

# Fiqry Revadiansyah

A data scientist with 5+ years of experience

Currently working at



Past Experiences:



bukalapak

[Profile & Portfolios](#)



LINKEDIN



SPEAKERDECK



**Fiqry Revadiansyah**

Data Scientist

# Python Intro & Data Types

(Dictionary and JSON)



## Outline Pembelajaran

# Dictionary and JSON

- Pengenalan Tipe Data Dictionary
- Operasi pada Dictionary
- Pengenalan pada JSON
- Import & Export JSON



## Outline Pembelajaran

# Dictionary and JSON



# Hands-On Required :

**Hands - On : 3. Dictionary dan JSON.ipynb**

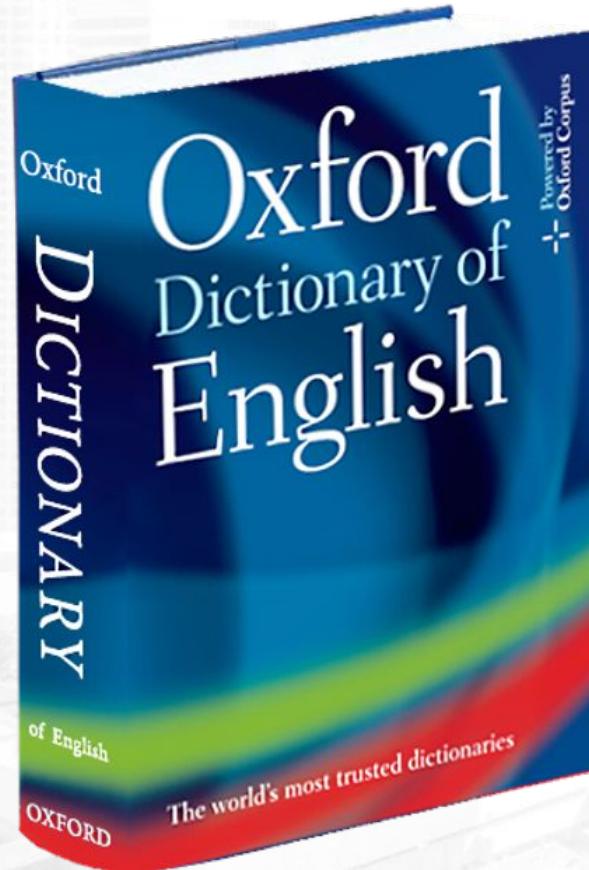
Klik disini untuk  
mengakses folder  
Hands-On

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. Jika sudah ada dictionary, bagaimana mengakses datanya

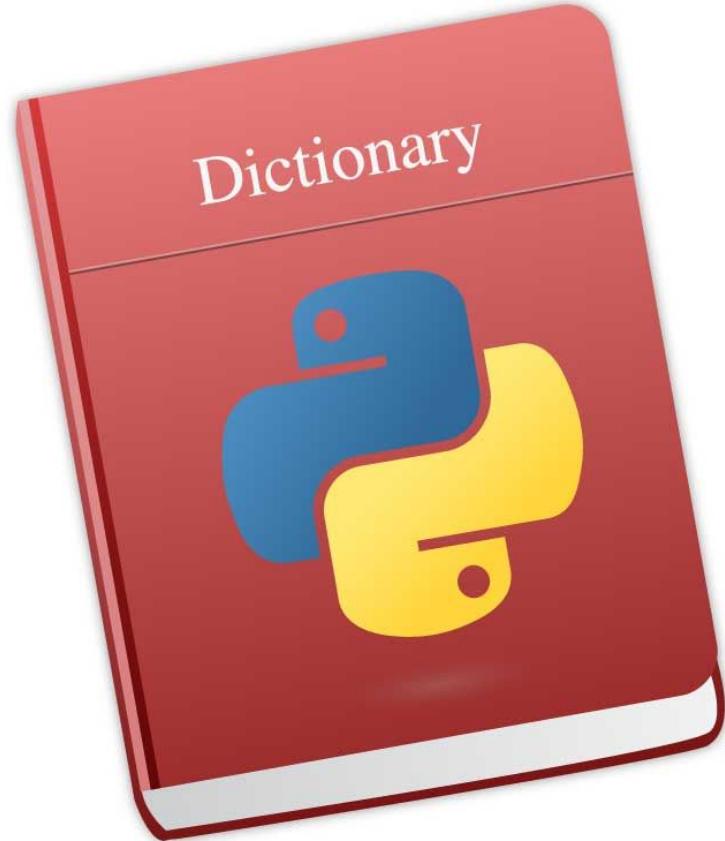
# Pengenalan Dictionary

1. **Apa itu dictionary dan bagaimana membuatnya**
2. Jika sudah ada dictionary, bagaimana mengakses datanya



Konsep Dictionary pada Python cukup mirip dengan Dictionary (kamus) pada umumnya

Kamus English digunakan untuk menyimpan sekumpulan definisi dari kata-kata dalam bahasa inggris



Sederhananya, dictionary pada Python digunakan untuk menyimpan kumpulan data, bisa berupa 1 value ataupun sebuah List

# Membuat Dictionary

## Pada Python

### Contoh Dictionary

```
{ 'nama' : 'dadang',  
  'umur' : 30,  
  'hobi' : 'memasak',  
  'sepatu' : 'nike' }
```

# Membuat Dictionary

## Pada Python

### Format Dictionary

```
{ 'nama' : 'dadam',  
  'umur' : 30,  
  'hobi' : 'memasak',  
  'sepatu' : 'nike' }
```

Dibuka dengan  
kurung kurawal buka “ { ”

Ditutup dengan  
kurung kurawal tutup “ } ”

# Membuat Dictionary

## Pada Python

### Format Dictionary

```
{'nama': 'dadan',  
 'umur': 30,  
 'hobi': 'memasak',  
 'sepatu': 'nike'}
```

**Key**

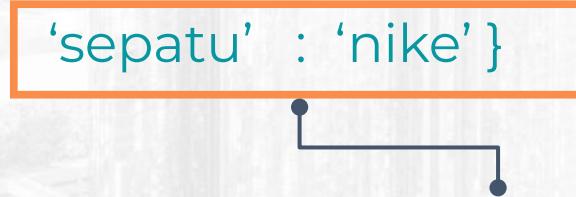
**Value**

# Membuat Dictionary

## Pada Python

### Format Dictionary

```
{ 'nama' : 'dadam' ,  
  'umur' : 30 ,  
  'hobi' : 'memasak' ,  
  'sepatu' : 'nike' }
```



key dan value  
dipisahkan oleh tanda  
titik dua (:)

