Web Scraping using Python
Python for Data Analytics

Peerapon Vateekul, Ph.D.
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University
Peerapon.v@chula.ac.th

Chula Big Data and IoT
Center of Excellence
(CUBIC)

Outlines

- What is **Web scraping?**
- Usage
- Advantages
- Workflows
- Libraries
- BeautifulSoup
- Collecting and Parsing a Web Page
- Pulling Text From a Web Page
- Example

Reference:
1. Manoj Pandey, https://slides.com/manojp/introws#/1
2. Paul Schreiber, Web Scraping with Python Slide, NICAR 2015
3. https://www.digitalocean.com/community/tutorials/how-to-scrape-web-pages-with-beautiful-soup-and-python-3 v

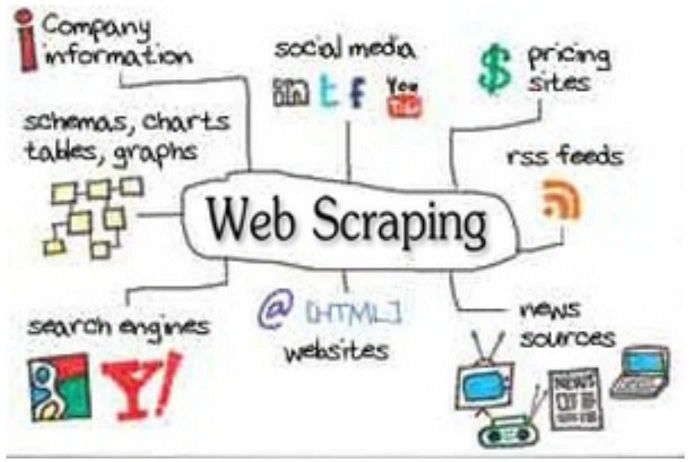
What is **Web scraping?**

- **Web scraping** is a technique for gathering data or information on web pages.
- You could revisit your favorite web site every time it updates for new information.
- Or you could write a **web scraper** to have it do it for you!
- It is a method to extract data from a website that does not have an **API** or we want to extract a LOT of data which we can not do through an API due to **rate limiting**.
- Through web scraping we can extract any data which we can see while browsing the web

Usage

WEB SCRAPING IN REAL LIFE

- Extract product information.
- Extract job postings and internships.
- Extract offers and discounts from deal-of-the-day websites.
- Crawl forums and social websites.
- Extract data to make a search engine.
- Gathering weather data etc.



