Lecture 16: Automated data retrieval II

Website investigators!

First, the libraries

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

HTML basics

- Makes the structure of websites
- A website consists of HTML elements
  - An open (start) tag
  - Some attributes
  - Content
  - A close (end) tag

HTML basics

- Makes the structure of websites
- A website consists of HTML elements
  - An open (start) tag
  - Some attributes
  - Content
  - A close (end) tag

HTML basics

- Makes the structure of websites
- A website consists of HTML elements
  - An open (start) tag
  - Some attributes
  - Content
  - A close (end) tag

HTML basics

- Makes the structure of websites
- A website consists of HTML elements
  - An open (start) tag
  - Some attributes
  - Content
  - A close (end) tag

Some common HTML tags

- <h1>, …, <h6>: headings
- <b>, <i>: bold, italic
- <p>: paragraph
- <a>: hyperlinks
- <img>: images
- <ul>, <ol>: unordered, ordered list
  - <li>: list element
- <table>: start of a table element
  - <tr>: table row
  - <th>: table header
  - <td>: table cell

Simple HTML example

```html
<!DOCTYPE html>
<html>
<head>
<title>Hello world!</title>
</head>
<body>

<h1>Welcome to Data Skills 1</h1>
<p>We are at the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
<p>The material is very interesting.</p>
<p>But <b>summer break</b> is coming!</p>

</body>
</html>
```

Simple HTML example

Content

Start tag

End tag

```
<!DOCTYPE html>
<html>
<head>
<title>Hello world!</title>
</head>
<body>

<h1>Welcome to Data Skills 1</h1>
<p>We are at the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
<p>The material is very interesting.</p>
<p>But <b>summer break</b> is coming!</p>

</body>
</html>
```

Simple HTML example

Start tag

Content

End tag

```html
<!DOCTYPE html>
<html>
<head>
<title>Hello world!</title>
</head>
<body>

<h1>Welcome to Data Skills 1</h1>
<p>We are at the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
<p>The material is very interesting.</p>
<p>But <b>summer break</b> is coming!</p>

</body>
</html>
```

