

# Python Fundamental

## Ch.3 : Data Type

Instruktur : Ahmad Rio Adriansyah  
Nurul Fikri Komputer

# Tipe Data Dasar

- Python memiliki tipe data dasar berikut :
  - Integer (int) : untuk menyimpan bilangan bulat
  - Floating point (float) : untuk menyimpan bilangan riil
  - Complex number (complex) : untuk menyimpan bilangan kompleks
  - String (str) : untuk menyimpan rangkaian karakter
  - Boolean (bool) : benar (**True**), atau salah (**False**)
  - Collection/Sequence : list, tuple, dictionary, stack

# Integer

- Pada python3, hampir tidak ada batasan seberapa banyak digit integer yang bisa disimpan (kecuali batasan memori dari sistem)
- Jadi menuliskan nilai integer seperti ini pun tidak masalah (L di akhir menandakan tipe data `long`)

```
>>> a = 123456789112345678921234567893123456789412345678951234567896123456789712
34567898123456789912345678910234567891123456789122345678913234567891423456789152
3456789162345678917234567891823456789192345678920
>>> a
12345678911234567892123456789312345678941234567895123456789612345678971234567898
12345678991234567891023456789112345678912234567891323456789142345678915234567891
62345678917234567891823456789192345678920L
>>> █
```

# Mengecek Tipe Data

- Gunakan perintah `type()` untuk memeriksa tipe data dari suatu konstanta atau variabel

```
>>> type(a)
```

# Integer

- Kita bisa menambahkan kode berikut di awal angkanya untuk menuliskan integer selain dalam bentuk desimal (basis 10)
  - Biner (basis 2) : 0b atau 0B
  - Oktal (basis 8) : 0o atau 0O
  - Heksadesimal (basis 16) : 0x atau 0X

# Latihan

```
>>> 0b100
4
>>> 0B1111
15
>>> 0o25
21
>>> 00100
64
>>> 0x100
256
>>> 0X13d
317
>>> 
```

# Floating Point

- Floating point / float ditentukan dengan adanya titik desimal (decimal point)

>>> 5.

>>> 3.14

- Dapat menggunakan e atau E sebagai eksponennya

>>> 2.9e8

>>> 1.1e-5

# Floating Point

- Berdasarkan standard IEEE 754, `float` pada python menggunakan 64 bit (double precision) yang nilai maksimalnya sekitar  $1.8e308$
- Nilai yang lebih tinggi dari itu akan direpresentasikan sebagai string `inf` (infinite, tak terhingga)
- Nilai paling kecilnya adalah sekitar  $5.0e-324$ , sehingga yang lebih kecil dari itu akan dianggap sebagai 0



# Complex Number

- Bilangan kompleks adalah pasangan berurut dari 2 buah bilangan riil, satu berperan sebagai bagian riil, yang satu lagi berperan sebagai bagian imajineranya (ditandai dengan karakter j).

- Mudahnya bilangan kompleks adalah  
 $\text{<riil> + <imajiner>j}$

`>>> 5+1j`

- Bisa juga dibuat dengan fungsi complex :  
 $\text{complex(<riil>, <imajiner>)}$

`>>> complex(5,1)`

# Complex Number

- Untuk memeriksa bagian riil dan imajiner dari bilangan kompleks dapat dipanggil dengan :

```
>>> z = 1.5+0.7j
```

```
>>> z.real
```

```
1.5
```

```
>>> z.imag
```

```
0.7
```

# Latihan

- Dalam melakukan pemrograman akan banyak berinteraksi dengan hal-hal yang berkaitan dengan bilangan
- Tentukan hal-hal apa saja yang direpresentasikan dengan kategori integer, float, atau complex!

# Operasi Numerik

- Untuk memproses bilangan, python menggunakan operator +, -, \*, /
- Ekspresinya diproses dari kiri ke kanan
- Hirarki \* dan / lebih tinggi (akan dikerjakan lebih dulu) daripada + dan -
- Kecuali jika diberikan tanda kurung ()
- Tanda sama dengan (=) digunakan untuk memasukkan nilai ke dalam variabel

# Latihan

- Bereksperimen dengan tipe data numerik
- Alokasi nilai ke dalam variabel juga dapat dilakukan secara simultan (sekaligus dalam satu baris)

```
>>> 2+2
4
>>> 5*3
15
>>> 10/4
2.5
>>> panjang = 20
>>> lebar = 10
>>> margin = 2
>>> luasisi = (panjang-2*margin)*(lebar-2*margin)
>>> print(luasisi)
96
>>> █
```

# Konversi Numerik

- Tipe data numerik yang satu dan yang lainnya dapat dipertukarkan, dalam contoh bilangan integer 1 dapat disimpan sebagai float dalam bentuk 1.0.
- Fungsi konversi `float()`, `int()`, `long()`\*, atau `complex()` dapat digunakan untuk mengubah ke dalam tipe data tersebut
  - \* hanya di python 2. di python 3 integer secara otomatis diubah menjadi long apabila terlalu besar nilainya.
- Fungsi `abs()` dapat digunakan untuk mendapatkan magnitude dari bilangan kompleks atau nilai mutlak dari bilangan riil (int atau float)

```
>>> a = 5
>>> b = 3.14
>>> c = -3+2j
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'complex'>
>>> █
```

```
>>> float(a)
5.0
>>> int(b)
3
>>> complex(a)
(5+0j)
>>> complex(b)
(3.14+0j)
>>> abs(c)
3.605551275463989
>>> █
```

# String

- String pada python dituliskan dengan diapit tanda petik ( ' ') atau petik dua ( " ")
- Jika di dalam string tersebut perlu ada tanda petik, gunakan variasi yang satu lagi atau escape character ( \ ) sebelum tanda petiknya.

```
>>> "A'la bin Muhammad"
```

```
>>> ' "Isn\'t", she said.'
```

```
>>> "Tanda petik satu \'\' atau petik dua \"\" bisa  
dituliskan dengan escape character \\\."
```



# Escape Character (\)

- Digunakan saat programmer ingin memerintahkan python untuk menginterpretasikan karakter pada string dengan cara berbeda, misalnya :
  - \n : enter, baris baru
  - \t : tab, tabulasi
- Escape character juga dapat digunakan untuk menuliskan perintah lebih dari satu baris (apabila satu baris terlalu panjang).

# Escape Character (\)

```
>>> pesan = "Kalimat ini terlalu panjang \
...     sehingga tidak bisa dituliskan dalam \
...     satu baris pada python, tetapi apabila \
...     dicetak (print) akan menjadi satu baris."

>>> pesanlagi = "Kalau ini sebaliknya. Ini baris
pertama\nIni baris kedua\nYang ini baris ketiga,
terakhir."
```

# Triple Quoted

- Selain menggunakan escape character di akhir baris, untuk menuliskan beberapa baris string juga dapat digunakan tanda petik tiga kali (atau petik dua tiga kali)
- Yang ditulis dan tercetak akan sama baris per barisnya.

>>> “Ini string

... yang terdiri dari beberapa baris

... tanpa menggunakan escape character. ””

# Menggabungkan String

- Dua buah string dapat digabungkan (concatenate) menjadi satu dengan operator +
- Bisa juga menduplikasinya sebanyak integer tertentu dengan operator \*

```
>>> "ini" + "python"
```

```
'inipython'
```

```
>>> "python"*3
```

```
'pythonpythonpython'
```

# Indeks dalam String

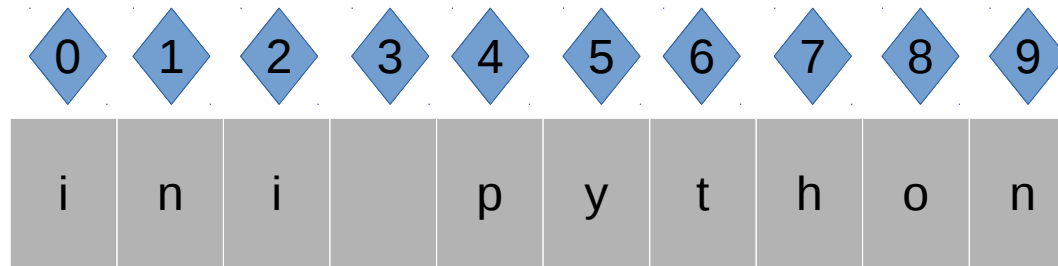
- String dapat dianggap sebagai rangkaian karakter
- Tiap bagiannya ditandai dengan angka indeks dan dapat diakses dengan men-slice-nya
- Indeks dalam python dimulai dari 0 hingga panjang stringnya dikurangi 1
- Untuk mengukur panjang string dapat menggunakan fungsi `len()`

```
>>> nama = "ini python"
```

