pertemuan 2

roadmap pembelajaran

jenis type data

1. numerik
2. string
3. list
4. tuple
5. dictionary

Jenis-Jenis Tipe Data Python

Ada 3 jenis type data yang sering dipakai sebagai awal belajar pemrograman python, yaitu:

1. Tipe data numerik (angka)
2. Tipe data string (teks)
3. Tipe data boolean (logik)

1. Tipe Data Numerik

1. int (integer) untuk menyatakan bilangan bulat, contoh: 1, 2, 4, 20, 40, 76, dll.
2. float untuk menyatakan bilangan pecahan, contoh: 1.5, 40.25, 80.3, dll.

2. Tipe Data String

1. char (character) untuk menyatakan 1 karakter, contoh: 'A'
2. string kumpulan karakter, contoh: 'Saya Belajar Python'

3. Tipe Data Boolean

Tipe data boolean digunakan untuk menentukan logika, karena tipe data ini hanya memiliki 2 kondisi yaitu True (Benar) dan False (salah)

macam2 tipe data

Tipe Data	Contoh	Penjelasan
Boolean	`True` atau `False`	Menyatakan benar `True` yang bernilai `1`, atau salah `False` yang bernilai `0`
String	`"Ayo belajar Python"`	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda `"` atau `'`)
Integer	`25` atau `1209`	Menyatakan bilangan bulat
Float	`3.14` atau `0.99`	Menyatakan bilangan yang mempunyai koma
Hexadecimal	`9a` atau `1d3`	Menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	`1 + 5j`	Menyatakan pasangan angka real dan imajiner
List	`['xyz', 786, 2.23]`	Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	`('xyz', 768, 2.23)`	Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	`{'nama': 'adi','id':2}`	Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

```
#tipe data Boolean
print(True)


#tipe data String
print(‘Ayo belajar Python‘)
print(’Belajar Python Sangat Mudah’)
```

```
#tipe data Integer
print(20)
```

```
#tipe data Float
print(3.14)
```

```
#tipe data Hexadecimal
nilaiBiasa = 23
nilaiHex = hex(nilaiBiasa)
print(nilaiHex)
print(type(nilaiHex))
```

```
#tipe data Complex
print(5j)
```

```
#tipe data List
print([1,2,3,4,5])
print([‘satu‘, ‘dua‘, ‘tiga‘])
```

```
#tipe data Tuple
print((1,2,3,4,5))
print((‘satu‘, ‘dua‘, ‘tiga‘))
```

```
#tipe data Dictionary
print({‘nama‘:‘Budi‘, ’umur’:20})

#tipe data Dictionary dimasukan ke dalam variabel biodata
biodata = {‘nama‘:‘Andi‘, ’umur’:21} #proses inisialisasi variabel biodata
print(biodata) #proses pencetakan variabel biodata yang berisi tipe data Dictionary
print(type(biodata)) #fungsi untuk mengecek jenis tipe data. akan tampil <class ’dict’>

yang berarti dict adalah tipe data dictionary
```

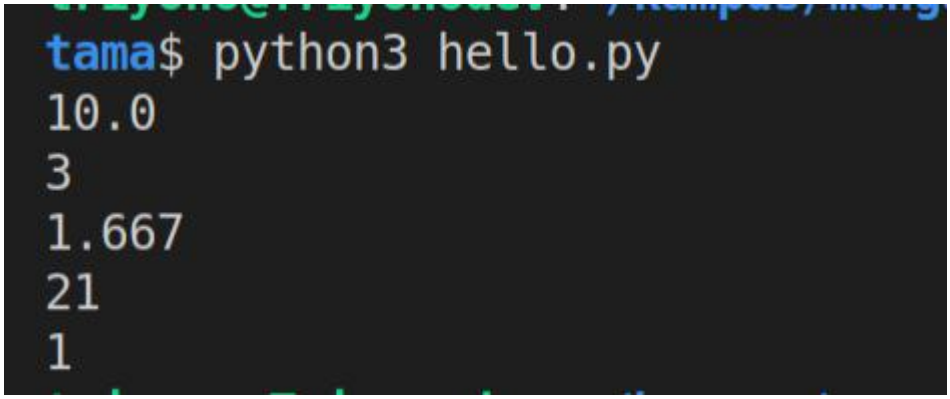
numerik

```
angkaInt = 10
print(float(angkaInt))

angkaFloat = 3.7
print(int(angkaFloat))
```

```
angkaDesimal = 1.66653456
print(round(angkaDesimal,3))
```

```
data = [1,5,3,21,4,5,6,7,7,8,]
maxdata =max(data)
mindata = min(data)
print(maxdata)
print(mindata)
```



string

mengakses nilai string

```
name = 'John Doe'
message = "John Doe belajar bahasa python di Dutabangsa"
print ("name[0]: ", name[0])
print ("message[1:4]: ", message[1:4])
```

mengupdate nilai string

```
message = 'Hello World'
print(message)
print ("Updated String :- ", message[:6] + 'Python')
```

operator format string

```
print ('My name is %s and weight is %d kg!' % ('Zara', 21))
```