+
Advantages

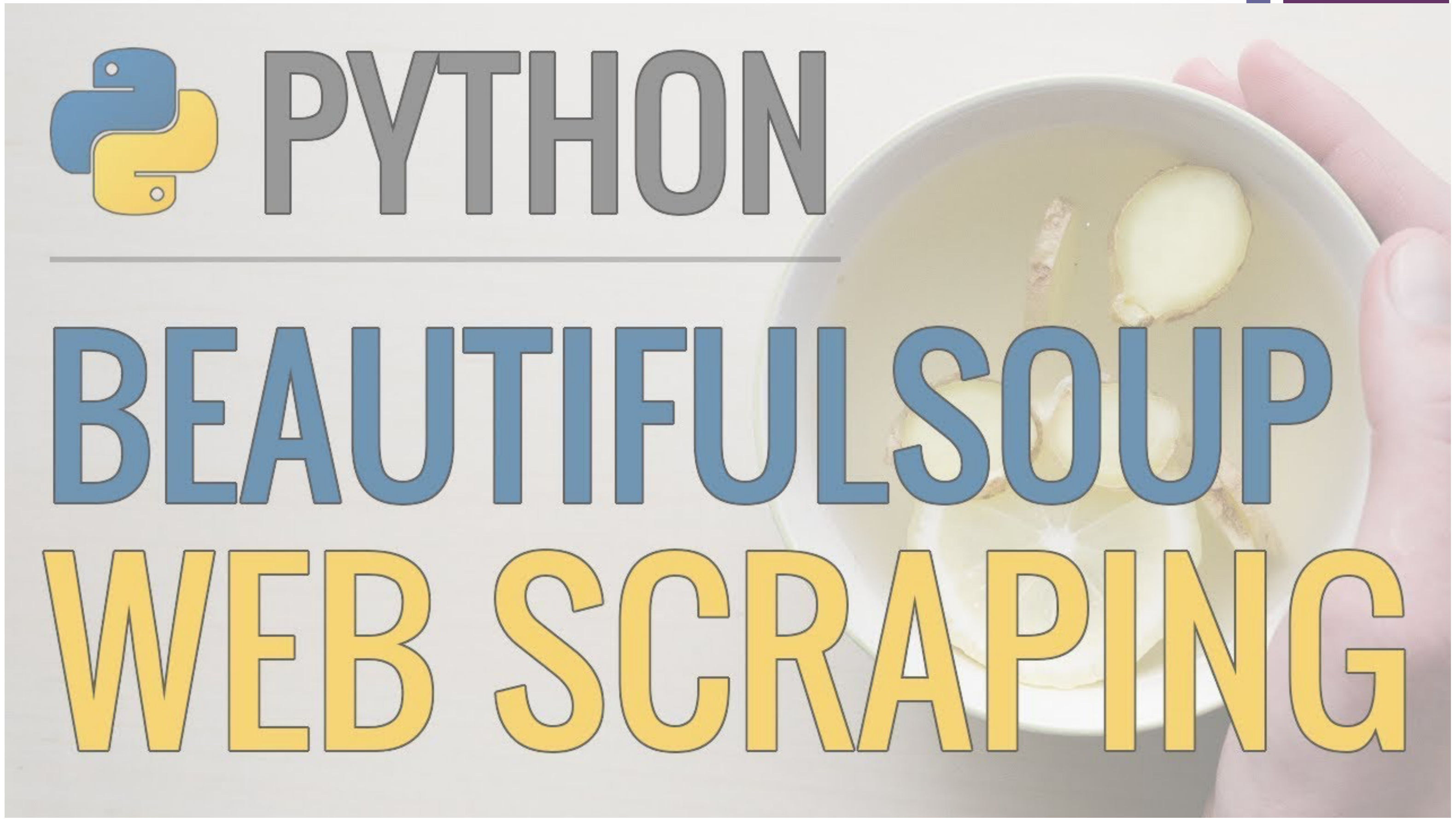
- Web Scraping is **not rate limited.**
- **Anonymously** access the website and gather data.
- Some websites do not have an API.
- Some data is not accessible through an API
- **and many more !**

Workflows

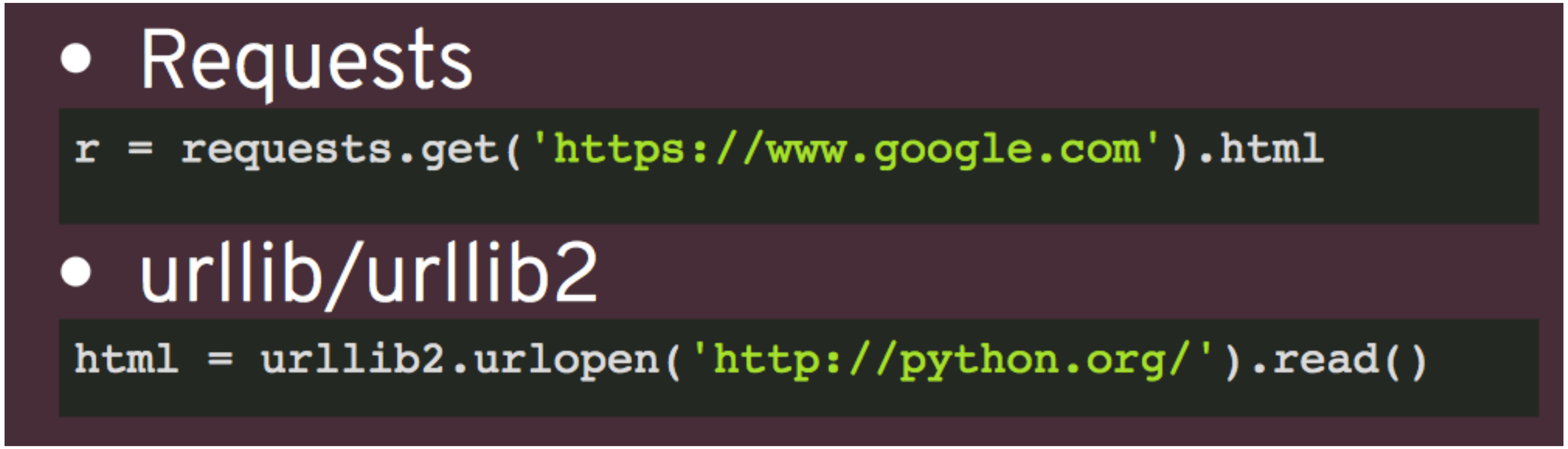
- **Web Scraping follows this workflow**:
  - **Get** the website - using **HTTP** library
  - **Parse** the html document - using any parsing library
  - **Store** the results - either a database, csv, text file, etc

