Selenium v.s Request And BeautifulSoup4

Detail

Presented by
Abraham Gale, Chi-Hui Lin

Outline

- Summary
- Installation
- Advantatges / Disadvatages
- Examples

Requests and Beautiful Soup4: Installation

- Install the Requests library
  - python -m pip install requests
- Install the BeautifulSoup4
  - python -m pip install beautifulsoup4
- (Optional) Install lxml
  - python -m pip install lxml

Requests: make HTTP requests

- Requests allows you to make HTTP requests
  - request = requests.get('https://www.example.com')

- You can also make other types of request
  - request = requests.post('https://httpbin.org/post', data = {'key':'value'})

- Usually just works, though does some interesting things behind the scenes

Response Object

- Most often, you will want the content of the reply
  - response = requests.get('https://www.example.com')
  - response.content
- You will also usually want to check the status code
  - response.status_code (== requests.codes.ok)
- Often you will want an error to be raised if the status is unexpected
  - response.raise_for_status()

Timeouts

- For code meant to be run in a production environment, use timeouts
    - requests.get('https://github.com/', timeout=10)

- WITHOUT TIMEOUT CODE CAN HANG FOREVER

- Timeout raises exception if NOTHING is sent in the given number of seconds
    - As long as first byte is sent in time, nothing will go wrong

Headers and cookies

- Sometimes you will want to change default header content
  - headers = {'user-agent': 'my-app/0.0.1'}
  - response = requests.get(“http://www.example.com”, headers=headers)
- Sometimes you might be interested in cookies
  - response.cookies[‘cookie_name']
- Cookies can put in (and are sent back in) a cookie jar, which is mostly like a dict
  - jar.set('tasty_cookie', 'yum', domain='httpbin.org', path='/cookies')
  - jar.set('gross_cookie', 'blech', domain='httpbin.org', path='/elsewhere')
  - response = requests.get(url, cookies=jar)

Sessions

- Sometimes you want a lot of requests to one URL domain
  - sess = requests.Session()
- Most importantly, this lets us use one TCP connection
- Session level cookies and default headers can be set
  - sess.auth = ('user', 'pass') #See last weeks lecture
  - sess.headers.update({'x-test': 'true'})
  - # both 'x-test' and 'x-test2' are sent
  - sess.get('https://httpbin.org/headers', headers={'x-test2': 'true'})

BeautifulSoup4: Parse HTML

- Normally, you want some specific part of the webpage, not just the entire thing
  - Maybe price, birth year, list of prices
- Import the package like this:
  - from bs4 import BeautifulSoup
- bs4 allows for many different backends, they recommend lxml as the fastest and most flexible
  - if installing extra dependency is hard, built in parser is fine

Making and navigating the soup

- For web scraping we must first make the soup
  - soup = BeautifulSoup(page.content, 'lxml')
  - soup = BeautifulSoup(page.content, 'html.parser')

- We can navigate the soup directly like this
  - soup.a

- We can get the content of the individual attributes like so
  - soup.a['title']

Find

- Most often we want to use find to find the object we are looking for
  - soup.find(‘a’)
- We can even use it iteratively
  - soup.find(‘a’).find(‘span’)
- We often want to filter by attributes, pass in as arguments
  - soup.find('a', href="/about-iaaf")
- Sometimes things have special meanings (name, class) so we need to pass them as dict
  - soup.find("a", attrs= {"class": "dropdown-toggle"})

Find_all

- If we want all instances of the given tag
  - soup.find_all(‘a’)
- We can also use regular expressions or lists to search for multiple types of tags
  - soup.find_all([‘title’, ‘a’])
  - soup.find_all(re.compile("^b"))
- We can even use boolean functions
  - def has_class_but_no_id(tag):
  -         return tag.has_attr('class') and not tag.has_attr('id')
  - soup.find_all(has_class_but_no_id)

