CIS 1100

Scraping Tables &
Requesting Pages

Python
Fall 2024
University of Pennsylvania

Review: Practice from last time

```
<table class="table table-striped">
        <tr><th>Date</th><th>Topics</th><th>Slides</th><th>Example Code</th><th>Due Dates</th></tr>
        <tr class="">
            <td>Wed, Aug 28, 2024</td>
            <td>Introduction</td>
            <td><a target="_blank" href="../intro.pdf">✏️</a></td>
            <td></td><td> </td>
        </tr>
        <tr class="">
            <td>Fri, Aug 30, 2024</td>
            <td>Hello, World!</td>
            <td><a target="_blank" href="../hello_world.pdf">🐭</a>
                <a target="_blank" href="../hello_world_lecture.pdf">✏️</a></td>
            <td><a target="_blank" href="./hello_world.py">hello_world.py</a><br /></td><td> </td>
        </tr>
```

From the previous step, get:

1. a list of all dates
2. a list of all lecture topics

Practice Part 1: (C12)

```
<tr class="">
    <td>Mon, Sep 9, 2024</td><td>Variables &amp;amp; Types</td>
    <td><a target="_blank" href="../datatypes.pdf">📽</a>
        <a target="_blank" href="../types_lecture2.pdf">✏️</a></td>
    <td></td><td> </td>
</tr>
<tr class="success">
    <td>Wed, Sep 11, 2024</td><td>Conditionals</td>
    <td><a target="_blank" href="../conditionals.pdf">📽</a>
        <a target="_blank" href="../conditionals_lecture.pdf">✏️</a></td>
    <td></td><td>HW00 @ 11:59pm </td>
</tr>
```

1. Get a list of all row tags, but only when an assignment is due

   (Is there a pattern you notice? One of the two rows above has a hw due)

2. Populate a dictionary that maps the date (string) to the HW due message (e.g. one of the entries should map `"Wed, Sep11, 2024"` to `"HW00 @ 11:59pm"`

`requests`

`pip install requests` to get access to a library that allows you to:

- programmatically "visit" websites
- get responses (HTML) within your program
- do all kinds of advanced stuff like *upload information to servers* or *communicate with APIs*

The Very Very Very Basics

- `get("my.url.com")` queries the website at that URL and returns a `Response`
- A `Response` is a dense object that contains information about what the remote server "said"
  - response code: a number that indicates whether your request was processed properly
  - information about the data encoding
  - the text of the response, i.e. all the HTML (or JSON...)

A Minimal Request

```
import requests

url = "https://www.cis.upenn.edu/~cis110/current/py/homework/homework.html"
r = requests.get(url)
print(r)
```

🖨️ ▼

```
<Response [200]>
```

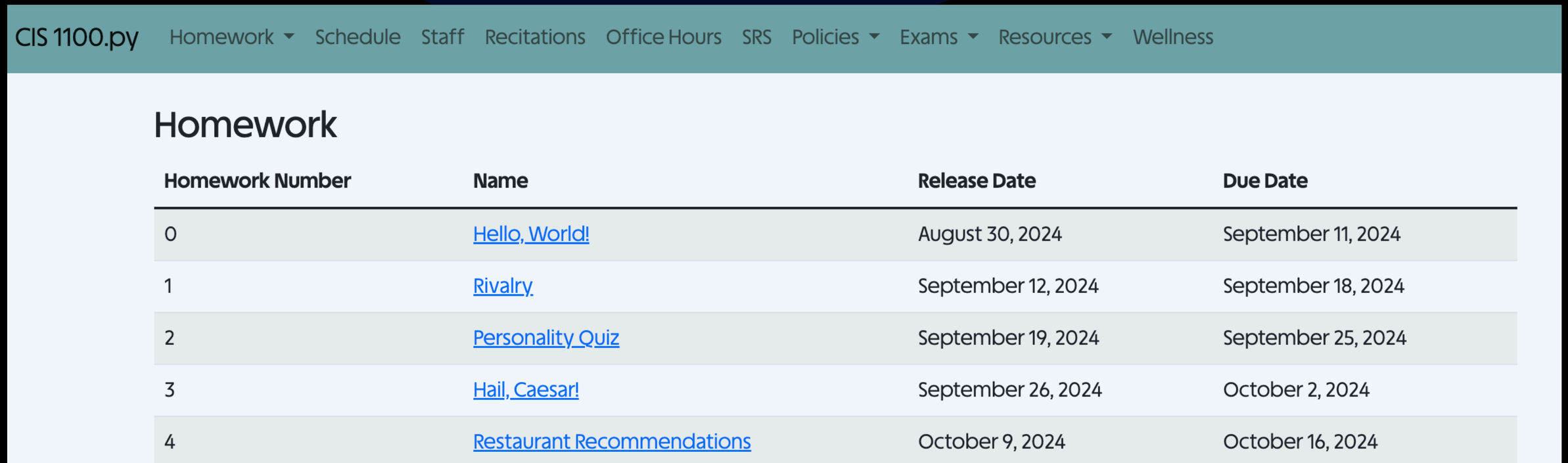
✅

A Minimal Request

```
import requests

url = "https://www.cis.upenn.edu/~cis110/current/py/homework/homework.html"
r = requests.get(url)
print(r.text)
```

`r.text` is just a string containing HTML, though. We know what to do with that...



CIS 1100.py Homework Schedule Staff Recitations Office Hours SRS Policies Exams Resources Wellness

Homework

Homework Number	Name	Release Date	Due Date
0	Hello, World!	August 30, 2024	September 11, 2024
1	Rivalry	September 12, 2024	September 18, 2024
2	Personality Quiz	September 19, 2024	September 25, 2024
3	Hail, Caesar!	September 26, 2024	October 2, 2024
4	Restaurant Recommendations	October 9, 2024	October 16, 2024

A Minimal Request

```
import requests
from bs4 import BeautifulSoup

url = "https://www.cis.upenn.edu/~cis110/current/py/homework/homework.html"
r = requests.get(url)
soup = BeautifulSoup(r.text, 'html.parser')
links = soup.table.find_all('a')
print([link.text for link in links])
```

```
['Hello, World!', 'Rivalry', 'Personality Quiz', 'Hail, Caesar!', 'Restaurant Recommendations']
```

Practice: (L11)

Add one or two lines to make it so that we download the simple syllabus html and get its contents ready to parse into a soup object.

```
import requests
from bs4 import BeautifulSoup

url = "https://www.cis.upenn.edu/~cis110/current/py/lectures/examples/simple_syllabus.html"
________
________
soup = BeautifulSoup(html_content, 'html.parser')
links = soup.table.find_all('a')
print([link.text for link in links])
```

Practice: (C14)

```
<tr><th>Date</th> <th>Topics</th> <th>Slides</th> <th>Example Code</th> <th>Due Dates</th>
</tr>
<tr class="">
    <td>Mon, Sep 16, 2024</td>
    <td>Sequences</td>
    <td><a target="_blank" href="../sequences.pdf">🐭</a>
        <a target="_blank" href="../sequences_lecture.pdf">✏️</a></td>
    <td><a href="./guessing.py">guessing.py</a><a href="./timer.py">timer.py</a></td>
    <td> </td>
</tr>
```

- Given the same simple syllabus and soup from L11:
  - get all the data rows of the table
  - find all the links to example code (Hint: how do we handle there being more than one example code? `find_all` may be useful)
  - get a list of all the lecture example code contents. (Using `requests` to do this)