kita bisa menulis list[58].

Untuk mengakses element ke-59, kita bisa menulis list[59].

Untuk mengakses element ke-60, kita bisa menulis list[60].

Untuk mengakses element ke-61, kita bisa menulis list[61].

Untuk mengakses element ke-62, kita bisa menulis list[62].

Untuk mengakses element ke-63, kita bisa menulis list[63].

Untuk mengakses element ke-64, kita bisa menulis list[64].

Untuk mengakses element ke-65, kita bisa menulis list[65].

Untuk mengakses element ke-66, kita bisa menulis list[66].

Untuk mengakses element ke-67, kita bisa menulis list[67].

Untuk mengakses element ke-68, kita bisa menulis list[68].

Untuk mengakses element ke-69, kita bisa menulis list[69].

Untuk mengakses element ke-70, kita bisa menulis list[70].

Untuk mengakses element ke-71, kita bisa menulis list[71].

Untuk mengakses element ke-72, kita bisa menulis list[72].

Untuk mengakses element ke-73, kita bisa menulis list[73].

Untuk mengakses element ke-74, kita bisa menulis list[74].

Untuk mengakses element ke-75, kita bisa menulis list[75].

Untuk mengakses element ke-76, kita bisa menulis list[76].

Untuk mengakses element ke-77, kita bisa menulis list[77].

Untuk mengakses element ke-78, kita bisa menulis list[78].

Untuk mengakses element ke-79, kita bisa menulis list[79].

Untuk mengakses element ke-80, kita bisa menulis list[80].

Untuk mengakses element ke-81, kita bisa menulis list[81].

Untuk mengakses element ke-82, kita bisa menulis list[82].

Untuk mengakses element ke-83, kita bisa menulis list[83].

Untuk mengakses element ke-84, kita bisa menulis list[84].

Untuk mengakses element ke-85, kita bisa menulis list[85].

Untuk mengakses element ke-86, kita bisa menulis list[86].

Untuk mengakses element ke-87, kita bisa menulis list[87].

Untuk mengakses element ke-88, kita bisa menulis list[88].

Untuk mengakses element ke-89, kita bisa menulis list[89].

Untuk mengakses element ke-90, kita bisa menulis list[90].

Untuk mengakses element ke-91, kita bisa menulis list[91].

Untuk mengakses element ke-92, kita bisa menulis list[92].

Untuk mengakses element ke-93, kita bisa menulis list[93].

Untuk mengakses element ke-94, kita bisa menulis list[94].

Untuk mengakses element ke-95, kita bisa menulis list[95].

Untuk mengakses element ke-96, kita bisa menulis list[96].

Untuk mengakses element ke-97, kita bisa menulis list[97].

Untuk mengakses element ke-98, kita bisa menulis list[98].

Untuk mengakses element ke-99, kita bisa menulis list[99].

Untuk mengakses element ke-100, kita bisa menulis list[100].

### Tuple Python

Sekarang tuple adalah urutan objek Python yang tidak berubah. Tuple adalah urutan, seperti daftar. Perbedaan utama antara tuple dan daftarnya adalah bahwa bahwa tuple tidak dapat dimodifikasi seperti list. Tuple hanya adalah sekumpulan objek Python menggunakan tanda kurung tajuk.

tup1 = ('Riskya', 'kimia', 1993, 2017)

tup2 = (1, 2, 3, 4, 5)

tup3 = "a", "b", "c", "d"

### Dictionary Python

Dictionary Python adalah teknologi yang memungkinkan kita untuk menyimpan data dalam bentuk pasangan kunci-nilai. Dalam Python, dictionary adalah tipe data yang terdiri dari pasangan kunci-nilai. Kunci adalah objek yang unik dan nilainya adalah objek yang dapat diakses melalui kunci.

Company Logo

# *String Python*

*String adalah jenis yang paling populer di bahasa pemrograman. Kita bisa membuatnya hanya dengan melampirkan karakter dalam tanda kutip. Python memperlakukan tanda kutip tunggal sama dengan tanda kutip ganda. Membuat string semudah memberi nilai pada sebuah variabel.*

*Dibawah ini adalah contoh sederhana dari sebuah string pada bahasa pemrograman Python.*

```
print("Hello World")
```

# *Mengakses Nilai dalam String*

*Python tidak menggunakan tipe karakter titik koma ;  
Ini diperlakukan sebagai string dengan panjang satu,  
sehingga juga dianggap sebagai substring.*

---

*Untuk mengakses substring, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan substring Anda. Sebagai contoh :*

```
name = 'Aidil' message = "Aidil Saputra Kirsan"  
print ("name[0]: ", name[0])  
print ("message[2:9]: ", message[2:9])
```

```
name = 'Aidil' message = "Aidil Saputra Kirsan"  
print ("name[0]: ", name[0])  
print ("message[2:9]: ", message[2:9])
```