# Membuat Dictionary

## Pada Python

### Format Dictionary

```
{ 'nama' : 'dadah',  
  'umur' : 30,  
  'hobi' : 'memasak',  
  'sepatu' : 'nike' }
```

Pisahkan tiap key  
dan value  
dengan tanda koma

# Membuat Dictionary

## Pada Python

### Format Dictionary

```
profile = { 'nama' : 'dadam',  
            'umur' : 30,  
            'hobi' : 'memasak',  
            'sepatu' : 'nike' }
```

- Simpan dictionary ke dalam object bernama **profile**

# Membuat Dictionary

## Pada Python

**Contoh Dictionary** setelah kita run

```
profile = {'nama' : 'dadan',
           'umur' : 30 ,
           'hobi' : 'memasak',
           'sepatu' : 'nike' }

profile
{'nama': 'dadan', 'umur': 30, 'hobi': 'memasak', 'sepatu': 'nike'}
```

# Membuat Dictionary

## Pada Python

Perbedaan Dictionary dengan List : **Format**

Dictionary

```
{ 'nama' : 'dadang',  
  'umur' : 30,  
  'hobi' : 'memasak',  
  'sepatu' : 'nike' }
```

List

```
[ 'dadang', 30, 'memasak', 'nike' ]
```

# Membuat Dictionary

## Pada Python

Perbedaan Dictionary dengan List : **Index**

### Dictionary

```
{ 'nama' : 'dad',  
  'umur' : 30,  
  'hobi' : 'memasak',  
  'sepatu' : 'nike' }
```

Index nya adalah key itu sendiri

### List

```
[ 'dad', 30, 'memasak', 'nike' ]
```

---

0      1      2      4

Index nya adalah nomor urut  
yang dimulai dari 0

# Apakah setiap Key hanya boleh 1 value ?

```
{ 'nama' : 'dadan' ,  
  'umur' : 30 ,  
  'hobi' : 'memasak' ,  
  'sepatu' : 'nike' }
```

**Key**                    **Value**

Apakah setiap Key  
hanya boleh 1 value ?  
Boleh lebih, dan dibuat dalam format List

# Membuat Dictionary

## Pada Python

### Contoh Dictionary

```
{ 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' , 'deden' ] ,  
  'umur' : [ 30 , 40 , 26 , 25 , 33 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' , 'kayang' ] ,  
  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' , 'Asics' ] }
```

# Membuat Dictionary

## Pada Python

### Contoh Dictionary

```
{ 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' , 'deden' ] ,  
  'umur' : [ 30 , 40 , 26 , 25 , 33 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' , 'kayang' ] ,  
  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' , 'Asics' ] }
```

value pada setiap key berupa List

# Membuat Dictionary

## Pada Python

### Contoh Dictionary

```
{ 'nama' : 'dadam' ,  
  'umur' : 30 ,  
  'hobi' : 'memasak' ,  
  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' , 'Asics' ] }
```

Bisa juga kasusnya..  
Hanya beberapa key saja yang  
berupa List, sedangkan sisanya  
tetap 1 value saja

# Membuat Dictionary

## Pada Python

### Contoh Dictionary

```
{ 'nama'    : 'dadam' ,  
  'umur'    : 30 ,  
  'hobi'     : 'memasak' ,  
  'sepatu'   : ['nike' , 'adidas' , 'puma' , 'rebook' , 'Asics' ] }
```

Pada kasus ini,  
mungkin si Dadan ini punya  
banyak sepatu yang dia punya

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan 1 key

```
{ 'nama' : [ 'dadan' , 'didin' , 'dodon' ] }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan 2 key

```
{ 'nama' : [ 'dadan' , 'didin' , 'dodon' ],  
  'umur' : [ 30 , 40 , 26 ] }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan 3 key

```
{ 'nama' : [ 'dadan' , 'didin' , 'dodon' ] ,  
  'umur' : [ 30 , 40 , 26 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan n key

```
{ 'nama' : [ 'dadang' , 'didin' , 'dodon' ] ,  
  'umur' : [ 30 , 40 , 26 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' , 'basket' ] ,  
  |  
  'ipk' : [ 3.45,3.60,3.52,...] }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary kosong (tanpa key)

```
{ }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan total value berbeda

```
{ 'nama' : [ 'dadan' , 'didin' , 'dodon' , 'dudun' , 'deden' ] ,  
  'umur' : [ 30 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' ] }
```

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan total value berbeda

```
{'nama' : [ 'dadan' , 'didin' , 'dodon' , 'dudun' , 'deden' ] ,  
 'umur' : [ 30 ] ,  
 'hobi' : [ 'memasak' , 'berkebun' ] }
```

value berisi 5 element

# Membuat Dictionary

Pada Python

**Contoh :** Dictionary dengan total value berbeda

```
{ 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudung' , 'dedeh' ] ,  
  'umur' : [ 30 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' ] }
```

value berisi 1 element

# Membuat Dictionary

## Pada Python

**Contoh :** Dictionary dengan total value berbeda

```
{ 'nama' : [ 'dadan' , 'didin' , 'dodon' , 'dudun' , 'deden' ] ,  
  'umur' : [ 30 ] ,  
  'hobi' : [ 'memasak' , 'berkebun' ] }
```

value berisi 2 element

# Membuat Dictionary

## Pada Python

### Simpan / Assign Dictionary ke Object Baru

```
mydictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                 'umur' : [ 30 , 40 , 26 , 25 ] ,  
                 'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                 'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

# Membuat Dictionary

## Pada Python

### Simpan / Assign Dictionary ke Object Baru

```
mydictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```



Nama dictionary (bebas) ,  
berguna saat operasi pada  
dictionary dilakukan

# Membuat Dictionary

## Pada Python

### Python Implementasi

```
mydictionary = { 'nama' : [ 'dadan' , 'didin' , 'dodon' , 'dudun' ] ,  
                 'umur' : [ 30 , 40 , 26 , 25 ] ,  
                 'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                 'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

```
mydictionary
```

```
{'nama': ['dadan', 'didin', 'dodon', 'dudun'],  
 'umur': [30, 40, 26, 25],  
 'hobi': ['memasak', 'berkebun', 'basket', 'musik'],  
 'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

# Challenge Time

Buat dictionary bernama **customer** dengan 3 key yaitu "**id**", "**nama**", dan "**umur**".

Value dari setiap key mengikuti 3 list berikut.

1. list **id** = ["C1","C2","C3"]
2. list **nama** = ["Adi", "Budi", "Dani"]
3. list **umur** = [25,19,30]

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**
  - Mengambil semua Key pada Dictionary
  - Mengambil List dari Dictionary
  - Mengambil element pada List dari Dictionary
  - Meng-update element pada List dari Dictionary

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**
  - **Mengambil semua Key pada Dictionary**
    - Mengambil List dari Dictionary
    - Mengambil element pada List dari Dictionary
    - Meng-update element pada List dari Dictionary

# Mengambil Key

Dari sebuah Dictionary

**Format Code :**

```
nama_dictionary.keys()
```

# Mengambil Key

Dari sebuah Dictionary

**Contoh Code :**

```
profile.keys()  
  
dict_keys(['nama', 'umur', 'hobi', 'sepatu'])
```

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**
  - Mengambil semua Key pada Dictionary
  - **Mengambil List dari Dictionary**
  - Mengambil element pada List dari Dictionary
  - Meng-update element pada List dari Dictionary

## Dictionary Sebelumnya

```
my_dictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

## Dictionary Sebelumnya

```
my_dictionary = { 'nama' : [ 'dadan' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

- 
- 
- 
- 

Dictionary sudah ada,  
bagaimana mengambil kembali  
list dari Dictionary tersebut ?

# Mengambil List

dari Dictionary yang sudah ada

**Format Code :**

```
nama_dictionary['nama_key']
```

# Mengambil List

dari Dictionary yang sudah ada

**Contoh Code 1:** Ambil data nama pada my\_dictionary

```
my_dictionary['nama']
```

```
my_dictionary['nama']
```

```
['dadon', 'didin', 'dodon', 'dudun']
```

# Mengambil List

dari Dictionary yang sudah ada

**Contoh Code 2 :** Ambil data umur pada my\_dictionary

```
my_dictionary['umur']
```

```
my_dictionary['umur']  
[30, 40, 26, 25]
```

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**
  - Mengambil semua Key pada Dictionary
  - Mengambil List dari Dictionary
  - Mengambil element pada List dari Dictionary**
  - Meng-update element pada List dari Dictionary

# Mengambil element pada List

Dari sebuah Dictionary

```
my_dictionary['nama']
```

```
my_dictionary['nama']
```

```
['dadon', 'didin', 'dodon', 'dudun']
```

# Mengambil element pada List

Dari sebuah Dictionary

```
my_dictionary['nama']
```

```
my_dictionary['nama']
```

```
['dadon', 'didin', 'dodon', 'dudun']
```

Bisa dikatakan, format ini  
menjadi sebuah List

# Mengambil element pada List

Dari sebuah Dictionary

my\_dictionary['nama']

```
my_dictionary['nama']  
['dadam', 'didin', 'dodon', 'dudun']
```

Saya hanya ingin  
mengambil element  
atas nama didin saja



Lakukan operasi pada List

# Mengambil element pada List

Dari sebuah Dictionary

**Format Code :**

nama\_dictionary['nama\_key'][index]



Index dari element pada List  
(dimulai dari 0)

# Mengambil element pada List

Dari sebuah Dictionary

**Contoh Code 1:** Ambil data nama pada List pada urutan kedua dari my\_dictionary

```
my_dictionary['nama']  
['dadon', 'didin', 'dodon', 'dudun']
```

```
my_dictionary['nama'][1]  
'didin'
```

# Mengambil element pada List

Dari sebuah Dictionary

**Contoh Code 2 :** Ambil data nama pada List pada urutan keempat dari my\_dictionary

```
my_dictionary['nama']
```

```
['dadun', 'didin', 'dodon', 'dudun']
```

```
my_dictionary['nama'][3]
```

```
'dudun'
```

# Challenge Time

Dari dictionary yang tadi telah dibuat yaitu dictionary **customer**:

1. ambil **element/item pertama** dari **key 'nama'** pada dictionary customer
2. ambil list **umur**
3. ambil **element/item kedua** dari **key 'id'** pada dictionary customer
4. hitung rata-rata **umur**

# Pengenalan Dictionary

1. Apa itu dictionary dan bagaimana membuatnya
2. **Jika sudah ada dictionary, bagaimana cara mengakses datanya**
  - Mengambil semua Key pada Dictionary
  - Mengambil List dari Dictionary
  - Mengambil element pada List dari Dictionary
  - Meng-update element pada List dari Dictionary**

## Dictionary Sebelumnya

```
my_dictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

## Dictionary Sebelumnya

```
my_dictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

Terdapat kesalahan data (misalnya)  
Harusnya "Memancing", bukan "berkebun"

# Meng-update element pada List

Dari sebuah Dictionary

**Format Code :**

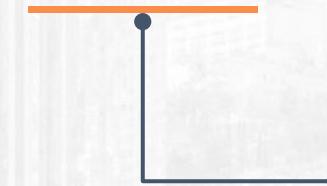
```
nama_dictionary['nama_key'][index] = value baru
```

# Meng-update element pada List

Dari sebuah Dictionary

**Format Code :**

nama\_dictionary['nama\_key'][index] = value baru



nama key yang element  
list-nya ingin diupdate

# Meng-update element pada List

Dari sebuah Dictionary

**Format Code :**

nama\_dictionary['nama\_key'][index] = value baru



Index element pada List yang  
ingin diupdate

# Meng-update element pada List

Dari sebuah Dictionary

**Format Code :**

nama\_dictionary['nama\_key'][index] = value baru

Nilai baru dari  
element

## Dictionary Sebelumnya

```
my_dictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25 ] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

Element berasal dari Key hobi  
dan pada index = 1 (urutan kedua)