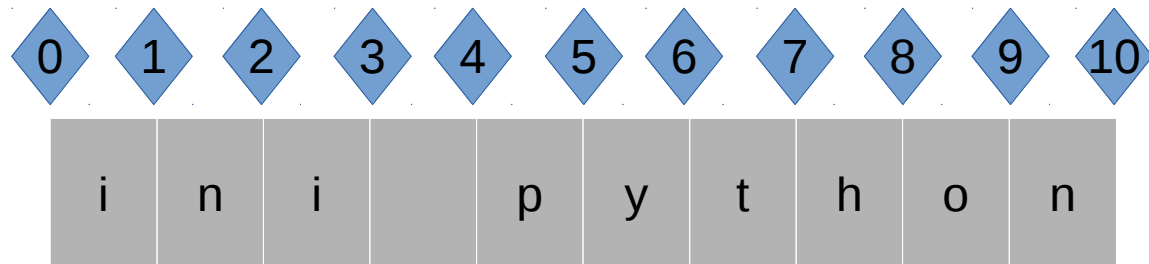
```
>>> len(nama)
```

# Indeks dalam String

## Mengakses Indeks



## Slicing



# Indeks dalam String

- Untuk mengakses indeks string tertentu digunakan kurung siku ( [ ] )  
`>>> nama[5] #huruf ke-6 (indeks 5) pada string nama`
- Pemenggalan string (slice) dilakukan dengan kurung siku ( [ ] ) dan dipisahkan dengan titik dua ( : )  
`>>> nama[0:5] #huruf ke-1 sampai ke-5 (indeks 0 s.d. 4)`
- Jika tidak ditulis defaultnya adalah indeks paling awal (yang depan) dan indeks paling akhir (yang belakang)  
`>>> nama[:5]`  
`>>> nama[3:]`

# Pemenggalan (Slice) String

- Ada beberapa cara slicing

>>> nama[mulai:stop] #indeks stop tidak ikut

>>> nama[mulai:] #dari indeks mulai sampai akhir

>>> nama[:stop] #dari awal sampai stop-1

>>> nama[:] #menyalin semua isi string

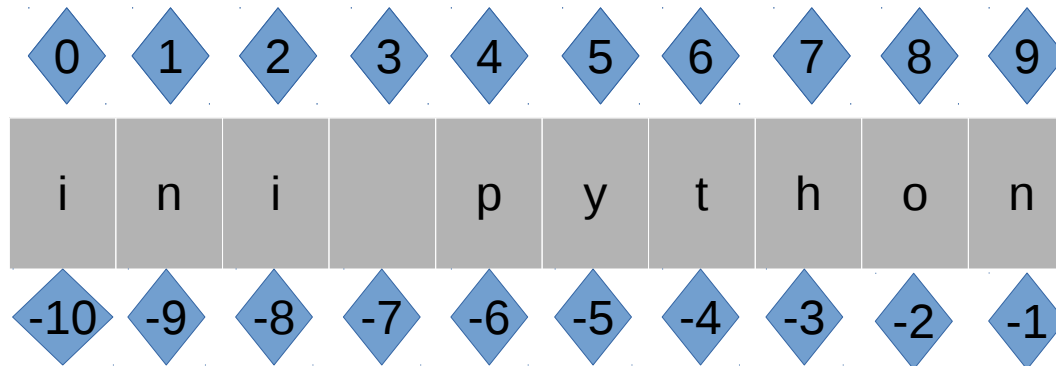
>>> nama[mulai:stop:jeda] #dari mulai hingga stop-1, lompat sebanyak jeda tiap langkahnya



