

```

{
  "metadata": {
    "kernel_spec": {
      "name": "python",
      "display_name": "Python (Pyodide)",
      "language": "python"
    },
    "language_info": {
      "codemirror_mode": {
        "name": "python",
        "version": 3
      },
      "file_extension": ".py",
      "mimetype": "text/x-python",
      "name": "python",
      "nbconvert_exporter": "python",
      "pygments_lexer": "ipython3",
      "version": "3.8"
    }
  },
  "nbformat_minor": 4,
  "nbformat": 4,
  "cells": [
    {
      "cell_type": "code",
      "source": "#Extract Stock Data",
      "metadata": {
        "trusted": true
      },
      "outputs": [],
      "execution_count": null
    },
    {
      "cell_type": "code",
      "source": "import yfinance as yf\n\n# Extract Tesla stock data\nntesla_data = yf.download('TSLA', start='2020-01-01', end='2024-01-01')\n\n# Reset the index, save, and display the first five rows\nntesla_data.reset_index(inplace=True)\nntesla_data.head()",
      "metadata": {
        "trusted": true
      },
      "outputs": [],
      "execution_count": null
    },
    {
      "cell_type": "code",
      "source": "import requests\nfrom bs4 import BeautifulSoup\nimport pandas as pd\n\n# Scrape revenue data\nurl = 'URL_TO_TESLA_REVENUE_PAGE'\nresponse = requests.get(url)\nsoup = BeautifulSoup(response.text, 'html.parser')\n\n# Extract revenue data",
      "metadata": {
        "trusted": true
      },
      "outputs": [],
      "execution_count": null
    }
  ]
}

```

```

and create a DataFrame (modify based on the page structure)
\nrevenue_data = []\nfor row in soup.find_all('tr'):\n    columns =
row.find_all('td')\n    if columns:\n
revenue_data.append([col.text for col in columns])\n\n# Create a
DataFrame\ntesla_revenue = pd.DataFrame(revenue_data, columns=['Year',
'Revenue'])\ntesla_revenue.tail() # Display the last five rows\n",
    "metadata": {
        "trusted": true
    },
    "outputs": [],
    "execution_count": null
},
{
    "cell_type": "code",
    "source": "# Extract GameStop stock data\ngme_data =
yf.download('GME', start='2020-01-01', end='2024-01-01')\n\n# Reset
the index, save, and display the first five
rows\ngme_data.reset_index(inplace=True)\ngme_data.head()",
    "metadata": {
        "trusted": true
    },
    "outputs": [],
    "execution_count": null
},
{
    "cell_type": "code",
    "source": "# Scrape GameStop revenue data (similarly to Tesla)
\nurl = 'URL_TO_GME_REVENUE_PAGE'\nresponse = requests.get(url)\nsoup
= BeautifulSoup(response.text, 'html.parser')\n\n# Extract revenue
data and create a DataFrame\nrevenue_data = []\nfor row in
soup.find_all('tr'):\n    columns = row.find_all('td')\n    if
columns:\n        revenue_data.append([col.text for col in columns])
\n\n# Create a DataFrame\ngme_revenue = pd.DataFrame(revenue_data,
columns=['Year', 'Revenue'])\ngme_revenue.tail() # Display the last
five rows",
    "metadata": {
        "trusted": true
    },
    "outputs": [],
    "execution_count": null
},
{
    "cell_type": "code",
    "source": "import matplotlib.pyplot as plt\n\n# Plot Tesla stock
data\ndef make_graph(data, title):\n    plt.figure(figsize=(10, 6))\n
plt.plot(data['Date'], data['Close'], label='Close Price')\n
plt.title(title)\n    plt.xlabel('Date')\n    plt.ylabel('Stock
Price')\n    plt.grid(True)\n    plt.legend()\n    plt.show()
\n\nmake_graph(tesla_data, 'Tesla Stock Price')\n",
    "metadata": {

```

```
    "trusted": true
  },
  "outputs": [],
  "execution_count": null
},
{
  "cell_type": "code",
  "source": "# Plot GameStop stock data\nmake_graph(gme_data,\n'GameStop Stock Price')",
  "metadata": {
    "trusted": true
  },
  "outputs": [],
  "execution_count": null
}
]
```