Simple HTML example

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Hello world!</title>
5  </head>
6  <body>
7
8  <h1>Welcome to Data Skills 1</h1>
9  <p>We are at the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
10 <p>The material is very interesting.</p>
11 <p>But <b>summer break</b> is coming!</p>
12
13 </body>
14 </html>
```

Start tag

Content

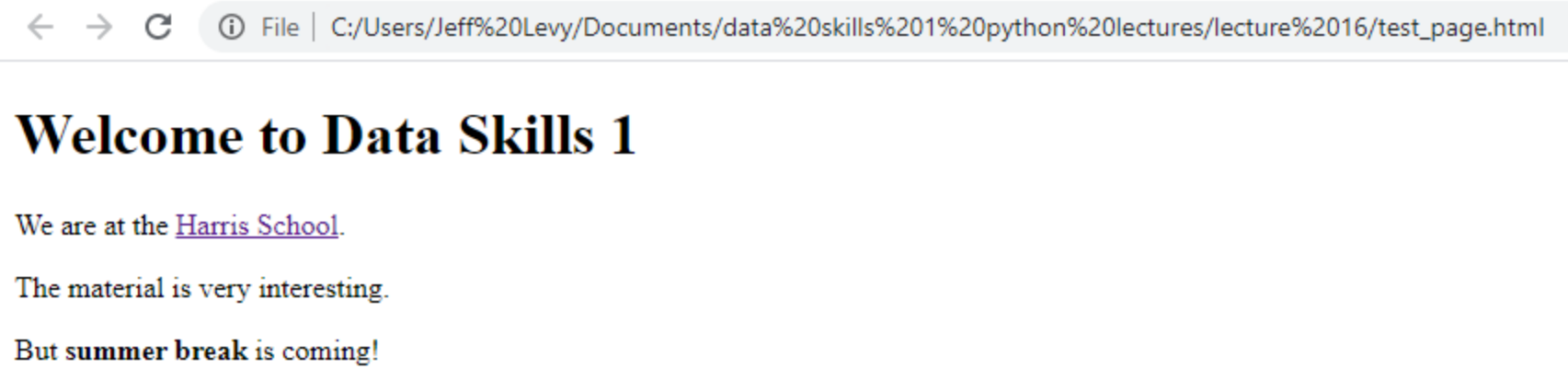
End tag

End tag

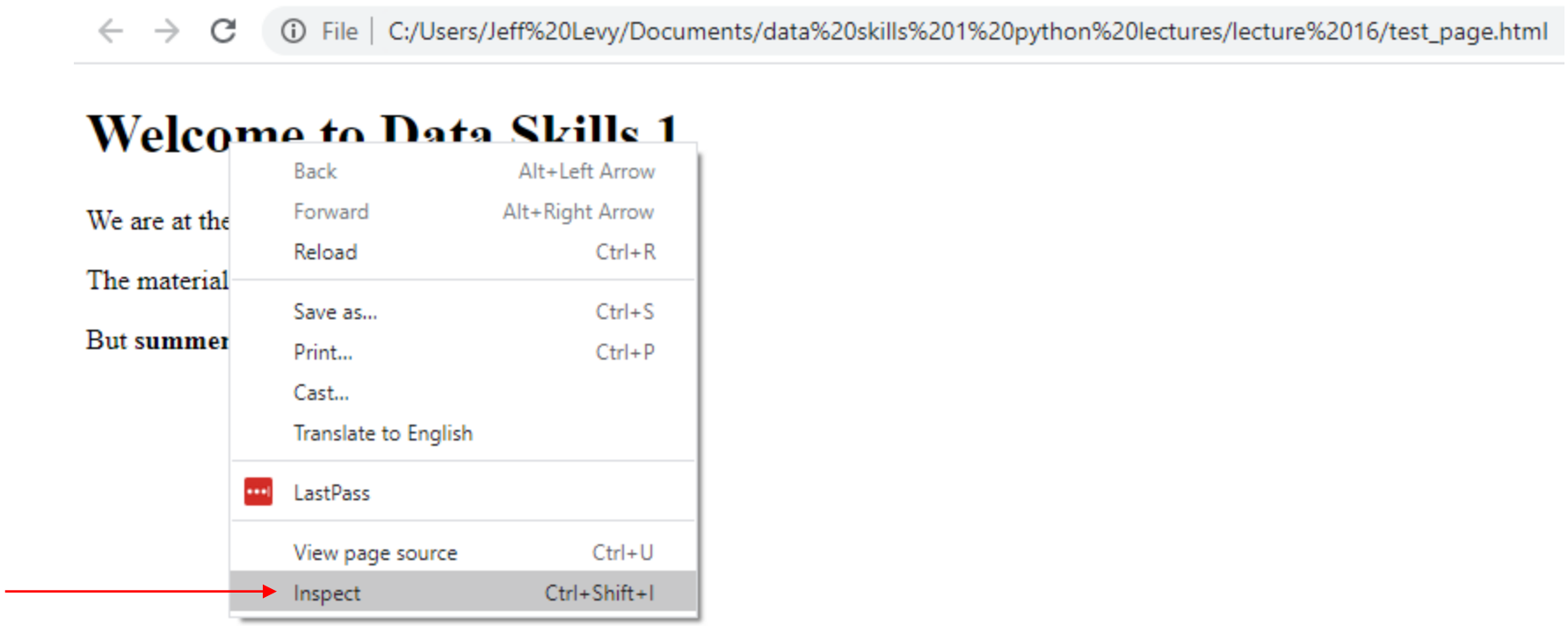
Attribute

Start tag

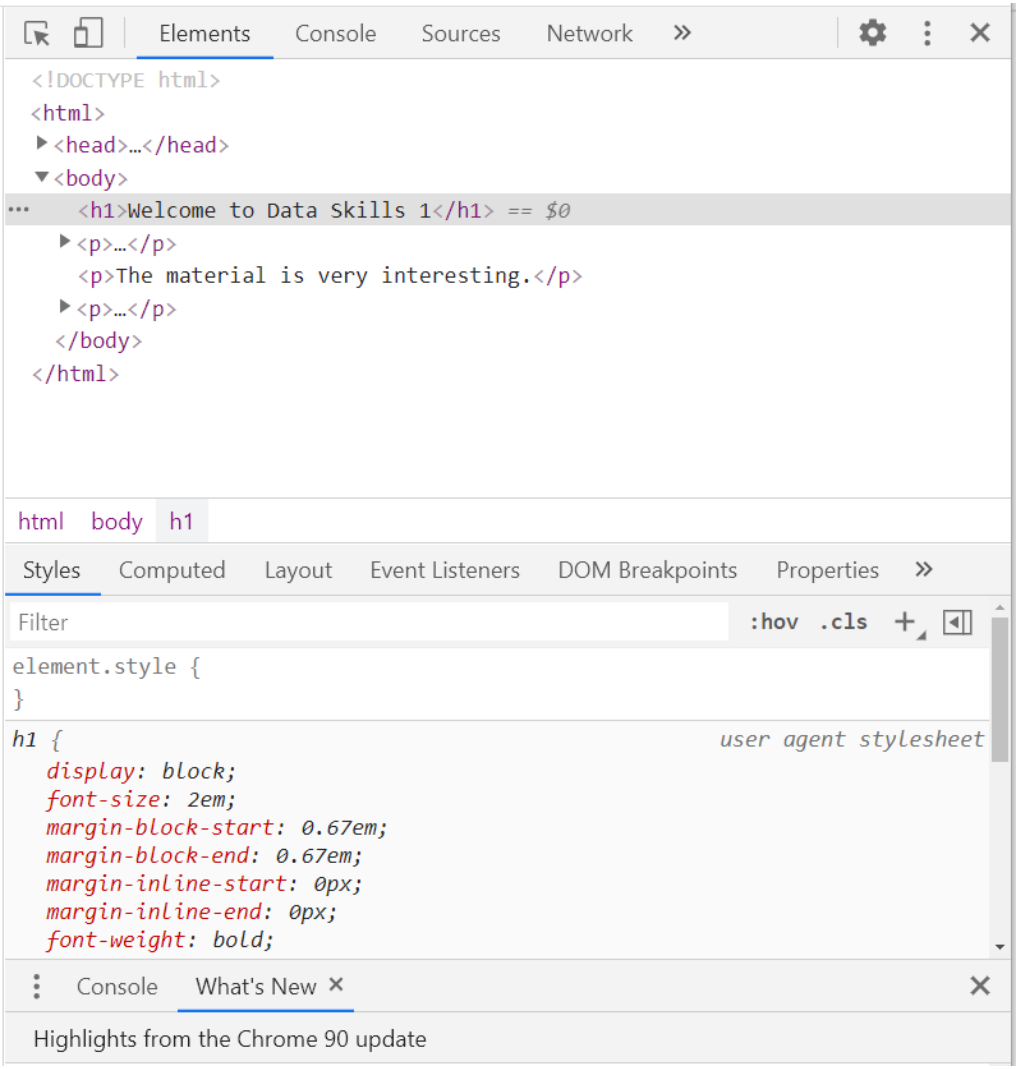
Simple HTML example



Browser “inspect” tool



Browser “inspect” tool



Browser “inspect” tool

Elements Console Sources Network »

