

# **WEEK 3 – HOMEWORK 3**

## **INTRODUCTION TO PYTHON AND PROGRAMMING**



By  
Anugrah Yazid Ghani

**DATA SCIENCE - BATCH 11**

# TIPE DATA PADA PYTHON

Dasar dari mempelajari Bahasa Pemrograman yang baru adalah pemahaman terhadap tipe data. Berikut tipe-tipe data yaitu:

## A) Numbers

Tipe numerik pada Python dibagi menjadi 3:

- int (Integer)
- float (Float)
- complex (Complex)

Contoh script:

```
1. a = 10
2. print(a, "bertipe", type(a))
3. b = 1.7
4. print(b, "bertipe", type(b))
5. c = 1+3j
6. print(c, " Bertipe bilangan kompleks? ", isinstance(1+3j,complex))
```

Output:

```
10 bertipe <class 'int'>
1.7 bertipe <class 'float'>
(1+2j) Bertipe bilangan kompleks? True
```

Integer tidak dibatasi oleh angka atau panjang tertentu, namun dibatasi oleh memori yang tersedia. Sehingga tidak perlu menggunakan variabel yang menampung big number misalnya long long (C/C++), biginteger, atau sejenisnya. Contoh kode untuk menunjukkan bahwa Python tidak membatasi output integer adalah pencarian bilangan ke-10.000 pada deret fibonacci (catatan: bilangan ke-10.000 pada deret fibonacci memiliki panjang 2.090 digit) sebagai berikut:

```
1. x=[0]*10005;      #inisialisasi array 0 sebanyak 10005; x[0]=0
2. x[1]=1;          #x[1]=1
3.
```

```
4. for j in range(2, 10001):
5.     x[j]=x[j-1]+x[j-2] # Fibonacci
6. print(x[10000])
```

Output:

```
33644764876431783266621612005107543310302148460680063906564769974680081
44216666236815559551363373402558206533268083615937373479048386526826304
08924630564318873545443695598274916066020998841839338646527313000888302
69235673613135117579297437854413752130520504347701602264758318906527890
85515436615958298727968298751063120057542878345321551510387081829896979
16131278562650331954871402142875326981879620469360978799003509623022910
26368131493195275630227837628441540360584402572114334961180023091208287
04608892396232883546150577658327125254609359112820392528539343462090424
52489294039017062338889910858410651831733604374707379085526317643257339
93712871937587746897479926305837065742830161637408969178426378624212835
25811282051637029808933209990570792006436742620238978311147005407499845
92503606335609338838319233867830561364353518921332797329081337326426526
33989763922723407882928177953580570993691049175470808931841056146322338
21746563732124822638309210329770164805472624384237486241145309381220656
49140327510866433945175121615265453613331113140424368548051067658434935
23836959653428071768775328348234345557366719731392746273629108210679280
78471803532913117677892465908993863545932789452377767440619224033763867
40040213303432974969020283281459334188268176838930720036347956231171031
01291953169794607632737589253530772552375943788434504067715555779056450
44301664011946258097221672975861502696844314695203461493229110597067624
32685159928347098912847067408620085871350162603120719031720860940812983
21581077282076353186624611278245537208532365305775956430072517744315051
53960090516860322034916322264088524885243315805153484962243484829938090
50704834824493274537326245677558790891871908036620580095947431500524025
32709746995318770724376825907419939632265984147498193609285223945039707
16544315642132815768890805878318340491743455627052022356484649519611246
02683139709750693826487066132645076650746115126775227486215986425307112
98441182622661057163515069260029861704945425047491378115154139941550671
25627119713325276363193960690289565028826860836224108205056243070179497
6171121233066073310059947366875
```

### Batasan akurasi variabel bertipe float

Python melakukan pemotongan pada digit ke 16 pada variabel float. Float atau bilangan pecahan dibatasi akurasi pada 15 desimal. Yang membedakan Integer dan Float adalah titik (decimal points). Misalnya dalam penulisan angka 1 jenisnya Integer, tapi jika dituliskan sebagai 1.0 artinya berjenis Float atau pecahan.

```
1. b = 0.1234567890123456789
```

```
2. print(b)
```