Libraries

- **BeautifulSoup**
- Lxml
- Scrapy
- Selenium



Http Libraries: Get Web Content (html)



- Requests

```
r = requests.get('https://www.google.com').html
```

- urllib/urllib2

```
html = urllib2.urlopen('http://python.org/').read()
```

Parsing Libraries: Get Title

- beautifulsoup

```
tree = BeautifulSoup(html_doc)
tree.title
```

- lxml

```
tree = lxml.html.fromstring(html_doc)
title = tree.xpath('/title/text()')
```

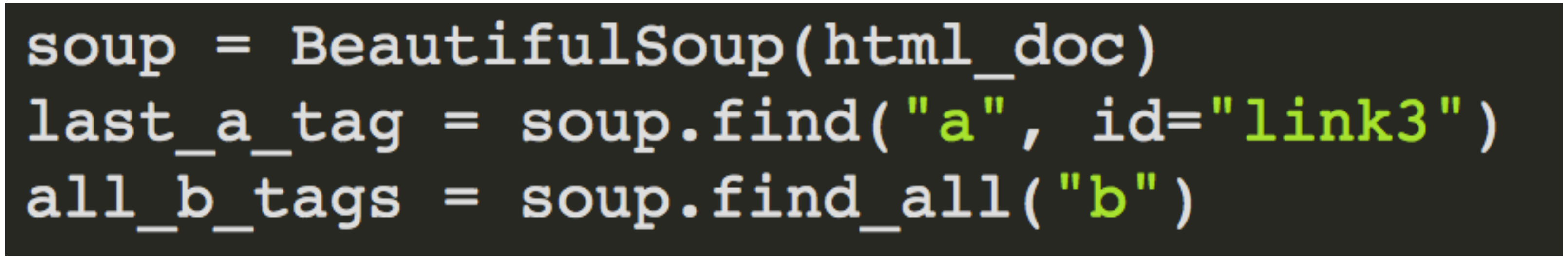
- re

```
title = re.findall('<title>(.*?)</title>', html_doc)
```

