EMERITUS
INSTITUTE OF MANAGEMENT

WEEK 8

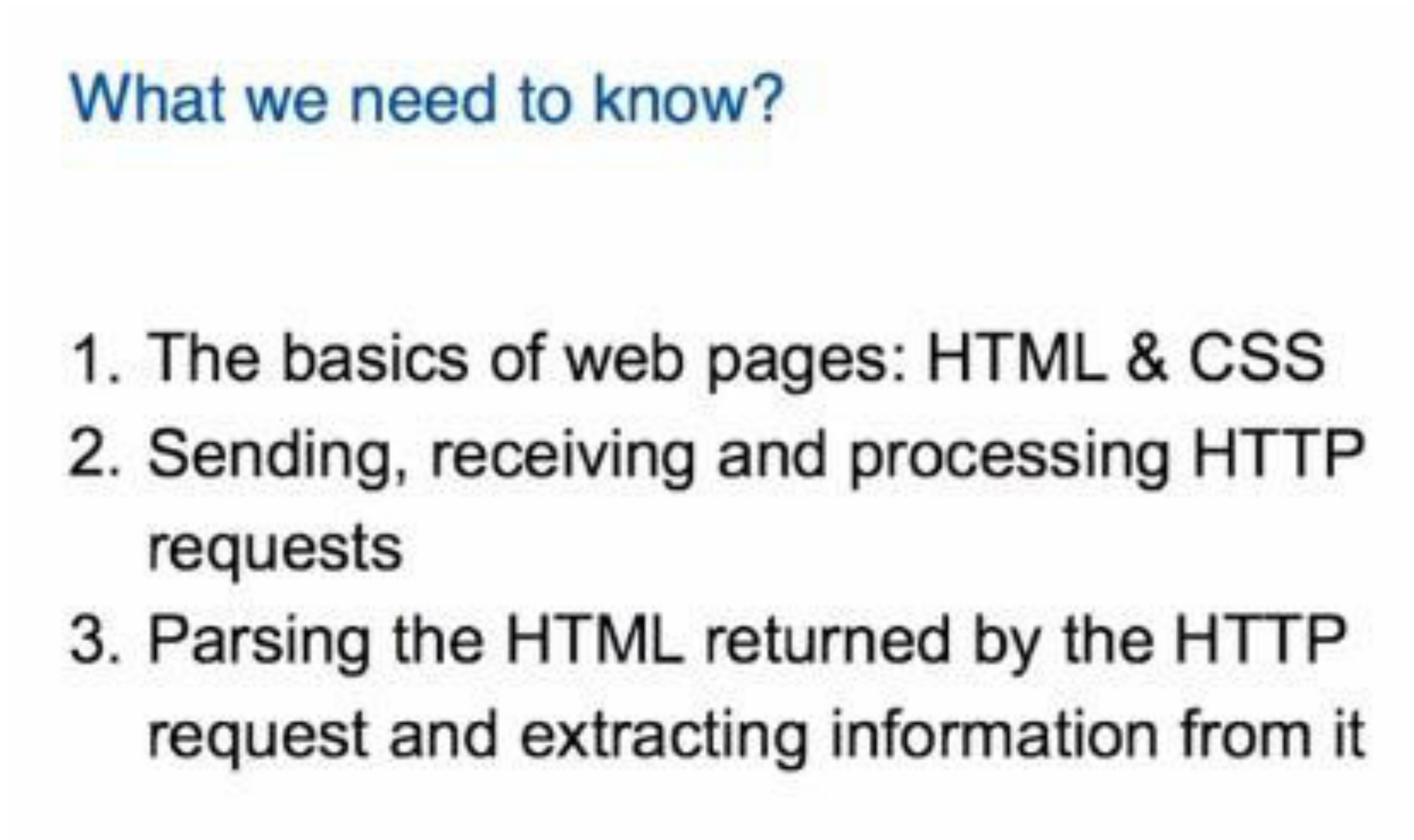
DATA EXTRACTION- GETTING DATA FROM THE INTERNET- PART 2

MITSloan Executive Education

Columbia Business School
AT THE VERY CENTER OF BUSINESS™
EXECUTIVE EDUCATION

Tuck
EXECUTIVE EDUCATION
at Dartmouth

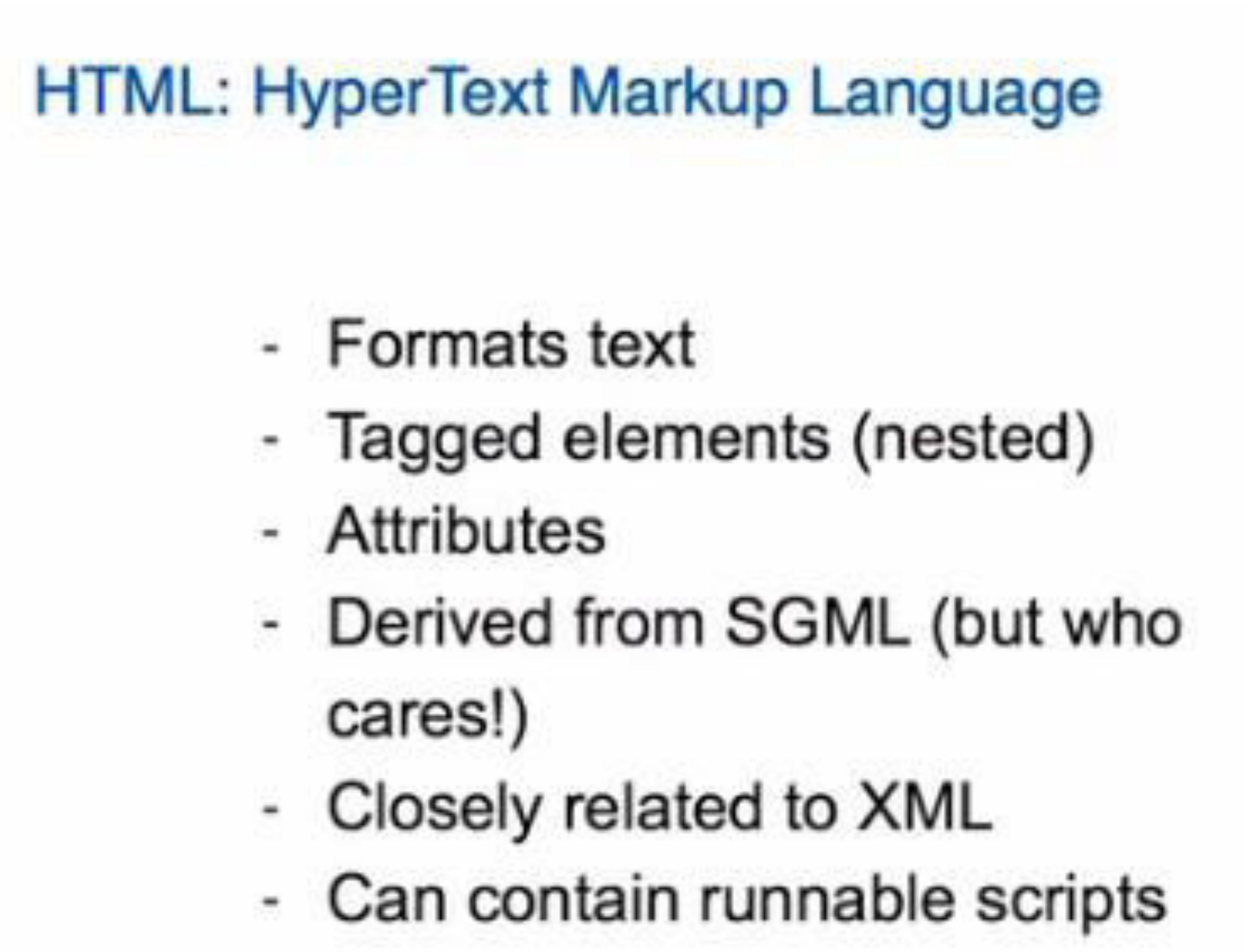
Getting Data from Web: Part 1



What we need to know?

1. The basics of web pages: HTML & CSS
2. Sending, receiving and processing HTTP requests
3. Parsing the HTML returned by the HTTP request and extracting information from it

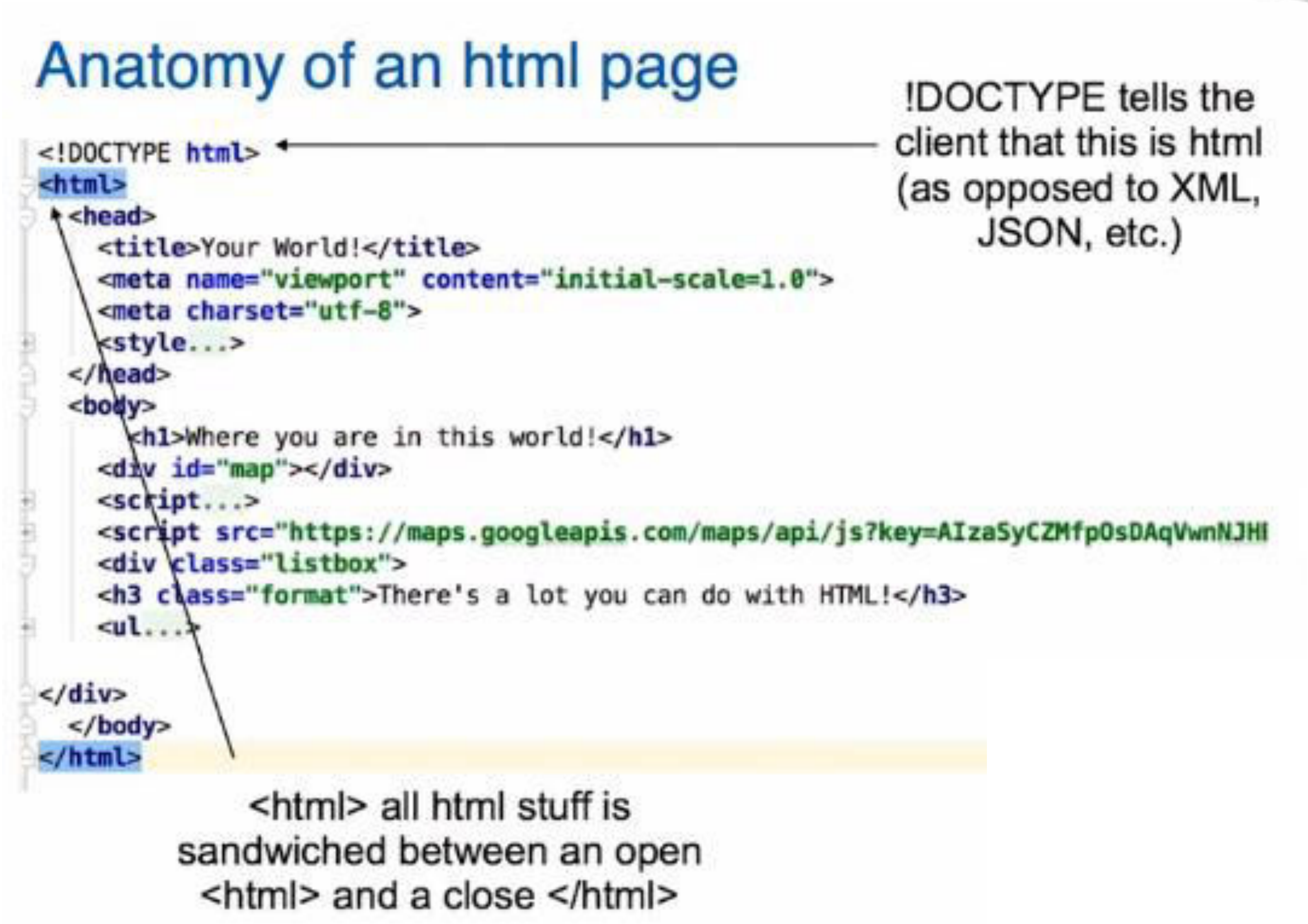
Getting Data from Web: Part 1



HTML: HyperText Markup Language

- Formats text
- Tagged elements (nested)
- Attributes
- Derived from SGML (but who cares!)
- Closely related to XML
- Can contain runnable scripts

