

MODUL V

LISTS & TUPLE

Tujuan :

1. Dapat mengetahui dan memahami lists dan tuple dalam Python
2. Dapat menggunakan dan mendeklarasikan lists dan tuple
3. Dapat membuat program sederhana untuk lists dan tuple

Tugas Pendahuluan

1. Sebutkan kelebihan dan kekurangan List dan Tuple pada python.!
2. Apa manfaat dari List dan Tuple pada python?

1. Dasar Teori

1.1. Pengertian lists

A. List Python

Dalam bahasa pemrograman Python, struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeks. 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.

B. 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 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5 ];  
list3 = ["a", "b", "c", "d"]
```

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 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5, 6, 7 ];  
print ("Nilai list1[0]:", list1[0])  
print ("Nilai list2[1:5]:", list2[1:5])
```

Output

```
Nilai list1[0]: physics  
Nilai list2[1:5]: [2, 3, 4, 5]
```

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 ().

```
list = ['physics', 'chemistry', 1997, 2000];  
print ("Nilai indeks ke-2 saat ini:")  
print (list[2])  
list[2] = 2001;  
print ("Nilai indeks ke-2 setelah diperbaharui:")  
print (list[2])
```

Output:

```
Nilai indeks ke-2 saat ini:  
1997  
Nilai indeks ke-2 setelah diperbaharui:  
2001
```

E. Menghapus List

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.

```
list1 = ['physics', 'chemistry', 1997, 2000];
print (list1)
del list1[2];
print ("Setelah nilai indeks ke-2 dihapus:")
print (list1)
```

Output:
physics, chemistry, 1997, 2000
Setelah nilai indeks ke-2 dihapus:
physics, chemistry, 2000

F. Operasi Dasar Pada List Python

List Python merespons operator `+` dan `*` seperti string; Itu artinya penggabungan dan pengulangan di sini juga berlaku, kecuali hasilnya adalah list baru, bukan sebuah [String](#).

Sebenarnya, list merespons semua operasi urutan umum yang kami gunakan pada [String](#) di bab sebelumnya. Dibawah ini adalah tabel daftar operasi dasar pada list python

Python Expression	Hasil	Penjelasan
<code>len([1, 2, 3, 4])</code>	4	Length
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	Concatenation
<code>['Halo!'] * 4</code>	<code>['Halo!', 'Halo!', 'Halo!', 'Halo!']</code>	Repetition
<code>2 in [1, 2, 3]</code>	True	Membership
<code>for x in [1,2,3] : print (x,end = ' ')</code>	1 2 3	Iteration

Indexing, Slicing dan Matrix Pada List Python

Karena list adalah urutan, pengindeksan dan pengiris bekerja dengan cara yang sama untuk list seperti yang mereka lakukan untuk [String](#).

Dengan asumsi input berikut :

```
L = ['C++', 'Java', 'Python']
```

Python Expression	Hasil	Penjelasan
<code>L[2]</code>	<code>'Python'</code>	Offset mulai dari nol
<code>L[-2]</code>	<code>'Java'</code>	Negatif: hitung dari kanan
<code>L[1:]</code>	<code>['Java', 'Python']</code>	Slicing mengambil bagian

Method dan Fungsi Build-in Pada List Python

Python menyertakan fungsi built-in sebagai berikut

Python Function	Penjelasan
<code>cmp(list1, list2)</code>	# Tidak lagi tersedia dengan Python 3
<code>len(list)</code>	Memberikan total panjang list.
<code>max(list)</code>	Mengembalikan item dari list dengan nilai maks.
<code>min(list)</code>	Mengembalikan item dari list dengan nilai min.
<code>list(seq)</code>	Mengubah tuple menjadi list.

Python menyertakan methods built-in sebagai berikut

Python Methods	Penjelasan
<code>list.append(obj)</code>	Menambahkan objek obj ke list
<code>list.count(obj)</code>	Jumlah pengembalian berapa kali obj terjadi dalam list
<code>list.extend(seq)</code>	Tambahkan isi seq ke list
<code>list.index(obj)</code>	Mengembalikan indeks terendah dalam list

Python Methods	Penjelasan
	yang muncul obj
<code>list.insert(index, obj)</code>	Sisipkan objek obj ke dalam list di indeks offset
<code>list.pop(obj = list[-1])</code>	Menghapus dan mengembalikan objek atau obj terakhir dari list
<code>list.remove(obj)</code>	Removes object obj from list
<code>list.reverse()</code>	Membalik list objek di tempat
<code>list.sort([func])</code>	Urutkan objek list, gunakan compare func jika diberikan

1.2 Pengertian Tuple

A. Tuple

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

Membuat tuple semudah memasukkan nilai-nilai yang dipisahkan koma. Secara opsional, Anda dapat memasukkan nilai-nilai yang dipisahkan koma ini di antara tanda kurung juga.

```
tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5 );
tup3 = "a", "b", "c", "d";
```