+

BeautifulSoup

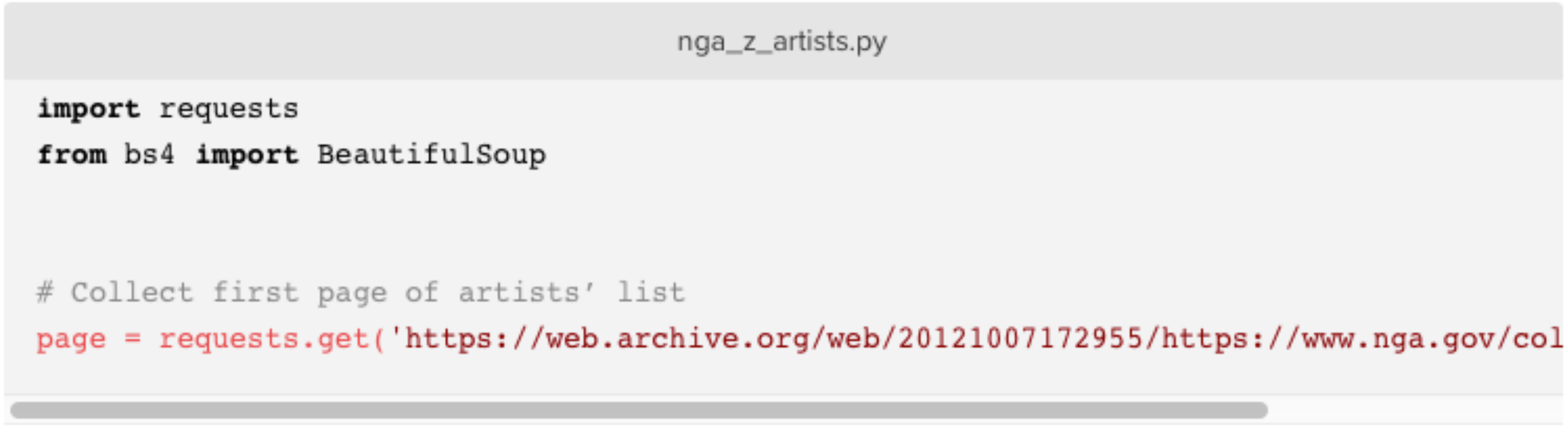
- A beautiful API
- Very easy to use
- Can handle broken markup
- Purely in Python



```
soup = BeautifulSoup(html_doc)
last_a_tag = soup.find("a", id="link3")
all_b_tags = soup.find_all("b")
```

Collecting and Parsing a Web Page

- The next step we will need to do is collect the URL of the first web page with Requests. We'll assign the URL for the first page to the variable page by using the method requests.get().



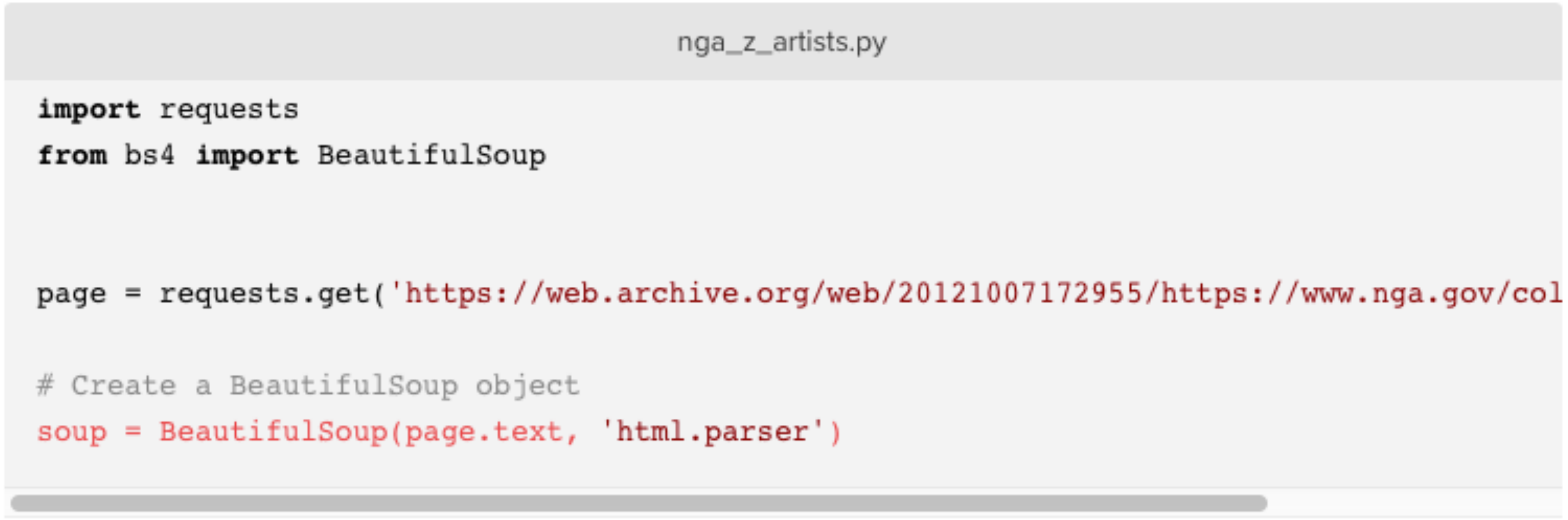
nga_z_artists.py

```
import requests
from bs4 import BeautifulSoup


# Collect first page of artists' list
page = requests.get('https://web.archive.org/web/20121007172955/https://www.nga.gov/col
```

page = requests.get('https://web.archive.org/web/20121007172955/https://www.nga.gov/collection/anZ1.htm')

Collecting and Parsing a Web Page (cont.)