# Meng-update element pada List

Dari sebuah Dictionary

**Contoh Code :**

```
my_dictionary['hobi'][1] = 'memancing'
```

```
my_dictionary
```

```
{'nama': ['dadang', 'didin', 'dodon', 'dudun'],
 'umur': [30, 40, 26, 25],
 'hobi': ['memasak', 'memancing', 'basket', 'musik'],
 'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

# Challenge Time

Dari dictionary yang tadi telah dibuat yaitu dictionary **customer**:

1. Ternyata element pertama pada **key ‘umur’** (umur Adi) seharusnya bernilai **20**. Update lah umur tersebut
2. Ternyata nama customer terakhir seharusnya adalah **‘Dina’**. Update lah nama tersebut

## Outline Pembelajaran

# Dictionary and JSON



Pengenalan Tipe Data  
Dictionary



Operasi pada Dictionary



Pengenalan pada JSON



Import & Export JSON

# Beberapa hal penting :

1. Mengetahui apakah suatu key ada pada dictionary
2. Menambah Key dan Value pada dictionary
3. Menghapus key pada dictionary
4. Menghapus value pada dictionary

# Beberapa hal penting :

1. **Mengetahui apakah suatu key ada pada dictionary**
2. Menambah Key dan Value pada dictionary
3. Menghapus key pada dictionary
4. Menghapus value pada dictionary

## Format Code :

'nama\_key' in nama\_dictionary

## Format Code :

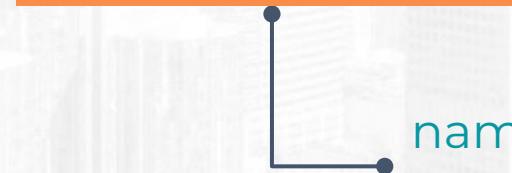
'nama\_key' in nama\_dictionary



nama key yang ingin  
dicari apakah ada dalam  
dictionary

## Format Code :

'nama\_key' in nama\_dictionary



nama dictionary  
yang dipunya

## Contoh Code :

```
'umur' in my_dictionary
```

True

```
'hobi' in my_dictionary
```

True

```
'usia' in my_dictionary
```

False

```
'cita_cita' in my_dictionary
```

False

# Beberapa hal penting :

1. Mengetahui apakah suatu key ada pada dictionary
- 2. Menambah Key dan Value pada dictionary**
3. Menghapus key pada dictionary
4. Menghapus value pada dictionary

```
profile = { 'nama' : 'dadam',  
           'umur' : 30,  
           'hobi' : 'memasak',  
           'sepatu' : 'nike' }
```



Kita ingin menambah key baru dari dadan,  
sebut saja warna\_favorite = 'Biru'

## Format Code :

```
nama_dictionary.update( { 'key_baru' : 'value_baru' } )
```

## Contoh Code :

```
profile.update( { 'warna_favorite': 'biru' } )
```

```
profile.update({ 'warna_favorite' : 'biru' })
```

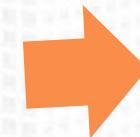
```
profile
```

```
{'nama': 'dadang',  
'umur': 30,  
'hobi': 'memasak',  
'sepatu': 'nike',  
'warna_favorite': 'biru'}
```

## Kasus 1 :

### Menambah pada value nya saja

```
profile = { 'nama' : 'dadang' ,  
           'umur' : 30 ,  
           'hobi' : 'memasak' ,  
           'sepatu' : 'nike' }
```



Ternyata, Menari juga hobinya  
dadang

**Format Code :** value tersebut diubah menjadi sebuah List

```
nama_dictionary['nama_key'] = ['value_1' , 'value_2']
```

## Contoh Code :

```
profile['hobi'] = ['memasak', 'menari']
```

```
profile['hobi'] = ['memasak', 'menari']
```

```
profile
```

```
{'nama': 'dadang', 'umur': 30, 'hobi': ['memasak', 'menari'], 'sepatu': 'nike'}
```

## Kasus 2 :

Menambah value saja  
dan berupa List

```
my_dictionary = { 'nama' : [ 'dadang' , 'didin' , 'dodon' , 'dudun' ] ,  
                  'umur' : [ 30 , 40 , 26 , 25] ,  
                  'hobi' : [ 'memasak' , 'berkebun' , 'basket' , 'musik' ] ,  
                  'sepatu' : [ 'nike' , 'adidas' , 'puma' , 'rebook' ] }
```

Menambah satu value (Hendra)  
baru pada key "Nama"



## Format Code :

```
nama_dictionary['nama_key'].append('value baru')
```

## Format Code :

nama\_dictionary['nama\_key'].append('value baru')

Sebuah List

## Format Code :

nama\_dictionary['nama\_key'].append('value baru')

Sebuah List

Konsepnya mirip dengan  
menambah element pada List



## Contoh Code :

```
my_dictionary['nama'].append('Hendra')
```

```
my_dictionary
```

```
{'nama': ['dadang', 'didin', 'dodon', 'dudun', 'Hendra'],
 'umur': [30, 40, 26, 25],
 'hobi': ['memasak', 'berkebun', 'basket', 'musik'],
 'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

# Challenge Time

Dari dictionary yang tadi telah dibuat yaitu dictionary **customer** dan telah diupdate kesalahan pada umur dan nama:

1. Tambahkan key dan value baru yaitu:  
'tempat\_tinggal' : ['Bandung', 'Jakarta', 'Bogor']
  
2. Tambahkan value baru pada setiap key yaitu:
  - untuk 'id': 'C4'
  - untuk 'nama': 'Ina'
  - untuk 'umur': 22
  - untuk 'tempat\_tinggal': 'Tangerang'

# Beberapa hal penting :

1. Mengetahui apakah suatu key ada pada dictionary
2. Menambah Key dan Value pada dictionary
- 3. Menghapus key pada dictionary**
4. Menghapus value pada dictionary

## Format Code :

```
del nama_dictionary['nama_key']
```

catatan :

setelah dihapus, key tersebut **secara total hilang dan tidak bisa dikembalikan.**

Sehingga harus ditambahkan lagi dari awal

## Contoh Code 1: Menghapus key hobi

```
del my_dictionary['hobi']

my_dictionary

{'nama': ['dadang', 'didin', 'dodon', 'dudun'],
 'umur': [30, 40, 26, 25],
 'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

## Contoh Code 2: Menghapus key nama

```
del my_dictionary['nama']
```

```
my_dictionary
```

```
{'umur': [30, 40, 26, 25],  
 'hobi': ['memasak', 'berkebun', 'basket', 'musik'],  
 'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

