Python Pandas working with Json

###In Part we are going to learn about - How to read json data

1.read Json (read_json)

2.To Json(to_json)

3.Json normalize

In [2]:
```
data='{"employee_name":"james","email":"james@gmail.com","job_profile":[{"title
```

In [3]:
```
type(data)
```

Out[3]:
```
str
```

In [4]:
```
import pandas as pd
```

In [5]:
```
pd.read_json(data)
```

Out[5]:

	employee_name	email	job_profile
0	james	james@gmail.com	{'title1': 'Team Lead', 'title2': 'Sr. Develop...

In [6]:
```
pd.read_json(data,orient='record')
```

Out[6]:

	employee_name	email	job_profile
0	james	james@gmail.com	{'title1': 'Team Lead', 'title2': 'Sr. Develop...

In [7]:
```
pd.read_json(data,orient='index')
```

Out[7]:

	0
employee_name	james
email	james@gmail.com
job_profile	[{'title1': 'Team Lead', 'title2': 'Sr. Develo...

In [9]:
```
pd.read_json(data,orient='columns')     ## assignment for the uses of split fo
```

Out[9]:

	employee_name	email	job_profile
0	james	james@gmail.com	{'title1': 'Team Lead', 'title2': 'Sr. Develop...

In [ ]:
```

```

In [10]:
```
df=pd.DataFrame([['a','b'],['c','d']],
                  index=['row1','row2'],
                   columns=['col1','col2'])
```

In [11]:
```
df
```

Out[11]:

	col1	col2
row1	a	b
row2	c	d

In [12]:
```
df.to_json()
```

Out[12]:
```
'{"col1":{"row1":"a","row2":"c"},"col2":{"row1":"b","row2":"d"}}'
```

In [13]:
```
df.to_json(orient='index')
```

Out[13]:
```
'{"row1":{"col1":"a","col2":"b"},"row2":{"col1":"c","col2":"d"}}'
```

In [15]:
```
df.to_json(orient='columns')
```

Out[15]:
```
'{"col1":{"row1":"a","row2":"c"},"col2":{"row1":"b","row2":"d"}}'
```

In [16]:
```
df.to_json(orient='records')
```

Out[16]:
```
'[{"col1":"a","col2":"b"},{"col1":"c","col2":"d"}]'
```

In [17]:
```
df.to_json(orient='split')
```

Out[17]:
```
'{"columns":["col1","col2"],"index":["row1","row2"],"data":[["a","b"],
["c","d"]]}'
```

In [18]:
```
df.to_json(orient='table')
```

Out[18]:
```
'{"schema":{"fields":[{"name":"index","type":"string"},{"name":"col1","typ
e":"string"},{"name":"col2","type":"string"}],"primaryKey":["index"],"pandas_
version":"1.4.0"},"data":[{"index":"row1","col1":"a","col2":"b"},{"index":"ro
w2","col1":"c","col2":"d"}]}'
```

In [19]:
```
schema='{"schema":{"fields":[{"name":"index","type":"string"},{"name":"col1","
```

In [21]:
```
pd.read_json(schema,orient='table')
```

Out[21]:

	col1	col2
row1	a	b
row2	c	d

In [31]:
```
data=[{"employee_name":"james","email":"james@gmail.com","job_profile":{"title
```

In [32]:
```
type(data)
```

Out[32]:
```
list
```

In [35]:
```
pd.json_normalize(data)
```

Out[35]:

	employee_name	email	job_profile.title1	job_profile.title2
0	james	james@gmail.com	Team Lead	Sr. Developer

In [36]:
```
data=[
    {
        "id":1,
        "name":"cole volk",
        "fitness":{"height":130,"weight":60},
    },
    {"name":"Mark Reg","fitness":{"height":130,"weight":60}},
    {
        "id":2,
        "name":"Faye Raker",
        "fitness":{"height":130,"weight":60},
    }
]
```

In [37]:
```
pd.json_normalize(data)
```

Out[37]:

	id	name	fitness.height	fitness.weight
0	1.0	cole volk	130	60
1	NaN	Mark Reg	130	60
2	2.0	Faye Raker	130	60

In [38]:
```
# Max level parameter
pd.json_normalize(data,max_level=0)
```

Out[38]:

	id	name	fitness
0	1.0	cole volk	{'height': 130, 'weight': 60}
1	NaN	Mark Reg	{'height': 130, 'weight': 60}
2	2.0	Faye Raker	{'height': 130, 'weight': 60}

In [39]:
```
pd.json_normalize(data,max_level=1)
```

Out[39]:

	id	name	fitness.height	fitness.weight
0	1.0	cole volk	130	60
1	NaN	Mark Reg	130	60
2	2.0	Faye Raker	130	60

In [40]:
```
data = [
    {
        "state": "Florida",
        "shortname": "FL",
        "info": {"governor": "Rick Scott"},
        "counties": [
            {"name": "Dade", "population": 12345},
            {"name": "Broward", "population": 40000},
            {"name": "Palm Beach", "population": 60000},
        ],
    },
    {
        "state": "Ohio",
        "shortname": "OH",
        "info": {"governor": "John Kasich"},
        "counties": [
            {"name": "Summit", "population": 1234},
            {"name": "Cuyahoga", "population": 1337},
        ],
    },
]
```

In [41]:
```
type(data)
```

Out[41]:
```
list
```

In [42]:
```
pd.json_normalize(data)
```

Out[42]:

	state	shortname	counties	info.governor
0	Florida	FL	[{'name': 'Dade', 'population': 12345}, {'name...	Rick Scott
1	Ohio	OH	[{'name': 'Summit', 'population': 1234}, {'nam...	John Kasich

In [45]:
```
pd.json_normalize(data,"counties",["state","shortname",["info","governor"]])
```

Out[45]:

	name	population	state	shortname	info.governor
0	Dade	12345	Florida	FL	Rick Scott
1	Broward	40000	Florida	FL	Rick Scott
2	Palm Beach	60000	Florida	FL	Rick Scott
3	Summit	1234	Ohio	OH	John Kasich
4	Cuyahoga	1337	Ohio	OH	John Kasich

In [ ]:
```

```