

AI Mastery Course

Module 1

Introduction to everything



Section

Python data types



Learning Objectives

Di akhir modul ini, kita akan dapat:

- Memahami python data types
 - Mengeksplorasi primitive data types di python
 - Mempelajari bagaimana mendefinisikan, memanipulasi dan menggunakan non-primitive data types
 - Memahami bagaimana menggunakan functions untuk type conversion
- 
- 

Agenda

01

DATA TYPES

- Data types di Python
- Primitive data types

02

SEQUENCES

- List
- Tuple
- String

03

MAPPING & SET

- Dictionary
- Set

04

CONVERSION

- Type conversion – primitive
- Type conversion – non-primitive

05

KESIMPULAN

- Kuis
- Ringkasan



01

Data Types

- Data types di Python
- Primitive data types

Data structures in python

- Cara untuk mengatur dan menyimpan data
- Tipe data variabel menentukan operasi mana yang dapat diterapkan padanya



Primitive data types

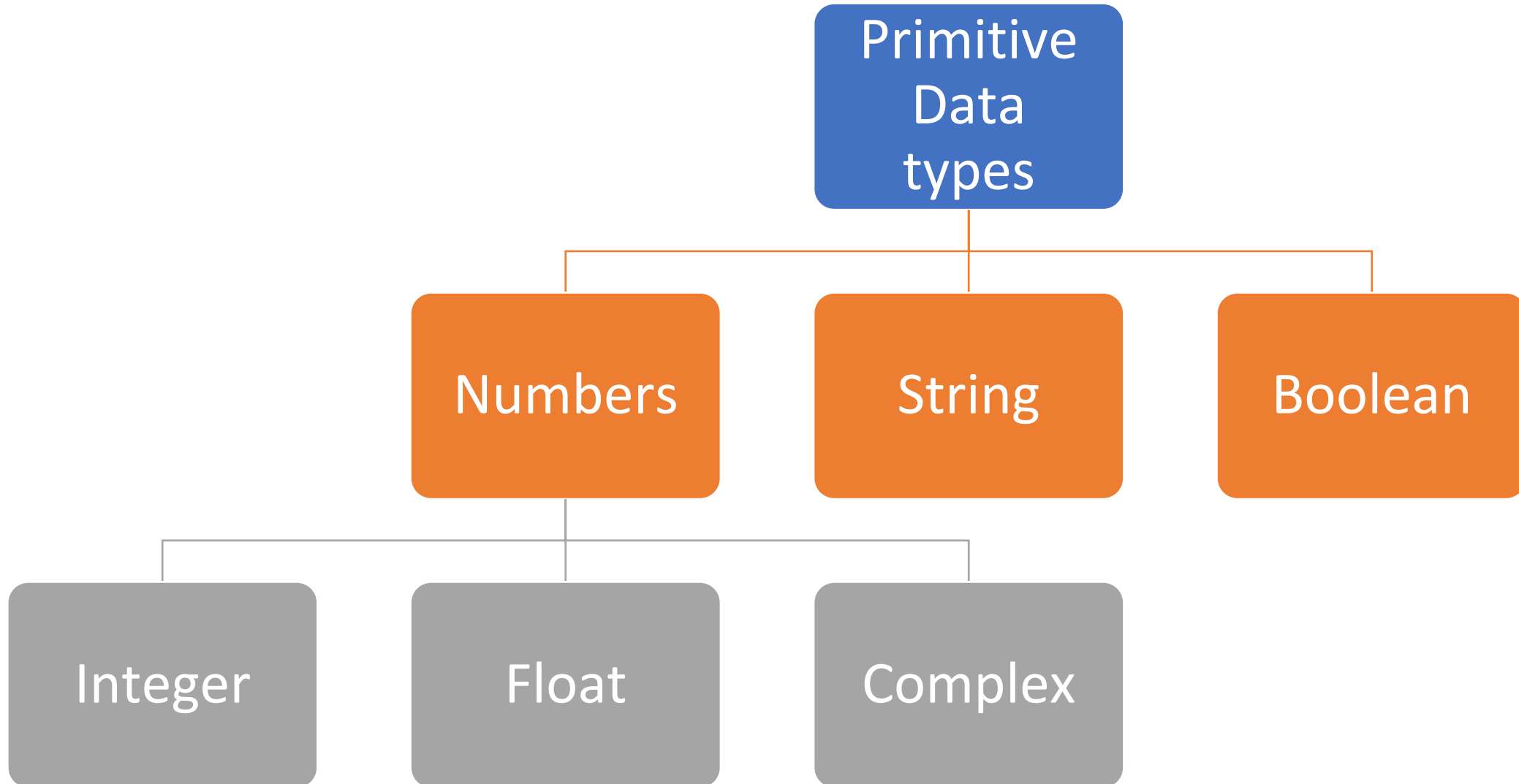
Struktur data dasar, blok bangunan untuk manipulasi data dan berisi nilai data yang murni dan sederhana.