# Beberapa hal penting :

1. Mengetahui apakah suatu key ada pada dictionary
2. Menambah Key dan Value pada dictionary
3. Menghapus key pada dictionary
- 4. Menghapus value pada dictionary**

## **Format Code 1:** Menghapus value terakhir pada salah satu Key

```
nama_dictionary['nama_key'].pop()
```

## Contoh Code : Menghapus value terakhir pada key name

Before

```
my_dictionary
```

```
{'nama': ['dad', 'didin', 'dodon', 'dudun'],
'umur': [30, 40, 26, 25],
'hobi': ['memasak', 'berkebun', 'basket', 'musik'],
'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

After

```
my_dictionary['nama'].pop()
```

```
my_dictionary
```

```
{'nama': ['dad', 'didin', 'dodon'],
'umur': [30, 40, 26, 25],
'hobi': ['memasak', 'berkebun', 'basket', 'musik'],
'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

## **Format Code 2 :** Menghapus value tertentu pada salah satu Key

```
nama_dictionary['nama_key'].remove()
```

## Format Code : Menghapus value tertentu pada salah satu Key

Before

```
my_dictionary
{'nama': ['dadam', 'didin', 'dodon', 'dudun'],
'umur': [30, 40, 26, 25],
'hobi': ['memasak', 'berkebun', 'basket', 'musik'],
'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

After

```
my_dictionary['nama'].remove('didin')
my_dictionary
{'nama': ['dadam', 'dodon', 'dudun'],
'umur': [30, 40, 26, 25],
'hobi': ['memasak', 'berkebun', 'basket', 'musik'],
'sepatu': ['nike', 'adidas', 'puma', 'rebook']}
```

# Challenge Time

Dari dictionary **customer** terakhir yang telah ditambahkan key dan value baru.

1. Ternyata data umur tidak dapat disimpan sehingga **key 'umur' harus dihapus**
2. Hapus element terakhir dan pertama dari **key 'tempat\_tinggal'**

# Challenge Time Lanjutan (I)

Buatlah dictionary awal dengan nama "esport\_player" yang memiliki keys sebagai berikut:

```
nama = ["Monoka chan", "Avroz", "IMLG.Heliportpoe", "Misha", "Si AA"]  
role = ["Angela Floryn Nana", "MM", "Professor", "EXP", "Core Indomaret"]  
kill = [100, 2, 99, 2, 5]  
death = [9999, 1, 9998, 3, 5]  
assist = [0, 15, 200, 5, 5]
```

Pertanyaannya:

1. Hitunglah **KDA** pada setiap orang, dengan rumus:  
**(Kill + Assist) / Death**, serta sebutkan siapa yang memiliki **KDA** tertinggi dan terendah?
2. Berdasarkan rata-rata dari **kill**, **death**, atau **assist**, sebutkan siapa saja yang memiliki nilai **kill**, **death**, dan **assist** yang nilainya diatas **rata-rata** pada masing-masing kolom/keys?

# Challenge Time Lanjutan (II)

Masih dengan dictionary sebelumnya, **esport\_player**, coba bantu admin untuk melakukan modifikasi data sehingga data **esport\_player** berubah menjadi seperti ini:

```
{'IPK': [3.51, 3.8, 99999, 3.54, 3.92, 4.0],  
 'KDA': [0.01, 17.0, 0.03, 2.33, 2.0, 100],  
 'nama': ['Monoka chan',  
           'Avroz',  
           'IMLG.Heliportpoe',  
           'Misha',  
           'Si AA',  
           'Supreme Leader'],  
 'role': ['Cuma Angela Aja',  
          'MM/Core',  
          'Tank/EXP',  
          'EXP Laner',  
          'Core/MM',  
          'All Role']}
```

## Outline Pembelajaran

# Dictionary and JSON



Pengenalan Tipe Data  
Dictionary



Operasi pada Dictionary



Pengenalan pada JSON



Import & Export JSON

# JSON (JavaScript Object Notation)

JSON kini menjadi standar untuk **menyimpan dan bertukar data** antar platform / aplikasi.

Salah satu alasannya adalah karena JSON format termasuk **human-readable** yang bisa diakses dengan logika sederhana

# JSON (JavaScript Object Notation)

Contoh JSON

```
{  
    "nama": "Budi Santoso",  
    "hobi": ["berlari", "membaca", "memasak"],  
    "usia": 35,  
    "anak": [  
        {  
            "nama": "Nisa Ayu",  
            "usia": 6  
        },  
        {  
            "nama": "Gusti Bagus",  
            "age": 10  
        }  
    ]  
}
```

Konsepnya mirip  
dengan Dictionary  
Python

# JSON (JavaScript Object Notation)

## Contoh JSON