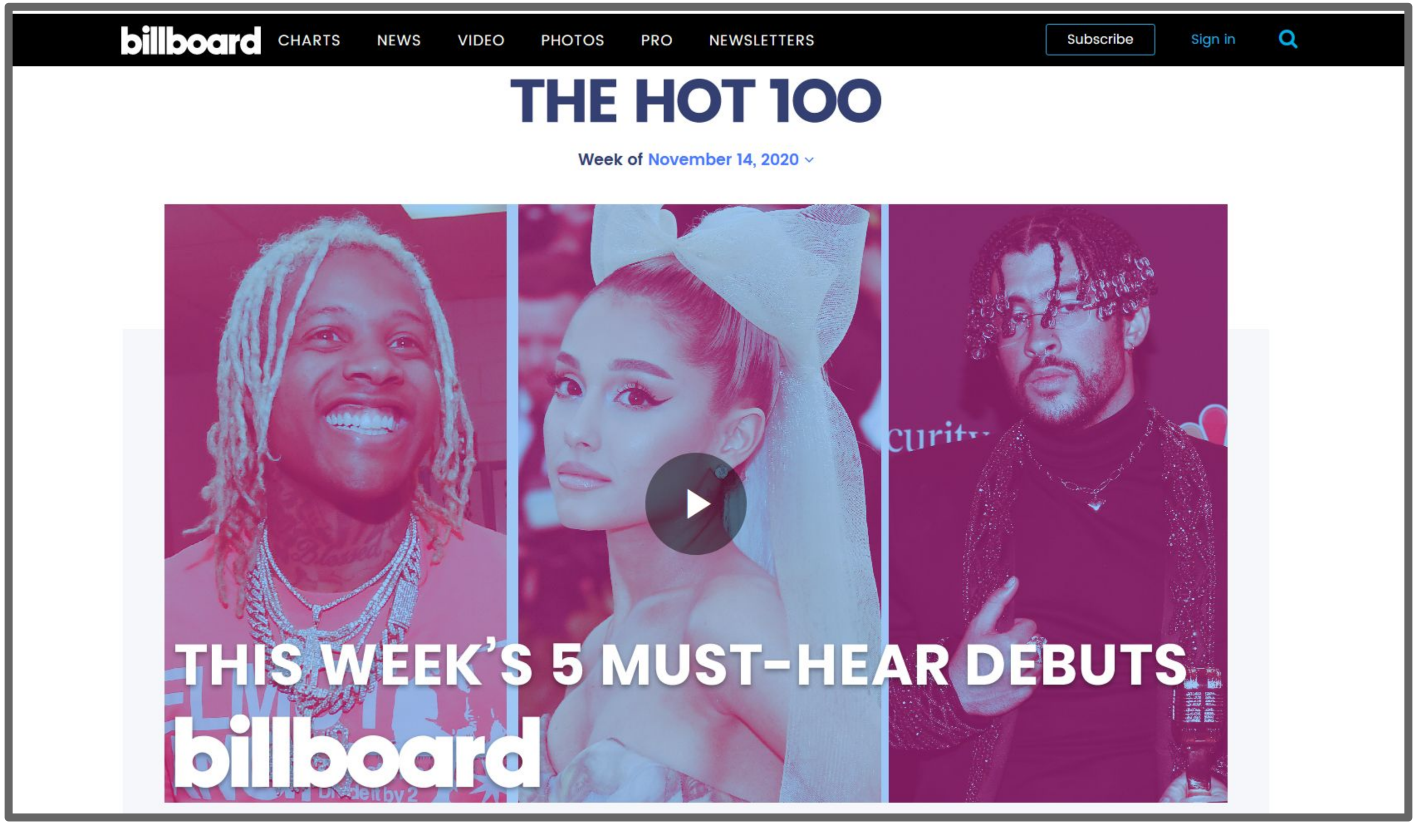
Outline

- Selenium
- Requests and Selenium
- Installation process for Selenium
- 5 steps to use Selenium
- Examples

Selenium

- Selenium Python bindings provides a simple API to write functional/acceptance tests using Selenium WebDriver.

