

1. The HTML file has a single table listing all DCS minors.
 - a. Each student appears in one <tr> (table row).
 - b. All other <tr> elements contain the student data in <td> cells, in this order:

Student name (Last, First)

Email address (inside a link)

Class year (e.g., 2025)

Majors (some have multiple)

Minors

GECs

Advisor name (Last, First)

2. We will use BeautifulSoup's find_all('tr') to get all table rows.
3. For each row after the header, we'll call find_all('td') to get each cell.

Interpreter Test:

with open("dcs_minor_roster.html", "r", encoding="utf-8") as f:

```
soup = BeautifulSoup(f, "html.parser")
```

```
rows = soup.find_all("tr")
```

```
print("Number of table rows:", len(rows))
```

```
cells = rows[1].find_all("td")
```

```
print("Number of data cells:", len(cells))
```

```
print("Raw text for first row:", [c.text.strip() for c in cells])
```

4. From each td cell:

Name: read the text directly (e.g., "Beeham, Zoowee").

Email: get the <a> tag inside the second cell, use a['href'], and remove the "mailto:" prefix.

Year: convert the third cell text to an integer or leave as a string.

Majors, Minors, GECs: split the text by commas and strip whitespace. If a field contains "0000" or is empty, replace it with an empty list or empty string.

Advisor: read as plain text and strip any spaces.

For each student, we'll create a Student object with:

```
Student(name, email, year, majors, minors, geecs, advisor)
```

```
majors_text = cells[3].text.strip()
```

```
majors = [m.strip() for m in majors_text.split(",") if m.strip() and m != "0000"]
```

```
print("Extracted majors list:", majors)
```

5. We'll store each Student in two dictionaries:

by_year: key = class year, value = list of Student objects for that year.

by_advisor: key = advisor's name, value = list of Student objects they advise.

Before returning, we'll sort:

Each year's student list by last name (and first name if needed).

Each advisor's student list alphabetically by student name.

The function will return a tuple:

(by_year, by_advisor)