Output:

```
0.12345678901234568
```

Contoh jika berupa integer:

```
1. a = 1234567890123456789
2. print(a)
```

Output:

```
1234567890123456789
```

Karena Python banyak digunakan juga oleh matematikawan, tipe bilangan di Python juga mendukung bilangan imajiner dan bilangan kompleks. Nilai bilangan kompleks (complex) dituliskan dalam formulasi  $x + yj$ , yakni bagian  $x$  adalah bilangan real dan  $y$  adalah bilangan imajiner. Contohnya adalah sebagai berikut:

```
1. c = 1+5j
2. print(c)
```

Output:

```
(1+5j)
```

## B) Strings

String adalah urutan dari karakter unicode yang dideklarasikan dengan petik tunggal atau ganda. String > 1 baris dapat ditandai dengan tiga petik tunggal atau ganda `'''` atau `"""`.

```
1. s = "Ini adalah string baris tunggal"
```

```
1. s = """Ini adalah string
2. yang memiliki baris pertama
3. dan selanjutnya baris kedua"""
```

## C) Bool/Boolean

Tipe data bool atau Boolean merupakan turunan dari bilangan bulat (integer atau int) yang hanya punya dua nilai konstanta: True dan False.

### Nilai Boolean

Nilai konstanta False dan True merepresentasikan nilai kebenaran (truth values), meskipun ada nilai-nilai lain yang juga dianggap benar atau salah. Di dalam konteks angka, misalnya digunakan sebagai argumen dari operator matematika aritmatika, kedua nilai ini berlaku seperti halnya bilangan bulat 0 dan 1, sesuai False dan True.

Ada fungsi bawaan bool() yang dapat mengubah nilai menjadi nilai Boolean, apabila nilai tersebut dapat direpresentasikan sebagai **nilai kebenaran (truth values)**.

**Nilai kebenaran** adalah sebuah nilai yang dapat diuji sebagai benar atau salah, untuk digunakan di sintaksis kondisi if atau while atau sebagai operan dari operasi Boolean.

Berikut adalah objek bawaan yang didefinisikan bernilai salah dalam pengujian nilai kebenaran:

- Konstanta yang sudah didefinisikan bernilai salah: None dan False.
- Angka nol dari semua tipe numeric: 0, 0.0, 0j, Decimal(0), Fraction(0, 1).
- Urutan (sequence) dan koleksi (collection) yang kosong: "", (), {}, set(), range(0).

Untuk objek yang didefinisikan sendiri, representasi nilai Boolean akan bergantung dari definisi metode (method) khusus bernama \_\_bool\_\_(self). Jika metode ini mengembalikan True maka interpretasi nilai dari objeknya akan True, demikian juga sebaliknya.

### Operasi Boolean

Operasi dan fungsi bawaan yang memiliki hasil Boolean akan selalu mengembalikan 0 atau False untuk yang bernilai salah, serta 1 atau True untuk yang bernilai benar, kecuali dinyatakan berbeda dalam dokumentasi.

Operasi untuk tipe Boolean akan dijelaskan lebih lanjut di modul Operator, Operands, dan Expressions.

## D) List

List adalah jenis kumpulan data terurut (ordered sequence), dan merupakan salah satu variabel yang sering digunakan pada Python. Serupa, namun tak sama dengan array pada bahasa pemrograman lainnya. Bedanya, elemen List pada Python tidak harus memiliki tipe data yang sama. Mendeklarasikan List cukup mudah dengan kurung siku dan elemen yang dipisahkan dengan koma.

**Setiap data di dalamnya dapat diakses dengan indeks yang dimulai dari 0.**

```
1. a = [1, 2.2, 'python']
```

Python mengenal slicing operator `[]` yang dapat melakukan ekstraksi sebuah item atau beberapa item yang berada dalam range tertentu pada tipe data urutan (sequences), misalnya list, string dan tuple. Beberapa tipe urutan juga mendukung "extended slicing" dengan parameter ketiga berupa "step".

