

JSON

# Python JSON to dict

- JSON (JavaScript Object Notation) merupakan format data populer yang digunakan untuk mewakili data terstruktur.
- JSON dapat digunakan untuk mengirim dan menerima data antara server dan aplikasi web.
- Dalam Python, JSON ada sebagai string. Contoh:
  - `x = {"nama": "Budi", "hobby": ["Makan", "Tidur"]}`
- JSON objek dapat juga disimpan ke dalam file
- Untuk bekerja dengan JSON, perlu di-import modul json

# Python JSON to dictionary

- Method **json.loads()** digunakan untuk parse JSON string
- Method ini akan mengembalikan tipe dictionary

## Script

```
1 import json
2
3 mhs = '{"nama": "Budi", "hobby": ["Makan", "Tidur"]}'
4 mhs_dict = json.loads(mhs)
5
6 print(mhs_dict)
7
8 print(mhs_dict['hobby'])
```

- **mhs** adalah JSON string
- **mhs\_dict** adalah dictionary

## Hasil:

```
{'nama': 'Budi', 'hobby': ['Makan', 'Tidur']}
['Makan', 'Tidur']
```

# Python Read JSON File

- Method **json.load()** digunakan untuk membaca file yang mengandung objek JSON

## File .json

```
mhs.json x
1 {"nama": "Budi",
2  "hobby": ["Makan", "Tidur"]}
3 }
```

## Script

```
1 import json
2
3 with open('mhs.json') as f:
4     data = json.load(f)
5
6 print(data)
```

## Hasil

```
{'nama': 'Budi', 'hobby': ['Makan', 'Tidur']}
```

# Python Convert to JSON String

- Method **json.dumps()** digunakan untuk mengkonversi dictionary ke JSON string

## Script

```
1 import json
2
3 mhs_dict = {'nama': 'Budi',
4 'alamat': 'Jakarta'
5 }
6
7 mhs_json = json.dumps(mhs_dict)
8
9 print(mhs_json)
```

## Hasil

```
{"nama": "Budi", "alamat": "Jakarta"}
```

# Python Convert to JSON String

- Konversi Python Object ke JSON

Python	JSON
dict	Object
list	Array
tuple	Array
str	String
int	Number
float	Number
True	true
False	false
None	null

## Script

```
1 import json
2
3 mhs = {
4     "nama" : "Budi",
5     "umur" : 19,
6     "bekerja" : True,
7     "menikah" : False,
8     "hobby" : ("Travelling", "Gaming"),
9     "beasiswa" : None,
10    "kendaraan" : [
11        {"jenis" : "motor", "km/l" : 25.3},
12        {"jenis" : "mobil", "km/l" : 11.7},
13    ]
14 }
15
16 print(json.dumps(mhs))
```

## Hasil

```
{"nama": "Budi", "umur": 19, "bekerja": true, "menikah": false, "hobby": ["Travelling", "Gaming"], "beasiswa": null, "kendaraan": [{"jenis": "motor", "km/l": 25.3}, {"jenis": "mobil", "km/l": 11.7}]}
```

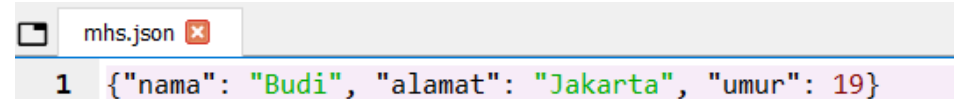
# Writing JSON to a File

- Method `json.dump()` digunakan untuk menulis ke dalam file

## Script

```
1 import json
2
3 mhs = {"nama": "Budi",
4       "alamat": "Jakarta",
5       "umur": 19}
6
7
8 with open('mhs.json', 'w') as json_file:
9     json.dump(mhs, json_file)
```

## Hasil



```
mhs.json
1 {"nama": "Budi", "alamat": "Jakarta", "umur": 19}
```

# Prettify JSON

- Untuk membuat tampilan JSON lebih rapi dapat diatur menggunakan indent dan sort\_keys

## Script

```
1 import json
2
3 mhs = '{"nim": "1911500123", "nama": "Budi", "hobby": ["Makan", "Tidur"]}'
4
5 mhs_dict = json.loads(mhs)
6
7 print(json.dumps(mhs_dict, indent = 4))
8
9 print(json.dumps(mhs_dict, indent = 4, sort_keys=True))
```

## Hasil

```
{
  "nim": "1911500123",
  "nama": "Budi",
  "hobby": [
    "Makan",
    "Tidur"
  ]
}
{
  "hobby": [
    "Makan",
    "Tidur"
  ],
  "nama": "Budi",
  "nim": "1911500123"
}
```



XML

# XML

- XML adalah singkatan dari eXtensible Markup Language.
- Dirancang untuk menyimpan dan mengirimkan data dalam jumlah kecil hingga menengah dan digunakan secara luas untuk berbagi informasi yang terstruktur.
- Python memungkinkan untuk mengurai dan memodifikasi dokumen XML dengan kelas minidom XML.