Selenium on Data Exploration



Comparison between Requests and Selenium for Data Exploration

LIBRARIES	Requests	Selenium
Get a web page / Extract the content	**Faster**	**Slower**
Interactions with objects in web pages	**NO**	**YES**

Comparison between Requests and Selenium for Data Exploration

- Tasks
  - Visit the Google page (https://www.google.com)
    - Requests: 0.167s
    - Selenium: 0.792 s

Installation Process of Selenium

- Install the driver of the browser
- Install the Selenium python library

Selenium: Driver Installation for Windows Users

- Selenium requires a driver to interface with the chosen browser
  - Download the driver for your browser, like Chrome or Firefox
  - Set the *Path* variable
    - Control Panel > System and Security > System > Advanced system settings > Advanced tab-Environment Variables Button
    - Add the path to the driver directory to *Path* variables
    - Restart the computer

Selenium: Driver Installation for Linux Users

- Download the browser driver
  - Set the *PATH* variable
    - *echo %PATH* -> show you which directory shell will search for executable files
    - nano ~/.bashrc -> Edit the PATH variables
      - export PATH="$PATH:/common/users/cl1288/bin"
  - Install it!
    - python3 -m pip3 install webdrivermanager
    - webdrivermanager firefox --linkpath /common/users/cl1288/bin

Selenium: The Python Library Installation

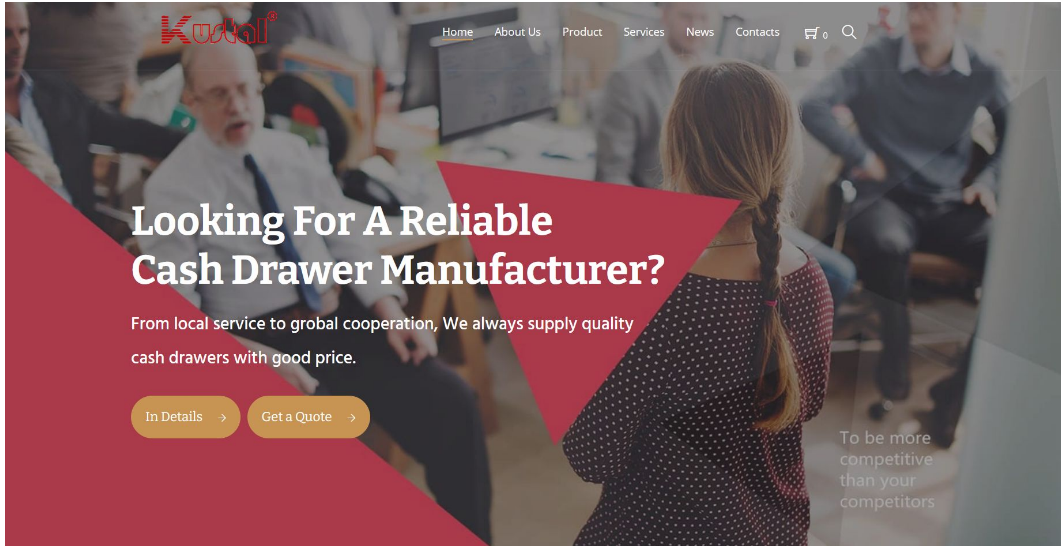
- Install the Selenium library
  - python -m pip install selenium
  - Note: make sure python is python3

5 steps to use Selenium

1. Navigate to a website
2. Wait Until the page loading the element
3. Locate the element
4. Interact with the element
5. Close the browser windows

