

```
import numpy as np
import pandas as pd
list=[[1,'Smith',50000],[2,'Jones',60000]]
```

```
df=pd.DataFrame(list)
df
```

	0	1	2
0	1	Smith	50000
1	2	Jones	60000

```
df.columns=['Empd','Name','Salary']
df
```

	Empd	Name	Salary
0	1	Smith	50000
1	2	Jones	60000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2 entries, 0 to 1
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  -
0   Empd     2 non-null      int64
1   Name     2 non-null      object
2   Salary   2 non-null      int64
dtypes: int64(2), object(1)
memory usage: 176.0+ bytes
```

```
df=pd.read_csv("50_Startups.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
#   Column              Non-Null Count  Dtype
---  -
0   R&D Spend           50 non-null     float64
1   Administration      50 non-null     float64
2   Marketing Spend     50 non-null     float64
3   State               50 non-null     object
4   Profit              50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

```
df.head()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83

1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

```
df.tail()
```

	R&D Spend	Administration	Marketing Spend	State	Profit
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

```
import numpy as np
import pandas as pd
df=pd.read_csv("employee.csv")
```

```
-----
-----
FileNotFoundError                                Traceback (most recent call
last)
```

```
~\AppData\Local\Temp\ipykernel_42792\3506309008.py in <module>
```

```
1 import numpy as np
2 import pandas as pd
```

```
----> 3 df=pd.read_csv("employee.csv")
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-
packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
```

```
309         stacklevel=stacklevel,
310     )
```

```
--> 311         return func(*args, **kwargs)
```

```
312
```

```
313         return wrapper
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-
packages\pandas\io\parsers\readers.py in read_csv(filepath_or_buffer,
sep, delimiter, header, names, index_col, usecols, squeeze, prefix,
mangle_dupe_cols, dtype, engine, converters, true_values,
false_values, skipinitialspace, skiprows, skipfooter, nrows,
na_values, keep_default_na, na_filter, verbose, skip_blank_lines,
parse_dates, infer_datetime_format, keep_date_col, date_parser,
dayfirst, cache_dates, iterator, chunksize, compression, thousands,
decimal, lineterminator, quotechar, quoting, doublequote, escapechar,
comment, encoding, encoding_errors, dialect, error_bad_lines,
warn_bad_lines, on_bad_lines, delim_whitespace, low_memory,
memory_map, float_precision, storage_options)
```

```
584     kwds.update(kwds_defaults)
```

```
585
```

```
--> 586     return _read(filepath_or_buffer, kwds)
```

587  
588

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-  
packages\pandas\io\parsers\readers.py in _read(filepath_or_buffer,  
kwds)
```

```
480  
481     # Create the parser.  
--> 482     parser = TextFileReader(filepath_or_buffer, **kwds)  
483  
484     if chunksize or iterator:
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-  
packages\pandas\io\parsers\readers.py in __init__(self, f, engine,  
**kwds)
```

```
809         self.options["has_index_names"] =  
kwds["has_index_names"]  
810  
--> 811         self._engine = self._make_engine(self.engine)  
812  
813     def close(self):
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-  
packages\pandas\io\parsers\readers.py in _make_engine(self, engine)
```

```
1038         )  
1039         # error: Too many arguments for "ParserBase"  
-> 1040         return mapping[engine](self.f, **self.options) #  
type: ignore[call-arg]  
1041  
1042     def _failover_to_python(self):
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-  
packages\pandas\io\parsers\c_parser_wrapper.py in __init__(self, src,  
**kwds)
```

```
49  
50     # open handles  
--> 51     self._open_handles(src, kwds)  
52     assert self.handles is not None  
53
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-  
packages\pandas\io\parsers\base_parser.py in _open_handles(self, src,  
kwds)
```

```
227         memory_map=kwds.get("memory_map", False),  
228         storage_options=kwds.get("storage_options", None),  
--> 229         errors=kwds.get("encoding_errors", "strict"),  
230     )  
231
```

```
c:\users\asus\appdata\local\programs\python\python37\lib\site-
```

```
packages\pandas\io\common.py in get_handle(path_or_buf, mode,
encoding, compression, memory_map, is_text, errors, storage_options)
    705             encoding=ioargs.encoding,
    706             errors=errors,
--> 707             newline="",
    708         )
    709     else:
```

FileNotFoundError: [Errno 2] No such file or directory: 'employee.csv'