PANDAS Kütüphanesi kullanılarak Dosya Türlerinin Okutulması:

read_excel : Excel XLS veya XLSX dosya tipindeki verileri tablo şeklinde okur
read_html: HTML belgesinde bulunan tüm tabloları okur
read_json: Bir JSON (JavaScript Object Notation) dize gösteriminden verileri okur.
read_sql: Bir SQL sorgusunun sonuçlarını veri şablonu olarak okur
read_csv: csv ve txt uzantılı dosyaları okur. Varsayılan ayırıcı virgül olarak kullanılan ve dosyadan, URL den ayrılmış verileri okur.

In [2]:
```
import pandas as pd
```

In [3]:
```
df = pd.read_csv('/Users/nerdem/Downloads/Mobiles_Dataset.csv')
df.head()
```

```
---------------------------------------------------------------------------
UnicodeDecodeError                        Traceback (most recent call last)
Cell In[3], line 1
----> 1 df = pd.read_csv('/Users/nerdem/Downloads/Mobiles_Dataset.csv')
      2 df.head()

File /opt/anaconda3/lib/python3.12/site-packages/pandas/io/parsers/readers.py:1026, in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, 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, date_format, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage_options, dtype_backend)
   1013 kwds_defaults = _refine_defaults_read(
   1014     dialect,
   1015     delimiter,
   (...)
   1022     dtype_backend=dtype_backend,
   1023 )
   1024 kwds.update(kwds_defaults)
-> 1026 return _read(filepath_or_buffer, kwds)

File /opt/anaconda3/lib/python3.12/site-packages/pandas/io/parsers/readers.py:620, in _read(filepath_or_buffer, kwds)
    617 _validate_names(kwds.get("names", None))
    619 # Create the parser.
--> 620 parser = TextFileReader(filepath_or_buffer, **kwds)
    622 if chunksize or iterator:
    623     return parser
```

```
File /opt/anaconda3/lib/python3.12/site-packages/pandas/io/parsers/readers.py:1620, in TextFileReader.__init__(self, f, engine, **kwds)
   1617     self.options["has_index_names"] = kwds["has_index_names"]
   1619 self.handles: IOHandles | None = None
-> 1620 self._engine = self._make_engine(f, self.engine)

File /opt/anaconda3/lib/python3.12/site-packages/pandas/io/parsers/readers.py:1898, in TextFileReader._make_engine(self, f, engine)
   1895     raise ValueError(msg)
   1897 try:
-> 1898     return mapping[engine](f, **self.options)
   1899 except Exception:
   1900     if self.handles is not None:

File /opt/anaconda3/lib/python3.12/site-packages/pandas/io/parsers/c_parser_wrapper.py:93, in CParserWrapper.__init__(self, src, **kwds)
     90 if kwds["dtype_backend"] == "pyarrow":
     91     # Fail here loudly instead of in cython after reading
     92     import_optional_dependency("pyarrow")
---> 93 self._reader = parsers.TextReader(src, **kwds)
     95 self.unnamed_cols = self._reader.unnamed_cols
     97 # error: Cannot determine type of 'names'

File parsers.pyx:574, in pandas._libs.parsers.TextReader.__cinit__()

File parsers.pyx:663, in pandas._libs.parsers.TextReader._get_header()

File parsers.pyx:874, in pandas._libs.parsers.TextReader._tokenize_rows()

File parsers.pyx:891, in pandas._libs.parsers.TextReader._check_tokenize_status()

File parsers.pyx:2053, in pandas._libs.parsers.raise_parser_error()

File <frozen codecs>:322, in decode(self, input, final)

UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa5 in position 139946: invalid start byte
```

In [13]:
```
pip install chardet
```

```
Requirement already satisfied: chardet in /opt/anaconda3/lib/python3.12/site-packages (4.0.0)
Note: you may need to restart the kernel to use updated packages.
```