2 steps to use Selenium

1. Navigate to a website
2. ___
3. ___
4. ___
5. Close the browser windows



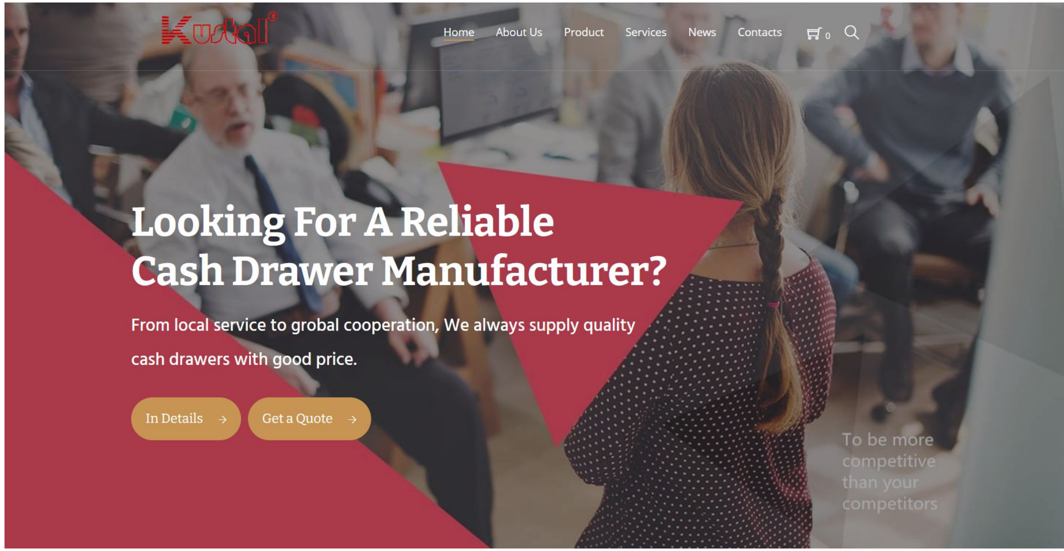
Looking For A Reliable Cash Drawer Manufacturer?

From local service to grobal cooperation, We always supply quality cash drawers with good price.

To be more competitive than your competitors

4 steps to use Selenium: Skip "Wait"

1. Navigate to a website
2. __
3. Locate the element
4. Interact with the element
5. Close the browser windows



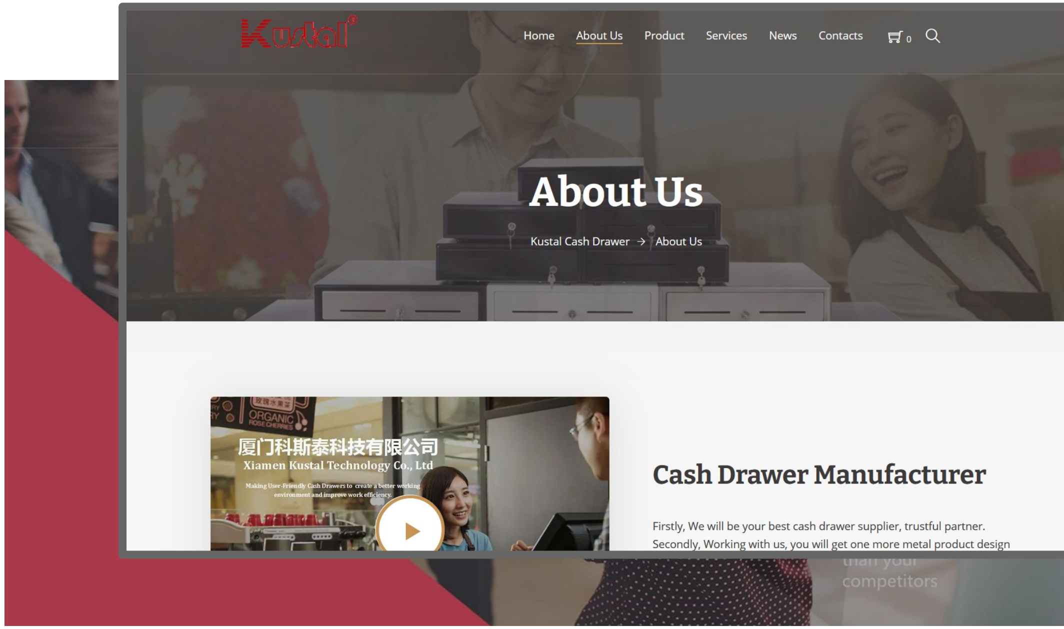
Looking For A Reliable
Cash Drawer Manufacturer?