Getting Data from Web: Part 1



Anatomy of an html page

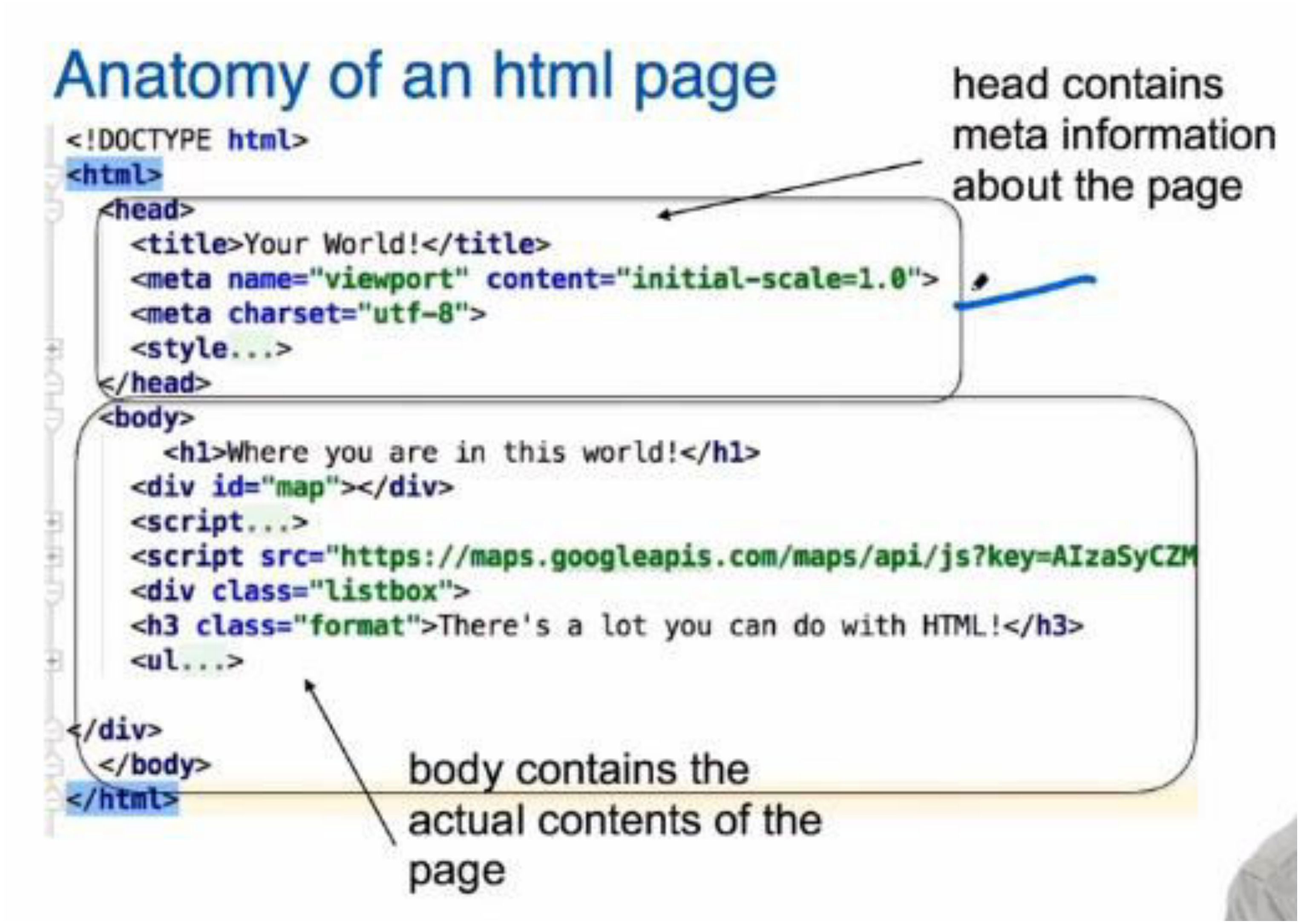
```
<!DOCTYPE html>
<html>
  <head>
    <title>Your World!</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style...>
  </head>
  <body>
      <h1>Where you are in this world!</h1>
    <div id="map"></div>
    <script...>
    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCZMfpOsDAqVwnNJHI
    <div class="listbox">
    <h3 class="format">There's a lot you can do with HTML!</h3>
    <ul...>

</div>
  </body>
</html>
```

!DOCTYPE tells the client that this is html (as opposed to XML, JSON, etc.)

<html> all html stuff is sandwiched between an open <html> and a close </html>

Getting Data from Web: Part 1



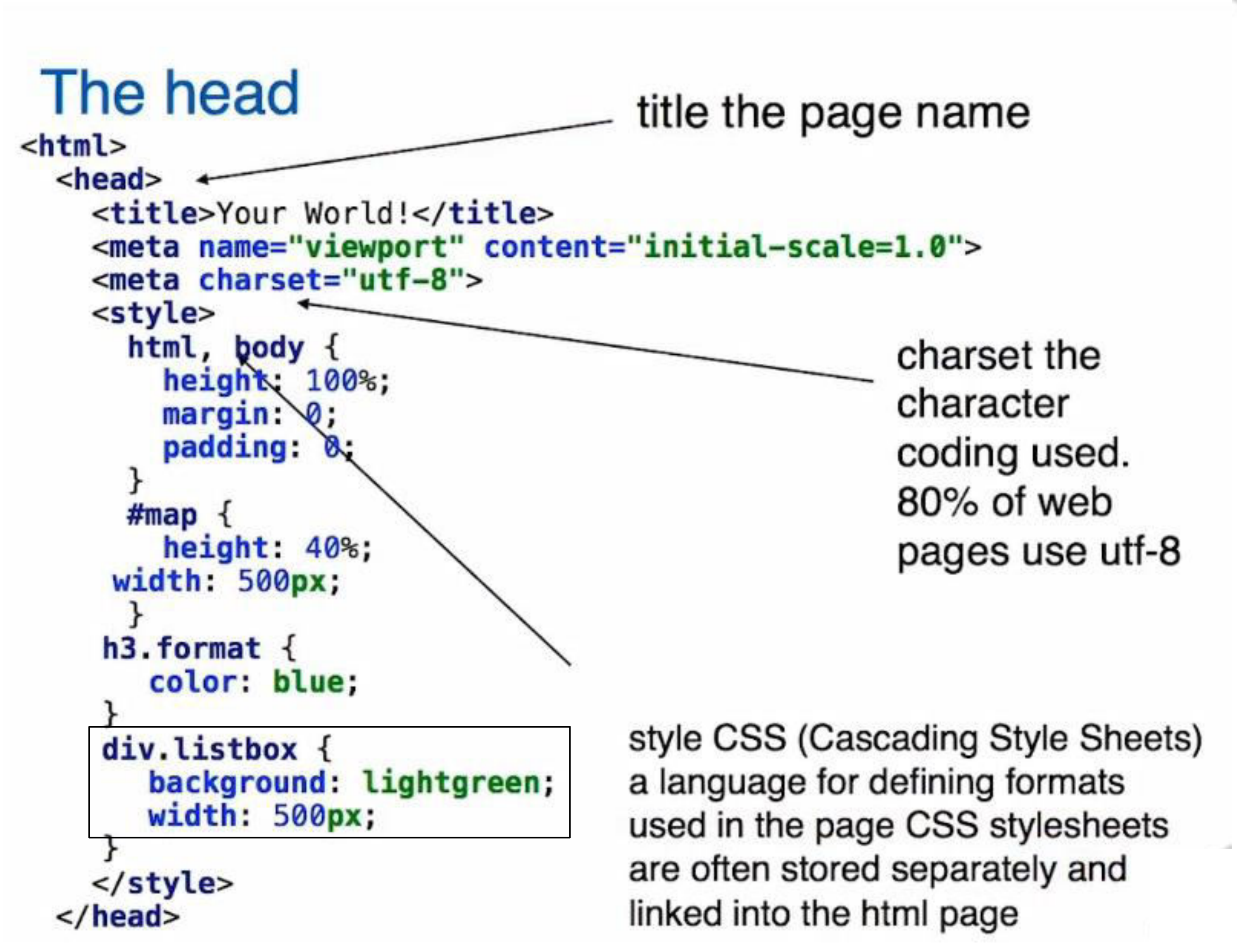
Anatomy of an html page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Your World!</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style...>
  </head>
  <body>
      <h1>Where you are in this world!</h1>
    <div id="map"></div>
    <script...>
    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCZM
    <div class="listbox">
    <h3 class="format">There's a lot you can do with HTML!</h3>
    <ul...>

</div>
  </body>
</html>
```

head contains meta information about the page
body contains the actual contents of the page

Getting Data from Web: Part 1



The head

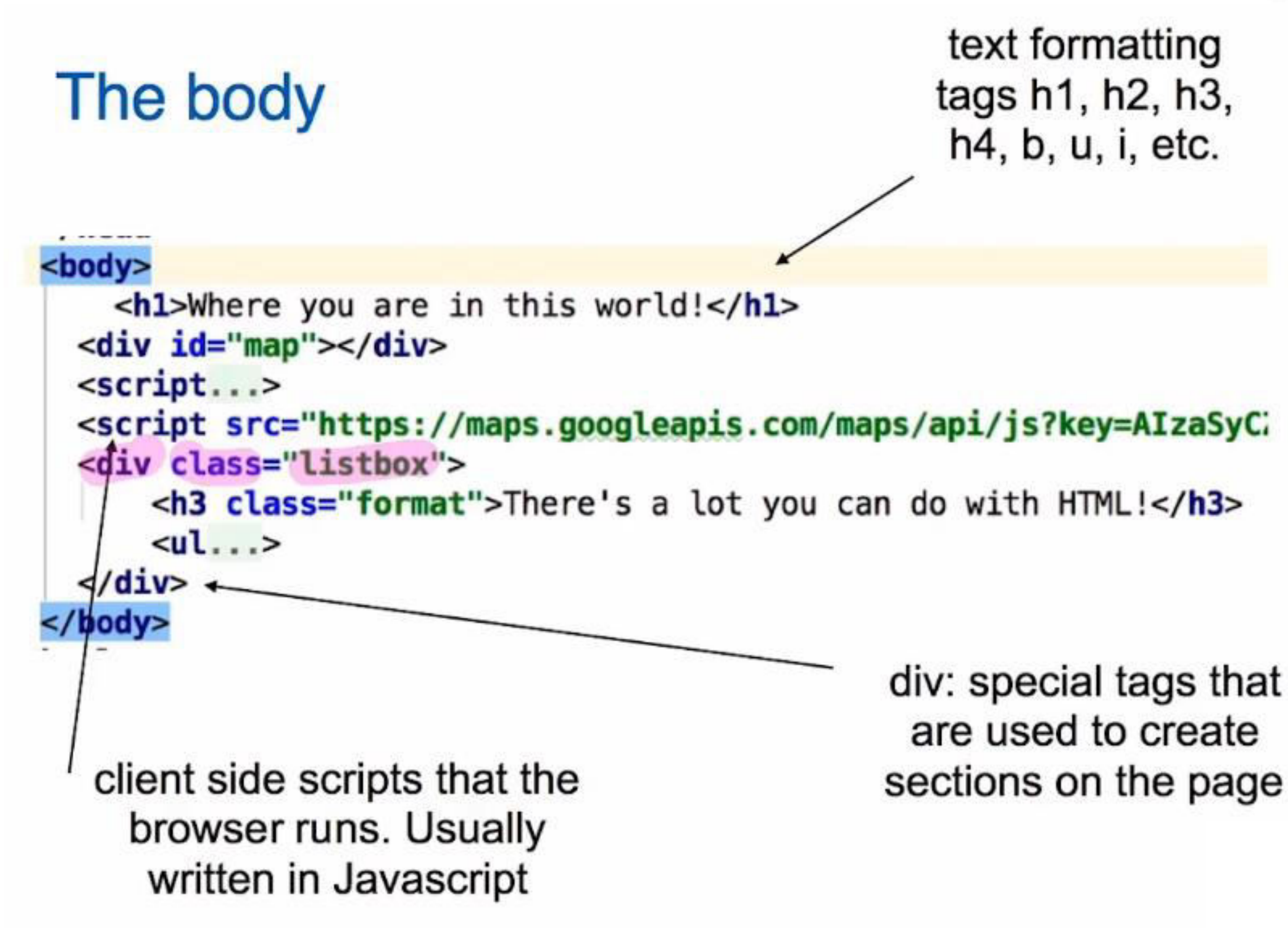
title the page name