- We'll now create a BeautifulSoup object, or a parse tree.
- This object takes as its arguments the page.text document from Requests (the content of the server's response) and then parses it from Python's built-in html.parser.



nga_z_artists.py

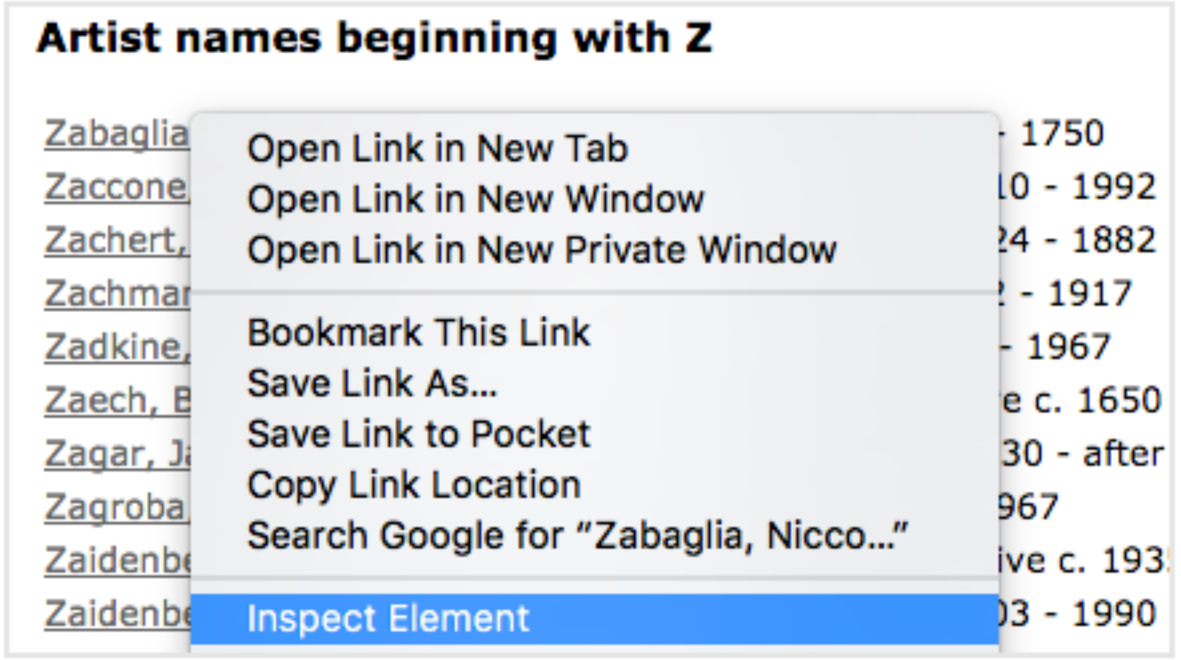
```
import requests
from bs4 import BeautifulSoup


page = requests.get('https://web.archive.org/web/20121007172955/https://www.nga.gov/col

# Create a BeautifulSoup object
soup = BeautifulSoup(page.text, 'html.parser')
```

- With our page collected, parsed, and set up as a BeautifulSoup object, we can move on to collecting the data that we would like.

Pulling Text From a Web Page

- For this project, we'll collect artists' names and the relevant links available on the website. You may want to collect different data, such as the artists' nationality and dates. Whatever data you would like to collect, you need to find out how it is described by the DOM of the web page.