```
{  
    "nama": "Budi Santoso",  
    "hobi": ["berlari", "membaca", "memasak"],  
    "usia": 35,  
    "anak": [  
        {  
            "nama": "Nisa Ayu",  
            "usia": 6  
        },  
        {  
            "nama": "Gusti Bagus",  
            "age": 10  
        }  
    ]  
}
```

- Key **anak** memiliki value berupa List,  
\*\* ditandai dengan kurung siku buka “[ “ dan tutup “ ] “
- dalamList terdapat 2 element  
\*\* ditandai tanda koma “ , “
- element pada List tersebut juga sebuah Dictionary

# JSON (JavaScript Object Notation)



**Deserialization**  
**Json → Python**



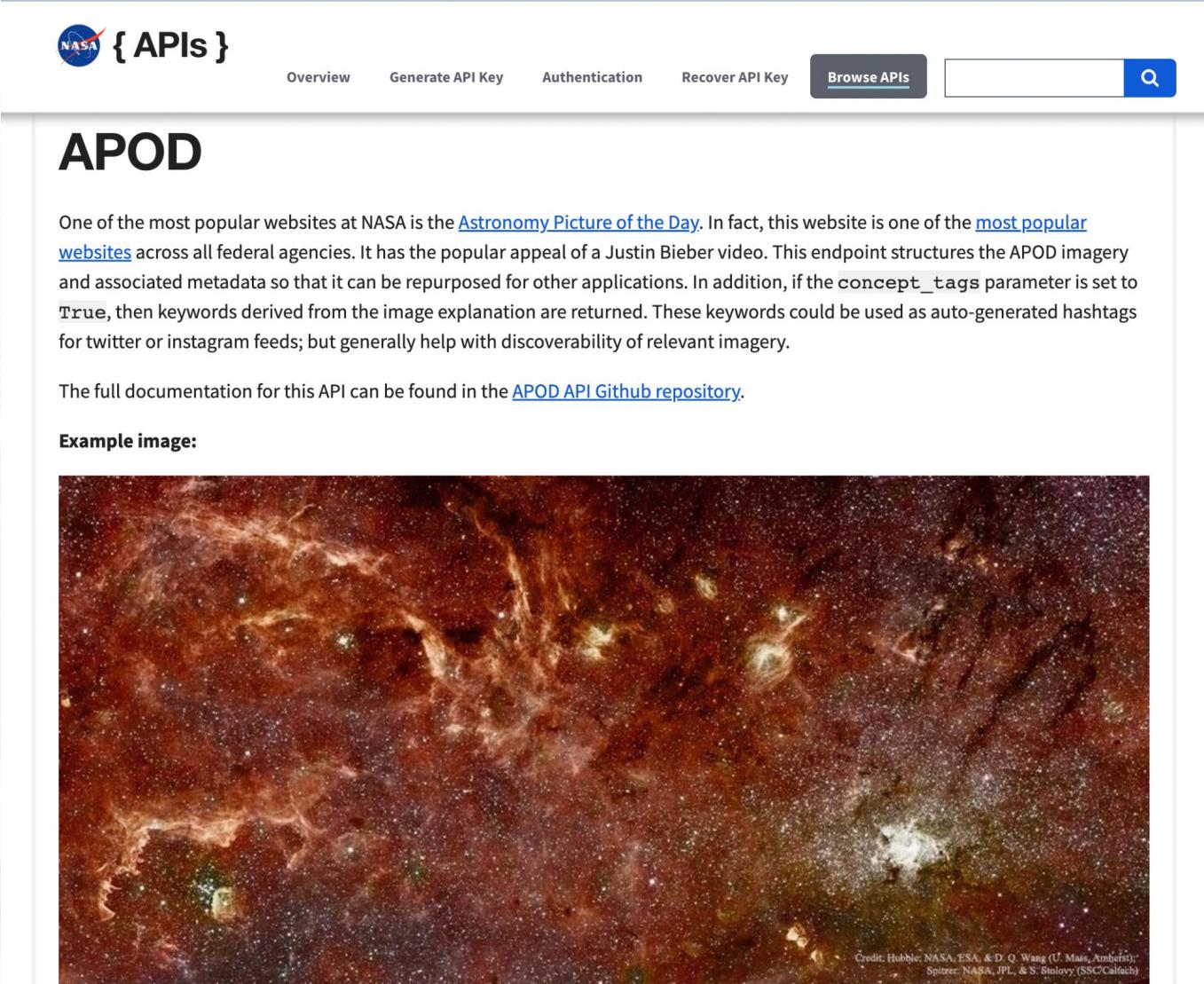
**Serialization**  
**Python → Json**

# Penggunaan JSON to Python

Pertanyaan:

Bagaimana cara mendapatkan gambar-gambar tentang astronomi yang resmi dari NASA ?

# Penggunaan JSON to Python



The screenshot shows the NASA APIs website with the APOD endpoint selected. The page title is "APOD". The content describes the APOD website as one of the most popular at NASA, mentioning its popularity and how it structures imagery and metadata. It also notes that setting the `concept_tags` parameter to `True` returns keywords from image explanations. Below this text is a link to the [APOD API Github repository](#). An example image is displayed, which is a colorful nebula.

One of the most popular websites at NASA is the [Astronomy Picture of the Day](#). In fact, this website is one of the [most popular websites](#) across all federal agencies. It has the popular appeal of a Justin Bieber video. This endpoint structures the APOD imagery and associated metadata so that it can be repurposed for other applications. In addition, if the `concept_tags` parameter is set to `True`, then keywords derived from the image explanation are returned. These keywords could be used as auto-generated hashtags for twitter or instagram feeds; but generally help with discoverability of relevant imagery.

The full documentation for this API can be found in the [APOD API Github repository](#).

**Example image:**



Credit: Hubble: NASA, ESA, & D. Q. Wang (U. Mass, Amherst).  
Spitzer: NASA, JPL, & S. Stolovy (SSC/Calech)

<https://api.nasa.gov/>

# Penggunaan JSON to Python

Solusi :

NASA memberikan kita akses untuk mendapatkan gambar nebula via API Request dengan code berikut :

```
import requests

r = requests.get('https://api.nasa.gov/planetary/apod?',
                  params={"api_key": "DEMO_KEY"})
print(r)

<Response [200]>
```

# Penggunaan JSON to Python

r.content

```
b'{"copyright":"Urs Leutenegger","date":"2022-08-11","explanation":"On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.", "hdurl":"https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg", "media_type": "image", "service_version": "v1", "title": "Perseids and MAGIC", "url": "https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg"}\n'
```

Outputnya mirip  
dengan Python  
Dictionary

# Penggunaan JSON to Python

```
# Simpan
dataset = r.content

# Ambil satu key
dataset["copyright"]
```

```
-----  
TypeError                                 Traceback (most recent call last)
/var/folders/sc/ww14556141j7kr5wc6q114340000gn/T/ipykernel_30740/3034218078.py in <module>
      3
      4 # Ambil satu key
----> 5 dataset["copyright"]

TypeError: byte indices must be integers or slices,
```

Sayangnya operasi  
datanya berbeda,  
karena memang  
bukan Dictionary

# Penggunaan JSON to Python

**Format Code** : Untuk convert JSON format ke Dictionary

```
import json
```

```
json.loads(json_data)
```

# Penggunaan JSON to Python

## Contoh Code :

```
import json

json.loads(dataset)

{'copyright': 'Urs Leutenegger',
'date': '2022-08-11',
'explanation': "On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.",
'hdurl': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg',
'media_type': 'image',
'service_version': 'v1',
'title': 'Perseids and MAGIC',
'url': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg'}
```

# Penggunaan JSON to Python

## Contoh Code :

```
import json
json.loads(dataset)