```
<html>
  <head>
    <title>Your World!</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style>
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
      #map {
        height: 40%;
     width: 500px;
      }
     h3.format {
        color: blue;
     }
    div.listbox {
        background: lightgreen;
        width: 500px;
     }
    </style>
  </head>
```

charset the character coding used. 80% of web pages use utf-8

style CSS (Cascading Style Sheets) a language for defining formats used in the page CSS stylesheets are often stored separately and linked into the html page

Getting Data from Web: Part 1



The body

text formatting tags h1, h2, h3, h4, b, u, i, etc.

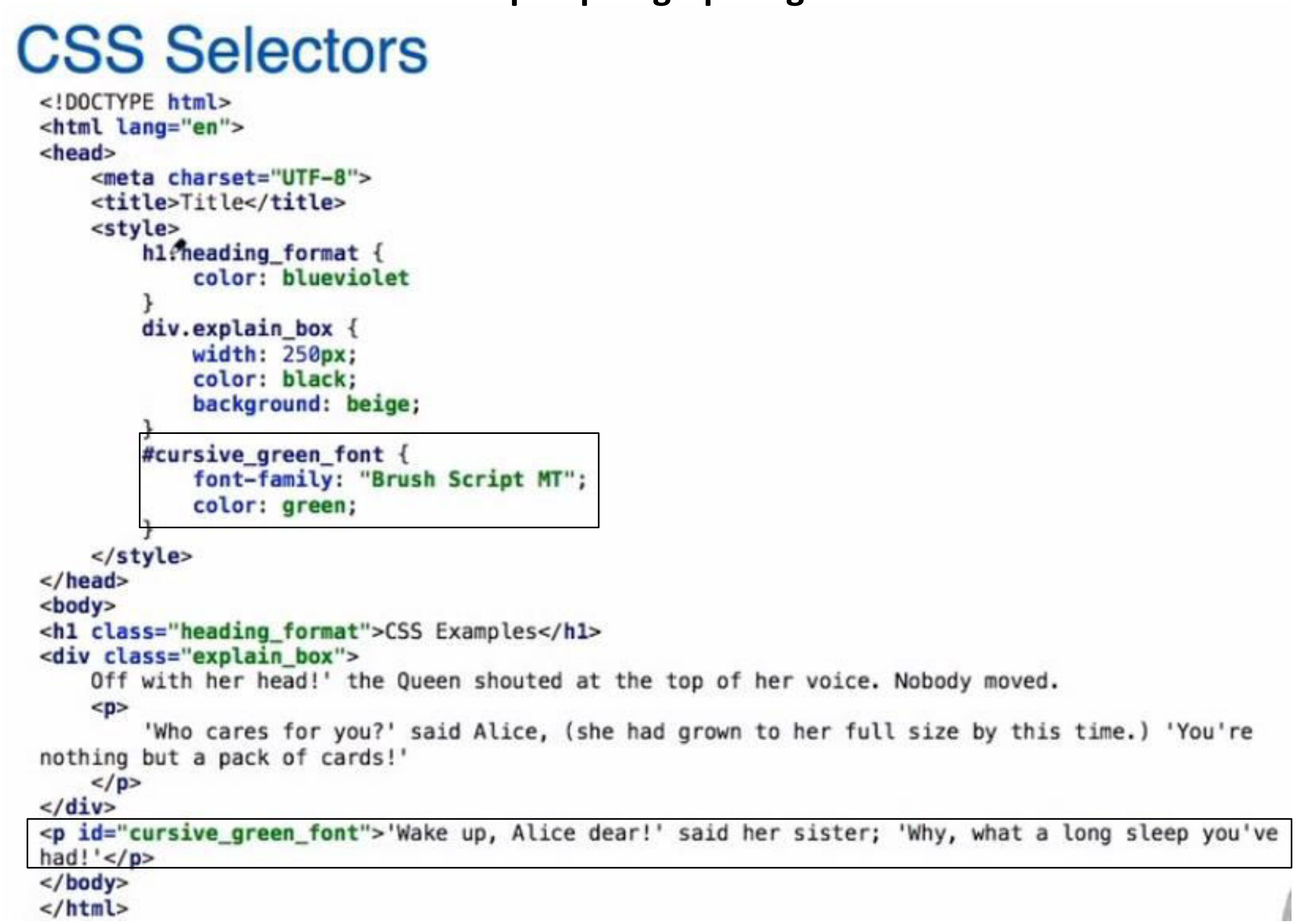
```
<body>
    <h1>Where you are in this world!</h1>
  <div id="map"></div>
  <script...>
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC
  <div class="listbox">
      <h3 class="format">There's a lot you can do with HTML!</h3>
      <ul...>
  </div>
</body>
```

div: special tags that are used to create sections on the page

client side scripts that the browser runs. Usually written in Javascript

Getting Data from Web: Part 2

<p> - paragraph tag

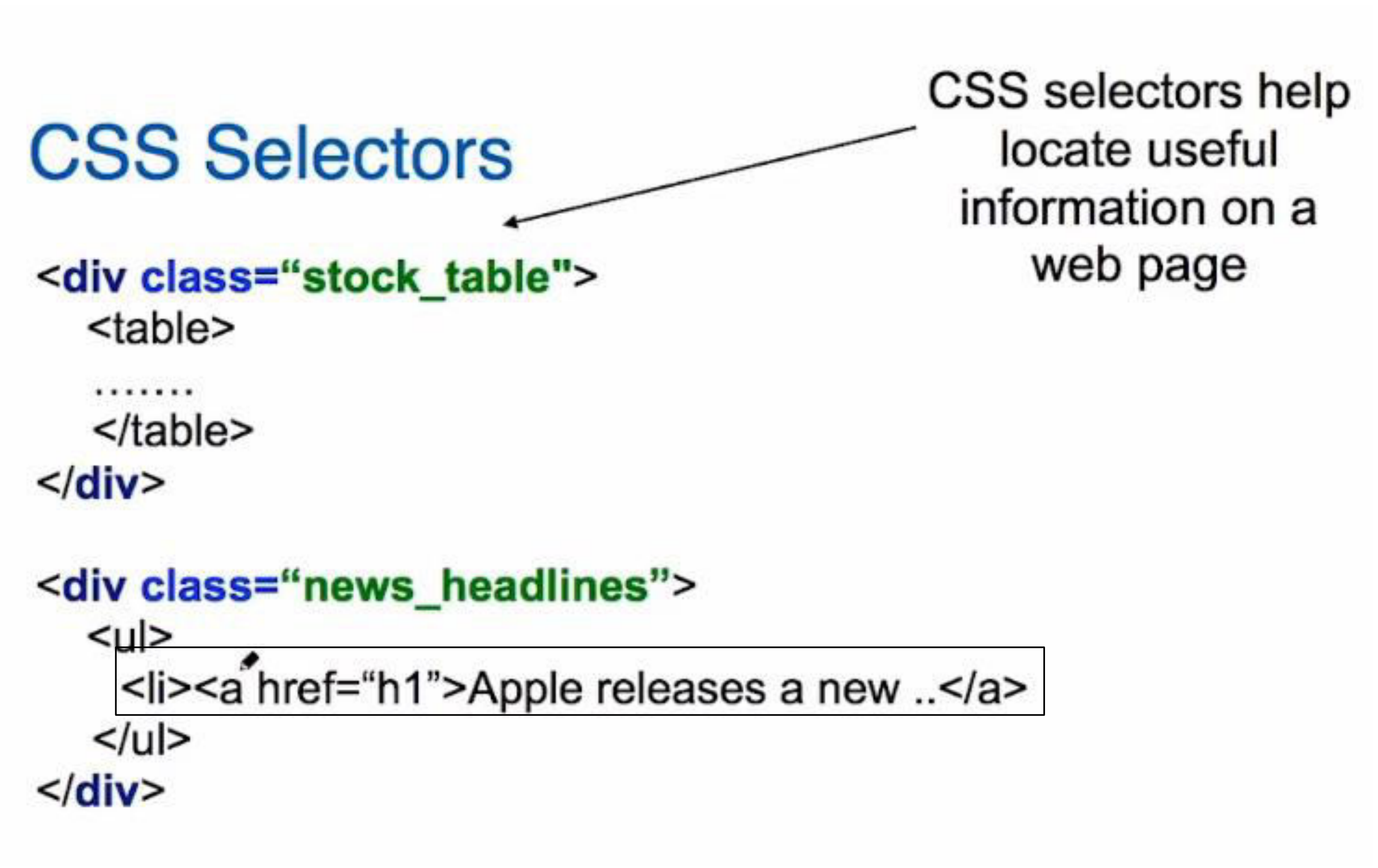


CSS Selectors

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
        h1.heading_format {
            color: blueviolet
        }
        div.explain_box {
            width: 250px;
            color: black;
            background: beige;
        }
        #cursive_green_font {
            font-family: "Brush Script MT";
            color: green;
        }
    </style>
</head>
<body>
<h1 class="heading_format">CSS Examples</h1>
<div class="explain_box">
    Off with her head!' the Queen shouted at the top of her voice. Nobody moved.
    <p>
        'Who cares for you?' said Alice, (she had grown to her full size by this time.) 'You're
nothing but a pack of cards!'
    </p>
</div>
<p id="cursive_green_font">'Wake up, Alice dear!' said her sister; 'Why, what a long sleep you've
had!'</p>
</body>
</html>
```

Getting Data from Web: Part 2

<a> - annotate tag



CSS Selectors

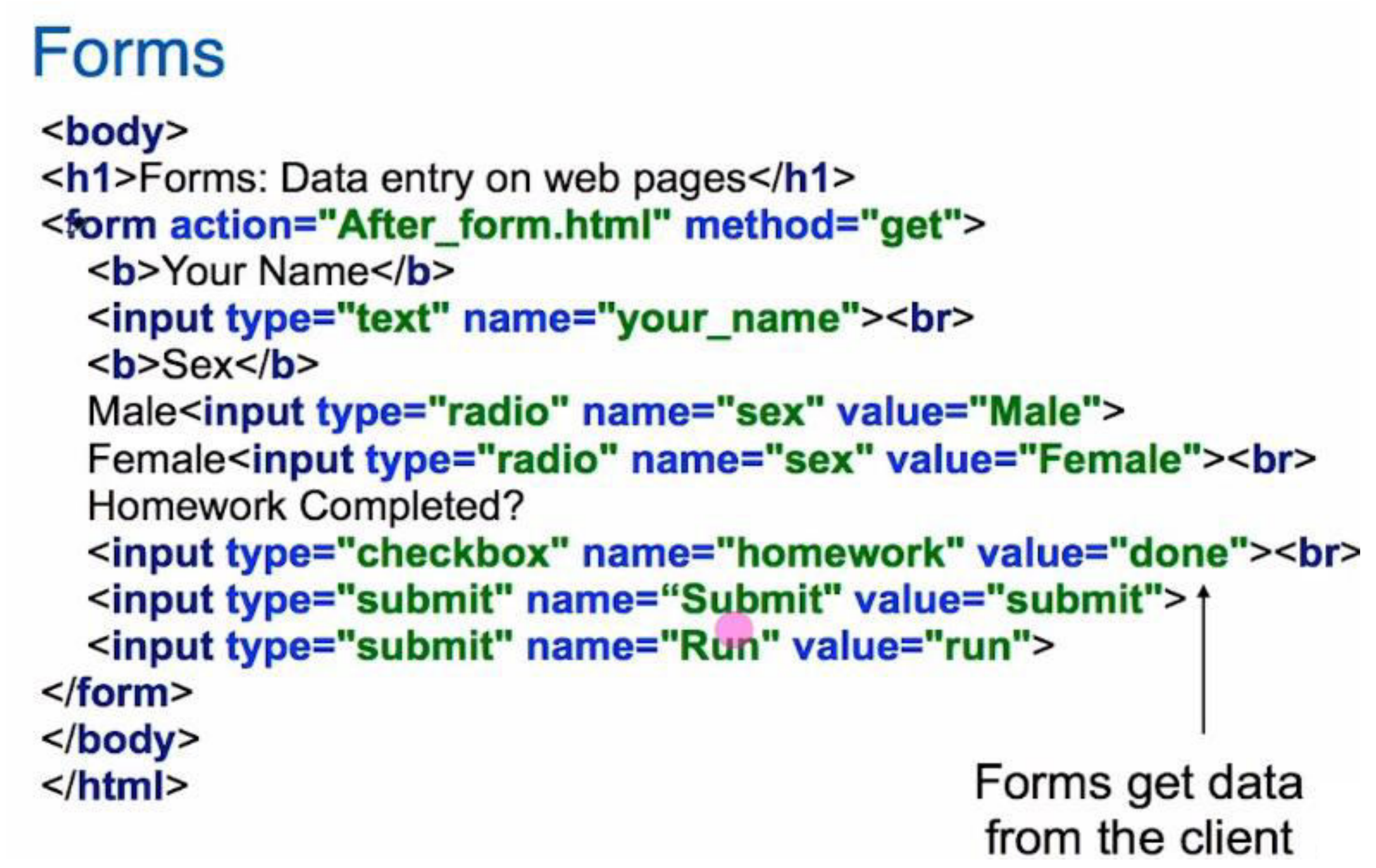
CSS selectors help locate useful information on a web page

```
<div class="stock_table">
    <table>
    .......
    </table>
</div>

<div class=“news_headlines”>
    <ul>
    <li><a href=“h1”>Apple releases a new ..</a>
    </ul>
</div>
```

Getting Data from Web: Part 2

Under ‘forms’ tag, ‘input type’ tag holds different types of data like text, numbers, radio buttons, checkbox etc.

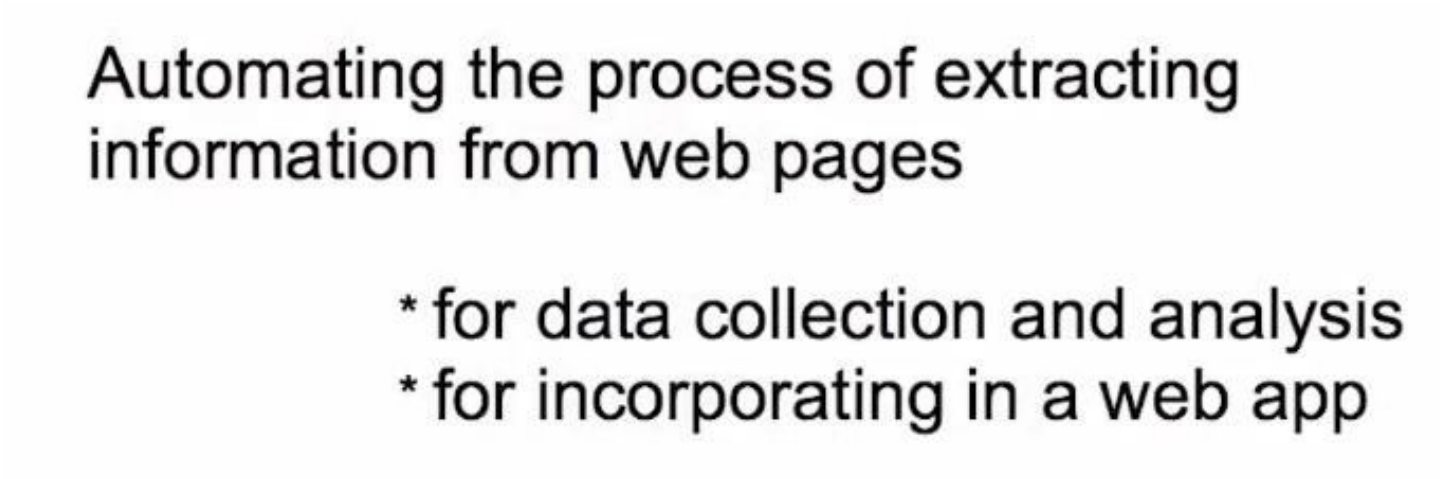


Forms

```
<body>
<h1>Forms: Data entry on web pages</h1>
<form action="After_form.html" method="get">
   <b>Your Name</b>
   <input type="text" name="your_name"><br>
   <b>Sex</b>
   Male<input type="radio" name="sex" value="Male">
   Female<input type="radio" name="sex" value="Female"><br>
   Homework Completed?
   <input type="checkbox" name="homework" value="done"><br>
   <input type="submit" name="Submit" value="submit">
   <input type="submit" name="Run" value="run">
</form>
</body>
</html>
```

Forms get data from the client

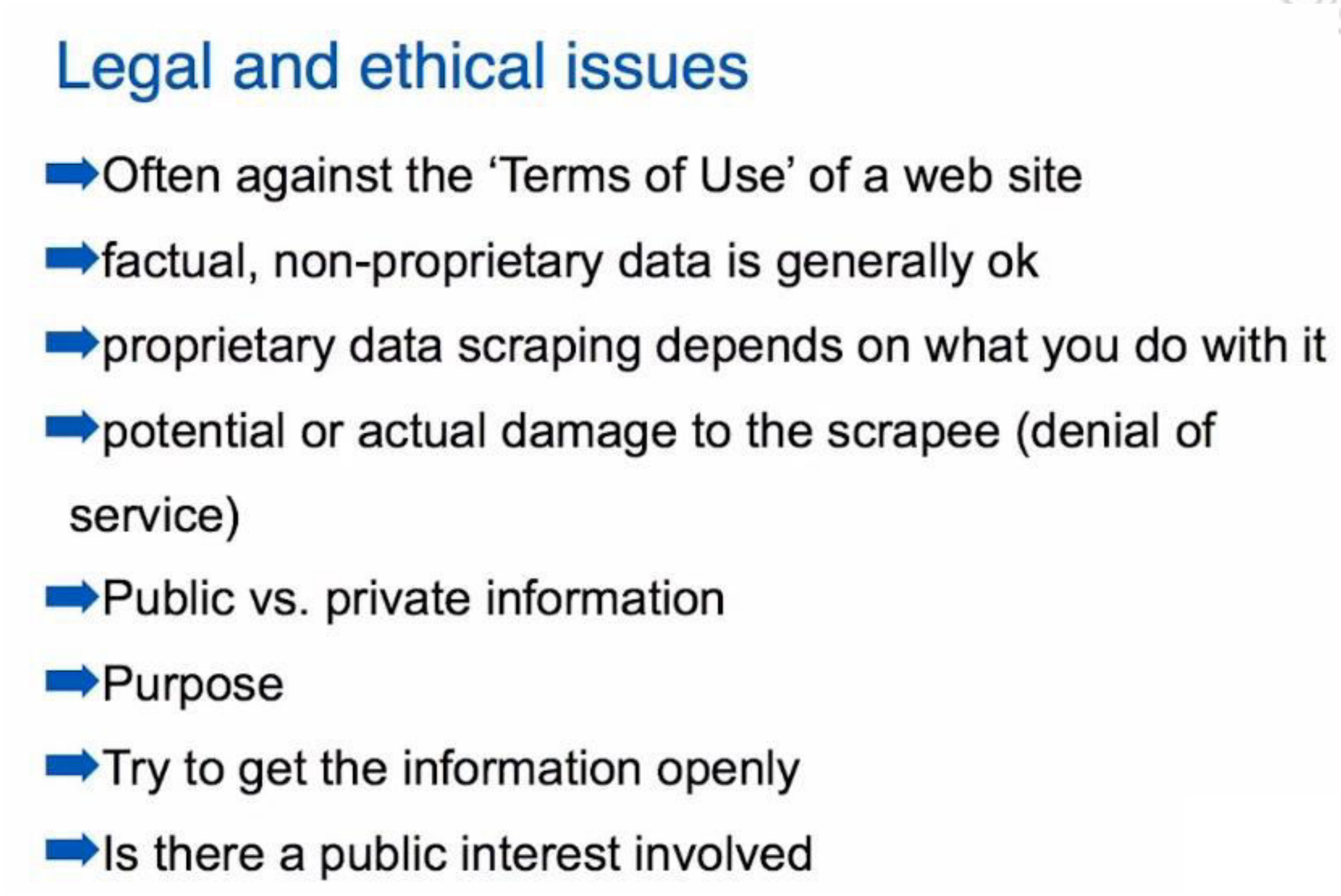
Web Scraping



Automating the process of extracting information from web pages

* for data collection and analysis
* for incorporating in a web app

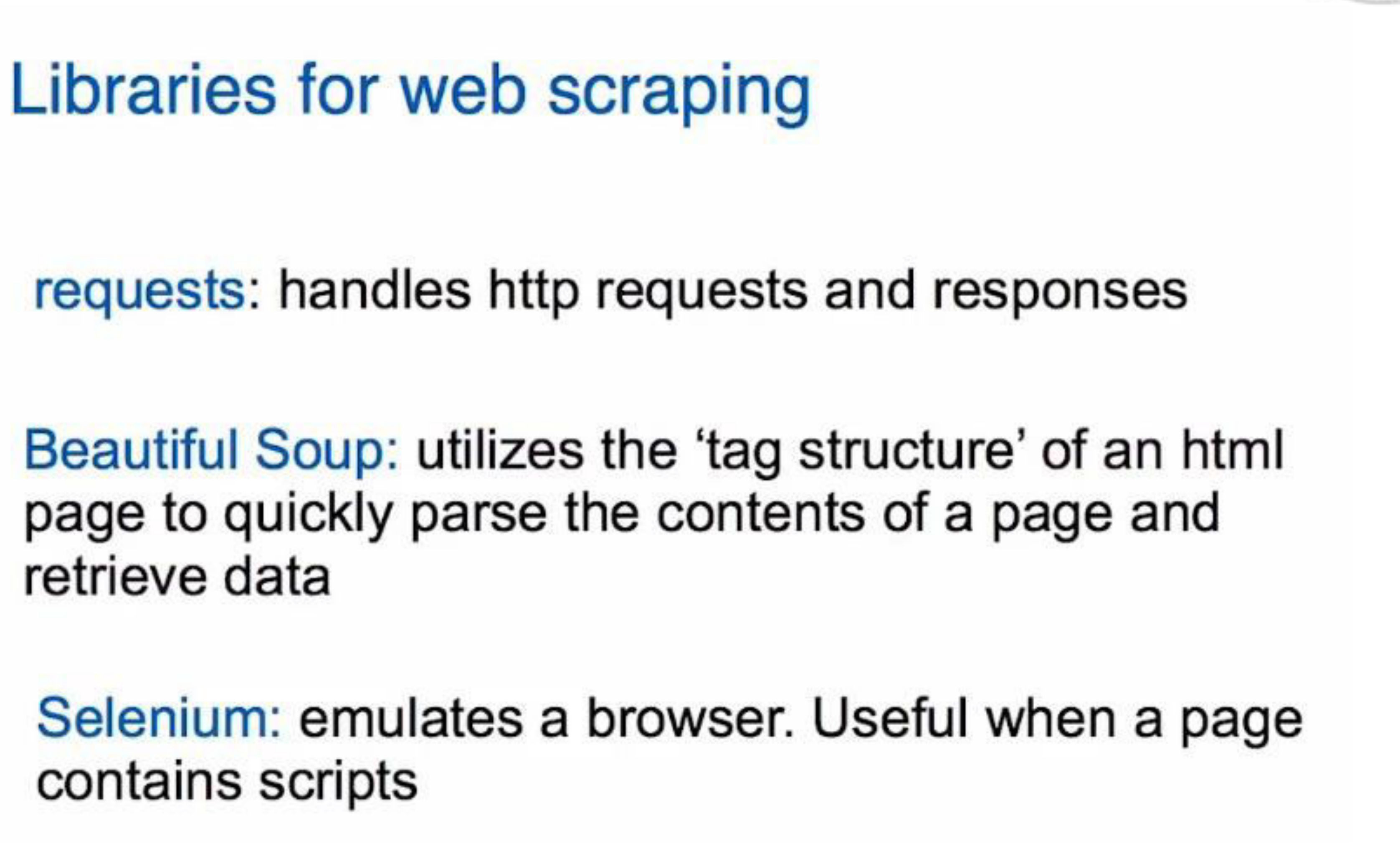
Web Scraping: Issues



Legal and ethical issues

- Often against the ‘Terms of Use’ of a web site
- factual, non-proprietary data is generally ok
- proprietary data scraping depends on what you do with it
- potential or actual damage to the scrapee (denial of service)
- Public vs. private information
- Purpose
- Try to get the information openly
- Is there a public interest involved

Web Scraping: Libraries



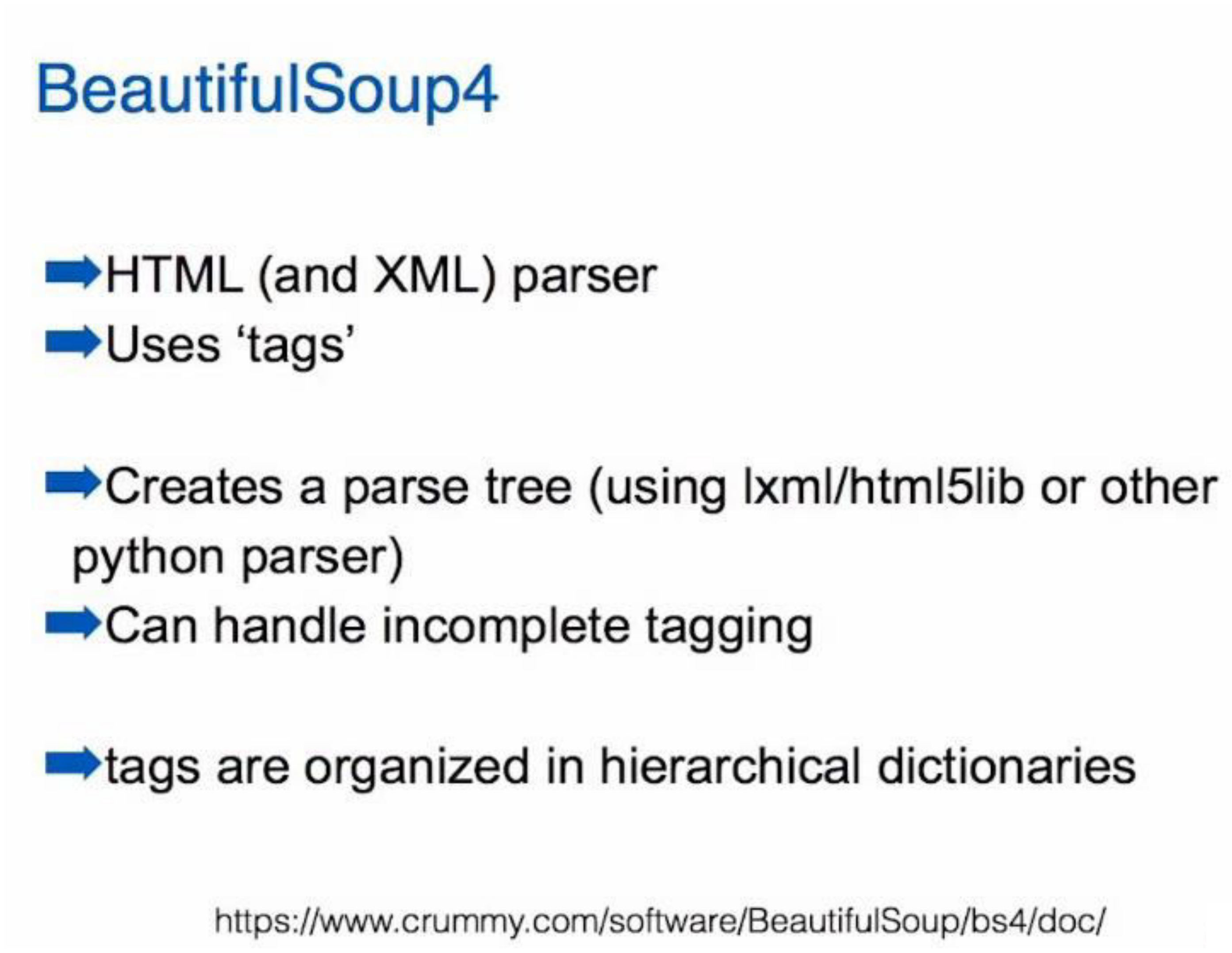
Libraries for web scraping

requests: handles http requests and responses

Beautiful Soup: utilizes the ‘tag structure’ of an html page to quickly parse the contents of a page and retrieve data

Selenium: emulates a browser. Useful when a page contains scripts

Web Scraping



BeautifulSoup4

➡HTML (and XML) parser

➡Uses ‘tags’

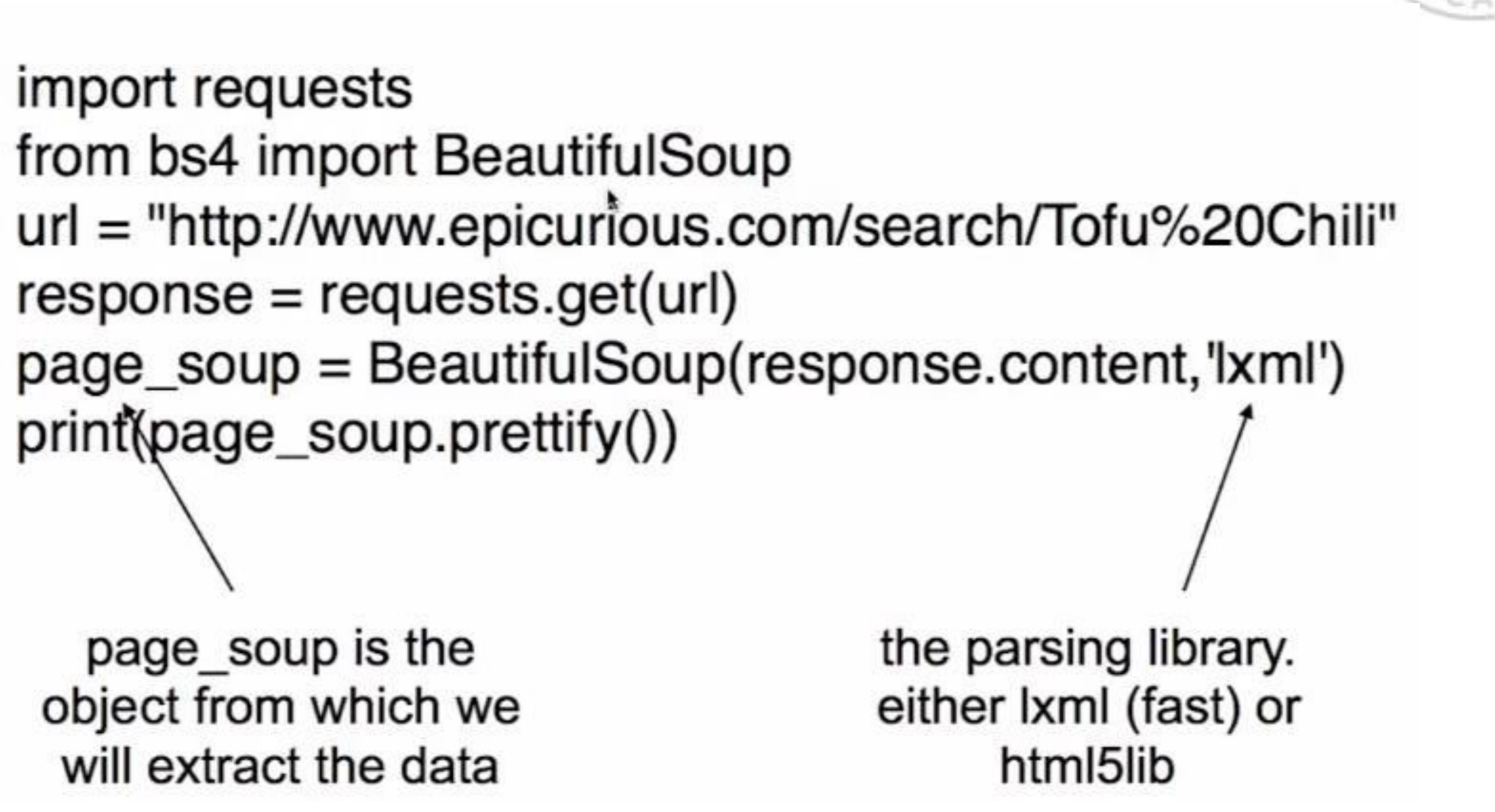
➡Creates a parse tree (using lxml/html5lib or other python parser)

➡Can handle incomplete tagging

➡tags are organized in hierarchical dictionaries

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

BeautifulSoup4: Part 1

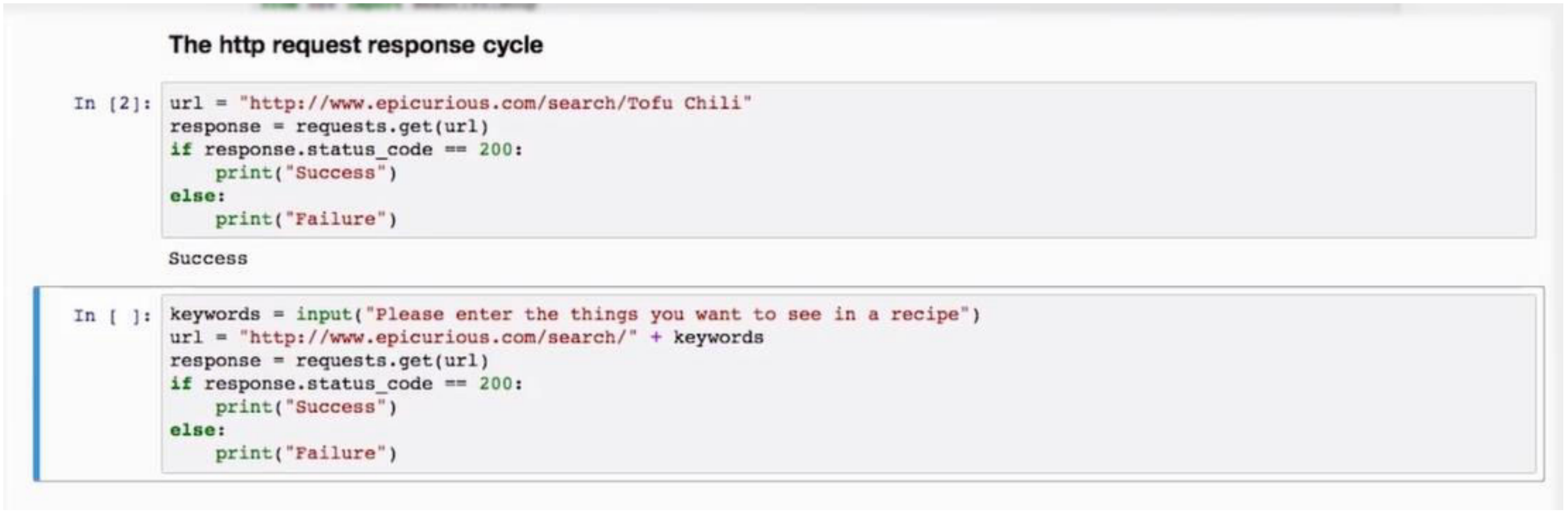


```
import requests
from bs4 import BeautifulSoup
url = "http://www.epicurious.com/search/Tofu%20Chili"
response = requests.get(url)
page_soup = BeautifulSoup(response.content,'lxml')
print(page_soup.prettify())
```

page_soup is the object from which we will extract the data

the parsing library. either lxml (fast) or html5lib

BeautifulSoup4: Part 1



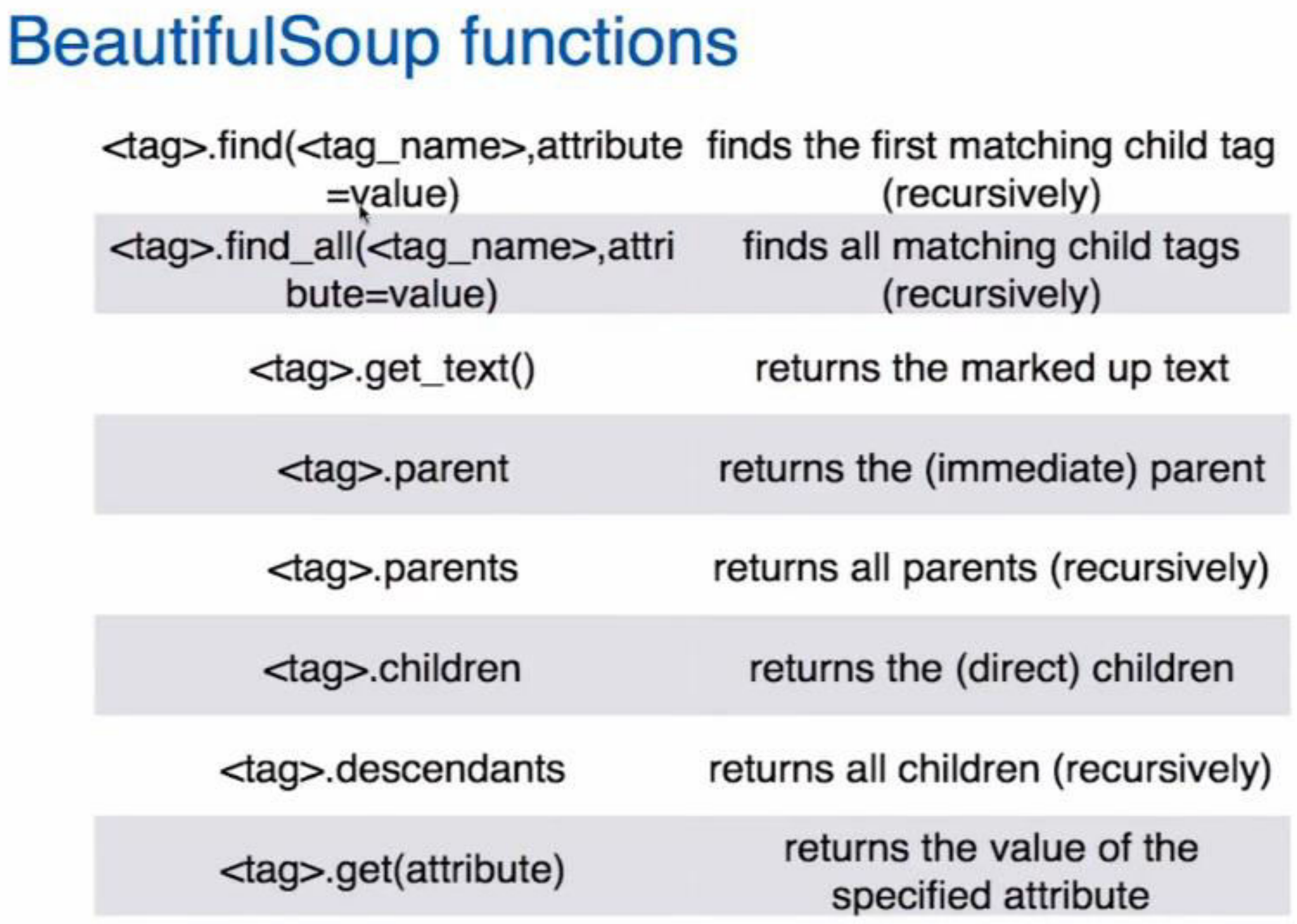
The http request response cycle

```
In [2]: url = "http://www.epicurious.com/search/Tofu Chili"
        response = requests.get(url)
        if response.status_code == 200:
            print("Success")
        else:
            print("Failure")

        Success

In [ ]: keywords = input("Please enter the things you want to see in a recipe")
        url = "http://www.epicurious.com/search/" + keywords
        response = requests.get(url)
        if response.status_code == 200:
            print("Success")
        else:
            print("Failure")
```

BeautifulSoup4: Part 2

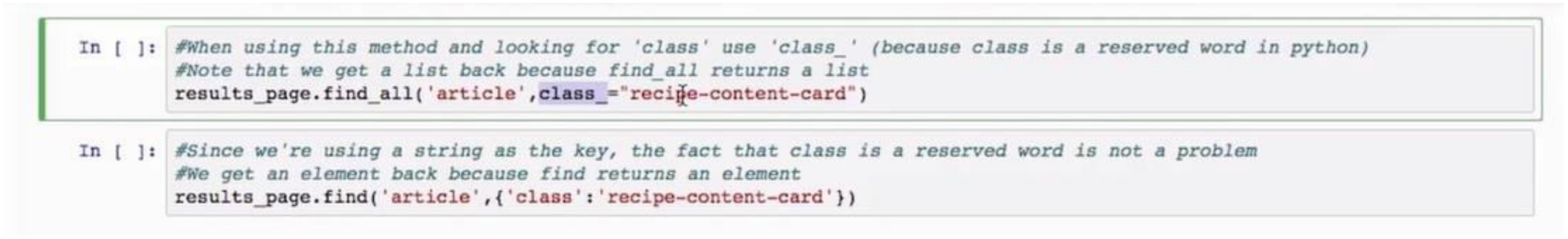


BeautifulSoup functions

<tag>.find(<tag_name>,attribute =value)	finds the first matching child tag (recursively)
<tag>.find_all(<tag_name>,attribute=value)	finds all matching child tags (recursively)
<tag>.get_text()	returns the marked up text
<tag>.parent	returns the (immediate) parent
<tag>.parents	returns all parents (recursively)
<tag>.children	returns the (direct) children
<tag>.descendants	returns all children (recursively)
<tag>.get(attribute)	returns the value of the specified attribute

BeautifulSoup4: Part 3

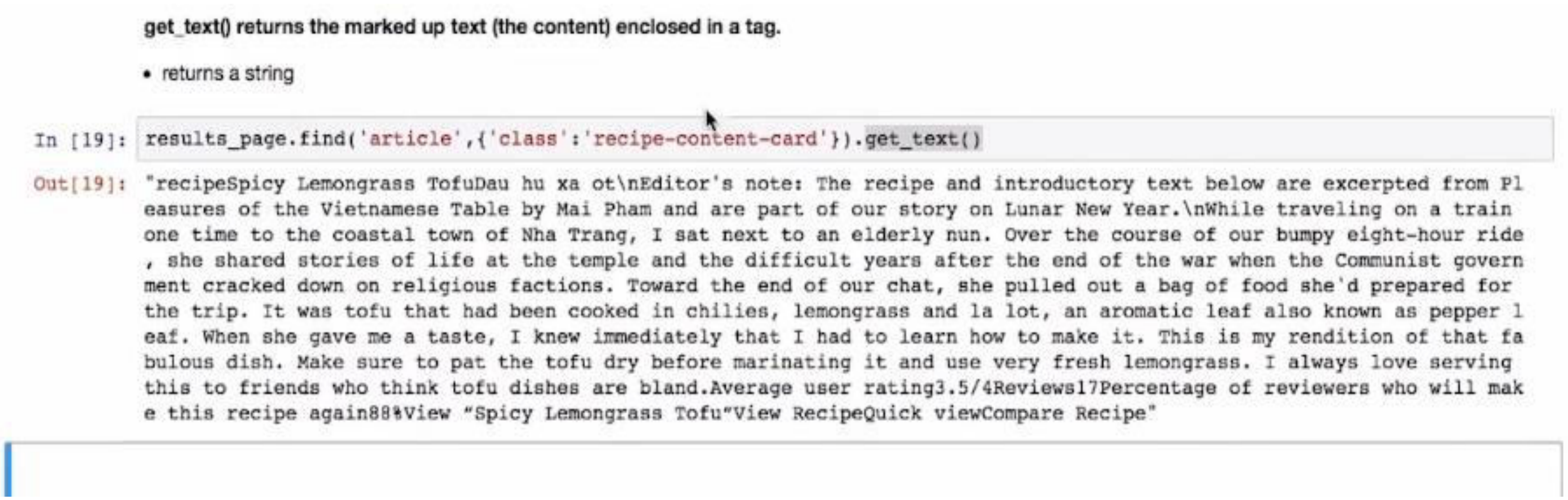
Obtain specific tags and values visible on site



```
In [ ]: #When using this method and looking for 'class' use 'class_' (because class is a reserved word in python)
        #Note that we get a list back because find_all returns a list
        results_page.find_all('article',class_="recipe-content-card")

In [ ]: #Since we're using a string as the key, the fact that class is a reserved word is not a problem
        #We get an element back because find returns an element
        results_page.find('article',{'class':'recipe-content-card'})
```

Add .get text function to get content visible on the site



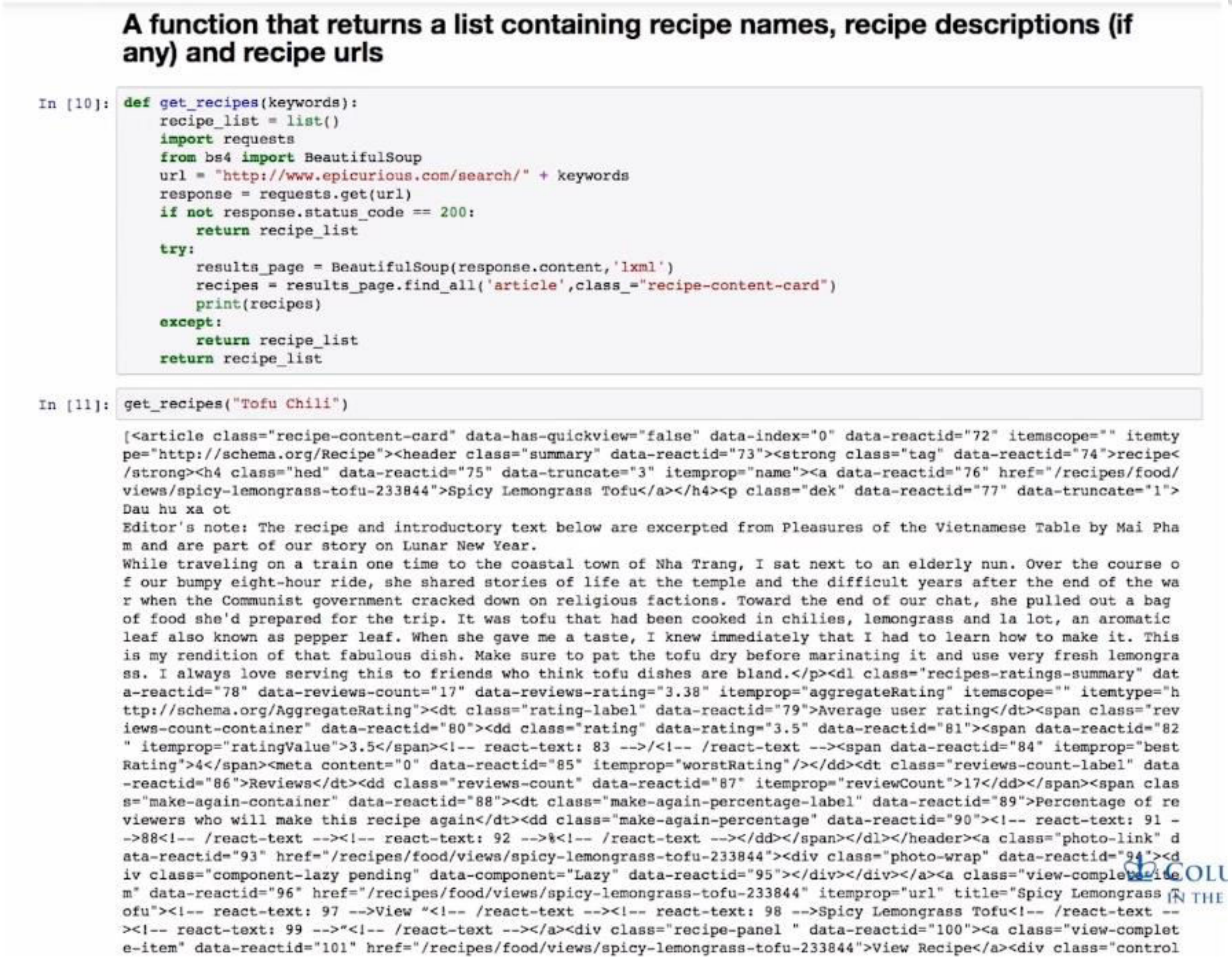
get_text() returns the marked up text (the content) enclosed in a tag.