- **`x[0]`** artinya mengambil elemen paling awal, dengan index 0 dari List x.
- **`x[5]`** artinya mengambil elemen dengan index 5 dari List x.
- **`x[-1]`** artinya mengambil elemen dengan index paling belakang ke-1 dari List x.
- **`x[3:5]`** artinya membuat list dari anggota elemen List x dengan index 3 hingga sebelum index 5 (tidak termasuk elemen dengan index 5, dalam hal ini hanya index 3-4).
- **`x[:5]`** artinya membuat list dari anggota elemen List x paling awal hingga sebelum index 5 (tidak termasuk elemen dengan index 5, dalam hal ini hanya index 0-4).
- **`x[-3:]`** artinya membuat list dari anggota elemen List x mulai index ke-3 dari belakang hingga paling belakang.
- **`x[1:7:2]`** artinya membuat list dari anggota elemen List x dengan index 1 hingga sebelum index 7, dengan "step" 2 (dalam hal ini hanya index 1, 3, 5).

```
1. x = [5,10,15,20,25,30,35,40]
2. print(x[5])
3. print(x[-1])
4. print(x[3:5])
5. print(x[:5])
6. print(x[-3:])
7. print(x[1:7:2])
```

Output:

```
30
40
[20, 25]
[5, 10, 15, 20, 25]
[30, 35, 40]
[10, 20, 30]
```

Elemen pada list dapat diubah atau ditambahkan. Misalnya untuk melakukan perubahan kemudian penambahan:

```
1. x = [1,2,3]
2. x[2]=4
3. print(x)
```

Output:

```
[1, 2, 4]
```

```
1. x = [1,2,3]
2. x[2]=4
3. x.append(5)
4. print(x)
```

Output:

```
[1, 2, 4, 5]
```

**Untuk menghapus item pada list, gunakan fungsi del.** Ingat bahwa Indeks Python dimulai dari 0:

```
1. binatang = ['kucing', 'rusa', 'badak', 'gajah']
2. del binatang[2]
3. print(binatang)
```

Output:

```
['kucing', 'rusa', 'gajah']
```

Coba tambahkan kembali:

1. `del binatang [2]`
2. `print(binatang)`

Pada baris terbawah kode di atas, maka output akan menjadi:

Output:

```
['kucing', 'rusa']
```

## Slicing pada String

Karena string mirip dengan list, maka slicing operator [ ] juga dapat digunakan pada string untuk mengambil isinya atau bahkan substring. Sebuah string utuh bersifat mutable (bisa diubah), namun elemennya bersifat immutable (tidak bisa diubah).

1. `s = "Hello World!"`
2. `print(s[4])`      #ambil karakter kelima dari string s
3. `print(s[6:11])`    #ambil karakter ketujuh hingga sebelas dari string s
4. `s[5]="d"`          #ubah karakter keenam dari string s menjadi "d", seharusnya  
                          #gagal karena immutable
5. `s = "Halo Dunia!"` #ubah isi string s menjadi "Halo Dunia!", seharusnya berhasil  
                          #karena mutable
6. `print(s)`

Output:

```
'o'
```

```
'World'
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: 'str' object does not support item assignment
```

```
'Halo Dunia!'
```



## E) Tuple

Tuple adalah jenis dari list yang tidak dapat diubah elemennya. Umumnya tuple digunakan untuk data yang bersifat sekali tulis, dan dapat dieksekusi lebih cepat. Tuple didefinisikan dengan kurung dan elemen yang dipisahkan dengan koma.

```
1. t = (5,'program', 1+3j)
```

Seperti list, kita dapat melakukan slicing, namun pada tuple kita tidak dapat melakukan perubahan:

```
1. t = (5,'program', 1+3j)
2. print(t[1])
3. print(t[0:3])
4. print(t[0]=10)
```

Output:

```
'program'
(5, 'program', (1+3j))
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

## F) Set

Set adalah kumpulan item bersifat unik dan tanpa urutan (unordered collection). Didefinisikan dengan kurawal dan elemennya dipisahkan dengan koma. Pada Set kita dapat melakukan union dan intersection, sekaligus otomatis melakukan penghapusan data duplikat.