Non primitive data types

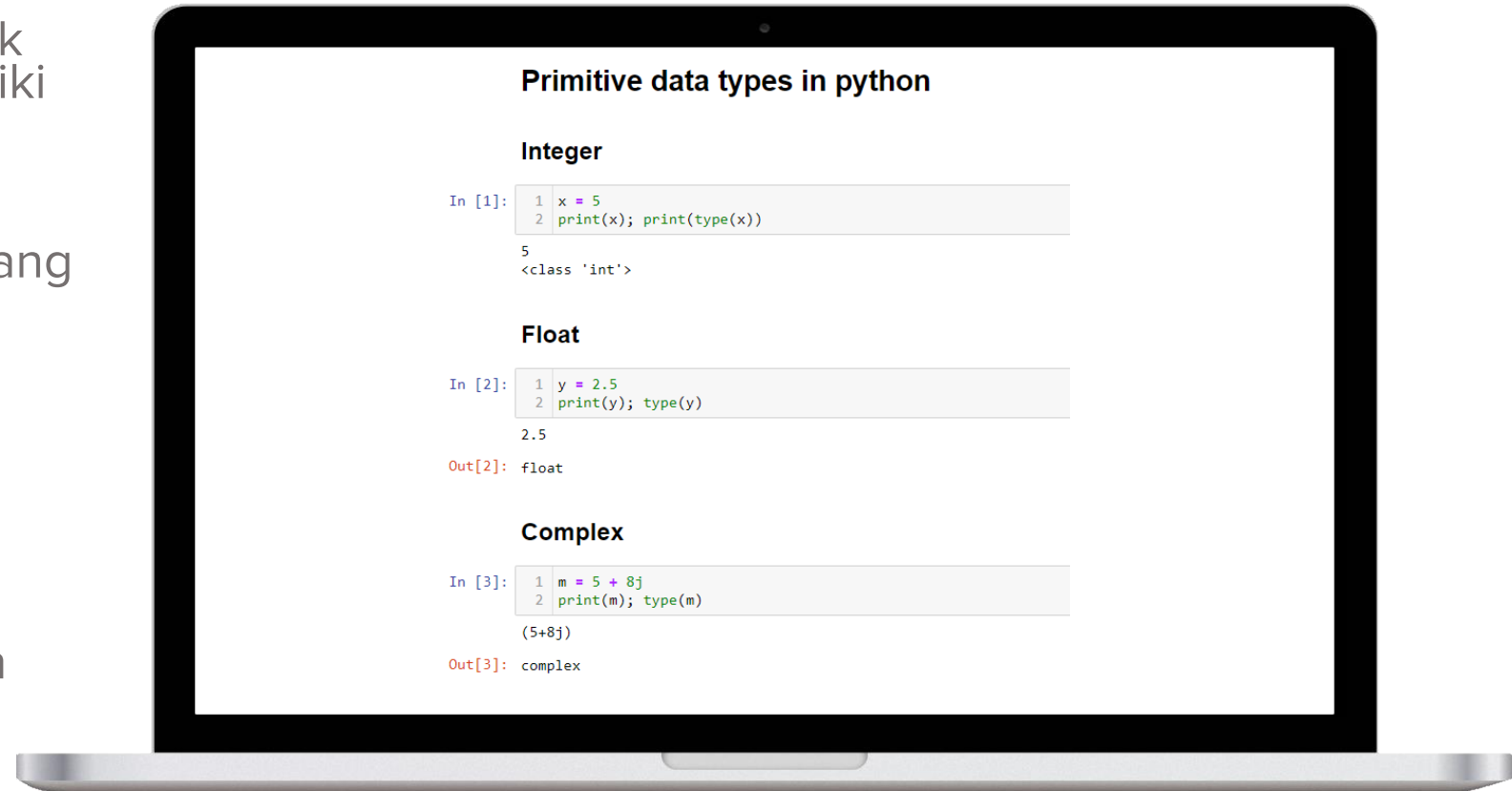
Berasal dari primitive data, digunakan untuk menyimpan kumpulan nilai.

Primitive data types di Python



Primitive data types

- `type()` fungsi dapat digunakan untuk mengetahui kelas mana yang dimiliki variabel atau nilai
- Bilangan bulat dapat memiliki panjang berapa pun, hanya dibatasi oleh memori yang tersedia
- Float in python akurat hingga 15 tempat desimal
- 5 adalah bilangan bulat, 5.0 adalah bilangan floating-point
- Bilangan kompleks ditulis dalam bentuk, $a + bj$, di mana a adalah bagian real dan b adalah bagian imajiner



Primitive data types

String

- Teks diapit dengan tanda kutip tunggal atau ganda

Boolean

- Berupa dua nilai Benar atau Salah

String

```
In [4]: 1 k = "Hello from Python"
        2 print(k); type(k)
```

Hello from Python

Out[4]: str

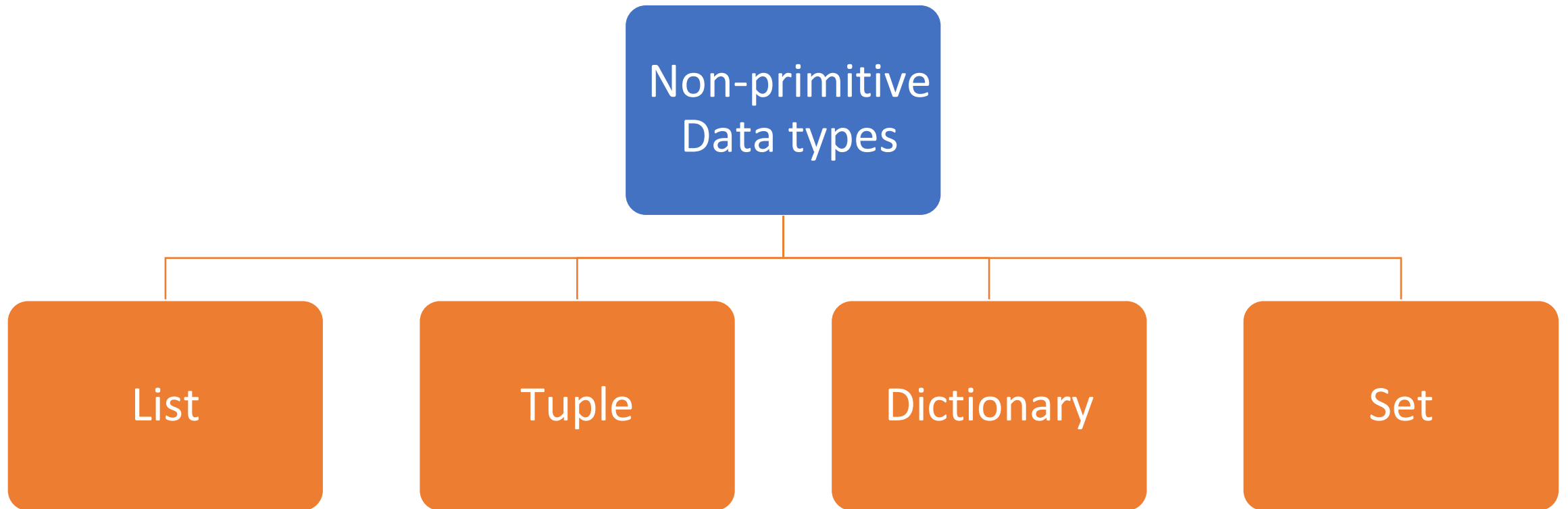
Boolean

```
In [5]: 1 m = True
        2 n = False
        3 print(m); type(m)
        4 print(n); type(n)
```