In [17]:
```
import chardet
```

In [19]:
```
with open('/Users/nerdem/Downloads/Mobiles_Dataset.csv', 'rb') as f
    result = chardet.detect(f.read())
    encoding_detected = result['encoding']

print("Detected Encoding:", encoding_detected)
```

```
df = pd.read_csv('/Users/nerdem/Downloads/Mobiles_Dataset.csv', enc
```

```
Detected Encoding: ISO-8859-1
```

In [21]:
```
df = pd.read_csv('/Users/nerdem/Downloads/Mobiles_Dataset.csv', enc
```

In [23]:
```
df.head()
```

Out[23]:

	Company Name	Model Name	Mobile Weight	RAM	Front Camera	Back Camera	Processor	Battery Capacity
0	Apple	iPhone 16 128GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
1	Apple	iPhone 16 256GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
2	Apple	iPhone 16 512GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
3	Apple	iPhone 16 Plus 128GB	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh
4	Apple	iPhone 16 Plus 256GB	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh

In [25]:
```
with open('/Users/nerdem/Downloads/Mobiles_Dataset.csv', 'rb') as f
    data = f.read()  # Dosyanın tüm baytlarını oku

# 0xA5 içeren kısımları göster
indexes = [i for i, b in enumerate(data) if b == 0xA5]
print(f"0xA5 byte şu pozisyonlarda bulundu: {indexes}")
```

```
0xA5 byte şu pozisyonlarda bulundu: [139946]
```

In [27]:
```
# CSV dosyasını ISO-8859-1 ile okuma
with open('/Users/nerdem/Downloads/Mobiles_Dataset.csv', 'r', encod
    # Her satır için
    for i, line in enumerate(f, start=1):
        # Satırdaki her kolon
        columns = line.split(',')  # Virgül ile ayır
        for j, value in enumerate(columns, start=1):
            if '¥' in value:  # 0xA5 baytı genellikle '¥' olarak gö
                print(f"0xA5 baytı {i}. satır, {j}. sütunda bulunuy
                break
```

```
0xA5 baytı 929. satır, 15. sütunda bulunuyor.
```

In [29]:
```
df.RAM.head()
```

Out[29]:
```
0    6GB
1    6GB
2    6GB
3    6GB
4    6GB
Name: RAM, dtype: object
```

In [45]:
```
df[['Model Name','Front Camera']].head()
```

Out[45]:

	Model Name	Front Camera
0	iPhone 16 128GB	12MP
1	iPhone 16 256GB	12MP
2	iPhone 16 512GB	12MP
3	iPhone 16 Plus 128GB	12MP
4	iPhone 16 Plus 256GB	12MP

Dosya Dönüştürme: Pandas kütüphanesi yardımıyla dosyalar arasında istenilen şekilde dönüşüm yapılabilir. Bazen verilerin dosyalarda veya veri tabanlarında depolanma şekli uygulama için doğru formatta olmayabilir. Örneğin Json formatındaki bir dosya csv formatına dönüştürülmek istenebilir.

In [58]:
```
# CSV dosyasını oku
df = pd.read_csv('/Users/nerdem/Downloads/Mobiles_Dataset.csv', enc

# Excel dosyası olarak kaydet
df.to_excel('/Users/nerdem/Downloads/Mobiles_Dataset.xlsx', index=Fa


print("Dönüştürme işlemi tamamlandı!")
```

```
Dönüştürme işlemi tamamlandı!
```

In [ ]:
```
```

In [64]:
```
df=pd.read_excel('/Users/nerdem/Downloads/Mobiles_Dataset.xlsx')
df.head()
```

Out[64]:

	Company Name	Model Name	Mobile Weight	RAM	Front Camera	Back Camera	Processor	Battery Capacity
0	Apple	iPhone 16 128GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
1	Apple	iPhone 16 256GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
2	Apple	iPhone 16 512GB	174g	6GB	12MP	48MP	A17 Bionic	3,600mAh
3	Apple	iPhone 16 Plus 128GB	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh
4	Apple	iPhone 16 Plus 256GB	203g	6GB	12MP	48MP	A17 Bionic	4,200mAh

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: