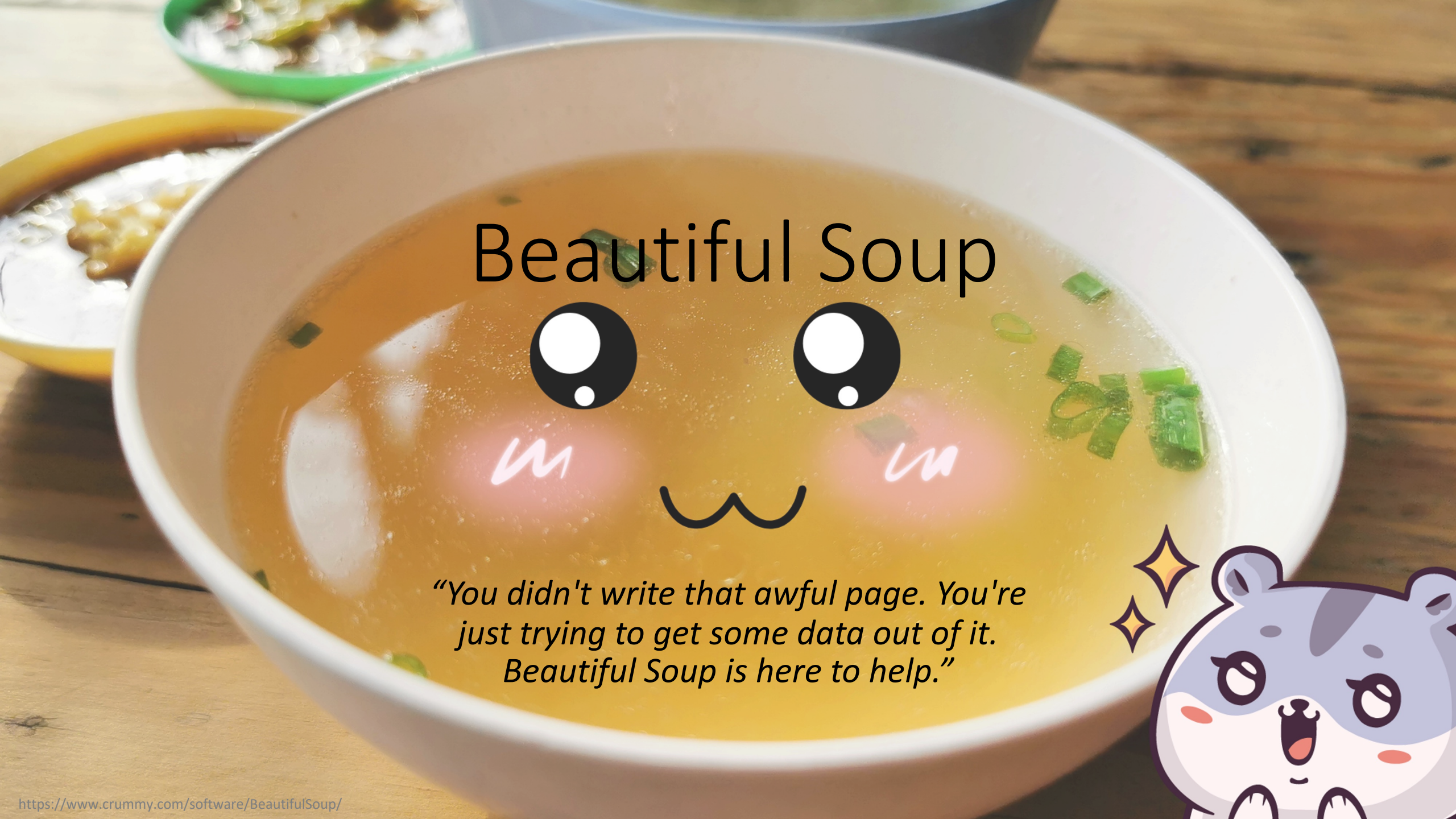
Beautiful Soup



“You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help.”

https://www.crummy.com/software/BeautifulSoup/

What is Beautiful Soup?

- Beautiful Soup is a library that makes it easy to scrape information from web pages.
- It sits atop an HTML or XML parser, providing Pythonic idioms for iterating, searching, and modifying the parse tree.

https://pypi.org/project/beautifulsoup4/

HyperText Markup Language (HTML)

- While performing web scarping, we deal with html tags.
- Thus, we must have good understanding of them.