True
False

Out[5]: bool

Non-primitive data types di Python










02

SEQUENCES

- List
- Tuple
- String

List

- | | | | | | |
|---|--|---|--|---|--|
|  | <p>berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung siku ([]).</p> |  | <p>Mengakomodasi pengulangan elemen dan elemen dari tipe data yang berbeda</p> |  | <p>Tipe data yang bisa berubah – daftar dapat dimodifikasi</p> |
|  | <p>nilai yang disimpan dalam daftar dapat diakses menggunakan operator irisan ([] dan [:]) dengan indeks mulai dari 0 di awal daftar.</p> |  | <p>Mendukung pengindeksan negatif tanda plus (+) adalah operator rangkaian daftar, dan tanda asterisk (*) adalah operator pengulangan.</p> | | |

List

Pengindeksan dimulai dengan 0 (kiri ke kanan). Python juga mendukung pengindeksan negatif (kanan ke kiri).

```
1  mylist = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
2  tinylist = [123, 'john']
3
4  print(mylist)           # Prints complete List
5  print(type(mylist))     # Prints type of object as List
6  print(len(mylist))      # Prints length of the List
7  print(mylist[0])        # Prints first element of the List
8  print(mylist[1:3])      # Prints elements starting from 2nd till 3rd
9  print(mylist[2:])       # Prints elements starting from 3rd element
10 print(tinylist * 2)      # Prints list two times
11 print(mylist + tinylist) # Prints concatenated lists
```

```
['abcd', 786, 2.23, 'john', 70.2]
<class 'list'>
5
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```

List is mutable

Daftar memiliki berbagai metode yang tersedia yang dapat digunakan untuk memanipulasi daftar.

```
1 myList = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
2 tinylst = [123, 'john']
3
4 print(mylist)
5 myList.append('wiley') # appending an element to a list
6 print(mylist)
7 myList.insert(2,'python') # inserting an element at index 2
8 print(mylist)
9 myList.remove(786) # removing an element from a list
10 print(mylist)
11 myList.pop(3) # removing an element from a specific position
12 print(mylist)
13 myList.extend(tinylst) # extending a list with elements from another list
14 print(mylist)
```

```
['abcd', 786, 2.23, 'john', 70.2]
['abcd', 786, 2.23, 'john', 70.2, 'wiley']
['abcd', 786, 'python', 2.23, 'john', 70.2, 'wiley']
['abcd', 'python', 2.23, 'john', 70.2, 'wiley']
['abcd', 'python', 2.23, 70.2, 'wiley']
['abcd', 'python', 2.23, 70.2, 'wiley', 123, 'john']
```

Tuple



berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung ().



Mengakomodasi pengulangan elemen dan elemen dari tipe data yang berbeda



Objek yang tidak dapat diubah – hanya baca



nilai yang disimpan dalam tuple dapat diakses menggunakan operator irisan ([] dan [:]) dengan indeks mulai dari 0 di awal tuple.



Mendukung pengindeksan negatif tanda plus (+) adalah operator rangkaian tuple, dan tanda asterisk (*) adalah operator pengulangan.

Tuple

Tuple tidak mendukung modifikasi; itu adalah objek yang tidak dapat diubah.

```
tuple_items = ('Cars',786,2.23,'toyota',70.2)
tinytuple = (123,'camry')

print(tuple_items)           #Prints complete tuple
print(tuple_items[0])        #Prints first element of complete tuple
print(tuple_items[1:3])      #prints elements starting from second till third
print(tuple_items[2:])       #Prints elemnts starting from third till end
print(tinytuple*2)           #Prints tuple two times
print(tuple_items+tinytuple) #Prints concatenated tuples |

('Cars', 786, 2.23, 'toyota', 70.2)
Cars
(786, 2.23)
(2.23, 'toyota', 70.2)
(123, 'camry', 123, 'camry')
('Cars', 786, 2.23, 'toyota', 70.2, 123, 'camry')
```

List v/s Tuple

List	Tuple
Dapat didefinisikan menggunakan tanda kurung siku []	Dapat didefinisikan menggunakan tanda kurung ()
Bisa berubah	Tidak bisa berubah
Implikasi dari iterasi memakan waktu dalam list	Implikasi dari iterasi jauh lebih cepat dalam tuple
Kurang hemat memori	Lebih hemat memori
Kesalahan dan perubahan yang tidak terduga dapat dengan mudah terjadi dalam list	Kesalahan dan perubahan yang tidak terduga jarang terjadi pada tuple

String



Merupakan primitive data types tapi mendukung banyak fungsi dari koleksi (list and tuple).



String adalah kumpulan abjad, kata, atau karakter lainnya.



Dapat dibuat dengan melampirkan urutan karakter dalam sepasang tanda kutip tunggal atau ganda.



Tidak dapat diubah, kuat – memiliki banyak metode bawaan untuk eksplorasi dan manipulasi.

String

- String multi-baris dilambangkan menggunakan tanda kutip tiga, `'''` atau `"""`

Strings are immutable objects which means once we declare a string we can't update the already declared string

```
string = "Data Science !"

print(string)           # Print complete string
print(string[0])        # Prints first character of the string
print(string[2:5])      # Prints characters starting from third till fifth
print(string[2:])       # Prints characters starting from third character
print(string * 2)       # Prints strings two times
print(string + "Foundation") # Prints concatenated string
```

```
Data Science !
D
ta
ta Science !
Data Science !Data Science !
Data Science !Foundation
```



03

MAPPING & SET

- Dictionary
- Set

Dictionary



berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung kurawal {}.



Setiap item memiliki dua komponen – kunci dan nilai



'Kunci' mengidentifikasi item, dan 'nilai' menyimpan nilai item. Nilai dapat diakses dengan memasukkan kunci dalam tanda kurung siku ([]).



kunci tidak dapat diubah (angka, String, atau Tuples) dan unik, nilainya dapat berupa jenis apa pun.



Objek yang dapat berubah – dapat dimodifikasi

Dictionary

- String multi-baris dilambangkan menggunakan tanda kutip tiga, ''' atau '''
- String adalah objek yang tidak dapat diubah yang berarti setelah kami mendeklarasikan string, kami tidak dapat memperbarui string yang sudah dideklarasikan.

```
dict={}
dict['one'] = "This is one"
dict[2] = "This is two"

tinydict = {'name':'john','code':6734 , 'dept':'sales'}

print(dict['one'])           # Prints value for 'one' key
print(dict[2])              # Prints value for 2 key
print(tinydict)             # Prints complete dictionary
print(tinydict.keys())      # Prints all keys of the dictionary
print(tinydict.values())    # Prints all the values of each key

This is one
This is two
{'name': 'john', 'code': 6734, 'dept': 'sales'}
dict_keys(['name', 'code', 'dept'])
dict_values(['john', 6734, 'sales'])
```

Set



Kumpulan nilai yang dipisahkan dengan koma dan diapit dalam tanda kurung kurawal {}.



Tidak berurutan, bisa berubah dan tidak mendukung pengulangan elemen



Terutama digunakan untuk operasi teori himpunan seperti serikat pekerja, persimpangan dll.



Set adalah tipe data yang bisa berubah, sehingga dapat dimodifikasi. Frozenset tidak dapat diubah.

Set

- Sets bersifat iretable
- Set juga bisa berubah, yaitu, kita dapat memodifikasi elemen set setelah pembuatan

```
1 setA = {2,4,5,6,5,6,5,4}      # Defining setA
2 setB = {1,2,4,8,8}           # Defining setB
3
4 print(setA, type(setA))      # Prints the variable setA and its data type
5 print(len(setA))             # Prints the length of the setA
6 setA.add(9) ; print(setA)    # Adding new value of the setA
7 setA.remove(6) ; print(setA) # Removing 6 from setA
