In [3]:
```
import pandas as pd
data = {
    "calories": [420,380,390],
    "duration":[50,40,45]
}
#load data into a DATAFRAME object:
df=pd.DataFrame(data)

print(df)
```

```
   calories  duration
0       420        50
1       380        40
2       390        45
```

In [5]:
```
#locate row
#dataframe is like a table with rows and columns
#pandas use the loc attribute to return one or more specified row

#refer to the row index:
print(df.loc[0])
```

```
calories    420
duration     50
Name: 0, dtype: int64
```

In [6]:
```
#retun row 0 and 1
print(df.loc[[0, 1]])
#nb- when using [], the result is a pandas dataframe
```

```
   calories  duration
0       420        50
1       380        40
```

In [8]:
```
#NAMED INDEX-with the index argument you can name your own indexes

import pandas as pd
data = {
    "calories":[420,380,390],
    "duration": [50,40,45]
}
df= pd.DataFrame(data, index = ["day1","day2", "day3"])
print(df)
```

```
      calories  duration
day1       420        50
day2       380        40
day3       390        45
```

In [10]:
```
#locate named indexes
#use the named index in the loc attribute to return the specified row
print(df.loc["day2"])
```

```
calories    380
duration     40
Name: day2, dtype: int64
```

In [11]:
```
#       PANDAS READS CSV FILES
```

In [13]:
```
import pandas as pd

df = pd.read_csv('data.csv')

print(df.to_string())
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
C:\Users\BERYLA~1\AppData\Local\Temp/ipykernel_4276/956099774.py in <module>
      1 import pandas as pd
      2
----> 3 df = pd.read_csv('data.csv')
      4
      5 print(df.to_string())

C:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\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:\ProgramData\Anaconda3\lib\site-packages\pandas\io\parsers\base_parser.py in _open_handles(self, src, kwds)
    220         Let the readers open IOHandles after they are done with their potential raises.
    221         """
--> 222         self.handles = get_handle(
    223             src,
    224             "r",

C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    700         if ioargs.encoding and "b" not in ioargs.mode:
    701             # Encoding
--> 702             handle = open(
    703                 handle,
    704                 ioargs.mode,

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

In [14]:
```

```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\BERYLA~1\AppData\Local\Temp/ipykernel_4276/3826719451.py in <module>
      1 import pandas as pd
----> 2 df =pd.read__csv('data.csv')
      3 print(df.to_string)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\__init__.py in __getattr__(name)
    242         return _SparseArray
    243
--> 244     raise AttributeError(f"module 'pandas' has no attribute '{name}'")
    245
    246

AttributeError: module 'pandas' has no attribute 'read__csv'
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```