```
1. a = {1,2,2,3,3,3}
2. print(a)
```

Output:

```
{1, 2, 3}
```

Karena set bersifat unordered, maka kita tidak bisa mengambil sebagian data / elemen datanya menggunakan proses slicing.

```
1. a = {1,2,3}
2. print(a[1])
```

Output:

```
Traceback (most recent call last):
File "<string>", line 301, in runcode
File "<interactive input>", line 1, in <module>
TypeError: 'set' object does not support indexing
```

## G) Dictionary

Dictionary pada Python adalah kumpulan pasangan kunci-nilai (pair of key-value) yang bersifat tidak berurutan. Dictionary dapat digunakan untuk menyimpan data kecil hingga besar. Untuk mengakses datanya, kita harus mengetahui kuncinya (key). Pada Python, dictionary didefinisikan dengan kurawal dan tambahan definisi berikut:

1. Setiap elemen pair key-value dipisahkan dengan koma (,).
2. Key dan Value dipisahkan dengan titik dua (:).
3. Key dan Value dapat berupa tipe variabel/obyek apapun.

```
1. d = {1:'value','key':2}
2. print(type(d))
```

Output:

```
<class 'dict'>
```

```
1. d = {1:'value','key':2}
2. print(type(d))
3. print("d[1] = ", d[1]);
4. print("d['key'] = ", d['key']);
```

Output:

```
<class 'dict'>
d[1] = value
d['key'] = 2
```

Dictionary bukan termasuk dalam implementasi urutan (sequences), sehingga tidak bisa dipanggil dengan urutan indeks. Misalnya dalam contoh berikut dicoba dengan indeks 2, tetapi menghasilkan error (KeyError) karena tidak ada kunci (key) 2:

```
1. d = {1:'value','key':2}
2. print(type(d))
3. print("d[1] = ", d[1]);
4. print("d['key'] = ", d['key']);
5.
6. # Generates error
7. print("d[2] = ", d[2]);
```

Output:

```
<class 'dict'>
d[1] = value
d['key'] = 2

-----

KeyError Traceback (most recent call last)
<ipython-input-7-4b566e677ca2> in <module>()
1 d = {1:'value','key':2}
----> 2 print("d[2] = ", d[2]);

KeyError: 2
```

## H) Konversi (conversion, cast) antar tipe data

Kita dapat melakukan konversi tipe data bawaan dengan menggunakan fungsi konversi tipe bawaan (standard type) misalnya: int(), float(), str(), dll.

```
1. print(float(5))
```

Output:

```
5.0
```

Konversi float ke int akan bersifat floor/truncating atau menghilangkan nilai di belakang koma.

```
1. print(int(10.6))
```

Output:

```
10
```

```
1. print(int(-10.6))
```

Output:

```
-10
```

Konversi dari-dan-ke string akan melalui pengujian dan dipastikan validitasnya.

```
1. print(float('2.5'))
```

Output:

```
2.5
```

```
1. print(str(25))
```

Output:

```
'25'
```

```
1. print(int('1p'))
```

Output:

```
Traceback (most recent call last):  
File "<string>", line 301, in runcode  
File "<interactive input>", line 1, in <module>  
ValueError: invalid literal for int() with base 10: '1p'
```

Anda juga dapat melakukan konversi kumpulan data (set, list, tuple).

```
1. print(set([1,2,3]))
```

Output:

```
{1, 2, 3}
```

```
1. print(tuple({5,6,7}))
```

Output:

```
(5, 6, 7)
```

```
1. print(list('hello'))
```

Output:

```
['h', 'e', 'l', 'l', 'o']
```

Untuk konversi ke dictionary, data harus memenuhi persyaratan key-value. Berikut adalah dua contoh konversi:

List dari beberapa List yang isinya pasangan nilai menjadi Dictionary.

Serta konversi List dari beberapa Tuple yang isinya pasangan nilai menjadi Dictionary.

```
1. print(dict([[1,2],[3,4]]))
```

Output:

```
{1: 2, 3: 4}
```

```
1. print(dict([(3,26),(4,44)]))
```

Output:

```
{3: 26, 4: 44}
```