- HTML is the standard **markup language** for Web pages.
  - Markup language is a term used in computer text processing to refer to an organized annotation system (i.e. language) that marks certain parts or elements of a document as different from plain text.
  - Essentially, markup language is used in web documents or applications to format text and to give it a specific structure.

https://www.w3schools.com/html/
https://www.seobility.net/en/wiki/Markup_Language

HTML – Tags

- `<!DOCTYPE html>` : HTML documents must start with a type declaration
- HTML document is contained between `<html>` and `</html>`
- The visible part of the HTML document is between `<body>` and `</body>`
- HTML headings are defined with the `<h1>` to `<h6>` tags
- HTML paragraphs are defined with the `<p>` tag

```
<!DOCTYPE html>

<html>
    <body>
        <h1>This is the first heading</h1>
        <p>And here is a paragraph.</p>
    </body>
</html>
```

https://www.analyticsvidhya.com/blog/2015/10/beginner-guide-web-scraping-beautiful-soup-python/

HTML – Tags

- HTML links are defined with the <a> tag, for example:
  This is a link for uio.no
- HTML tables are defined with <Table>, row as <tr> and rows are divided into data as <td>
- HTML list starts with <ul> (unordered) and <ol> (ordered). Each item of list starts with

https://www.analyticsvidhya.com/blog/2015/10/beginner-guide-web-scraping-beautiful-soup-python/

```html
<!DOCTYPE html>

<html>
    <body>
        <h1>This is the first heading</h1>
        <p>And here is a paragraph.</p>
        <h2>This is my second heading</h2>
        <p>Here is a table:</p>
        <table>
            <tr>
                <th>Name</th>
                <th>Course</th>
                <th>Points</th>
            </tr>
            <tr>
                <td>Peter</td>
                <td>INF3331</td>
                <td>50</td>
            </tr>
            <tr>
                <td>George</td>
                <td>INF4331</td>
                <td>94</td>
            </tr>
        </table>
        <p>And here are some lists:</p>
        <p>The first one, is a unordered list.</p>
        <ul>
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ul>
        <p>And the second one is ordered.</p>
        <ol>
            <li>Coffee</li>
            <li>Tea</li>
            <li>Milk</li>
        </ol>
        <p>Links are kind of cool! <a href="https://nbviewer.org/github/UiO-IN3110/UiO-IN3110.github.io/blob/HEAD/lectures/07-web-programming/Introduction%20to%20HTML.ipynb">Just look here!</a></p>
    </body>
</html>

```

And the final result is this: tag.html

This is the first heading

And here is a paragraph.

This is my second heading

Here is a table:

Name	Course	Points
Peter	INF3331	50
George	INF4331	94

And here are some lists:

The first one, is a unordered list.

- Coffee
- Tea
- Milk

And the second one is ordered.

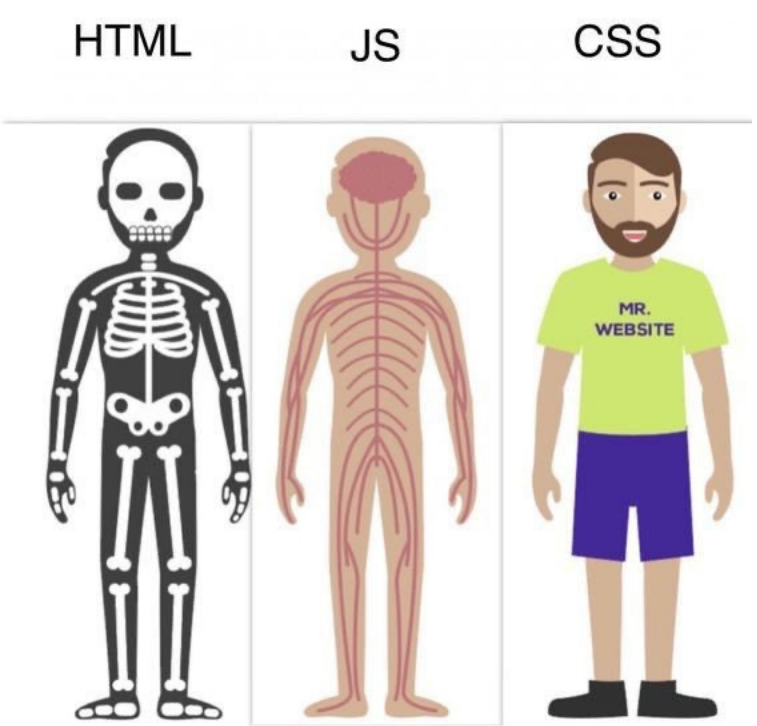
1. Coffee
2. Tea
3. Milk

Links are kind of cool! Just look here!

Why does it not look like a “normal” website?

- HTML is Skeleton of your website.
- CSS (or Cascading Style Sheets) gives all the nice fleshy covering to the website.
- JavaScript (JS) provides basic functionality to the website to come alive.