```
<!DOCTYPE html>
<html>
▼<head>
   <title>Hello world!</title>
 </head>
▼<body>
   <h1>Welcome to Data Skills 1</h1>
…  ▼<p> == $0
      "We are at the "
      <a href="https://harris.uchicago.edu/">Harris School</a>
      "."
    </p>
    <p>The material is very interesting.</p>
  ▼<p>
      "But "
      <b>summer break</b>
      " is coming!"
    </p>
 </body>
</html>
```

Introducing the BeautifulSoup Library

```
5   with open(r'test_page.html', 'r') as page:
6       text = page.read()
7
8   soup = BeautifulSoup(text, 'lxml')
```

Introducing the BeautifulSoup Library

```
5   with open(r'test_page.html', 'r') as page:
6       text = page.read()
7
8   soup = BeautifulSoup(text, 'lxml')
```

“lxml” is an external resource used by browsers to parse HTML; if you have issues with using it here, change to the default Python parser, ‘html.parser’

Introducing the BeautifulSoup Library

```
5  with open(r'test_page.html', 'r') as page:
6      text = page.read()
7
8  soup = BeautifulSoup(text, 'lxml')
```

The “soup” object is the website content, parsed into an easy-to-use reference

Introducing the BeautifulSoup Library

```
In [64]: text
Out[64]: '<!DOCTYPE html>\n<html>\n<head>\n<title>Hello world!</title>
\n</head>\n<body>\n\n<h1>Welcome to Data Skills 1</h1>\n<p>We are at
the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
\n<p>The material is very interesting.</p>\n<p>But <b>summer break</b>
is coming!</p>\n\n</body>\n</html>\n\n'
```

Introducing the BeautifulSoup Library

```
In [64]: text
Out[64]: '<!DOCTYPE html>\n<html>\n<head>\n<title>Hello world!</title>
\n</head>\n<body>\n\n<h1>Welcome to Data Skills 1</h1>\n<p>We are at
the <a href="https://harris.uchicago.edu/">Harris School</a>.</p>
\n<p>The material is very interesting.</p>\n<p>But <b>summer break</b>
is coming!</p>\n\n</body>\n</html>\n\n'
```

```
In [66]: soup
Out[66]:
<!DOCTYPE html>
<html>
<head>
<title>Hello world!</title>
</head>
<body>
<h1>Welcome to Data Skills 1</h1>
<p>We are at the <a href="https://harris.uchicago.edu/">Harris
School</a>.</p>
<p>The material is very interesting.</p>
<p>But <b>summer break</b> is coming!</p>
</body>
</html>
```

Searching a “soup” object

```
In [69]: soup.find_all('p')
Out[69]:
[<p>We are at the <a href="https://harris.uchicago.edu/">Harris
School</a>.</p>,
 <p>The material is very interesting.</p>,
 <p>But <b>summer break</b> is coming!</p>]
```

Searching a “soup” object

```
In [69]: soup.find_all('p')
Out[69]:
[<p>We are at the <a href="https://harris.uchicago.edu/">Harris
School</a>.</p>,
 <p>The material is very interesting.</p>,
 <p>But <b>summer break</b> is coming!</p>]
```

```
In [70]: soup.find('a')
Out[70]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

Searching a “soup” object

```
In [79]: soup.find_all('a', href="https://harris.uchicago.edu/")
Out[79]: [<a href="https://harris.uchicago.edu/">Harris School</a>]
```

Searching a “soup” object

```
In [79]: soup.find_all('a', href="https://harris.uchicago.edu/")
Out[79]: [<a href="https://harris.uchicago.edu/">Harris School</a>]
```

```
In [78]: soup.find_all('a', href=lambda h: 'uchicago' in h)
Out[78]: [<a href="https://harris.uchicago.edu/">Harris School</a>]
```

Working with a “tag” object

```
tag = soup.find('a')
```

```
In [72]: tag
Out[72]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

Working with a “tag” object

```
14    tag = soup.find('a')
```

```
In [72]: tag
Out[72]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

```
In [73]: tag.attrs
Out[73]: {'href': 'https://harris.uchicago.edu/'}
```

Working with a “tag” object

```
14    tag = soup.find('a')
```

```
In [72]: tag
Out[72]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

```
In [73]: tag.attrs
Out[73]: {'href': 'https://harris.uchicago.edu/'}
In [74]: tag.contents
Out[74]: ['Harris School']
```

Working with a “tag” object

```
14    tag = soup.find('a')
```

```
In [72]: tag
Out[72]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

```
In [73]: tag.attrs
Out[73]: {'href': 'https://harris.uchicago.edu/'}
In [74]: tag.contents
Out[74]: ['Harris School']
In [75]: tag.has_attr('href')
Out[75]: True
```

Working with a “tag” object

```
14    tag = soup.find('a')
```

```
In [72]: tag
Out[72]: <a href="https://harris.uchicago.edu/">Harris School</a>
```

```
In [73]: tag.attrs
Out[73]: {'href': 'https://harris.uchicago.edu/'}
In [74]: tag.contents
Out[74]: ['Harris School']
In [75]: tag.has_attr('href')
Out[75]: True
In [76]: tag.parent
Out[76]: <p>We are at the <a href="https://
harris.uchicago.edu/">Harris School</a>.</p>
```