8 print(setA.union(setB))      # Union operation = setA U setB
9 print(setA.intersection(setB)) # Intersection operation = setA n setB
```

```
{2, 4, 5, 6} <class 'set'>
4
{2, 4, 5, 6, 9}
{2, 4, 5, 9}
{1, 2, 4, 5, 8, 9}
{2, 4}
```

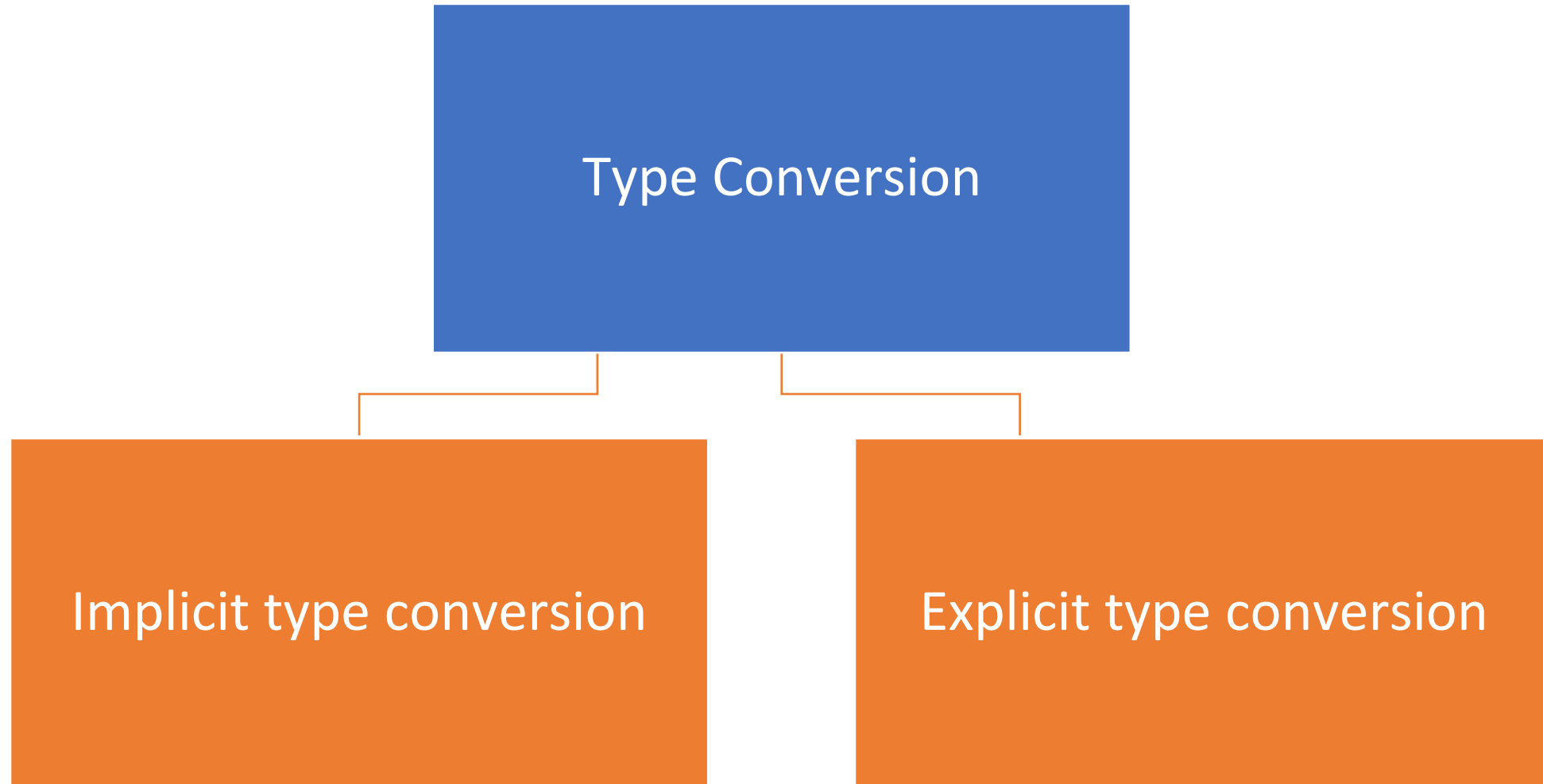


04

CONVERSION

- Type conversion – primitive
- Type conversion – non-primitive

Type Conversion



Type Conversion

Function	Description
int(x [,base])	Mengubah x menjadi integer. base menspesifikasi the base jika x adalah string.
long(x [,base])	Mengubah x menjadi long integer. base menspesifikasi the base jika x adalah string.
float(x)	Mengubah x menjadi floating-point number.
complex(real [,imag])	Membuat sebuah complex number.
str(x)	Mengubah object x menjadi sebuah representasi string.
eval(str)	Mengevaluasi sebuah string dan mengembalikan sebuah object.
tuple(s)	Mengubah s menjadi sebuah tuple.
list(s)	Mengubah s menjadi sebuah list.
set(s)	Mengubah s menjadi sebuah set.
dict(d)	Membuat sebuah dictionary. d merupakan sebuah sequence dari (key,value) tuples.
frozenset(s)	Mengubah s menjadi sebuah frozen set.
chr(x)	Mengubah sebuah integer menjadi sebuah character.
ord(x)	Mengubah sebuah single character menjadi integer value.
hex(x)	Mengubah sebuah integer menjadi sebuah hexadecimal string.
oct(x)	Mengubah sebuah integer menjadi sebuah octal string

Type Conversion : primitive