- returns a string

```
In [19]: results_page.find('article',{'class':'recipe-content-card'}).get_text()

Out[19]: "recipeSpicy Lemongrass TofuDau hu xa ot\nEditor's note: The recipe and introductory text below are excerpted from Pl
         easures of the Vietnamese Table by Mai Pham and are part of our story on Lunar New Year.\nWhile traveling on a train
         one time to the coastal town of Nha Trang, I sat next to an elderly nun. Over the course of our bumpy eight-hour ride
         , she shared stories of life at the temple and the difficult years after the end of the war when the Communist govern
         ment cracked down on religious factions. Toward the end of our chat, she pulled out a bag of food she'd prepared for
         the trip. It was tofu that had been cooked in chilies, lemongrass and la lot, an aromatic leaf also known as pepper l
         eaf. When she gave me a taste, I knew immediately that I had to learn how to make it. This is my rendition of that fa
         bulous dish. Make sure to pat the tofu dry before marinating it and use very fresh lemongrass. I always love serving
         this to friends who think tofu dishes are bland.Average user rating3.5/4Reviews17Percentage of reviewers who will mak
         e this recipe again88%View "Spicy Lemongrass Tofu"View RecipeQuick viewCompare Recipe"
```

Epicurious Example: Part 1



A function that returns a list containing recipe names, recipe descriptions (if any) and recipe urls

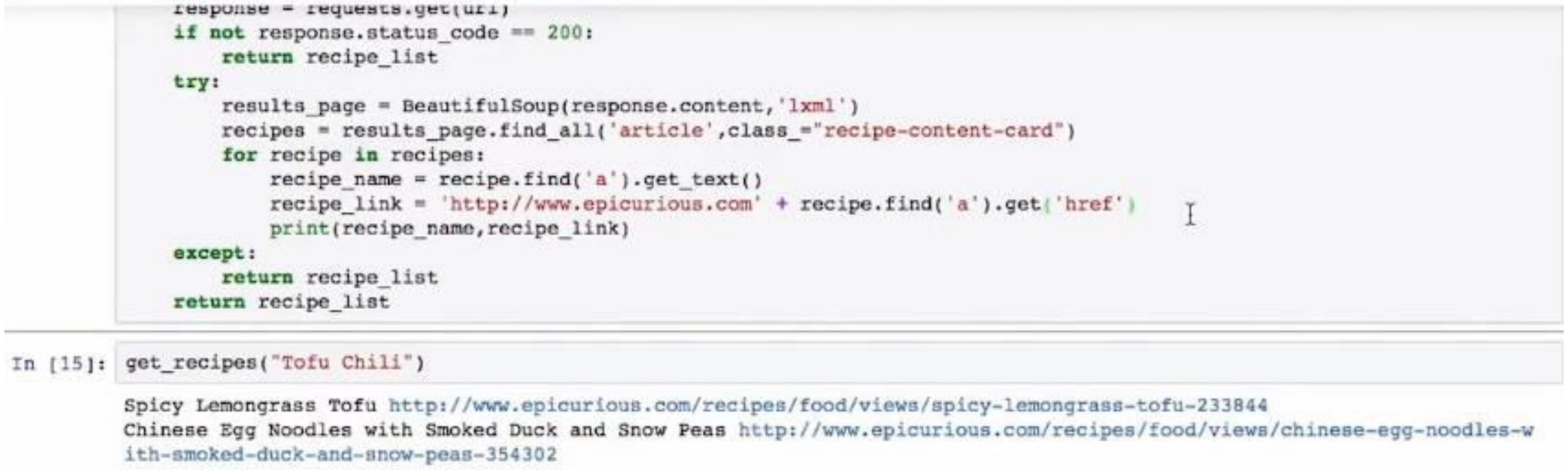
```
In [10]: def get_recipes(keywords):
             recipe_list = list()
             import requests
             from bs4 import BeautifulSoup
             url = "http://www.epicurious.com/search/" + keywords
             response = requests.get(url)
             if not response.status_code == 200:
                 return recipe_list
             try:
                 results_page = BeautifulSoup(response.content,'lxml')
                 recipes = results_page.find_all('article',class_="recipe-content-card")
                 print(recipes)
             except:
                 return recipe_list
             return recipe_list
```

```
In [11]: get_recipes("Tofu Chili")
```

```
[<article class="recipe-content-card" data-has-quickview="false" data-index="0" data-reactid="72" itemscope="" itemty
pe="http://schema.org/Recipe"><header class="summary" data-reactid="73"><strong class="tag" data-reactid="74">recipe<
/strong><h4 class="hed" data-reactid="75" data-truncate="3" itemprop="name"><a data-reactid="76" href="/recipes/food/
views/spicy-lemongrass-tofu-233844">Spicy Lemongrass Tofu</a></h4><p class="dek" data-reactid="77" data-truncate="1">
Dau hu xa ot
Editor's note: The recipe and introductory text below are excerpted from Pleasures of the Vietnamese Table by Mai Pha
m and are part of our story on Lunar New Year.
While traveling on a train one time to the coastal town of Nha Trang, I sat next to an elderly nun. Over the course o
f our bumpy eight-hour ride, she shared stories of life at the temple and the difficult years after the end of the wa
r when the Communist government cracked down on religious factions. Toward the end of our chat, she pulled out a bag
of food she'd prepared for the trip. It was tofu that had been cooked in chilies, lemongrass and la lot, an aromatic
leaf also known as pepper leaf. When she gave me a taste, I knew immediately that I had to learn how to make it. This
is my rendition of that fabulous dish. Make sure to pat the tofu dry before marinating it and use very fresh lemongra
ss. I always love serving this to friends who think tofu dishes are bland.</p><dl class="recipes-ratings-summary" dat
a-reactid="78" data-reviews-count="17" data-reviews-rating="3.38" itemprop="aggregateRating" itemscope="" itemtype="h
ttp://schema.org/AggregateRating"><dt class="rating-label" data-reactid="79">Average user rating</dt><span class="rev
iews-count-container" data-reactid="80"><dd class="rating" data-rating="3.5" data-reactid="81"><span data-reactid="82
" itemprop="ratingValue">3.5</span><!-- react-text: 83 -->/<!-- /react-text --><span data-reactid="84" itemprop="best
Rating">4</span><meta content="0" data-reactid="85" itemprop="worstRating"/></dd><dt class="reviews-count-label" data
-reactid="86">Reviews</dt><dd class="reviews-count" data-reactid="87" itemprop="reviewCount">17</dd></span><span clas
s="make-again-container" data-reactid="88"><dt class="make-again-percentage-label" data-reactid="89">Percentage of re
viewers who will make this recipe again</dt><dd class="make-again-percentage" data-reactid="90"><!-- react-text: 91 -
->88<!-- /react-text --><!-- react-text: 92 -->%<!-- /react-text --></dd></span></dl></header><a class="photo-link" d
ata-reactid="93" href="/recipes/food/views/spicy-lemongrass-tofu-233844"><div class="photo-wrap" data-reactid="94"><d
iv class="component-lazy pending" data-component="Lazy" data-reactid="95"></div></div></a><a class="view-complete-ite
m" data-reactid="96" href="/recipes/food/views/spicy-lemongrass-tofu-233844" itemprop="url" title="Spicy Lemongrass T
ofu"><!-- react-text: 97 -->View “<!-- /react-text --><!-- react-text: 98 -->Spicy Lemongrass Tofu<!-- /react-text --
><!-- react-text: 99 -->”<!-- /react-text --></a><div class="recipe-panel " data-reactid="100"><a class="view-complet
e-item" data-reactid="101" href="/recipes/food/views/spicy-lemongrass-tofu-233844">View Recipe</a><div class="control
```

Epicurious Example: Part 2

Obtain clickable links of URL

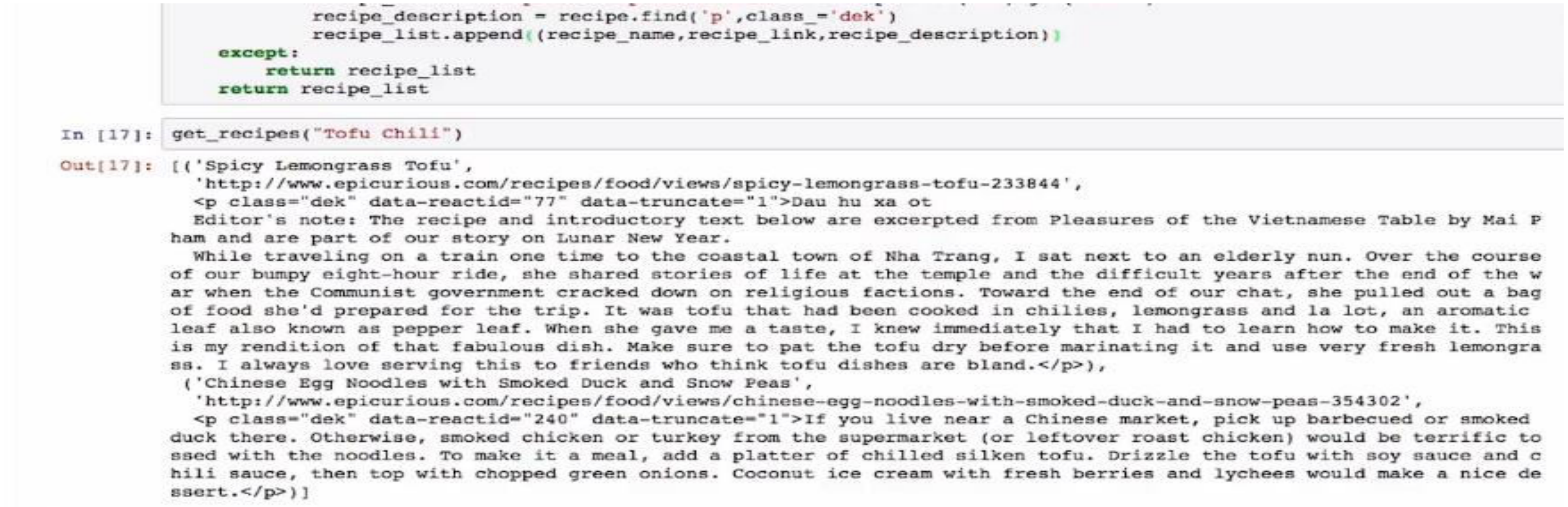


```
    response = requests.get(url)
    if not response.status_code == 200:
        return recipe_list
    try:
        results_page = BeautifulSoup(response.content,'lxml')
        recipes = results_page.find_all('article',class_="recipe-content-card")
        for recipe in recipes:
            recipe_name = recipe.find('a').get_text()
            recipe_link = 'http://www.epicurious.com' + recipe.find('a').get('href')
            print(recipe_name,recipe_link)
    except:
        return recipe_list
    return recipe_list
```

```
In [15]: get_recipes("Tofu Chili")

         Spicy Lemongrass Tofu http://www.epicurious.com/recipes/food/views/spicy-lemongrass-tofu-233844
         Chinese Egg Noodles with Smoked Duck and Snow Peas http://www.epicurious.com/recipes/food/views/chinese-egg-noodles-w
         ith-smoked-duck-and-snow-peas-354302
```

