In [2]:

```
import pandas

mydataset = {
    'cars': ["bmw","volvo","buggati"],
    'passings':[3,6,2]
}

mrcars = pandas.DataFrame(mydataset)
print(mrcars)
```

```
      cars  passings
0      bmw         3
1    volvo         6
2  buggati         2
```

In [17]:

```
import pandas
r = pandas.read_json("E:\\hm.json")
print(r.to_string())
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7096/96362869.py in <module>
      1 import pandas
----> 2 r = pandas.read_json("E:\\hm.json")
      3 print(r.to_string())

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kw
args)
    205                 else:
    206                     kwargs[new_arg_name] = new_arg_value
--> 207             return func(*args, **kwargs)
    208 
    209         return cast(F, wrapper)

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kw
args)
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312 
    313         return wrapper

~\anaconda3\lib\site-packages\pandas\io\json\_json.py in read_json(path_or_buf,
orient, typ, dtype, convert_axes, convert_dates, keep_default_dates, numpy, pre
cise_float, date_unit, encoding, encoding_errors, lines, chunksize, compressio
n, nrows, storage_options)
    612 
    613     with json_reader:
--> 614         return json_reader.read()
    615 
    616 

~\anaconda3\lib\site-packages\pandas\io\json\_json.py in read(self)
    746                 obj = self._get_object_parser(self._combine_lines(data_
lines))
    747         else:
--> 748             obj = self._get_object_parser(self.data)
    749         self.close()
    750         return obj

~\anaconda3\lib\site-packages\pandas\io\json\_json.py in _get_object_parser(sel
f, json)
    768         obj = None
    769         if typ == "frame":
--> 770             obj = FrameParser(json, **kwargs).parse()
    771 
    772         if typ == "series" or obj is None:

~\anaconda3\lib\site-packages\pandas\io\json\_json.py in parse(self)
    883 
    884         else:
--> 885             self._parse_no_numpy()
```

```
    886 
    887         if self.obj is None:

~\anaconda3\lib\site-packages\pandas\io\json\_json.py in _parse_no_numpy(self)
   1138         if orient == "columns":
   1139             self.obj = DataFrame(
-> 1140                 loads(json, precise_float=self.precise_float), dtype=No
ne
   1141             )
   1142         elif orient == "split":

ValueError: Expected object or value
```

In [8]:

```
import pandas as fam
family = {
    'father':['tuuu','30','male','5435345353'],
    'mother':['eyuu','30','female','6736737373'],
    'bro':['tuu','10','male','784384848'],
    'sister':['euu','10','female','737373737']
}
f=fam.DataFrame(family)
print(f)
print(fam.__version__)
```

```
       father      mother        bro     sister
0        tuuu        eyuu        tuu        euu
1          30          30         10         10
2        male      female       male     female
3  5435345353  6736737373  784384848  737373737
1.3.4
```

In [12]:

```
import pandas as pd

a = [1, 7, 2,6,9,9,0.6]

f = pd.Series(a)

print(f)
```

```
0    1.0
1    7.0
2    2.0
3    6.0
4    9.0
5    9.0
6    0.6
dtype: float64
```

In [13]:

```
import pandas as pd

calories = {"day1": 420, "day2": 380, "day3": 390}

m = pd.Series(calories)

print(m)
```

```
day1    420
day2    380
day3    390
dtype: int64
```

In [ ]: