isspace()	Mengembalikan true jika string hanya berisi karakter spasi dan false sebaliknya.
istitle()	Mengembalikan true jika string benar "titlecased" dan false sebaliknya.
isupper()	Mengembalikan true jika string memiliki setidaknya satu karakter casing dan semua karakter casing ada dalam huruf besar dan false sebaliknya.
join(seq)	Merges (concatenates) representasi string elemen dalam urutan seq menjadi string, dengan string pemisah.
len(string)	Mengembalikan panjang string
ljust(width[, fillchar])	Mengembalikan string berlapis ruang dengan string asli dibiarkan dibenarkan ke kolom lebar total.
lower()	Mengonversi semua huruf besar dalam bentuk string menjadi huruf kecil.
lstrip()	Menghapus semua spasi utama dalam string.
maketrans()	Mengembalikan tabel terjemahan untuk digunakan dalam fungsi terjemahan.
max(str)	Mengembalikan karakter alfabetik dari string str.
min(str)	Mengembalikan min karakter abjad dari string str.
replace(old, new [, max])	Menggantikan semua kemunculan lama dalam string dengan kejadian baru atau paling maksimal jika max diberikan.
rfind(str, beg = 0,end = len(string))	Sama seperti find (), tapi cari mundur dalam string.
rindex(str, beg = 0, end = len(string))	Sama seperti index (), tapi cari mundur dalam string.
rjust(width,[, fillchar])	Mengembalikan string berlapis ruang dengan senar asli benar-dibenarkan untuk total kolom lebar.
rstrip()	Menghapus semua spasi spasi string.

split(str="", num=string.count(str))	Membagi string sesuai dengan pemisah str (ruang jika tidak disediakan) dan mengembalikan daftar substring; Terpecah menjadi paling banyak substring jika diberikan.
Metode	**Penjelasan**
splitlines(num=string.count('\n'))	Membagi string sama sekali (atau num) NEWLINEs dan mengembalikan daftar setiap baris dengan NEWLINEs dihapus.
startswith(str, beg=0,end=len(string)	Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise.
strip([chars])	Lakukan kedua lstrip () dan rstrip () pada string
swapcase()	Kasus invers untuk semua huruf dalam string.
title()	Mengembalikan versi string "titlecased", yaitu, semua kata diawali dengan huruf besar dan sisanya huruf kecil.
translate(table, deletechars="")	Menerjemahkan string sesuai dengan tabel terjemahan str (256 karakter), menghapus string del.
upper()	Mengonversi huruf kecil dalam bentuk string ke huruf besar.
zfill (width)	Mengembalikan string asli yang tertinggal dengan angka nol ke total karakter lebar; Dimaksudkan untuk angka, zfill () mempertahankan tanda apapun yang diberikan (kurang satu nol).
isdecimal()	Mengembalikan nilai true jika string unicode hanya berisi karakter desimal dan false sebaliknya.

5.15 List

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.

```
#Contoh sederhana pembuatan list pada bahasa pemrograman python
list1 = ['kimia', 'fisika', 1993, 2017]
list2 = [1, 2, 3, 4, 5 ] list3 =
["a", "b", "c", "d"]
```

Akses Nilai Dalam List

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

Berikut adalah contoh cara mengakses nilai di dalam list python :

```
#Cara mengakses nilai di dalam list Python
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])
```

Setelah Anda mengeksekusi kode diatas, hasilnya akan seperti dibawah ini :

list1[0]: fisika

list2[1:5]: [2, 3, 4, 5]

Update Nilai Dalam List

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 (). Sebagai contoh :

```
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

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 :

```
#Contoh cara menghapus nilai pada list python list = ['fisika',
'kimia', 1993, 2017]
print (list) del list[2]
print ("Setelah dihapus nilai pada index 2 : ", list)
```

Operasi Dasar

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
len([1, 2, 3, 4])	**4**	Length
[1, 2, 3] + [4, 5, 6]	**[1, 2, 3, 4, 5, 6]**	Concatenation
['Halo!'] * 4	**['Halo!', 'Halo!', 'Halo!', 'Halo!']**	Repetition
2 in [1, 2, 3]	**True**	Membership
for x in [1,2,3] : print (x,end = ' ')	**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
L[2]	**'Python'**	Offset mulai dari nol
L[-2]	**'Java'**	Negatif: hitung dari kanan
L[1:]	**['Java', 'Python']**	Slicing mengambil bagian

Method dan Fungsi Build-in pada List Python Python menyertakan fungsi built-in sebagai berikut

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

Python menyertakan methods built-in sebagai berikut

Python Method	Penjelasan
list.append(obj)	Menambahkan objek obj ke list
list.count(obj)	Jumlah pengembalian berapa kali obj terjadi dalam list
list.extend(seq)	Tambahkan isi seq ke list
list.index(obj)	Mengembalikan indeks terendah dalam list yang muncul obj

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

5.16 Tuple

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. 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. Sebagai contoh :

```
#Contoh sederhana pembuatan tuple pada bahasa pemrograman python
tup1 = ('fisika', 'kimia', 1993, 2017)
tup2 = (1, 2, 3, 4, 5 ) tup3
= "a", "b", "c", "d"
```

Tupel kosong ditulis sebagai dua tanda kurung yang tidak berisi apa-apa, contohnya :

tup1 = ();

Untuk menulis tupel yang berisi satu nilai, Anda harus memasukkan koma, meskipun hanya ada satu nilai, contohnya :**tup1 = (50,)**

Seperti indeks String, indeks tuple mulai dari 0, dan mereka dapat diiris, digabungkan, dan seterusnya.

Akses Nilai Dalam Tuple

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 :

```
#Cara mengakses nilai tuple 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])
```

Setelah Anda mengeksekusi kode diatas, hasilnya akan seperti dibawah ini :

tup1[0]: fisika

tup2[1:5]: (2, 3, 4, 5)

Update 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')
# Aksi seperti dibawah ini tidak bisa dilakukan pada tuple python
# Karena memang nilai pada tuple python tidak bisa diubah
# tup1[0] = 100;
# Jadi, buatlah tuple baru sebagai berikut
tup3 = tup1 + tup2
print (tup3)
```

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

Sebagai contoh

```
tup = ('fisika', 'kimia', 1993, 2017) ;
print (tup) del tup;
print "Setelah menghapus tuple : "
print tup
```

Operasi Dasar Pada List 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, list merespons semua operasi urutan umum yang kami gunakan pada di bab sebelumnya. Dibawah ini adalah tabel daftar operasi dasar pada list python

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

Indexing, Slicing dan Matrix

Karena tupel adalah urutan, pengindeksan dan pengiris bekerja dengan cara yang sama untuk tupel seperti pada String, dengan asumsi masukan berikut Dengan asumsi input berikut :

T = ('C++', 'Java', 'Python')

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

Fungsi Build-in

Python menyertakan fungsi built-in sebagai berikut

Python Function	Penjelasan
cmp(tuple1, tuple2)	# Tidak lagi tersedia dengan Python 3
len(tuple)	Memberikan total panjang tuple.
max(tuple)	Mengembalikan item dari tuple dengan nilai maks.

min(tuple)	Mengembalikan item dari tuple dengan nilai min.
Python Function	**Penjelasan**
tuple(seq)	Mengubah tuple menjadi tuple.

5.17 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: {}. Nilai kamus bisa berupa tipe apa pun, namun key harus berupa tipe data yang tidak berubah seperti string, angka, atau tupel.

Akses Nilai

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

```
#Contoh cara membuat Dictionary pada Python dict =
{'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print ("dict['Name']: ", dict['Name']) print
("dict['Age']: ", dict['Age'])
```

Update Nilai

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.

```
#Update dictionary python
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 Nilai

Anda dapat menghapus elemen Dictionary individual atau menghapus keseluruhan isi Dictionary. Anda juga dapat menghapus seluruh Dictionary dalam satu operasi. Untuk menghapus seluruh Dictionary secara eksplisit, cukup gunakan **del** statement. Berikut adalah contoh sederhana :

```
#Contoh cara menghapus pada Dictionary Python dict = {'Name':
'Zara', 'Age': 7, 'Class': 'First'} 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['School']: ", dict['School'])
```

5.18 Tanggal dan Jam

Program Python yang dapat menangani tanggal dan waktu dalam beberapa cara. Mengkonversi antara format tanggal adalah tugas umum untuk komputer. Modul Python's waktu dan kalender membantu melacak tanggal dan waktu. Interval waktu adalah angka floating-point dalam satuan detik. Instants tertentu dalam waktu dinyatakan dalam satu detik sejak 12:00 am, 1 Januari 1970(epoch). Ini adalah waktu yang populer modul yang tersedia di Python yang menyediakan fungsi untuk bekerja dengan waktu, dan untuk mengkonversi antara pernyataan. Fungsi time.time() mengembalikan sistem saat ini waktu sejak 12:00 am, 1 Januari 1970.