B. Akses Nilai Dalam Tuple

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

```
tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5, 6, 7 );
print ("tup1[0]: ", tup1[0])
print ("tup2[1:5]: ", tup2[1:5])
```

Output:

```
tup1[0]: physics
tup2[1:5]: [2, 3, 4, 5]
```

C. Memperbarui Nilai Dalam Tuple

Tupel 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');
# Sintaks seperti dibawah tidak bisa diterapkan di Tuple
# tup1[0] = 100;
# Tuple hanya dapat mengambil isi dimasukkan ke Tuple baru
tup3 = tup1 + tup2;
print (tup3)
```

Output:

```
12, 34.56, 'abc', 'xyz'
```

D. Menghapus Nilai Dalam Tuple

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

dibaung.

Untuk secara eksplisit menghapus keseluruhan tuple, cukup gunakan del statement.

```
tup = ('physics', 'chemistry', 1997, 2000);
print (tup)
del tup;
print "Setelah tup dihapus: "
print (tup)
```

Output:

```
physics, chemistry, 1997, 2000
Setelah tup dihapus:
Traceback (most recent call last):
  File "test.py", line 9, in <module>
    print tup;
NameError: name 'tup' is not defined
```

E. Operasi Sederhana Pada Tuple

Tupel merespons operator + dan * sama seperti String; Mereka berarti penggabungan dan pengulangan di sini juga berlaku, kecuali hasilnya adalah tupel baru, bukan string.

Sebenarnya, Tuple merespons semua operasi urutan umum yang kami gunakan pada [String](#) di bab sebelumnya. Dibawah ini adalah tabel daftar operasi dasar pada Tuple python

Python Expression	Hasil	Penjelasan
<code>len((1, 2, 3))</code>	3	Length
<code>(1, 2, 3) + (4, 5, 6)</code>	<code>(1, 2, 3, 4, 5, 6)</code>	Concatenation
<code>('Halo!',) * 4</code>	<code>('Halo!', 'Halo!', 'Halo!', 'Halo!')</code>	Repetition
<code>3 in (1, 2, 3)</code>	True	Membership
<code>for x in (1,2,3) : print (x, end = ' ')</code>	1 2 3	Iteration

Indexing, Slicing dan Matrix Pada Tuple Python

Karena tupel adalah urutan, pengindeksan dan pengiris bekerja dengan cara yang sama untuk tupel seperti pada String, dengan asumsi masukan berikut
T = ('C++', 'Java', 'Python')

Python Expression	Hasil	Penjelasan
<code>T[2]</code>	<code>'Python'</code>	Offset mulai dari nol
<code>T[-2]</code>	<code>'Java'</code>	Negatif: hitung dari kanan
<code>T[1:]</code>	<code>('Java', 'Python')</code>	Slicing mengambil bagian

Fungsi Build-in Pada Tuple Python

Python menyertakan fungsi built-in sebagai berikut

Python Function	Penjelasan
<code>cmp(tuple1, tuple2)</code>	# Tidak lagi tersedia dengan Python 3
<code>len(tuple)</code>	Memberikan total panjang tuple.
<code>max(tuple)</code>	Mengembalikan item dari tuple dengan nilai maks.
<code>min(tuple)</code>	Mengembalikan item dari tuple dengan nilai min.
<code>tuple(seq)</code>	Mengubah tuple menjadi tuple.

2. Praktikum Latihan

3. Tugas Praktikum

1. Buatlah program List pada python (dinamis) dengan ada pertanyaan ingin mengulang iya atau tidak dan dengan Inputan:
List 1=[15 menu makanan]
List 2=[5 menu minuman]
List 3=[5 nama teman terdekat anda]
List 4=[5 bulan lahir teman tersebut]
List 5=[5 tanggal lahir teman tersebut]
Dengan Hasil:
Output 1= gabungkan list1 dan list 4 (list 1 hanya menampilkan 5 menu makanan)
Output 2= update list3 index 3 dan index 5
Output 3= hapus list5 index 1 dan index 5
Output 4= gabungkan list4, dan list5 dan tentukan nilai max dan min

2. Buatlah program Tuple pada python (dinamis) dengan inputan:

Tuple1=(5 nama kota)

Tuple2=(12,20,23,31,19,30,45,98)

Tuple3=(pEmograman, pemograman, Pemograman, PemOgRamAn, PEMOGRAMAN)

Dengan Hasil:

Output 1=gabungkan Tuple3 dan Tuple1 (Tuple3=index 1 dan index 3,4, Tuple1=index 3 dan index 1)

Output 2=tampilkan Tuple2 index 2 dan index 4 sebanyak 5 kali