From local service to grobal cooperation, We always supply quality cash drawers with good price.

To be more competitive than your competitors

5 steps to use Selenium

1. Navigate to a website
2. Wait Until the page loading the element
3. Locate the element
4. Interact with the element
5. Close the browser windows

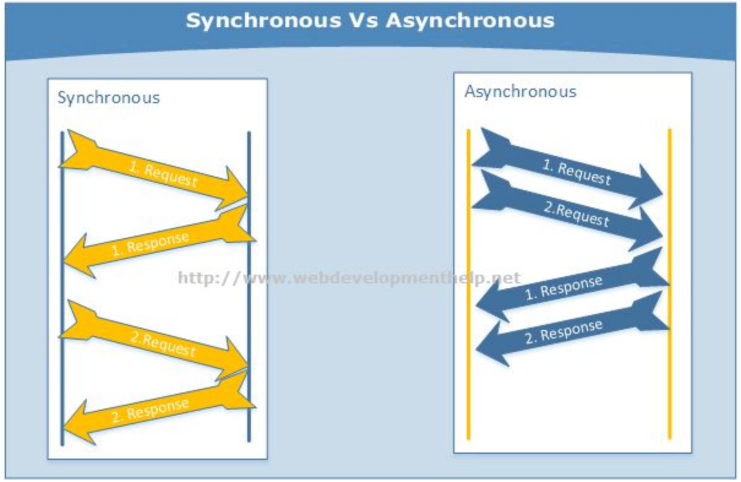


Making User-Friendly Cash Drawers to create a better working environment and improve work efficiency.

Firstly, We will be your best cash drawer supplier, trustful partner. Secondly, Working with us, you will get one more metal product design