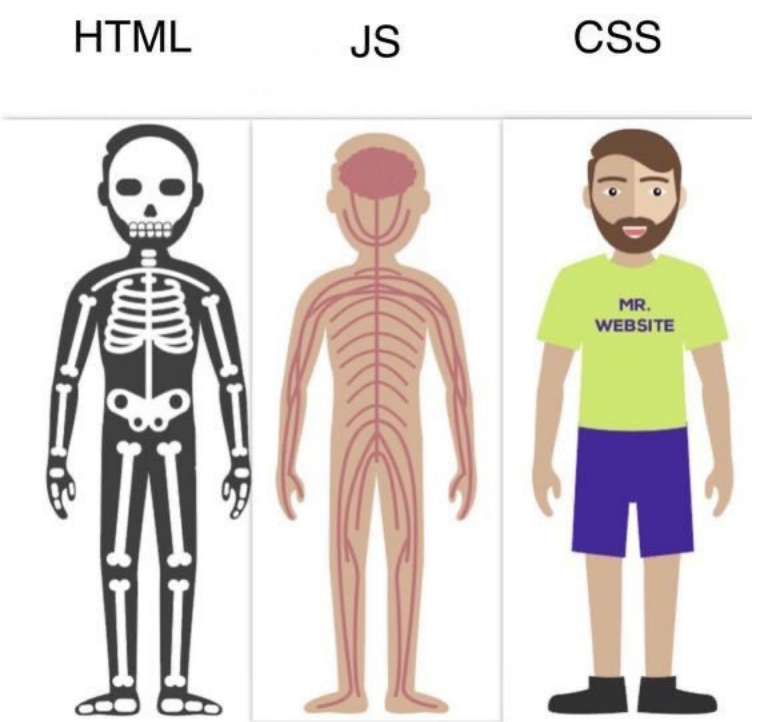
- Sounds intresting? Just wait to the next assignment! :D



https://www.quora.com/How-relevant-is-HTML-CSS-and-JavaScript-in-web-design

Why does it not look like a “normal” website?

- The class an id attribute is often used to point to a class name in a style sheet.
- It can also be used by a JavaScript to access and manipulate elements with the specific class name.



https://www.quora.com/How-relevant-is-HTML-CSS-and-JavaScript-in-web-design

Request

- First, we must make a request, which will download the HTML contents of a given web page for us.
- This will return a `Response object`
- This object has a `status_code` property, which indicates if the page was downloaded successfully:
  - A `status_code` of 200 means that the page downloaded successfully.
  - A status code starting with a 2 generally indicates success, and a code starting with a 4 or a 5 indicates an error.

https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Request

- We can print out the HTML content of the page using the content property.
- But this looks messy ☹

```python
# 1. Request
import requests

page = requests.get("https://raw.githubusercontent.com/fmwestby/IN3110_Group_Sessions/main/06_webScrapping/tags.html")

print(page.status_code)
# 200

print(page.content)
# b'<!DOCTYPE html>\n\n<html>\n    <body>\n        <h1>This is the first heading</h1>\n        <p>And here is a paragraph.</p>\n        <h2>This is my second hea
```

https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Beautiful Soup – The basics

- Using the Beautiful Soup `prettify()` method makes it more readable!

```
12  # 2. Beautiful Soup
13  from bs4 import BeautifulSoup
14
15  soup = BeautifulSoup(page.content, 'html.parser')
16
17  print(soup.prettify())
18  # <!DOCTYPE html>
19  # <html>
20  #  <body>
21  #   <h1>
22  #    This is the first heading
23  #   </h1>
24  #   <p>
25  #    .....
26
```

https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Beautiful Soup – Find the tags

- If we want to extract a single tag, we can instead use the `find_all` method, which will find all the instances of a tag on a page and put it in a list.

```
27  print(soup.find_all('p'))
28  # [<p>And here is a paragraph.</p>, <p>Here is a table:</p>, <p>And here are some lists:</p>, <p>The
```

https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Beautiful Soup – find id and class

- Beautiful Soup can also be used to find classes or id's

```
soup.find_all(id="id_name")

soup.find_all(class_="class_name")
```

https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Beautiful Soup (and Request) toturial

- A good toturial can be found here: https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/

Remember!

- On the Course homepage, you can find good resources!
  - There may also be some hints hiding there ☺

https://uio-in3110.github.io/

Higher Level Programming 2022

A course taught at the University of Oslo

Latest version - 2022

```
y = y_line + random.normal(0, 0.25, n)  # line with noise

# goal: fit a line to the data points x, y

# create a
A = array([x, ones(n)])
A = A.transpose()

result = linalg.lstsq(A, y)
# result is a 4-tuple, the
a, b = result[0]
```

```
from numpy import *
from matplotlib import pyplot
```

This lecture series introduces concepts of higher level programming. The lecture introduces essential tools to quickly and efficiently implement programming problems.

The assignments are available on the University course website . The images link to the lecture scripts. You can also run the lecture slides interactively on your browser launch binder . Video recordings of past lectures are available on this YouTube channel .

The lecture was initially created by Hans Petter Langtangen and extended by Joakim Sundnes, Ola Skavhaug, Jonathan Feinberg, Karl-Erik Holter, Vidar Tonaas Fauske, Benjamin Ragan-Kelley, Lisa Pankewitz, Sebastian Mitusch, Simon Funke, Ingeborg Gjerde, and Vegard Vinje. It is being taught on a yearly basis at University of Oslo under the name IN3110/IN4110 .