In [8]:
```
import pandas
mydataset ={
    'cars':["BMW", "Volvo"," Ford"],
    'passings':[3,7,2]
}
myvar= pandas.DataFrame(mydataset)

print(myvar)
```

```
  cars  passings
0   BMW         3
1 Volvo         7
2  Ford         2
```

In [10]:
```
import pandas as pd
mydataset = {
    "cars":["BMW","Volvo","Ford"],
    "passings":[3,7,2]
}
myvar = pd.DataFrame(mydataset)
print(myvar)
```

```
  cars  passings
0   BMW         3
1 Volvo         7
2  Ford         2
```

In [11]:
```
#series-pandas series are column in a table
import pandas as pd
a=[1,7,2]
myvar = pd.Series(a)
print(myvar)
```

```
0    1
1    7
2    2
dtype: int64
```

In [13]:
```
#labelsif nothing else is specified the values are labeled with the index numbers as shown above
print(myvar)
```

```
0    1
1    7
2    2
dtype: int64
```

In [19]:
```
#with the index argument you can name your own labels
import pandas as pd
a=[1,7,2]
myvar = pd.Series(a, index ={"x","y","z"})
print(myvar)
```

```
x    1
y    7
z    2
dtype: int64
```

In [20]:
```
import pandas as pd
calories = {"day1":420,"day2":380, "day3":390}
myvar = pd.Series(calories)
print(myvar)
```

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

In [21]:
```
import pandas as pd
calories = {"day1":420, "day2":380, "day3":390}
myvar = pd.Series(calories, index = ["day1","day2"])
print(myvar)
```

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

In [23]:
```
#datasets in pandas are usually multi-dimensional tables called DataFrames
import pandas as pd
data = {
    "calories":[420, 380, 390],
    "duration":[50,40,45]
}

myvar = pd.DataFrame(data)
print(myvar)
```

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

In [ ]:
```
#PANDA DATAFRAMES

#is a 2 dimensional data structure like 2dimensional array
```

In [29]:
```
import pandas as pd
data = {
    "calories":[420,380,390],
    "duration":[50,40,45]
}

df = pd.DataFrame(data)
print(df)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\BERYLA-1\AppData\Local\Temp/ipykernel_12956/4164611157.py in <module>
      5 }
      6
----> 7 df = pd.DataFrame(data)
      8 print(df)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\frame.py in __init__(self, data, index, columns, dtype, copy)
    612             elif isinstance(data, dict):
    613                 # GH#38939 de facto copy defaults to False only in non-dict cases
--> 614                 mgr = dict_to_mgr(data, index, columns, dtype=dtype, copy=copy, typ=manager)
    615             elif isinstance(data, ma.MaskedArray):
    616                 import numpy.ma.mrecords as mrecords

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in dict_to_mgr(data, index, columns, dtype, typ, copy)
    462         # TODO: can we get rid of the dt64tz special case above?
    463         return arrays_to_mgr(
--> 464             arrays, data_names, index, columns, dtype=dtype, typ=typ, consolidate=copy
    465         )
    466

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in arrays_to_mgr(arrays, arr_names, index, columns, dtype, verify_integrity, typ, consolidate)
    117         # figure out the index, if necessary
    118         if index is None:
--> 119             index = _extract_index(arrays)
    120         else:
    121             index = ensure_index(index)

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\internals\construction.py in _extract_index(data)
    633             lengths = list(set(raw_lengths))
    634             if len(lengths) > 1:
--> 635                 raise ValueError("All arrays must be of the same length")
    636
    637             if have_dicts:

ValueError: All arrays must be of the same length
```

In [31]:
```
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 [32]:
```
#use the name index in the loc attribute to return the specified rows

print(df.loc["day2"])
```

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

In [39]:
```
#Load a comma separated file (CSV file) into a DataFrame
import pandas as pd

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

print(df)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
C:\Users\BERYLA-1\AppData\Local\Temp/ipykernel_12956/3951487143.py in <module>
      2 import pandas as pd
      3
----> 4 df = pd.read_csv('data.csv')
      5
      6 print(df)

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             mapping[engine](self.f, **self.options)  # type: ignore[call-arg]
   1041         except Exception:
   1042             self.close()

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 [41]:
```
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_12956/2600213549.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             mapping[engine](self.f, **self.options)  # type: ignore[call-arg]
   1041         except Exception:
   1042             self.close()

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 [ ]:
```

```

In [ ]:
```

```