Into the wild: Parsing a table

maktaba.tetea.org/exam-results/FTNA2015/S0206.htm

THE NATIONAL EXAMINATIONS COUNCIL OF TANZANIA
FORM TWO NATIONAL ASSESSMENT (FTNA) 2015 RESULTS
CENTRE: S0206 - KILAKALA SECONDARY SCHOOL

School Overall Grade Summary

	A	B+	B	C	D	E	F
F	381	325	210	156	140	138	90
TOT	381	325	210	156	140	138	90

CNO	REPEATER	NAME OF CANDIDATE	SEX	CIVICS	HISTORY	GEOGRAPHY	E/D/KIISLAMU	KISWAHILI	ENGLISH	FRENCH	PHYSICS	CHEMISTRY	BIOLOGY	INFO & COMP STUD	B/MATH	FOOD	COMMERCE	B/KEEPING	GPA	CLASS
0001		AGNESS M NHENDE	F	C	D	D		B	C		E	E	C	F	E	D			1.7	CREDIT
0002		AGNESS P KAWIA	F	Absent																ABS
0003		AJESTA N MBWILO	F	A	B+	B		B+	A		C	A	B+	D	B		B+	A	4.6	DISTINCTION
0004		ALBERTINA A KITONKA	F	A	B+	A		B+	A		B	A	A	C	B+	B+			4.7	DISTINCTION
0005		ANGELINA H HAULE	F	A	B	B		B	A		C	B	A	E	F	C			3.9	DISTINCTION
0006		ANITHA M BUBERWA	F	A	A	A		A	A		B	A	A	C	A	B			5.0	DISTINCTION
0007		ANNA J HHAIMA	F	C	E	D		B	B+	E	F	F	C	E	F				1.7	CREDIT
0008		ASHA A MTUI	F	A	B+	A	B	B+	A		C	A	A	D	C	C			4.7	DISTINCTION
0009		AZIZA M NJIMWE	F	E	E	E	D	C	D		F	F	D	F	F	F			0.7	PASS
0010		BEATRICE G SOGOLELA	F	D	C	E		B	B+		E	E	C	E	D		C	E	2.1	CREDIT
0011		BEATRICE P RUGAIMBILA	F	A	B	B+		B	A		C	B	A	E	E		B+	C	4.1	DISTINCTION

Into the wild: Parsing a table

```
▼<body>
  ▼<center>
    <!-- Global site tag (gtag.js) - Google Analytics -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-0Q51
    VDK3K"></script>
    ▶<script>…</script>
    <!-- End Global site tag -->
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1
    52">
    <meta name="ROBOTS" content="NOINDEX">
    ▼<table>
      ▼<tbody>
        ▼<tr>
            <th class="head" colspan="23">THE NATIONAL EXAMINATIONS COUNCIL
            OF TANZANIA</th>
          </tr>
        ▶<tr>…</tr>
        ▶<tr>…</tr>
        ▶<tr>…</tr>
        ▶<tr>…</tr>
```

Into the wild: Parsing a table

CNO	ER	NAME OF CANDIDATE	EX	:	:	:	:	:	:	:	:	:	:	OD	:	:	:	:	GPA	CLASS
0001		AGNESS M NHENDE	F	C	D	D		B	C		E	E	C	F	E	D			1.7	CREDIT
0002		AGNESS P KAWIA	F	Absent																ABS
0003		AJESTA N MBWILO	F	A	B+	B		B+	A		C	A	B+	D	B		B+	A	4.6	DISTINCTION

```
▼<tr>
  <td align="center">0001</td>
  <td class="mark"></td>
  <td>AGNESS M NHENDE</td>
  <td class="mark">F</td>
  <td class="mark">C</td>
  <td class="mark">D</td>
  <td class="mark">D</td>
  <td class="mark"></td>
  <td class="mark">B</td>
  <td class="mark">C</td>
  <td class="mark"></td>
  <td class="mark">E</td>
  <td class="mark">E</td>
  <td class="mark">C</td>
  <td class="mark">F</td>
  <td class="mark">E</td>
```

Into the wild: Parsing a table

```
25  url = 'http://maktaba.tetea.org/exam-results/FTNA2015/S0206.htm'
26  path = r'c:\users\jeff levy\desktop\grades.csv'
27
28  response = requests.get(url)
29  soup = BeautifulSoup(response.text, 'lxml') #html.parser
```

Into the wild: Parsing a table

```
25  url = 'http://maktaba.tetea.org/exam-results/FTNA2015/S0206.htm'
26  path = r'c:\users\jeff levy\desktop\grades.csv'
27
28  response = requests.get(url)
29  soup = BeautifulSoup(response.text, 'lxml') #html.parser
```

How do I know the page loaded as I expect?

Into the wild: Parsing a table

```
25  url = 'http://maktaba.tetea.org/exam-results/FTNA2015/S0206.htm'
26  path = r'c:\users\jeff levy\desktop\grades.csv'
27
28  response = requests.get(url)
29  soup = BeautifulSoup(response.text, 'lxml') #html.parser
```

How do I know the page loaded as I expect?

```
In [85]: 'NHENDE' in soup.text
Out[85]: True
```

Into the wild: Parsing a table

```
25  url = 'http://maktaba.tetea.org/exam-results/FTNA2015/S0206.htm'
26  path = r'c:\users\jeff levy\desktop\grades.csv'
27
28  response = requests.get(url)
29  soup = BeautifulSoup(response.text, 'lxml') #html.parser
```

How do I know the page loaded as I expect?

```
In [85]: 'NHENDE' in soup.text
Out[85]: True
```

```
In [87]: soup.text[0:300]
Out[87]: '\n\n\n\n\n\n\n\nTHE NATIONAL EXAMINATIONS COUNCIL OF
TANZANIAFORM TWO NATIONAL ASSESSMENT (FTNA) 2015 RESULTSCENTRE: S0206
- KILAKALA SECONDARY SCHOOL School Overall Grade Summary \xa0 \n\n
\xa0AB
+BCDEFF38132521015614013890TOT38132521015614013890\xa0\xa0\xa0\xa0CHGE
KEFPCBIBFCB\xa0\xa0\xa0\xa0\xa0\xa0IIE/INRHHIN/OO/
\xa0\xa0\xa0\xa0\xa0\xa0VSODSGEYEOFMOMK\xa0\xa0\xa0\xa0\xa0\xa0ITG/'
```

Into the wild: Parsing a table

```
table = soup.find('table')
table.find_all('tr')[24].find_all('td')
```

```
[<td align="center">0001</td>,
 <td class="mark"></td>,
 <td>AGNESS M NHENDE</td>,
 <td class="mark">F</td>,
 <td class="mark">C</td>,
 <td class="mark">D</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark">B</td>,
 <td class="mark">C</td>,
 <td class="mark"></td>,
 <td class="mark">E</td>,
 <td class="mark">E</td>,
 <td class="mark">C</td>,
 <td class="mark">F</td>,
 <td class="mark">E</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark"></td>,
 <td class="mark">1.7</td>,
 <td class="mark">CREDIT</td>]
```

Into the wild: Parsing a table

First, I extract the table element as its own soup object

```
table = soup.find('table')
table.find_all('tr')[24].find_all('td')
```

```
[<td align="center">0001</td>,
 <td class="mark"></td>,
 <td>AGNESS M NHENDE</td>,
 <td class="mark">F</td>,
 <td class="mark">C</td>,
 <td class="mark">D</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark">B</td>,
 <td class="mark">C</td>,
 <td class="mark">></td>,
 <td class="mark">E</td>,
 <td class="mark">E</td>,
 <td class="mark">C</td>,
 <td class="mark">F</td>,
 <td class="mark">E</td>,
 <td class="mark">D</td>,
 <td class="mark">></td>,
 <td class="mark">></td>,
 <td class="mark">1.7</td>,
 <td class="mark">CREDIT</td>]
```

Into the wild: Parsing a table

```
table = soup.find('table')
table.find_all('tr')[24].find_all('td')
```