{'copyright': 'Urs Leutenegger',
 'date': '2022-08-11',
 'explanation': "On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.",
 'hdurl': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg',
 'media_type': 'image',
 'service_version': 'v1',
 'title': 'Perseids and MAGIC',
 'url': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg'}
```

\*\* Key copyright memiliki value “Urs Leutenegger”

# Penggunaan JSON to Python

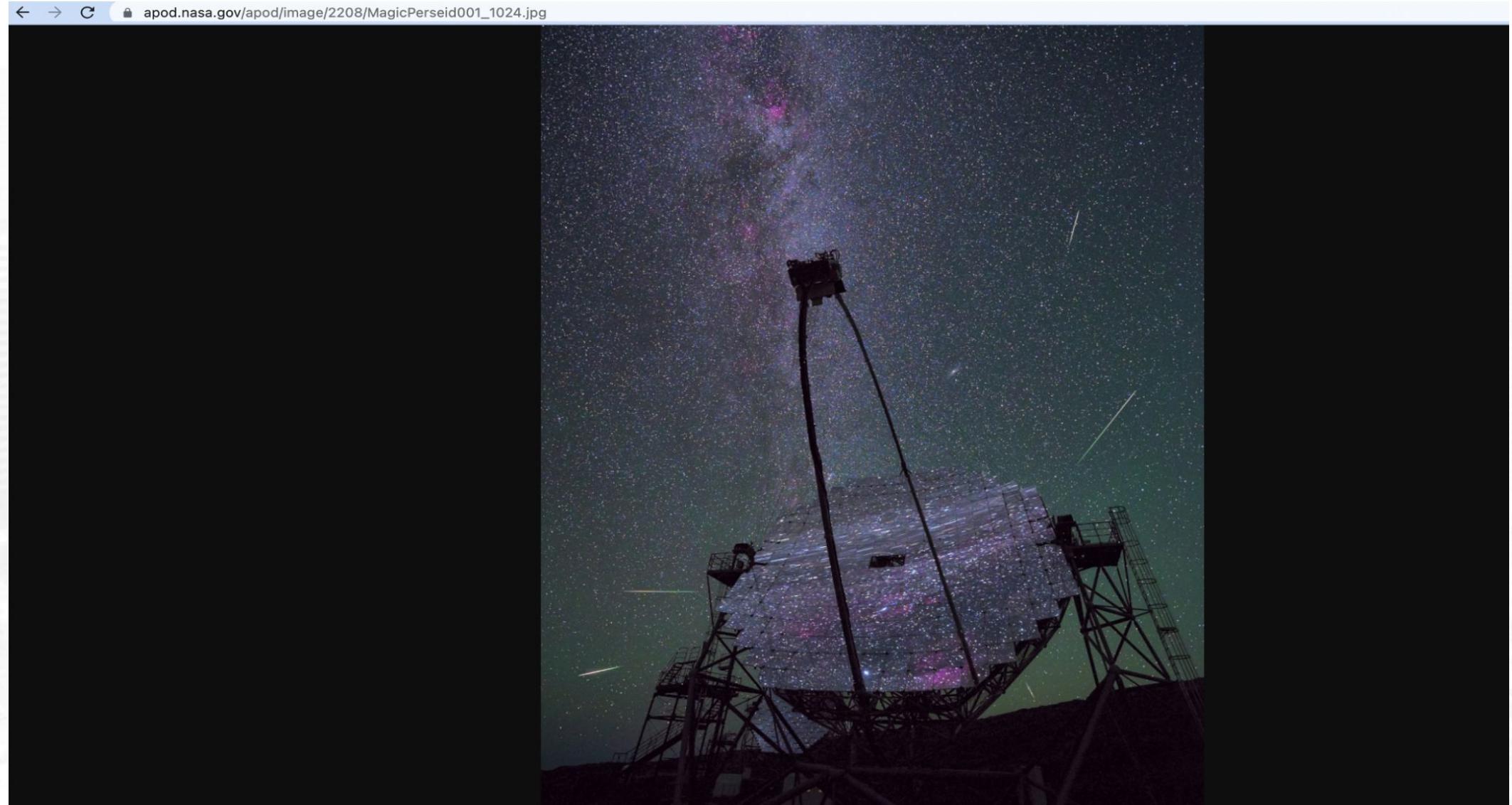
## Contoh Code :

```
import json

json.loads(dataset)

{'copyright': 'Urs Leutenegger',
'date': '2022-08-11',
'explanation': "On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.",
'hdurl': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg',
'media_type': 'image',
'service_version': 'v1',
'title': 'Perseids and MAGIC',
'url': 'https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg'}
```

\*\* Apabila key url ini kita buka valuenya di browser masing-masing, maka contohnya akan keluar seperti ini



\*Contoh value pada URL yang kita ambil via `request.get` python  
\*\*url gambar bisa berbeda-beda tergantung hasil dari `request` yg diambil

# Penggunaan JSON to Python

**Contoh Code :**

```
dataset_real = json.loads(dataset)
```

simpan hasil convert-an  
ke object yang kita namakan **dataset\_real**

# Penggunaan JSON to Python

## Contoh Code :

```
dataset_real[ "copyright" ]
```

Melihat value pada Key **copyright**

```
'Urs Leutenegger'
```

Melihat value pada Key **title**

```
dataset_real[ "title" ]
```

```
'Perseids and MAGIC'
```

# Penggunaan JSON to Python

## Contoh Code :

```
dataset_real["explanation"]
```

"On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon."

Mengingat value pada Key **explanation** berupa string,  
maka berlaku operasi pada string seperti biasa

# Penggunaan JSON to Python

## Contoh Code :

```
dataset_real["explanation"].split(".")

['On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma',
 ' MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov',
 " The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere",
 ' To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy',
 ' But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus',
 ' And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower',
 ' This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon',
 '']
```

Menggunakan fungsi `.split()` untuk memisahkan string dalam bentuk list

# Penggunaan JSON to Python

**Contoh Code 2 :** Menampilkan gambar url di notebook

```
from PIL import Image
import requests

im = Image.open(requests.get(dataset_real["url"], stream=True).raw)
im
```

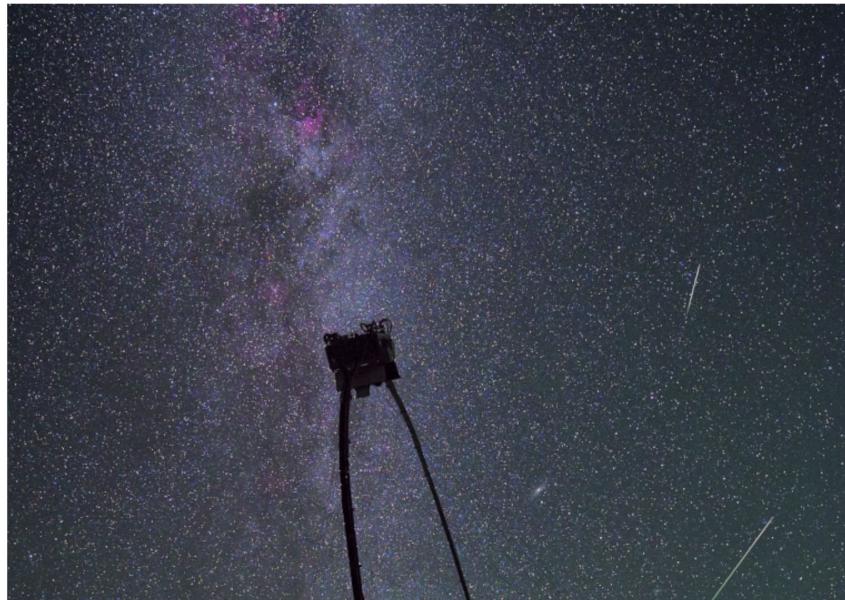
Gunakan fungsi **Image** dalam package **PIL**

# Penggunaan JSON to Python

## Contoh Code 2 : Menampilkan gambar url di notebook

```
from PIL import Image
import requests

im = Image.open(requests.get(dataset_real["url"], stream=True).raw)
im
```



Gambar akan muncul di bagian output pada cell

# Penggunaan Python to JSON

**Format Code :** Convert kembali dari Dictionary ke JSON

```
json.dumps(dictionary_data)
```

# Penggunaan Python to JSON

Contoh Code :

```
json.dumps(dataset_real)
```

dataset\_real adalah Python dictionary yang kita buat dengan meng-convert JSON to python sebelumnya.

# Penggunaan Python to JSON

Contoh Code :

```
json_data = json.dumps(dataset_real)
json_data
```

```
{"copyright": "Urs Leutenegger", "date": "2022-08-11", "explanation": "On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth's upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.", "hdurl": "https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg", "media_type": "image", "service_version": "v1", "title": "Perseids and MAGIC", "url": "https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg"}
```

# Challenge Time

Ambil isi Key '**date**' dari dataset\_real, Setelah itu, cari ubahlah penulisan format tanggal dari **YYYY-MM-DD** menjadi **DD/MM/YYYY** dengan menggunakan package **datetime**. Gunakan fungsi **strptime** dan **strftime** untuk mengconvert tipe data string -> datetime, dan datetime -> string

- \* kamu bisa mempelajari cara menggunakan package **datetime** terlebih dahulu di internet, contoh:
  - <https://www.geeksforgeeks.org/python-datetime-strptime-function/>
  - <https://www.programiz.com/python-programming/datetime/strftime>

## Outline Pembelajaran

# Dictionary and JSON



Pengenalan Tipe Data  
Dictionary



Operasi pada Dictionary



Pengenalan pada JSON



Import & Export JSON

# Export JSON : Menyimpan JSON file

**Format Code :**

```
with open( 'nama_file_saat_disimpan.json' , 'w' ) as write_file :  
    json.dump(nama_jason_di_python, write_file)
```

# Export JSON : Menyimpan JSON file

## Format Code :

```
with open( 'nama_file_saat_disimpan.json' , 'w' ) as write_file :  
    json.dump(nama_jason_di_python, write_file)
```

- pada kasus kita, namanya adalah json\_data

# Export JSON : Menyimpan JSON file

## Contoh Code :

```
▼ with open('data_NASA.json','w') as write_file :  
    json.dump(json_data, write_file)
```

# Import JSON : Mengambil JSON file dari folder kita

## Format Code :

```
with open('nama_file_di_folder.json', 'r') as read_file:  
    json_data = json.load(read_file)
```



simpan json format tersebut ke object bernama json\_data (bebas/bisa disesuaikan)

# Import JSON : Mengambil JSON file dari folder kita

## Contoh Code :

```
with open("data_NASA.json", "r") as read_file:  
    json_data = json.load(read_file)
```

```
json_data
```

```
{"copyright": "Urs Leutenegger", "date": "2022-08-11", "explanation": "On August 11, 2021 a multi-mirror, 17 meter-diameter MAGIC telescope reflected this starry night sky from the Roque de los Muchachos European Northern Observatory on the Canary Island of La Palma. MAGIC stands for Major Atmospheric Gamma Imaging Cherenkov. The telescopes can see the brief flashes of optical light produced in particle air showers as high-energy gamma rays impact the Earth\\'s upper atmosphere. To the dark-adapted eye the mirror segments offer a tantalizing reflection of stars and nebulae along the plane of our Milky Way galaxy. But directly behind the segmented mirror telescope, low on the horizon, lies the constellation Perseus. And on that date the dramatic composite nightscape also captured meteors streaming from the radiant of the annual Perseid meteor shower. This year the Perseid shower activity will again peak around August 13 but perseid meteors will have to compete with the bright light of a Full Moon.", "hdurl": "https://apod.nasa.gov/apod/image/2208/MagicPerseid001.jpg", "media_type": "image", "service_version": "v1", "title": "Perseids and MAGIC", "url": "https://apod.nasa.gov/apod/image/2208/MagicPerseid001_1024.jpg"}'
```

# Challenge Time

Budi memiliki data JSON sebagai berikut:

```
data_budi = "{  \"nama\": \"Budi Santoso\",  
              \"hobi\": [\"berlari\", \"membaca\", \"memasak\"],  
              \"usia\": 35,  
              \"anak\": [  
                {  
                  \"nama\": \"Nisa Ayu\",  
                  \"usia\": 6  
                },  
                {  
                  \"nama\": \"Gusti Bagus\",  
                  \"age\": 10  
                }  
              ]  
}"
```

Bantulah budi untuk:

1. Menyimpan JSON tersebut menjadi file "data\_budi.json"
2. Membaca file "data\_budi.json" lalu menyimpannya dengan nama variabel "data\_keluarga"



**Terima  
Kasih!**