Data Migration - Round 2

Team TheThree - Task 2

a) HTML to RST parser

- We implemented a class `MyHTMLParser` by inheriting from the `html.parser.HTMLParser` and overridden its handler methods to carry out the desired behavior when start tags, end tags, and data are encountered.
- List of tags that our parser can handle:
    - <body></body>
    - <div></div>
    - <span></span>
    - <p></p>
    - <em></em>, <b></b>, <code></code>
    - <br>, <a></a>
    - <ul>, <ol>, <li>
- Example:

```
text = '<ol><li>Coffee<ul><li>Arabica</li><li>Robusta</li></ul></li>
<li>Tea</li><li>Milk</li></ol>'
parser = MyHTMLParser()
parser.feed(text)
parser.get_rst()

### Output:
#
#
# #. Coffee
#
#     * Arabica
#     * Robusta
#
# #. Tea
# #. Milk
#
#
```

- Reference: python-html2rest

b) JSON to RST parser

- We implemented a class RstBuilder which have methods to write components to file
- Here is our implement

```
  class RstBuilder:
  def **init**(self, out: typing.TextIO = sys.stdout) -> None:
  self.\_out = out

      def _add(self, content: str) -> None:
          self._out.write(content + "\n")

      def title(self, title: str, border: str = "=") -> None:
          self._add(border * len(title))
          self._add(title)
          self._add(border * len(title))

      def newline(self) -> None:
          self._add("")

      def heading(self, heading: str, underline: str = "*") -> None:
          self._add(heading)
          self._add(underline * len(heading))

      def directive(self, name: str, fields: typing.List[typing.Tuple[str, str]]) -> None:
          self._add(f".. {name}::")
          for k, v in fields:
              self._add(f"   :{k}: {v}")

      def content(self, content: str) -> None:
          self._add(content)
```

c) Usage

The program have `2 arguments`. You can check the documentation by using the command

```
python main.py -h
```

```
usage: main.py [-h] [-i INPUT_FILE] [-o OUTPUT_FILE]

options:
  -h, --help            show this help message and exit
  -i INPUT_FILE, --input_file INPUT_FILE
                        Directory to input file. Accepts file *.json only
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Directory to output *.rst file.
```

Example command:

```
python main.py -i sample.json -o sample.rst
```