**Python with BeautifulSoup**

Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a **parse tree** for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

Beautiful Soup provides simple methods for navigating, searching, and modifying a parse tree in HTML, XML files. It transforms a complex HTML document into a tree of Python objects. It also automatically converts the document to Unicode, so you don't have to think about encodings.

Python library that allows for quick turnaround on web scraping projects. Currently available as Beautiful Soup 4 and compatible with both Python 2.7 and Python 3, Beautiful Soup creates a parse tree from parsed HTML and XML documents.

**Limitation:**

It can't interact with web pages like a human user. The library can just parse data. Therefore, you'll need to install other modules to extract the information, e.g., requests or httpx

**default encoding of BeautifulSoup**

The output from a BeautifulSoup is UTF-8 document.

1. **Importing the Libraries**

import requests

from bs4 import BeautifulSoup

1. **Collecting and Parsing a Web Page**

**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/collection/anZ1.htm')

1. **Removing Superfluous Data (unnecessary data)**

use the **decompose()** method to remove a tag from the parse tree and then destroy it along with its contents

1. **Pulling the Contents from a Tag**

want to target the contents of the <a> tags rather than print out the entire link tag. Beautiful Soup’s .contents, which will return the tag’s children.

1. **Writing the Data to a CSV File**

Collecting data that only lives in a terminal window is not very useful. Comma-separated values (CSV) files allow us to store tabular data in plain text, and is a common format for spreadsheets and databases**.**

**The four major and important objects are :**

* **BeautifulSoup :** The BeautifulSoup object represents the parsed document as a whole. So, it is the complete document which we are trying to scrape. For most purposes, you can treat it as a Tag object.

# importing the module

from bs4 import BeautifulSoup

# parsing the document

soup = BeautifulSoup('''<h1>Hello</h1>''', "html.parser")

print(type(soup))

* **Tag**

Beautiful Soup is not an HTTP client which means to scrap online websites you first have to download them using the requests module and then serve them to Beautiful Soup for scraping. Additionally, this object returns the first found tag if your document has multiple tags with the same name.

* **NavigableString**
* **Comments**

**Scrape Data From Local HTML Files using Python**

BeautifulSoup module in Python allows us to scrape data from local HTML files. For some reason, website pages might get stored in a local (offline environment), and whenever in need, there may be requirements to get the data from them. Sometimes there may be a need to get data from multiple Locally stored HTML files too. Usually HTML files got the tags like <h1>, <h2>,…<p>, <div> tags etc., by Using BeautifulSoup, we can scrap the contents and get the necessary details.

pip install beautifulsoup4

**The Prettify()** function in BeautifulSoup helps to view the tag nature and their nesting.