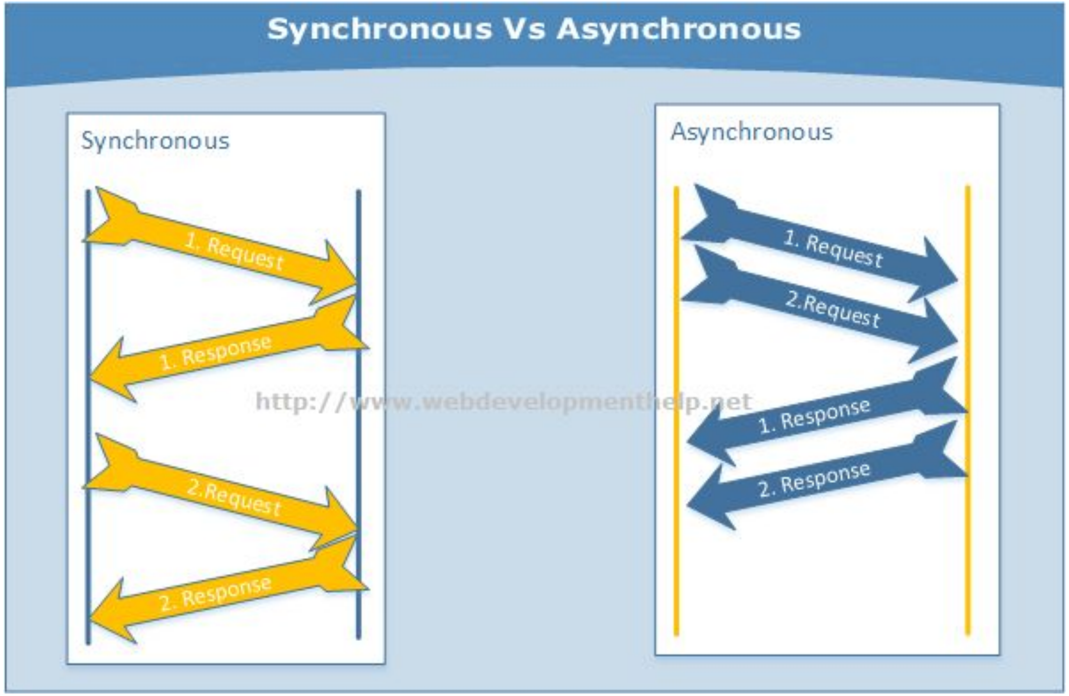
How to use Selenium: Wait

- Why do we need to “wait”?
  - Many web apps are using AJAX techniques.
- Async requests occur on a background thread
  - the UI is not going to be blocked while the request is processing



How to use Selenium: Wait

- Elements within the same page may load at different time intervals.



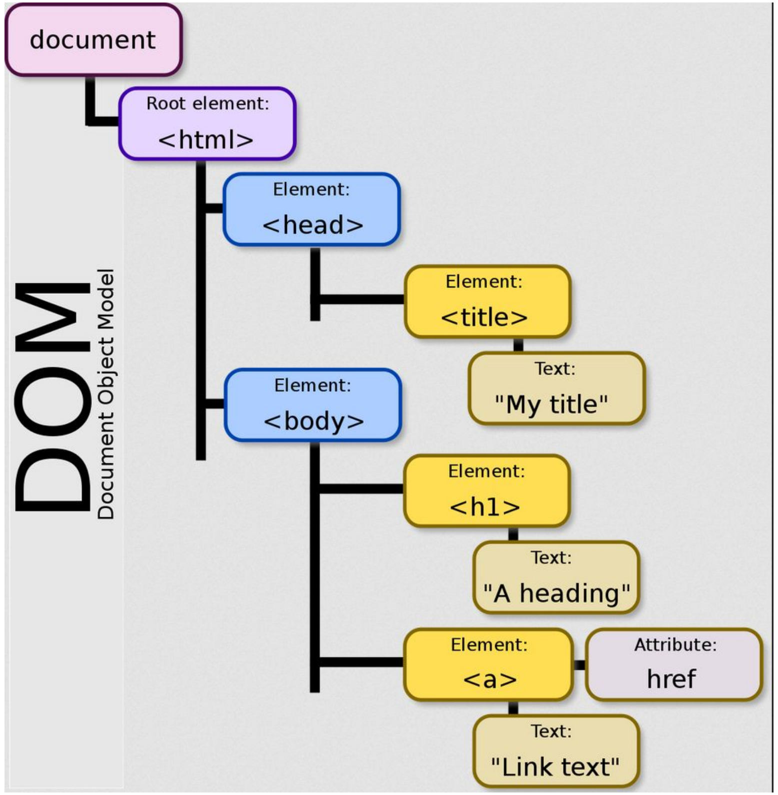
How to use Selenium: Wait

- Explicit Wait: Actively Wait
  - **Wait for a self-defined specific condition to occur**
    - before proceeding further
- Implicit Wait: Passively Wait
  - When trying to find **any element** not immediately available
    - **wait a certain amount of time**
  - Default is 0 second