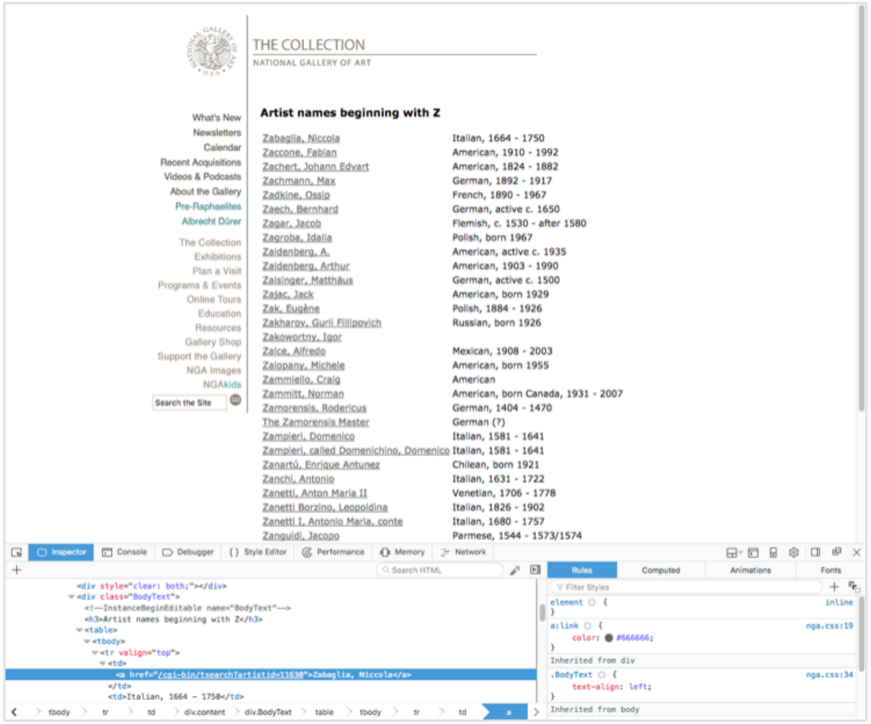
- To do this, in your web browser, **right-click — or CTRL + click on macOS** — on the first artist's name, **Zabaglia, Niccola**. Within the context menu that pops up, you should see a menu item similar to **Inspect Element** (Firefox) or **Inspect** (Chrome).


Artist names beginning with Z

Zabaglia - 1750
Zaccone 10 - 1992
Zachert 24 - 1882
Zachmar - 1917
Zadkine - 1967
Zaech, E e c. 1650
Zagar, J 30 - after
Zagroba 967
Zaidenbe ive c. 193
Zaidenbe 03 - 1990

Pulling Text From a Web Page (cont.)

- Once you click on the relevant **Inspect** menu item, the tools for web developers should appear within your browser.
- We want to look for the class and tags associated with the artists' names in this list.



THE COLLECTION
NATIONAL GALLERY OF ART

What's New
Newsletters
Calendar
Recent Acquisitions
Videos & Podcasts
About the Gallery
Pre-Raphaelites
Albrecht Dürer
The Collection
Exhibitions
Plan a Visit
Programs & Events
Online Tours
Education
Resources
Gallery Shop
Support the Gallery
NGA Images
NGAkids
Search the Site