```
1 x = 2.5; print(x,type(x),end="    >>> ")
2 y = int(x) ; print(y, type(y)) # converting float to int
3 x = 1; print(x,type(x),end="    >>> ")
4 y = bool(x) ; print(y, type(y)) # converting int to boolean
5 x = "124"; print(x,type(x),end="    >>> ")
6 y = int(x) ; print(y, type(y)) # converting string to int
7 x = "2.5"; print(x,type(x),end="    >>> ")
8 y = float(x) ; print(y, type(y)) # converting string to float
9 x = True; print(x,type(x),end="    >>> ")
10 y = str(x) ; print(y, type(y)) # converting boolean to str
11
```

```
2.5 <class 'float'>    >>>  2 <class 'int'>
1 <class 'int'>    >>>  True <class 'bool'>
124 <class 'str'>    >>>  124 <class 'int'>
2.5 <class 'str'>    >>>  2.5 <class 'float'>
True <class 'bool'>    >>>  True <class 'str'>
```

Type Conversion : primitive

```
1 x = [2, 5, 'Hi', 12, 2.5]; print(x, type(x), end = "\n >>> ")
2 y = tuple(x) ; print(y, type(y)) # converting list to tuple
3 x = [2, 5, 'Hi', 12, 2, 2.5, 5]; print(x, type(x), end = "\n >>> ")
4 y = set(x) ; print(y, type(y)) # converting list to set
5 x = (2, 5, 'Hi', 12, 2, 2.5, 5); print(x, type(x), end = "\n >>> ")
6 y = list(x) ; print(y, type(y)) # converting tuple to list
7 x = [['name','john'], ['age',22], ['city','delhi']]; print(x, type(x), end = "\n >>> ")
8 y = dict(x) ; print(y, type(y)) # converting list to dict
```

```
[2, 5, 'Hi', 12, 2.5] <class 'list'>
>>> (2, 5, 'Hi', 12, 2.5) <class 'tuple'>
[2, 5, 'Hi', 12, 2, 2.5, 5] <class 'list'>
>>> {2.5, 2, 5, 12, 'Hi'} <class 'set'>
(2, 5, 'Hi', 12, 2, 2.5, 5) <class 'tuple'>
>>> [2, 5, 'Hi', 12, 2, 2.5, 5] <class 'list'>
[['name', 'john'], ['age', 22], ['city', 'delhi']] <class 'list'>
>>> {'name': 'john', 'age': 22, 'city': 'delhi'} <class 'dict'>
```



05

KESIMPULAN

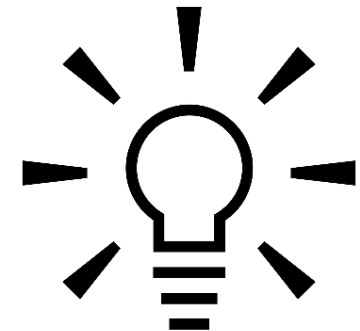
- Kuis
- Ringkasan

Kuis

Pertanyaan

Saat mengembangkan aplikasi kita ingin agar data diproses lebih cepat, dan membuat data hanya bisa dibaca saja. Tipe data yang dipilih untuk menyimpan data tersebut ?

- A. List
- B. Tuple
- C. Dictionary
- D. Set



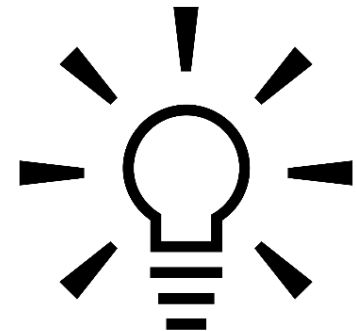
Kuis

Pertanyaan

Saat mengembangkan aplikasi kita ingin agar data diproses lebih cepat, dan membuat data hanya bisa dibaca saja. Tipe data yang dipilih untuk menyimpan data tersebut ?

- A. List
- B. Tuple
- C. Dictionary
- D. Set

Answer- B



Ringkasan

- Struktur data adalah cara untuk menyimpan data dalam bahasa pemrograman
- Python memiliki lima tipe data primitif yang tidak dapat diubah – integer, float, string, Boolean, dan kompleks
- Python memiliki 4 tipe data non primitif – daftar, tupel, kamus, dan set
- Tuple tidak dapat diubah sedangkan daftar, kamus, dan set dapat diubah.
- Kita dapat mengetikkan tipe data primitif cor dari satu ke yang lain.



Orbit Future Academy

PT Orbit Ventura Indonesia
Center of Excellence (Jakarta Selatan)
Gedung Veteran RI, Lt.15
Unit Z15-002, Plaza Semanggi
Jl. Jenderal Sudirman Kav.50, Jakarta
12930, Indonesia

- 📍 Jakarta Selatan/Pusat
- 📍 Jakarta Barat/BSD
- 📍 Kota Bandung
- 📍 Kab. Bandung
- 📍 Jawa Barat

Hubungi Kami

Director of Sales & Partnership
ira@orbitventura.com
+62 858-9187-7388

Social Media



TERIMA KASIH