# *Mengupdate String*

*Anda dapat “memperbarui” string yang ada dengan (kembali) menugaskan variabel ke string lain. Nilai baru dapat dikaitkan dengan nilai sebelumnya atau ke string yang sama sekali berbeda sama sekali. Sebagai contoh*

---

```
message = 'Aidil Saputra'  
print ("Updated String :- ", message[:6] + 'Kirsan')
```

# *Operator Format String*

*Salah satu fitur Python yang paling keren adalah format string operator %. Operator ini unik untuk string dan membuat paket memiliki fungsi dari keluarga printf C () C. berikut adalah contoh sederhananya :*

```
print ("My name is %s and weight is %d kg!" % ('Aidil', 70))
```



Operator	Penjelasan
%c	character
%s	Konversi string melalui str () sebelum memformat
%i	Dianggap sebagai bilangan bulat desimal
%d	Dianggap sebagai bilangan bulat desimal
%u	Unsigned decimal integer
%o	Bilangan bulat oktal
%x	Bilangan bulat heksadesimal (huruf kecil)
%X	Bilangan bulat heksadesimal (huruf besar)
%e	Notasi eksponensial (dengan huruf kecil 'e')
%E	Notasi eksponensial (dengan huruf besar 'E')
%f	Bilangan real floating point
%g	Yang lebih pendek dari% f dan% e
%G	Lebih pendek dari% f dan% E

# *List Python*

*Dalam bahasa pemrograman Python, struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeksnya. Indeks pertama dalam list adalah nol, indeks kedua adalah satu dan seterusnya.*

*Python memiliki enam jenis urutan built-in, namun yang paling umum adalah list dan tuple. Ada beberapa hal yang dapat Anda lakukan dengan semua jenis list. Operasi ini meliputi pengindeksan, pengiris, penambahan, perbanyak, dan pengecekan keanggotaan. Selain itu, Python memiliki fungsi built-in untuk menemukan panjang list dan untuk menemukan elemen terbesar dan terkecilnya.*

# Membuat List Python

*List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama jenisnya.*

*Membuat list sangat sederhana, tinggal memasukkan berbagai nilai yang dipisahkan koma di antara tanda kurung siku. Dibawah ini adalah contoh sederhana pembuatan list dalam bahasa Python.*

```
list1 = ["kimia", "fisika", 1993, 2017]
list2 = [1, 2, 3, 4, 5]
list3 = ["a", "b", "c", "d"]
```

## Akses Nilai Dalam List Python

*Untuk mengakses nilai dalam list python, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut.*

```
list1 = ["fisika", "kimia", 1993, 2017]
list2 = [1, 2, 3, 4, 5, 6, 7]

print ("list1[0]: ", list1[0])
print ("list2[1:5]: ", list2[1:5])
```

*Membuat list sangat sederhana, tinggal memasukkan berbagai nilai yang dipisahkan koma di antara tanda kurung siku. Dibawah ini adalah contoh sederhana pembuatan list dalam bahasa Python.*

```
list1 = ['kimia', 'fisika', 1993, 2017]  
list2 = [1, 2, 3, 4, 5 ]  
list3 = ["a", "b", "c", "d"]
```

# *Akses Nilai Dalam List Python*

*Untuk mengakses nilai dalam list python, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut.*

```
list1 = ['fisika', 'kimia', 1993, 2017]  
list2 = [1, 2, 3, 4, 5, 6, 7 ]
```

```
print ("list1[0]: ", list1[0])  
print ("list2[1:5]: ", list2[1:5])
```

# *Update Nilai Dalam List Python*

*Anda dapat memperbarui satu atau beberapa nilai di dalam list dengan memberikan potongan di sisi kiri operator penugasan, dan Anda dapat menambahkan nilai ke dalam list dengan metode append () .*

## *Update Nilai Dalam List Python*

```
list = ['Fisika', 'Kimia', 1993, 2017]
print("Nilai ada pada index 2 : ", list[2])
list[2] = 2001
print("Nilai baru ada pada index 2 : ", list[2])
```

## *Hapus Nilai Dalam List Python*

Untuk menghapus nilai di dalam list python, Anda dapat menggunakan salah satu pernyataan del jika anda tidak perlu elemen yang Anda hapus. Anda dapat menggunakan metode remove() jika Anda tidak tahu persis item mana yang akan dibuang. Sebagi contoh :

```
list = ['Fisika', 'Kimia', 1993, 2017]
print(list)
del list[2]
print("Setelah dihapus nilai pada index 2 : ", list)
```