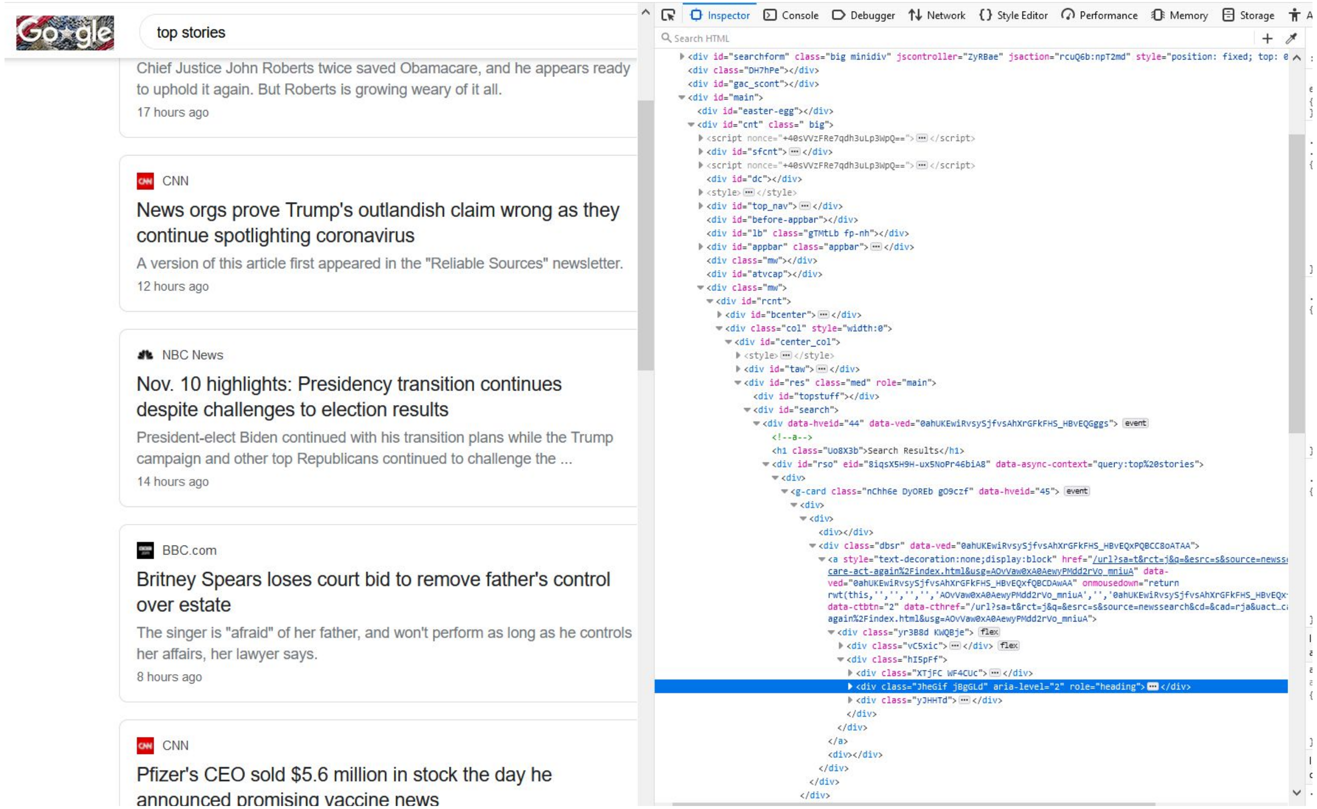
Locate the element

- Locate the element or elements with the following command
  - *find_element(s)_by_id*
    - *id, name, xpath, link_text, parital_link_text, tag_name, class_name*
- Read the element property
  - Inspect the element directly from the webpage
  - or Use Selenium IDE to simulate the actions on the element

Locate the element in an ideal condition



Locate the element in a practical condition



Chief Justice John Roberts twice saved Obamacare, and he appears ready to uphold it again. But Roberts is growing weary of it all.
17 hours ago

CNN
News orgs prove Trump's outlandish claim wrong as they continue spotlighting coronavirus
A version of this article first appeared in the "Reliable Sources" newsletter.
12 hours ago

NBC News
Nov. 10 highlights: Presidency transition continues despite challenges to election results
President-elect Biden continued with his transition plans while the Trump campaign and other top Republicans continued to challenge the ...
14 hours ago

BBC.com
Britney Spears loses court bid to remove father's control over estate
The singer is "afraid" of her father, and won't perform as long as he controls her affairs, her lawyer says.
8 hours ago