Second, I extract all the "tr" tags from within the table element

```
[<td align="center">0001</td>,
 <td class="mark"></td>,
 <td>AGNESS M NHENDE</td>,
 <td class="mark">F</td>,
 <td class="mark">C</td>,
 <td class="mark">D</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark">B</td>,
 <td class="mark">C</td>,
 <td class="mark">></td>,
 <td class="mark">E</td>,
 <td class="mark">E</td>,
 <td class="mark">C</td>,
 <td class="mark">F</td>,
 <td class="mark">E</td>,
 <td class="mark">D</td>,
 <td class="mark">></td>,
 <td class="mark">></td>,
 <td class="mark">1.7</td>,
 <td class="mark">CREDIT</td>]
```

Into the wild: Parsing a table

```
35  table = soup.find('table')
36  table.find_all('tr')[24].find_all('td')
```

Third, I take only the “tr” tag at the 24th index

```
[<td align="center">0001</td>,
 <td class="mark"></td>,
 <td>AGNESS M NHENDE</td>,
 <td class="mark">F</td>,
 <td class="mark">C</td>,
 <td class="mark">D</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark">B</td>,
 <td class="mark">C</td>,
 <td class="mark"></td>,
 <td class="mark">E</td>,
 <td class="mark">E</td>,
 <td class="mark">C</td>,
 <td class="mark">F</td>,
 <td class="mark">E</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark"></td>,
 <td class="mark">1.7</td>,
 <td class="mark">CREDIT</td>]
```

Into the wild: Parsing a table

```
table = soup.find('table')
table.find_all('tr')[24].find_all('td')
```

Finally, I extract all the "td" tags from within the single "tr" tag

```
[<td align="center">0001</td>,
 <td class="mark"></td>,
 <td>AGNESS M NHENDE</td>,
 <td class="mark">F</td>,
 <td class="mark">C</td>,
 <td class="mark">D</td>,
 <td class="mark">D</td>,
 <td class="mark"></td>,
 <td class="mark">B</td>,
 <td class="mark">C</td>,
 <td class="mark"></td>,
 <td class="mark">E</td>,
 <td class="mark">E</td>,
 <td class="mark">C</td>,
 <td class="mark">F</td>,
 <td class="mark">E</td>,
 <td class="mark">D</td>,
 <td class="mark">></td>,
 <td class="mark">></td>,
 <td class="mark">1.7</td>,
 <td class="mark">CREDIT</td>]
```

Whole document -> <table> -> <tr> -> <td>

Into the wild: Parsing a table

Whole document -> <table> -> <tr> -> <td>

Into the wild: Parsing a table

Whole document -> <table> -> <tr> -> <td> -> extract text from td tag

```
unparsed_rows = []
for row in table.find_all('tr'):
    td_tags = row.find_all('td')
    unparsed_rows.append([val.text for val in td_tags])

unparsed_rows[25]
```

Into the wild: Parsing a table

Whole document -> <table> -> <tr> -> <td> -> extract text from td tag

```
unparsed_rows = []
for row in table.find_all('tr'):
    td_tags = row.find_all('td')
    unparsed_rows.append([val.text for val in td_tags])

unparsed_rows[25]
```

```
['0002', '', 'AGNESS P KAWIA', 'F', 'Absent', 'ABS']
```

Problem: the “absent” entry spans multiple columns in the table

Into the wild: Parsing a table

Whole document -> <table> -> <tr> -> <td> -> extract text from td tag

```
unparsed_rows = []
for row in table.find_all('tr'):
    td_tags = row.find_all('td')
    unparsed_rows.append([val.text for val in td_tags])

unparsed_rows[25]
```

```
['0002', '', 'AGNESS P KAWIA', 'F', 'Absent', 'ABS']
```

```
def row_parser(row):
    if row[4] == 'Absent':
        row = row[:4] + ['Absent']*17
    return ','.join(row)
parsed_rows = [row_parser(row) for row in unparsed_rows[24:152]]
```