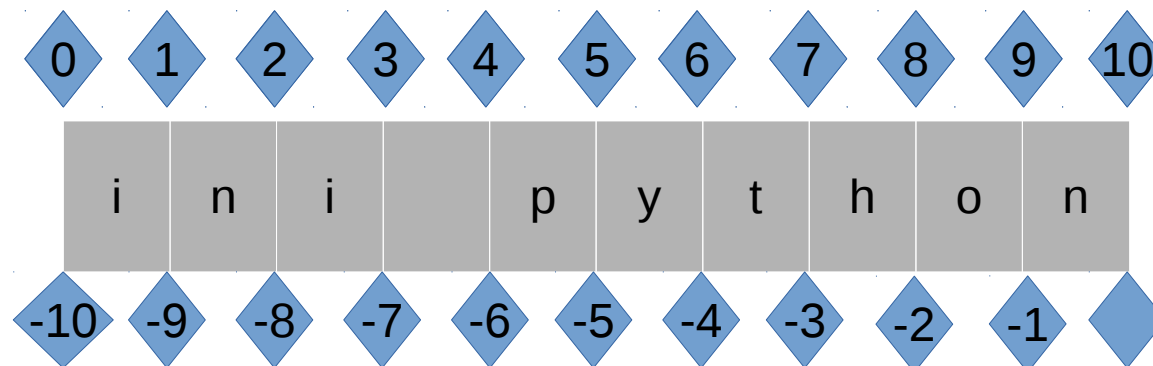
# Indeks Negatif

- Pemanggilan elemen string pada python dapat menggunakan angka negatif. Akan dihitung dari belakang jika negatif yang digunakan.

Mengakses Indeks



Slicing



# Indeks Negatif

```
>>> nama[3:-3]
```

```
>>> nama[-6:8]
```

```
>>> nama[-6:-1]
```

```
>>> nama[-5:]
```

```
>>> nama[: -5]
```

```
>>> nama[ : :-1]
```

```
>>> nama[9:3:-2]
```

# Latihan

- 1) Buat 3 variabel string **nama**, **nip** (10 digit angka), dan **tanggal lahir** (format yyyy-mm-dd).  
\*Note: python memiliki tipe data untuk tanggal (date), tapi dalam latihan ini, digunakan string.
- 2) Buat sebuah kode baru yang disebut nomor identifikasi, formatnya adalah huruf pertama dari nama, digabungkan dengan 2 digit tahun lahir, dan 3 digit terakhir dari nip.

Contoh : Jika namanya Uwais Al Qarni, lahir tanggal 25 Maret 1994, dan nipnya 1502348871, maka nomor identifikasinya adalah U94871

# Fungsi Lain dalam String

- `string.count()`
- `string.find()`
- `string.lower()`
- `string.upper()`
- `string.format()`
- `string.isalnum()`
- `string.replace()`
- `string.split()`
- `string.splitlines()`

# Help

- Untuk melihat dokumentasi terkait dengan objek tertentu (misalnya string) pada python dapat digunakan fungsi `help()` :

```
>>> help(str)
```

- Untuk keluar dari menunya, dapat diketik `q` atau `quit` (pada interactive help system).
- Fungsi ini akan sangat membantu saat bingung terkait penggunaan suatu objek/fungsi.

```
>>> pesan = "Selamat datang {0}, ini adalah  
kunjungan ke {1} pada hari ini"
```

```
>>> pesan.format("Rio",10)
```

```
>>> pesan.upper()
```

```
>>> pesan.find("kunjungan")
```

```
>>> pesan.replace("datang", "tinggal")
```

```
>>> daftarkata = pesan.split()
```

```
>>> type(daftarkata)
```

# Bool

- Tipe data bool atau boolean adalah tipe data binary berupa pernyataan benar (**True**) atau salah (**False**).
- Nilai-nilai berikut dianggap salah pada python
  - None
  - False
  - 0 pada tipe numerik apapun (0, 0.0, 0j)
  - Sequence atau dictionary kosong ([ ], ( ), "", { })
  - Kelas (class) dengan metode **\_\_bool\_\_()** atau **\_\_len\_\_()** bernilai 0

# Bool

```
>>> 4 > 1
```

```
>>> 10%3 == 0
```

```
>>> 5*12 == 60
```

```
>>> 2*2 != 5
```

```
>>> -10 < 6
```



# List

- Dalam pemrograman ada struktur data yang disebut dengan **array**, yaitu sekumpulan data yang bertipe sama dalam sebuah nama variabel.
- Contohnya : **string** adalah **array** dari **character**.
- Pada python, padanan untuk **array** adalah **list**.
- Bedanya, **list** dapat menyimpan data dengan tipe yang berbeda-beda.

# List

- Deklarasi list dituliskan dengan kurung siku ( [ ] ) dan dipisahkan tiap anggotanya dengan tanda koma ( , )

```
>>> buah = ["apel", "jeruk", "mangga"]
```

```
>>> kode = ["Ax7Hmm", "U94871", 12345,  
3.14159265,]
```

```
>>> listkosong = []
```

# List

- Mengakses indeks dan slicing pada list mirip seperti pada string, bedanya :
  - Elemen string tidak bisa diassign menggunakan indeksnya (immutable), list bisa.
  - Pada saat kita mengassign variabel dengan list, yang diberikan bukan isinya, tetapi alamat memorinya. Akibatnya jika kita mengubah salah satu elemen list, list yang satu lagi akan terpengaruh.

# List

```
>>> stringa = "ini string"
```

```
>>> stringb = stringa
```

```
>>> stringa[5] = "z"
```

```
>>> lista = list(stringa)
```

```
>>> listb = lista
```

```
>>> listc = lista[:]
```

```
>>> lista[5] = "z"
```

```
>>> listb[2] = "x"
```

# List

```
>>> kode[:2] = [ ]
```

```
>>> kode[:0] = [ 1.4, 2.7 ]
```

```
>>> kode[ : ] = [ 1, 2, 3, 4, 5, 6, 7 ]
```

```
>>> kode[3:3] = ["selipan"]
```

```
>>> kode[1:1] = "abc"
```

```
>>> kode[1:3] = [-1,-2,-3,-4,-5]
```

# Fungsi Lain dalam List

- `list.append(x)`
- `list.insert(i,x)`
- `list.remove(x)`
- `list.index(x)`
- `list.count(x)`
- `list.sort()`
- `list.reverse()`
- `list.pop(i)`

```
>>> sayur = ['bayam', 'tomat']  
>>> sayur.append('labu')  
>>> sayur.insert(0, 'wortel')  
>>> sayur.index('labu')  
>>> sayur.remove('tomat')  
>>> makanan = sayur + buah  
>>> makanan.reverse()  
>>> makanan.sort()  
>>> makanan.pop(0)
```

# Matriks

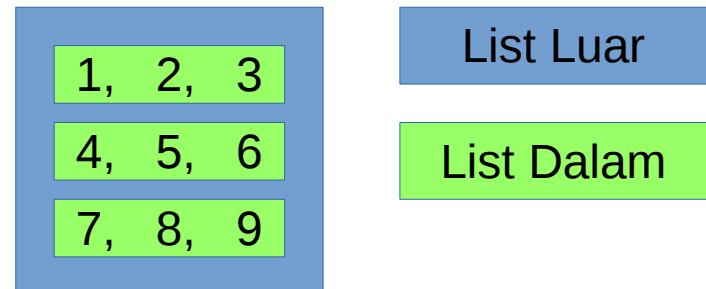
- Untuk membuat matriks (array multidimensi) pada python dapat digunakan list dalam list atau istilahnya list bersarang (nested list)
- Untuk matriks 2D, list luar merepresentasikan kolom, list dalam merepresentasikan baris-baris pada matriksnya.

```
>>> matriks = [ [1,2,3], [4,5,6], [7,8,9] ]
```

```
>>> print(matriks)
```

```
>>> for baris in matriks :
```

```
...     print(baris)
```





# Matriks

- Mengakses elemen matriks 2D sama seperti mengakses elemen list dua kali.
- $A_{i,j}$  = elemen matriks di baris ke-i kolom ke-j

```
>>> A01 = matriks[0][1]
```

```
>>> A12 = matriks[1][2]
```

```
>>> A20 = matriks[2][0]
```

```
>>> A22 = matriks[2][2]
```

# Tuple

- Data sequence lain pada python adalah tuple
- Tuple mirip dengan list, tetapi elemennya immutable (tidak dapat diubah)
- Diapit dengan tanda kurung biasa dan tiap elemennya dipisahkan dengan tanda koma
- Tanda kurung bisa tidak dituliskan

# Tuple

Penulisan	Interpretasi
()	Sebuah tuple kosong
(1, )	Sebuah tuple dengan satu item
(1 )	Suatu ekspresi bukan tuple
(1, 2, 3)	Sebuah tuple dengan tiga item
1, 2, 3	Sebuah tuple dengan tiga item tanpa tanda kurung
('nama', ('awal', 'akhir'))	Tuple dua item dengan tuple bersarang

# Tuple

```
>>> koordinat = (1.6140046,124.7245567,13)
>>> tempat = ("Bunaken",koordinat)
>>> transkrip = ('A', 'A', 'B', 'A', 'C', 'D', 'A', 'A')
>>> transkrip.count('A')
>>> transkrip.index('D')
```