# Parse XML

- Siapkan dokumen xml sbb:

```
mhs.xml ✕
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <mhs>
3   <nama>Budi</nama>
4   <alamat>Jakarta</alamat>
5   <hobby name="Makan"/>
6   <hobby name="Tidur"/>
7   <hobby name="Gaming"/>
8 </mhs>
```

## Script

```
1 import xml.dom.minidom as minidom
2
3 def main():
4     doc = minidom.parse("mhs.xml")
5
6     print(doc.nodeName)
7     print(doc.firstChild.tagName)
8
9 if __name__ == "__main__":
10     main()
```

## Hasil

```
#document
mhs
```

# Parse XML

- Script

```
1 import xml.dom.minidom as minidom
2
3 def main():
4     doc = minidom.parse("mhs.xml")
5
6     print (doc.nodeName)
7     print (doc.firstChild.tagName)
8
9     hobby_list = doc.getElementsByTagName("hobby")
10    print("%d Hobby:" %hobby_list.length)
11    for hobi in hobby_list:
12        print (hobi.getAttribute("name"))
13
14 if __name__ == "__main__":
15     main()
```

## Hasil

```
#document
mhs
3 Hobby:
Makan
Tidur
Gaming
```

# Menambahkan Node

- Script

```
1 import xml.dom.minidom as minidom
2
3 def main():
4     doc = minidom.parse("mhs.xml")
5
6     print (doc.nodeName)
7     print (doc.firstChild.tagName)
8
9
10    hobby_baru = doc.createElement("hobby")
11    hobby_baru.setAttribute("name", "Travelling")
12    doc.firstChild.appendChild(hobby_baru)
13
14    hobby_list = doc.getElementsByTagName("hobby")
15    print("%d Hobby:" %hobby_list.length)
16    for hobi in hobby_list:
17        print (hobi.getAttribute("name"))
18
19 if __name__ == "__main__":
20     main()
```

## Hasil

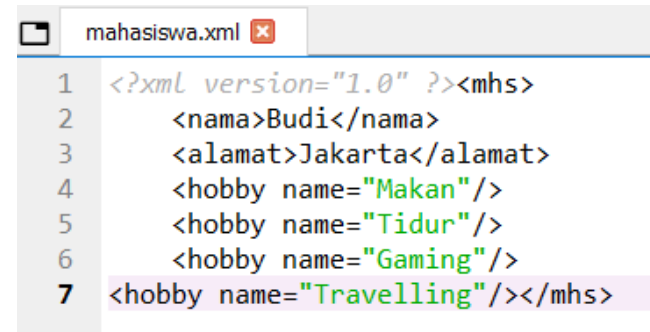
```
#document
mhs
4 Hobby:
Makan
Tidur
Gaming
Travelling
```

# Menyimpan ke File

- Script

```
1 import xml.dom.minidom as minidom
2
3 def main():
4     doc = minidom.parse("mhs.xml")
5
6     print (doc.nodeName)
7     print (doc.firstChild.tagName)
8
9
10    hobby_baru = doc.createElement("hobby")
11    hobby_baru.setAttribute("name", "Travelling")
12    doc.firstChild.appendChild(hobby_baru)
13
14    hobby_list = doc.getElementsByTagName("hobby")
15    print("%d Hobby:" %hobby_list.length)
16    for hobi in hobby_list:
17        print (hobi.getAttribute("name"))
18
19    file_xml = open("mahasiswa.xml", "w")
20    doc.writexml(file_xml)
21    file_xml.close()
22
23 if __name__ == "__main__":
24     main()
```

## Hasil



```
mahasiswa.xml x
1 <?xml version="1.0" ?><mhs>
2     <nama>Budi</nama>
3     <alamat>Jakarta</alamat>
4     <hobby name="Makan"/>
5     <hobby name="Tidur"/>
6     <hobby name="Gaming"/>
7 <hobby name="Travelling"/></mhs>
```

# Parse XML using ElementTree

- Script

```
1 import xml.etree.ElementTree as ET
2
3 tree = ET.parse('mhs.xml')
4 root = tree.getroot()
5
6 print(root.tag)
7
8 for child in root:
9     print(child.tag, child.attrib)
```

## Hasil

```
mhs
nama {}
alamat {}
hobby {'name': 'Makan'}
hobby {'name': 'Tidur'}
hobby {'name': 'Gaming'}
```