Into the wild: Parsing a table

```
In [92]: parsed_rows
Out[92]:
['0001,,AGNESS M NHENDE,F,C,D,D,,B,C,,E,E,C,F,E,D,,,1.7,CREDIT',
 '0002,,AGNESS P
KAWIA,F,Absent,Absent,Absent,Absent,Absent,Absent,Absent,Absent,Absent,Absent,A
ent,Absent,Absent,Absent,Absent,Absent',
 '0003,,AJESTA N MBWILO,F,A,B+,B,,B+,A,,C,A,B+,D,B,,B+,A,4.6,DISTINCTION',
 '0004,,ALBERTINA A KITONKA,F,A,B+,A,,B+,A,,B,A,A,C,B+,B+,,,4.7,DISTINCTION',
 '0005,,ANGELINA H HAULE,F,A,B,B,,B,A,,C,B,A,E,F,C,,,3.9,DISTINCTION',
 '0006,,ANITHA M BUBERWA,F,A,A,A,,A,A,,B,A,A,C,A,B,,,5.0,DISTINCTION',
 '0007,,ANNA J HHAIMA,F,C,E,D,,B,B+,E,F,F,C,E,F,,,,1.7,CREDIT',
 '0008,,ASHA A MTUI,F,A,B+,A,B,B+,A,,C,A,A,D,C,C,,,4.7,DISTINCTION',
 '0009,,AZIZA M NJIMWE,F,E,E,E,D,C,D,,F,F,D,F,F,F,,,0.7,PASS',
 '0010,,BEATRICE G SOGOLELA,F,D,C,E,,B,B+,,E,E,C,E,D,,C,E,2.1,CREDIT',
 '0011,,BEATRICE P RUGAIMBILA,F,A,B,B+,,B,A,,C,B,A,E,E,,B+,C,4.1,DISTINCTION',
 '0012,,BETINA B MWASOMOLA,F,B+,B+,B+,,B+,B,,D,D,B+,E,D,,B,C,3.7,DISTINCTION',
```

We now have a list of strings, where each string contains the same number of commas

Into the wild: Parsing a table

```python
header = 'CNO,Repeat,Name,Sex,CIV,HIST,GEO,ED
parsed_rows.insert(0, header)

document = '\n'.join(parsed_rows)

with open(path, 'w') as ofile:
    ofile.write(document)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	CNO	Repeat	Name	Sex	CIV	HIST	GEO	EDK	KIS	ENG	FRN	PHY	CHEM	BIO	COMP	MATH	FOOD	COMM	BKEEPING	GPA	CLASS	
2	1		AGNESS M	F	C	D	D		B	C		E	E	C	F	E	D			1.7	CREDIT	
3	2		AGNESS P	F	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	Absent	
4	3		AJESTA N	F	A	B+	B		B+	A		C	A	B+	D	B		B+	A	4.6	DISTINCTION	
5	4		ALBERTIN	F	A	B+	A		B+	A		B	A	A	C	B+	B+			4.7	DISTINCTION	
6	5		ANGELINA	F	A	B	B		B	A		C	B	A	E	F	C			3.9	DISTINCTION	
7	6		ANITHA M	F	A	A	A		A	A		B	A	A	C	A	B			5	DISTINCTION	
8	7		ANNA J HI	F	C	E	D		B	B+	E	F	F	C	E	F				1.7	CREDIT	
9	8		ASHA A M	F	A	B+	A	B	B+	A		C	A	A	D	C	C			4.7	DISTINCTION	
10	9		AZIZA M N	F	E	E	E	D	C	D		F	F	D	F	F	F			0.7	PASS	
11	10		BEATRICE	F	D	C	E		B	B+		E	E	C	E	D		C	E	2.1	CREDIT	
12	11		BEATRICE	F	A	B	B+		B	A		C	B	A	E	E		B+	C	4.1	DISTINCTION	
13	12		BETINA B	F	B+	B+	B+		B+	B		D	D	B+	E	D		B	C	3.7	DISTINCTION	
14	13		BITRES Y C	F	B+	B+	B+		B+	A	B+	B+	A	A	D	B+				4.4	DISTINCTION	
15	14		CATHERIN	F	A	B+	B+		B+	A	B+	C	B	A	D	B+				4.4	DISTINCTION	