# Range

- Range adalah fungsi built in (terpasang) pada python berupa sequence
- Biasanya digunakan untuk proses iterasi

```
>>> range(10)
```

```
>>> range(3,7)
```

```
>>> range(0,100,5)
```

# Range

- Ada 3 data descriptor yang bisa dipanggil pada fungsi range yaitu start, stop, dan step. Parameter yang diberikan satu, dua, atau tiga buah memberikan representasi yang berbeda
  - `range(stop)`
  - `range(start, stop)`
  - `range(start, stop, step)`
- Parameternya berupa integer
- start defaultnya 0, dan step defaultnya 1

# Range

- Pada python 3, range tidak langsung dijabarkan, tetapi dituliskan sebagai rangeobject
- Untuk menjabarkannya, dapat dikonversi ke dalam list

```
>>> hitung = range(10)
```

```
>>> list(hitung)
```

# Fungsi untuk Sequence

- Data sequence dapat berupa list, tuple, atau range.
- Ada 3 fungsi built in di python yang berguna jika digunakan oleh data sequence :
  - filter()
  - map()
  - reduce()



# Filter

- Mengembalikan sequence berisi item-item yang memenuhi fungsi tertentu

```
>>> def f(x): return x%2!=0 and x%3!=0
```

```
...
```

```
>>> filter(f, range(2,25))
```

- Python akan mengembalikan nilai-nilai dari range(2,25) yang memenuhi fungsi f. Jadi nilai-nilai yang bukan kelipatan 2 dan bukan kelipatan 3 akan dikembalikan dalam bentuk filter object.
- Untuk melihatnya dapat dikonversi ke dalam list dengan `list()` atau digunakan fungsi `filter.__next__()` untuk melihat satu per satu.

# Map

- Memanggil fungsi tertentu untuk setiap item pada sequence dan mengembalikan hasil fungsinya

```
>>> def kuadrat(x): return x*x
```

```
...
```

```
>>> map(kuadrat, range(10))
```

- Python akan mengembalikan nilai-nilai kuadrat dari range(10) dalam bentuk map object

# Reduce

- Mengembalikan sebuah nilai berupa hasil fungsi binari tertentu dari keseluruhan sequence

```
>>> def tambah(a,b): return a+b
```

```
...
```

```
>>> reduce(tambah, range(5,100,3))
```

- Python akan mengembalikan hasil penjumlahan pada range(5,100,3)

# Dictionary

- Dictionary disebut juga sebagai associative array karena mengaitkan dua buah nilai berupa key dan value.
- Dituliskan dalam tanda kurung kurawal dan tiap nilainya dipisahkan dengan koma
- Pemanggilan nilainya menggunakan key, bukan index seperti list atau tuple.
- Format penulisan :  
`dict = { key1 : value1, key2 : value2, ... }`

# Dictionary

```
>>> data = {"nama" : "Ahmad Rio",  
...         "tanggal lahir" : "1 Januari 1414",  
...         "buah favorit" : ["alpukat", "kiwi"],  
...         "berat" : 70,  
...         "banyak anak" : 3,  
...         1 : "Raihan", 2 : "Umar", 3 : "Habibi" }  
>>> data["nama"]
```

- Key di atas boleh mengandung spasi karena berbentuk string. Key dapat berupa data immutable seperti string, integer, bahkan tuple.

# Dictionary

Method	Efek
<code>dict.clear( )</code>	Menghapus seluruh item dari dictionary
<code>dict.copy( )</code>	Menghasilkan salinan dari dictionary
<code>dict.__contains__(k)</code>	Menguji apakah k merupakan key dari dictionary. Jika ada key k, maka python akan mengembalikan nilai True, jika tidak maka akan mengembalikan nilai False
<code>dict.items( )</code>	Menghasilkan salinan list pasangan key dan value dari dictionary
<code>dict.keys( )</code>	Menghasilkan salinan list key dari dictionary
<code>dict.update(otherdict)</code>	Melakukan update dictionary
<code>dict.values( )</code>	Menghasilkan salinan list value dari dictionary

# Latihan

- Buat sebuah dictionary berisi informasi tentang tempat wisata di Indonesia (boleh googling, setidaknya 3 buah tempat)
- Yang jadi key-nya adalah nama tempat wisatanya. Value-nya berupa dictionary yang berisi informasi tentang lokasi, koordinat, tipe tempat wisata, berbayar/tidaknya, tahun diresmikan, dll.