# *Update Nilai Dalam List Python*

```
list = ['fisika', 'kimia', 1993, 2017]
print ("Nilai ada pada index 2 : ", list[2])
```

```
list[2] = 2001
print ("Nilai baru ada pada index 2 : ", list[2])
```

## *Hapus Nilai Dalam List Python*

*Untuk menghapus nilai di dalam list python, Anda dapat menggunakan salah satu pernyataan del jika Anda tahu persis elemen yang Anda hapus. Anda dapat menggunakan metode remove() jika Anda tidak tahu persis item mana yang akan dihapus. Sebagai contoh :*

```
list = ['fisika', 'kimia', 1993, 2017]
```

```
print (list)
```

```
del list[2]
```

```
print ("Setelah dihapus nilai pada index 2 : ", list)
```

# Tuple Python

Sebuah tupel adalah urutan objek Python yang tidak berubah. Tupel adalah urutan, seperti daftar. Perbedaan utama antara tupel dan daftarnya adalah bahwa tupel tidak dapat diubah tidak seperti List Python. Tupel menggunakan tanda kurung, sedangkan List Python menggunakan tanda kurung siku.

```
tup1 = ('fisika', 'kimia', 1993, 2017)
tup2 = (1, 2, 3, 4, 5 )
tup3 = "a", "b", "c", "d"
```



## *Akses Nilai Dalam Tuple Python*

*Untuk mengakses nilai dalam tupel, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut. Sebagai contoh :*

```
tup1 = ('fisika', 'kimia', 1993, 2017)
tup2 = (1, 2, 3, 4, 5, 6, 7 )
```

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

## *Update Nilai Dalam Tuple Python*

*Tuple tidak berubah, yang berarti Anda tidak dapat memperbarui atau mengubah nilai elemen tupel. Anda dapat mengambil bagian dari tupel yang ada untuk membuat tupel baru seperti ditunjukkan oleh contoh berikut.*

```
tup1 = (12, 34.56)  
tup2 = ('abc', 'xyz')
```

```
tup3 = tup1 + tup2  
print (tup3)
```

## *Hapus Nilai Dalam Tuple Python*

*Menghapus elemen tuple individual tidak mungkin dilakukan. Tentu saja, tidak ada yang salah dengan menggabungkan tupel lain dengan unsur-unsur yang tidak diinginkan dibuang.*

```
tup = ('fisika', 'kimia', 1993, 2017)
print(tup)
```

```
# hapus tuple dengan statement del
del tup
```

```
# lalu buat kembali tuple yang baru dengan elemen
# yang diinginkan
tup = ('Bahasa', 'Literasi', 2020)
print("Setelah menghapus tuple :", tup)
```

# • Dictionary Python

*Dictionary Python berbeda dengan List ataupun Tuple. Karena setiap urutanya berisi key dan value. Setiap key dipisahkan dari value-nya oleh titik dua (:), item dipisahkan oleh koma, dan semuanya tertutup dalam kurung kurawal. Dictionary kosong tanpa barang ditulis hanya dengan dua kurung kurawal, seperti ini: {}*

## Akses Nilai Dalam Dictionary Python

Untuk mengakses elemen dictionary, Anda dapat menggunakan tanda kurung siku saja karena dalam format tersebut juga akan memproduksi sebuah baris. Baris tidak memiliki simbol selanjutnya.

```
dict = {"Name": "Zara", "Age": 7, "Class": "First"}  
print(dict["Name"]) # Output: Zara  
print(dict["Age"]) # Output: 7  
print(dict["Class"])
```

## Update Nilai Dalam Dictionary Python

Anda dapat memperbarui dictionary dengan menambahkan entri baru atau mengubah nilai salah satu entri yang ada. Atau menghapus entri yang ada sebelumnya dengan menggunakan del untuk memberikan nilai None pada entri yang dibuang.

```
dict = {"Name": "Zara", "Age": 7, "Class": "First"}  
dict["Age"] = 8 # Mengubah entri yang sudah ada  
dict["School"] = "DPS School" # Menambah entri baru  
  
print("dict[Age]: ", dict["Age"])  
print("dict[School]: ", dict["School"])
```

## Hapus Element Dictionary Python

Jika Anda menghapus elemen dictionary tersebut atau menghapus karakter di dictionary, maka anda dapat menghapus secara langsung menggunakan del untuk menghapus entri dictionary dalam satu operasi.

```
dict = {"Name": "Zara", "Age": 7, "Class": "First"}  
print("dict[Age]: ", dict["Age"])  
print("dict[Class]: ", dict["Class"])  
  
del dict["Name"] # Menghapus entri "Name"  
del dict["Age"] # Menghapus entri "Age"  
del dict["Class"] # Menghapus entri "Class"  
  
print("dict[Age]: ", dict["Age"])  
print("dict[Class]: ", dict["Class"])
```