Obtain description of the recipe



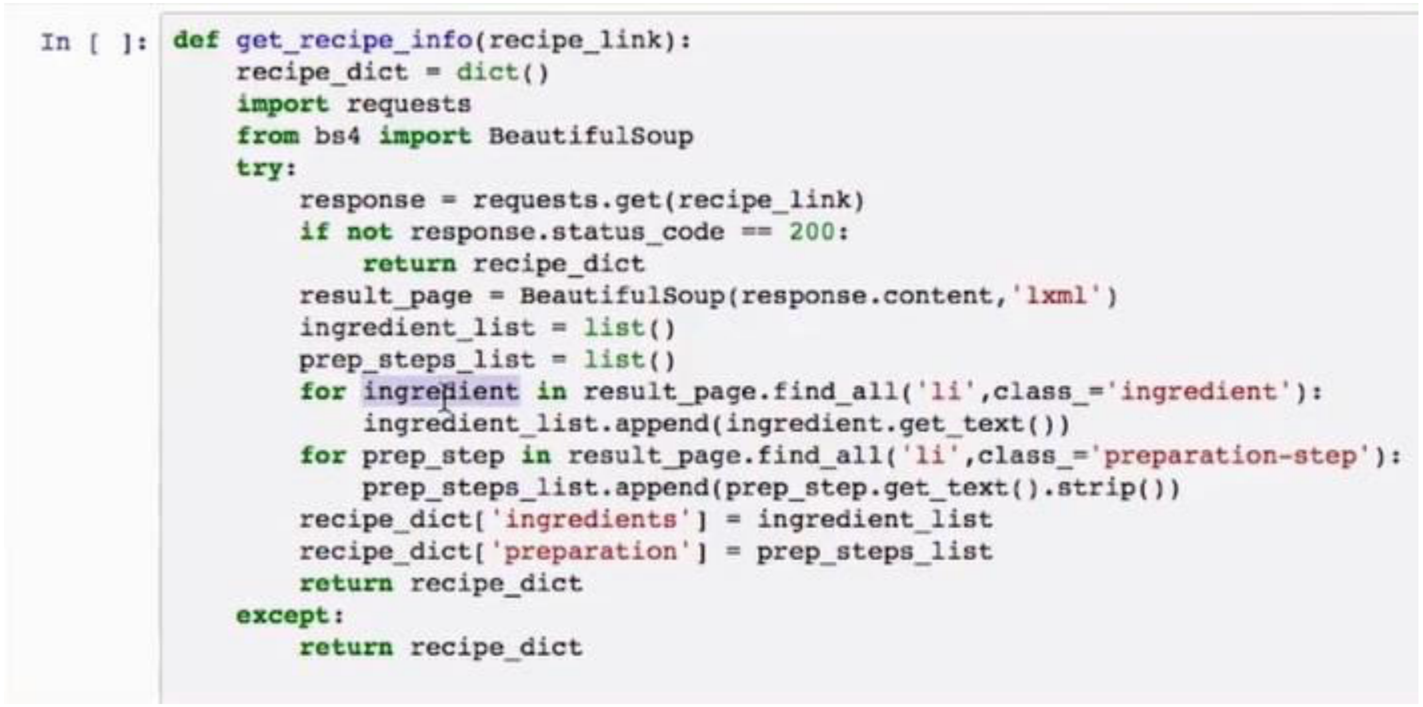
```
                recipe_description = recipe.find('p',class_='dek')
                recipe_list.append((recipe_name,recipe_link,recipe_description))
        except:
            return recipe_list
        return recipe_list
```

```
In [17]: get_recipes("Tofu Chili")

Out[17]: [('Spicy Lemongrass Tofu',
           'http://www.epicurious.com/recipes/food/views/spicy-lemongrass-tofu-233844',
           <p class="dek" data-reactid="77" data-truncate="1">Dau hu xa ot
           Editor's note: The recipe and introductory text below are excerpted from Pleasures of the Vietnamese Table by Mai P
          ham and are part of our story on Lunar New Year.
           While traveling on a train one time to the coastal town of Nha Trang, I sat next to an elderly nun. Over the course
          of our bumpy eight-hour ride, she shared stories of life at the temple and the difficult years after the end of the w
          ar when the Communist government cracked down on religious factions. Toward the end of our chat, she pulled out a bag
          of food she'd prepared for the trip. It was tofu that had been cooked in chilies, lemongrass and la lot, an aromatic
          leaf also known as pepper leaf. When she gave me a taste, I knew immediately that I had to learn how to make it. This
          is my rendition of that fabulous dish. Make sure to pat the tofu dry before marinating it and use very fresh lemongra
          ss. I always love serving this to friends who think tofu dishes are bland.</p>),
          ('Chinese Egg Noodles with Smoked Duck and Snow Peas',
           'http://www.epicurious.com/recipes/food/views/chinese-egg-noodles-with-smoked-duck-and-snow-peas-354302',
           <p class="dek" data-reactid="240" data-truncate="1">If you live near a Chinese market, pick up barbecued or smoked
          duck there. Otherwise, smoked chicken or turkey from the supermarket (or leftover roast chicken) would be terrific to
          ssed with the noodles. To make it a meal, add a platter of chilled silken tofu. Drizzle the tofu with soy sauce and c
          hili sauce, then top with chopped green onions. Coconut ice cream with fresh berries and lychees would make a nice de
          ssert.</p>)]
```

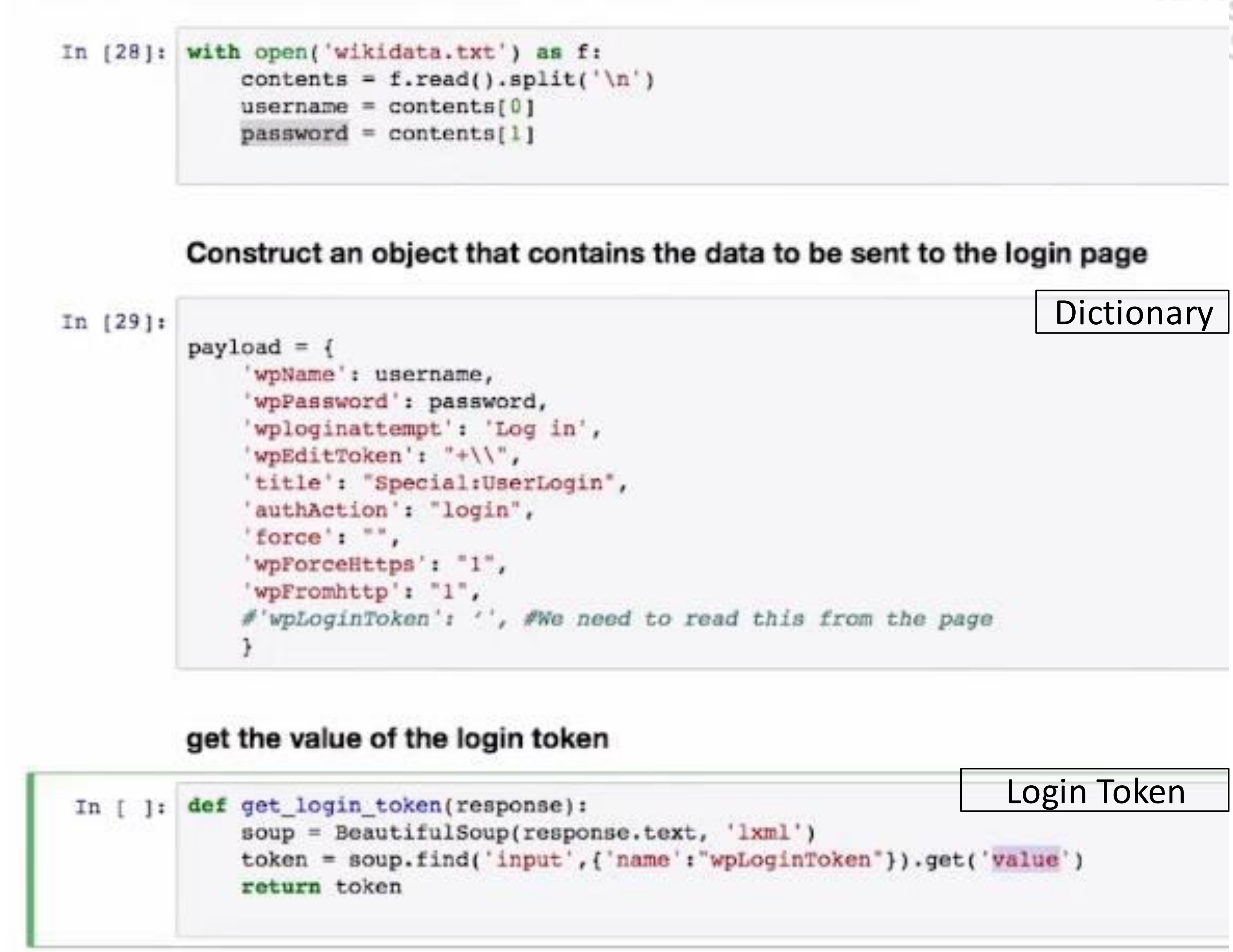
Epicurious Example: Part 3

Obtain ingredient and preparation steps



```
In [ ]: def get_recipe_info(recipe_link):
            recipe_dict = dict()
            import requests
            from bs4 import BeautifulSoup
            try:
                response = requests.get(recipe_link)
                if not response.status_code == 200:
                    return recipe_dict
                result_page = BeautifulSoup(response.content,'lxml')
                ingredient_list = list()
                prep_steps_list = list()
                for ingredient in result_page.find_all('li',class_='ingredient'):
                    ingredient_list.append(ingredient.get_text())
                for prep_step in result_page.find_all('li',class_='preparation-step'):
                    prep_steps_list.append(prep_step.get_text().strip())
                recipe_dict['ingredients'] = ingredient_list
                recipe_dict['preparation'] = prep_steps_list
                return recipe_dict
            except:
                return recipe_dict
```

Logging to a Web Server: Part 2



```
In [28]: with open('wikidata.txt') as f:
             contents = f.read().split('\n')
             username = contents[0]
             password = contents[1]
```

Construct an object that contains the data to be sent to the login page

Dictionary

```
In [29]:
         payload = {
             'wpName': username,
             'wpPassword': password,
             'wploginattempt': 'Log in',
             'wpEditToken': "+\\",
             'title': "Special:UserLogin",
             'authAction': "login",
             'force': "",
             'wpForceHttps': "1",
             'wpFromhttp': "1",
             #'wpLoginToken': '', #We need to read this from the page
             }
```

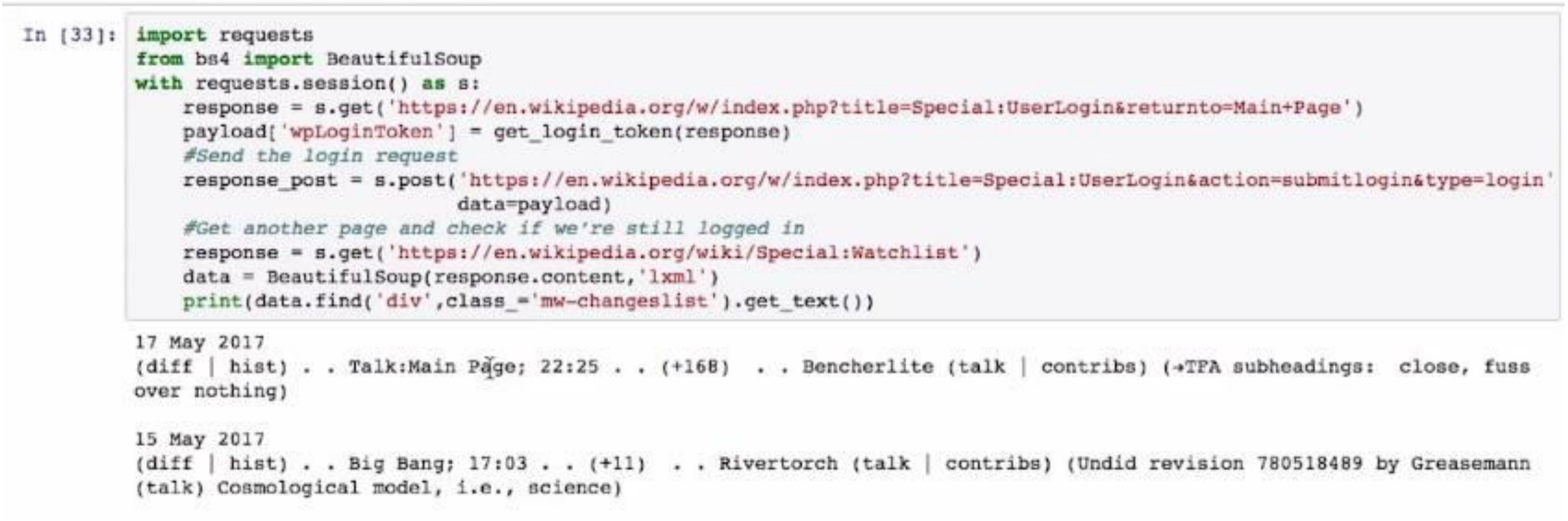
get the value of the login token

Login Token

```
In [ ]: def get_login_token(response):
            soup = BeautifulSoup(response.text, 'lxml')
            token = soup.find('input',{'name':"wpLoginToken"}).get('value')
            return token
```

Logging to a Web Server: Part 2

Set up a session, login and get data



```
In [33]: import requests
         from bs4 import BeautifulSoup
         with requests.session() as s:
             response = s.get('https://en.wikipedia.org/w/index.php?title=Special:UserLogin&returnto=Main+Page')
             payload['wpLoginToken'] = get_login_token(response)
             #Send the login request
             response_post = s.post('https://en.wikipedia.org/w/index.php?title=Special:UserLogin&action=submitlogin&type=login'
                                    data=payload)
             #Get another page and check if we're still logged in
             response = s.get('https://en.wikipedia.org/wiki/Special:Watchlist')
             data = BeautifulSoup(response.content,'lxml')
             print(data.find('div',class_='mw-changeslist').get_text())

         17 May 2017
         (diff | hist) . . Talk:Main Page; 22:25 . . (+168)  . . Bencherlite (talk | contribs) (→TFA subheadings:  close, fuss
         over nothing)

         15 May 2017
         (diff | hist) . . Big Bang; 17:03 . . (+11)  . . Rivertorch (talk | contribs) (Undid revision 780518489 by Greasemann
         (talk) Cosmological model, i.e., science)
```

GLOBAL IVY EDUCATION

www.emeritus.org