```
import time;  # harus menginclude modul time
ticks = time.time()
print "Number of ticks since 12:00am, January 1, 1970:", ticks
```

Bila kode diatas dieksekusi, maka akan menghasilkan hasil sebagai berikut :

Number of ticks since 12:00am, January 1, 1970: 7186862.73399

TimeTupple

Index	Penjelasan	Hasil
0	4-digit year	2008
1	Month	1 to 12
2	Day	1 to 31
3	Hour	0 to 23
4	Minute	0 to 59
5	Second	0 to 61 (60 or 61 are leap-seconds)
6	Day of Week	0 to 6 (0 is Monday)
7	Day of year	1 to 366 (Julian day)
8	Daylight savings	-1 , 0, 1, -1 means library determines DST

5.19 Fungsi

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action. Fungsi memberikan modularitas yang lebih baik untuk aplikasi Anda dan tingkat penggunaan kode yang tinggi.

Mendefinisikan Fungsi Python

Anda dapat menentukan fungsi untuk menyediakan fungsionalitas yang dibutuhkan. Berikut adalah aturan sederhana untuk mendefinisikan fungsi dengan Python.

- Fungsi blok dimulai dengan def kata kunci diikuti oleh nama fungsi dan tanda kurung (()).
- Setiap parameter masukan atau argumen harus ditempatkan di dalam tanda kurung ini. Anda juga dapat menentukan parameter di dalam tanda kurung ini.
- Pernyataan pertama dari sebuah fungsi dapat berupa pernyataan opsional string dokumentasi fungsi atau docstring.
- Blok kode dalam setiap fungsi dimulai dengan titik dua (:) dan indentasi.
- Pernyataan kembali [ekspresi] keluar dari sebuah fungsi, secara opsional menyampaikan kembali ekspresi ke pemanggil. Pernyataan pengembalian tanpa argumen sama dengan return None.

Contoh fungsi

```
def printme( str ):
   "This prints a passed string into this function"    print
(str)    return
```

5.20 Modul

Modul memungkinkan Anda mengatur kode Python secara logis. Mengelompokkan kode terkait ke dalam modul membuat kode lebih mudah dipahami dan digunakan. Modul adalah objek Python dengan atribut yang diberi nama yang bisa Anda bind dan dijadikan referensi. Secara sederhana modul adalah file yang terdiri dari kode Python. Modul dapat mendefinisikan fungsi, kelas dan variabel. Modul juga bisa menyertakan kode yang bisa dijalankan "runable".

Berikut adalah contoh modul sederhana pada Python :

```
def print_func( par ):
   print "Halo : ", par     return
```

Import Statement

Anda dapat menggunakan file sumber Python apapun sebagai modul dengan mengeksekusi pernyataan impor di file sumber Python lainnya. Impornya memiliki sintaks berikut. Ketika interpreter menemukan sebuah pernyataan import, ia mengimpor modul jika modul tersebut ada di jalur pencarian. Jalur pencarian adalah daftar direktori yang ditafsirkan juru bahasa sebelum mengimpor modul. Misalnya, untuk mengimpor modul hello.py, Anda perlu meletakkan perintah berikut di bagian atas script.

```
# Import module support import support
# Anda bisa memanggil fungsi defined sebagai
berikut support.print_func("Bara")
```

5.21 Membaca Input Keyboard

Python 2 memiliki dua fungsi built-in untuk membaca data dari input standar, yang secara default berasal dari keyboard. Fungsi ini adalah **input()** dan **raw_input()** Dengan Python 3, fungsi **raw_input()** tidak digunakan lagi. Selain itu,**input()** berfungsi membaca data dari keyboard sebagai string, terlepas dari apakah itu tertutup dengan tanda kutip (" atau "') atau tidak.

Input Python

Fungsi input([prompt]) setara dengan raw_input, kecuali mengasumsikan bahwa input adalah ekspresi Python yang valid dan mengembalikan hasil yang dievaluasi ke Anda.