## *Akses Nilai Dalam Dictionary Python*

*Untuk mengakses elemen Dictionary, Anda dapat menggunakan tanda kurung siku yang sudah dikenal bersama dengan key untuk mendapatkan nilainya. Berikut adalah contoh sederhananya*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
print ("dict['Name']: ", dict['Name'])  
print ("dict['Age']: ", dict['Age'])
```

## *Update Nilai Dalam Dictionary Python*

*Anda dapat memperbarui Dictionary dengan menambahkan entri baru atau pasangan nilai kunci, memodifikasi entri yang ada, atau menghapus entri yang ada seperti ditunjukkan pada contoh sederhana yang diberikan di bawah ini.*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
dict['Age'] = 8; # Mengubah entri yang sudah ada
dict['School'] = "DPS School" # Menambah entri baru
```

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

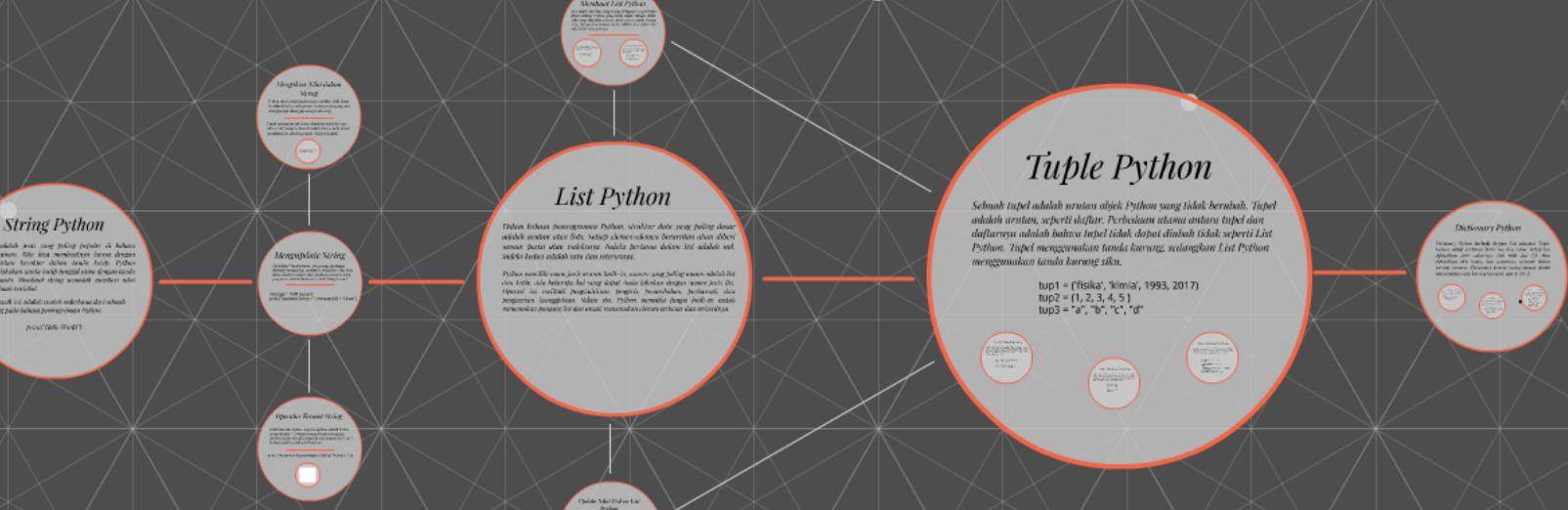
## *Hapus Elemen Dictionary Python*

*Anda dapat menghapus elemen Dictionary individual atau menghapus keseluruhan isi Dictionary. Anda juga dapat menghapus seluruh Dictionary dalam satu operasi.*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
  
print ("dict['Age']: ", dict['Age'])  
print ("dict['Name']: ", dict['Name'])  
print ("dict['Class']: ", dict['Class'])  
  
del dict['Name'] # hapus entri dengan key 'Name'  
#dict.clear() # hapus semua entri di dict  
#del dict # hapus dictionary yang sudah ada  
  
print ("dict['Age']: ", dict['Age'])  
print ("dict['Class']: ", dict['Class'])  
#print ("dict['Name']: ", dict['Name'])
```

# 3. Pemrograman Berorientasi Objek

# *String, List, Tuple, Dictionary*



Company Logo



Prezi