CNN
Pfizer's CEO sold $5.6 million in stock the day he announced promising vaccine news

```
<div id="searchform" class="big minidiv" jscontroller="ZyRBae" jsaction="rcuQ6b:npT2md" style="position: fixed; top: 0
<div class="DH7hPe"></div>
<div id="gac_scont"></div>
<div id="main">
  <div id="easter-egg"></div>
  <div id="cnt" class=" big">
    <script nonce="+40sVVzFRe7qdh3uLp3WpQ==">…</script>
    <div id="sfcnt">…</div>
    <script nonce="+40sVVzFRe7qdh3uLp3WpQ==">…</script>
    <div id="dc"></div>
    <style>…</style>
    <div id="top_nav">…</div>
    <div id="before-appbar"></div>
    <div id="lb" class="gTMtLb fp-nh"></div>
    <div id="appbar" class="appbar">…</div>
    <div class="mw"></div>
    <div id="atvcap"></div>
    <div class="mw">
      <div id="rcnt">
        <div id="bcenter">…</div>
        <div class="col" style="width:0">
          <div id="center_col">
            <style>…</style>
            <div id="taw">…</div>
            <div id="res" class="med" role="main">
              <div id="topstuff"></div>
              <div id="search">
                <div data-hveid="44" data-ved="0ahUKEwiRvsySjfvsAhXrGFkFHS_HBvEQGggs"> event
                  <!--a-->
                  <h1 class="Uo8X3b">Search Results</h1>
                  <div id="rso" eid="8iqsX5H9H-ux5NoPr46biA8" data-async-context="query:top%20stories">
                    <div>
                      <g-card class="nChh6e DyOREb gO9czf" data-hveid="45"> event
                        <div>
                          <div>
                            <div></div>
                            <div class="dbsr" data-ved="0ahUKEwiRvsySjfvsAhXrGFkFHS_HBvEQxPQBCC8oATAA">
                              <a style="text-decoration:none;display:block" href="/url?sa=t&rct=j&q=&esrc=s&source=newss
                              care-act-again%2Findex.html&usg=AOvVaw0xA0AewyPMdd2rVo_mniuA" data-
                              ved="0ahUKEwiRvsySjfvsAhXrGFkFHS_HBvEQxfQBCDAwAA" onmousedown="return
                              rwt(this,'','','','','AOvVaw0xA0AewyPMdd2rVo_mniuA','','0ahUKEwiRvsySjfvsAhXrGFkFHS_HBvEQx
                              data-ctbtn="2" data-cthref="/url?sa=t&rct=j&q=&esrc=s&source=newssearch&cd=&cad=rja&uact_c
                              again%2Findex.html&usg=AOvVaw0xA0AewyPMdd2rVo_mniuA">
                                <div class="yr3B8d KWQBje"> flex
                                  <div class="vC5xic">…</div> flex
                                  <div class="hI5pFf">
                                    <div class="XTjFC WF4CUc">…</div>
                                    <div class="JheGif jBgGLd" aria-level="2" role="heading">…</div>
                                    <div class="yJHHTd">…</div>
                                  </div>
                                </div>
                              </a>
                              <div></div>
                            </div>
                          </div>
                        </div>
```

Summary for Selenium

- Selenium provides API to interact objects on pages
- 5 steps to use Selenium
  - Navigate to a website
  - Wait Until the page loading the element
  - Locate the element
  - Interact with the element
  - Close the browser windows
- Selenium IDE provides an easy way to get the xpath or css information of objects

How to use Selenium: Interact with the element

- You could make selenium do things like a human does
  - Click a button
  - Type a keyword and then type a RETURN Key
- Read the element information
  - read the text
  - get to the link

How to use Selenium: Navigate to a website

- Set the PATH variable, if you did not do it for your system
- Get the driver
  - *driver = webdriver.Chrome()*
  - *driver = webdriver.Firefox()*
- Get to a website
  - *driver.get("https://www.google.com")*

How to use Selenium: Close the browser windows

- quit
    - Close all browser windows