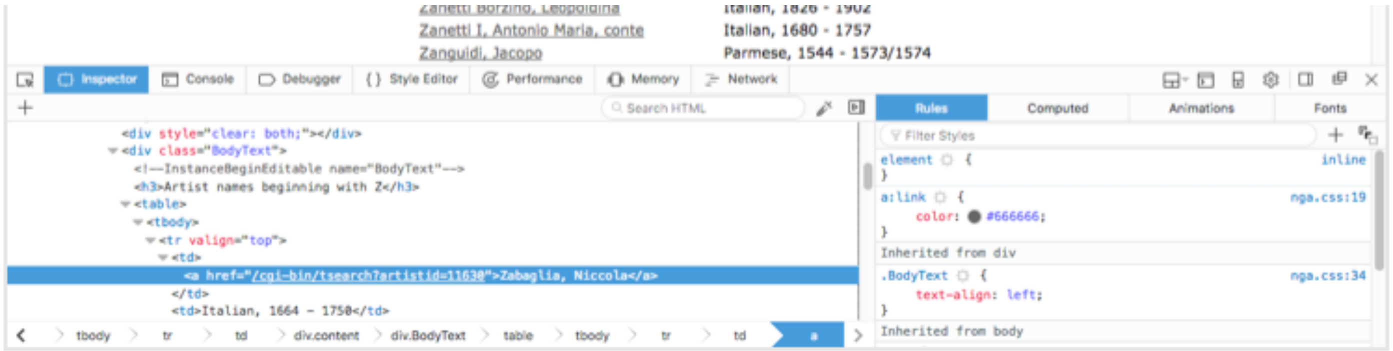
Artist names beginning with Z

Zabaglia, Niccola	Italian, 1664 - 1750
Zaccone, Fabian	American, 1910 - 1992
Zachert, Johann Edvart	American, 1824 - 1882
Zachmann, Max	German, 1892 - 1917
Zadkine, Ossip	French, 1890 - 1967
Zaech, Bernhard	German, active c. 1650
Zagar, Jacob	Flemish, c. 1530 - after 1580
Zagroba, Idalia	Polish, born 1967
Zaidenberg, A.	American, active c. 1935
Zaidenberg, Arthur	American, 1903 - 1990
Zaisinger, Matthäus	German, active c. 1500
Zajac, Jack	American, born 1929
Zak, Eugène	Polish, 1884 - 1926
Zakharov, Gurii Fillipovich	Russian, born 1926
Zakowortny, Igor	
Zalce, Alfredo	Mexican, 1908 - 2003
Zalopany, Michele	American, born 1955
Zammiello, Craig	American
Zammitt, Norman	American, born Canada, 1931 - 2007
Zamorensis, Rodericus	German, 1404 - 1470
The Zamorensis Master	German (?)
Zampieri, Domenico	Italian, 1581 - 1641
Zampieri, called Domenichino, Domenico	Italian, 1581 - 1641
Zanartú, Enrique Antunez	Chilean, born 1921
Zanchi, Antonio	Italian, 1631 - 1722
Zanetti, Anton Maria II	Venetian, 1706 - 1778
Zanetti Borzino, Leopoldina	Italian, 1826 - 1902
Zanetti I, Antonio Maria, conte	Italian, 1680 - 1757
Zanguidi, Jacopo	Parmese, 1544 - 1573/1574

```
<div style="clear: both;"></div>
▾<div class="BodyText">
   <!--InstanceBeginEditable name="BodyText"-->
   <h3>Artist names beginning with Z</h3>
  ▾<table>
    ▾<tbody>
      ▾<tr valign="top">
        ▾<td>
           <a href="/cgi-bin/tsearch?artistid=11630">Zabaglia, Niccola</a>
         </td>
         <td>Italian, 1664 – 1750</td>
```

```
element {                         inline
}
a:link {                          nga.css:19
    color: ● #666666;
}
Inherited from div
.BodyText {                       nga.css:34
    text-align: left;
}
Inherited from body
```

```
< tbody > tr > td > div.content > div.BodyText > table > tbody > tr > td > a >
```



Zanetti I, Antonio Maria, conte	Italian, 1680 - 1757
Zanguidi, Jacopo	Parmese, 1544 - 1573/1574

```
<div style="clear: both;"></div>
▾<div class="BodyText">
   <!--InstanceBeginEditable name="BodyText"-->
   <h3>Artist names beginning with Z</h3>
  ▾<table>
    ▾<tbody>
      ▾<tr valign="top">
        ▾<td>
           <a href="/cgi-bin/tsearch?artistid=11630">Zabaglia, Niccola</a>
         </td>
         <td>Italian, 1664 – 1750</td>
```

```
element {                         inline
}
a:link {                          nga.css:19
    color: ● #666666;
}
Inherited from div
.BodyText {                       nga.css:34
    text-align: left;
}
Inherited from body
```

```
< tbody > tr > td > div.content > div.BodyText > table > tbody > tr > td > a >
```

Any Questions?