```
>>>  x = input("sesuatu : " )
sesuatu : 20
>>>  x
20
>>>  x = input("sesuatu : " )
sesuatu : '20'
>>>  x
'20'
```

5.22 Exception

Python menyediakan dua fitur yang sangat penting untuk menangani kesalahan tak terduga dalam program Python Anda dan menambahkan kemampuan debugging di dalamnya.

- **Exception Handling**
- **Assertions**

Exception adalah sebuah peristiwa, yang terjadi selama pelaksanaan program yang mengganggu aliran normal instruksi program. Secara umum, ketika skrip Python menemukan situasi yang tidak dapat diatasi, hal itu menimbulkan pengecualian. Exception adalah objek Python yang mewakili kesalahan. Ketika skrip Python menimbulkan Exception, ia harus menangani Exception begitu saja sehingga berhenti dan berhenti.

Standard Exceptions

Nama	Penjelasan
Exception	Kelas dasar untuk semua pengecualian / exception
StopIteration	Dibesarkan ketika metode (iterator) berikutnya dari iterator tidak mengarah ke objek apa pun.
SystemExit	Dibesarkan oleh fungsi sys.exit ().
StandardError	Kelas dasar untuk semua pengecualian built-in kecuali StopIteration dan SystemExit.
ArithmeticError	Kelas dasar untuk semua kesalahan yang terjadi untuk perhitungan numerik.
OverflowError	Dibesarkan saat perhitungan melebihi batas maksimum untuk tipe numerik.
FloatingPointError	Dibesarkan saat perhitungan floating point gagal.
ZeroDivisonError	Dibesarkan saat pembagian atau modulo nol dilakukan untuk semua tipe numerik.
AssertionError	Dibesarkan jika terjadi kegagalan pernyataan Assert.
AttributeError	Dibesarkan jika terjadi kegagalan referensi atribut atau penugasan.
EOFError	Dibesarkan bila tidak ada input dari fungsi raw_input () atau input () dan akhir file tercapai.
ImportError	Dibesarkan saat sebuah pernyataan impor gagal.
KeyboardInterrupt	Dibesarkan saat pengguna menyela eksekusi program, biasanya dengan menekan Ctrl + c.
LookupError	Kelas dasar untuk semua kesalahan pencarian.

IndexError	Dibesarkan saat sebuah indeks tidak ditemukan secara berurutan.
KeyError	Dibesarkan saat kunci yang ditentukan tidak ditemukan dalam kamus.
NameError	Dibesarkan saat pengenal tidak ditemukan di namespace lokal atau global.
UnboundLocalError	Dibesarkan saat mencoba mengakses variabel lokal dalam suatu fungsi atau metode namun tidak ada nilai yang ditugaskan padanya.
EnvironmentError	Kelas dasar untuk semua pengecualian yang terjadi di luar lingkungan Python.
IOError	Dibesarkan saat operasi input / output gagal, seperti pernyataan cetak atau fungsi open () saat mencoba membuka file yang tidak ada.
OSError	Dibangkitkan untuk kesalahan terkait sistem operasi.
SyntaxError	Dibesarkan saat ada kesalahan dengan sintaks Python.
IndentationError	Dibesarkan saat indentasi tidak ditentukan dengan benar.
Nama	**Penjelasan**
SystemError	Dibesarkan saat penafsir menemukan masalah internal, namun bila kesalahan ini ditemui juru bahasa Python tidak keluar.
SystemExit	Dibesarkan saat juru bahasa Python berhenti dengan menggunakan fungsi sys.exit (). Jika tidak ditangani dalam kode, menyebabkan penafsir untuk keluar.
TypeError	Dibesarkan saat operasi atau fungsi dicoba yang tidak valid untuk tipe data yang ditentukan.
ValueError	Dibesarkan ketika fungsi bawaan untuk tipe data memiliki jenis argumen yang valid, namun argumen tersebut memiliki nilai yang tidak valid yang ditentukan.
RuntimeError	Dibesarkan saat kesalahan yang dihasilkan tidak termasuk dalam kategori apa pun.

NotImplementedError	Dibesarkan ketika metode abstrak yang perlu diimplementasikan di kelas warisan sebenarnya tidak dilaksanakan.