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

triple quote

```
kutipantiga = """this is a long string that is made up of
several lines and non-printable characters such as
TAB ( \t ) and they will show up that way when displayed.
NEWLINEs within the string, whether explicitly given like
this within the brackets [ \n ], or just a NEWLINE within
the variable assignment will also show up.
"""print (kutipantiga)
```

```
this is a long string that is made up of
several lines and non-printable characters such as
TAB (    ) and they will show up that way when displayed.
NEWLINEs within the string, whether explicitly given like
this within the brackets [
 ], or just a NEWLINE within
the variable assignment will also show up.
```

LIst

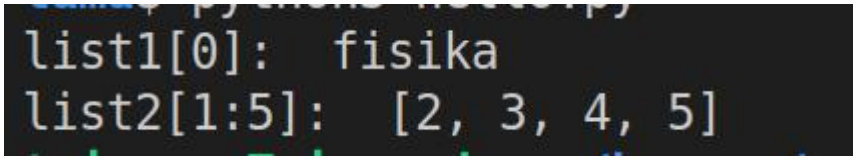
```
#Contoh sederhana pembuatan list pada bahasa pemrograman python
list1 = ['kimia', 'fisika', 1993, 2017]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]
```

akses nilai pada list

```
#Cara mengakses nilai di dalam list Python

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

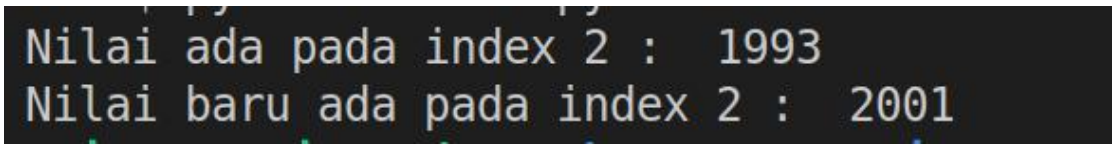
```
print ('list1[0]: ', list1[0])
print ('list2[1:5]: ', list2[1:5])
```



update nilai pada list

```
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 pada list

```
#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)
```

```
['fisika', 'kimia', 1993, 2017]
Setelah dihapus nilai pada index 2 :  ['fisika', 'kimia', 2017]
```

operasi dasar pada list

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

fungsi dan method build in pada list

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.
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 Methods	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

tuple

Sebuah tupel adalah urutan objek Python yang tidak berubah,Tupel menggunakan tanda kurung,

```
#Contoh sederhana pembuatan tuple pada bahasa pemrograman python

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

akses nilai pada tuple

```
#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])
```

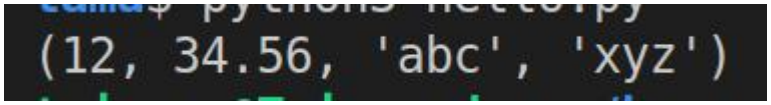
```
tup1[0]:  fisika
tup2[1:5]:  (2, 3, 4, 5)
```

update nilai pada tuple

```
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 tuple

Menghapus elemen tuple individual tidak mungkin dilakukan

```
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)
```

operasi dasar tuple

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

fungsi build in tuple

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.
`tuple(seq)`	Mengubah seq menjadi tuple.

dictionary

akses nilai dictionary

```
#Contoh cara membuat Dictionary pada Python

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

```
dict['Name']:  Zara
dict['Age']:  7
```

update nilai dictionary

```
#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'])
```

```
print(dict)
```

```
dict['Age']:  8
dict['School']:  DPS School
{'Name': 'Zara', 'Age': 8, 'Class': 'First', 'School': 'DPS School'}
```

hapus nilai dictionary

```
#Contoh cara menghapus pada Dictionary Python

del dict['Name'] # hapus entri dengan key 'Name'
dict.clear() # hapus semua entri di dict
del dict # hapus dictionary yang sudah ada
```

fungsi build in

Fungsi Python	Penjelasan
cmp(dict1, dict2)	Membandingkan unsur keduanya.
len(dict)	Memberikan panjang total Dictionary. Ini sama dengan jumlah item dalam Dictionary.
str(dict)	Menghasilkan representasi string yang dapat dicetak dari Dictionary
type(variable)	Mengembalikan tipe variabel yang lulus. Jika variabel yang dilewatkan adalah Dictionary, maka akan mengembalikan tipe Dictionary.

methode build in

Method Python	Penjelasan
dict.clear()	Menghapus semua elemen Dictionary
dict.copy()	Mengembalikan salinan Dictionary
dict.fromkeys()	Buat Dictionary baru dengan kunci dari seq dan nilai yang disetel ke nilai.
dict.get(key, default=None)	For key, nilai pengembalian atau default jika tombol tidak ada dalam Dictionary
dict.has_key(key)	Mengembalikan true jika key dalam Dictionary, false sebaliknya
dict.items()	Mengembalikan daftar dari pasangan tuple dictionary (key, value)
dict.keys()	Mengembalikan daftar key dictionary
dict.setdefault(key, default=None)	Mirip dengan get (), tapi akan mengatur dict [key] = default jika kunci belum ada di dict
dict.update(dict2)	Menambahkan pasangan kunci kata kunci dict2 ke dict
dict.values()	Mengembalikan daftar nilai dictionary