

***WEB SCRAPING***

**Web scraping** is the process of automatically extracting data from websites. It involves fetching web pages and then extracting specific pieces of information from the HTML content. This technique is widely used for various purposes, such as gathering data for research, monitoring prices, aggregating reviews, and more.

**Types of Data**

1. **Structured Data**

Ex: CSV, Excel Spreadsheets, Databases

1. **Unstructured Data**

Ex: Word Documents, PDF Files, Text Files, Media Logs

1. **Semi**- **Structured Data**

Ex: HTML,XML,JSON

**How Web Scraping Works/Components**

1. **Sending a Request:**

* A web scraper sends an HTTP request to a web server, requesting a particular web page.
* The server responds with the HTML content of the page.

1. **Parsing the HTML:**

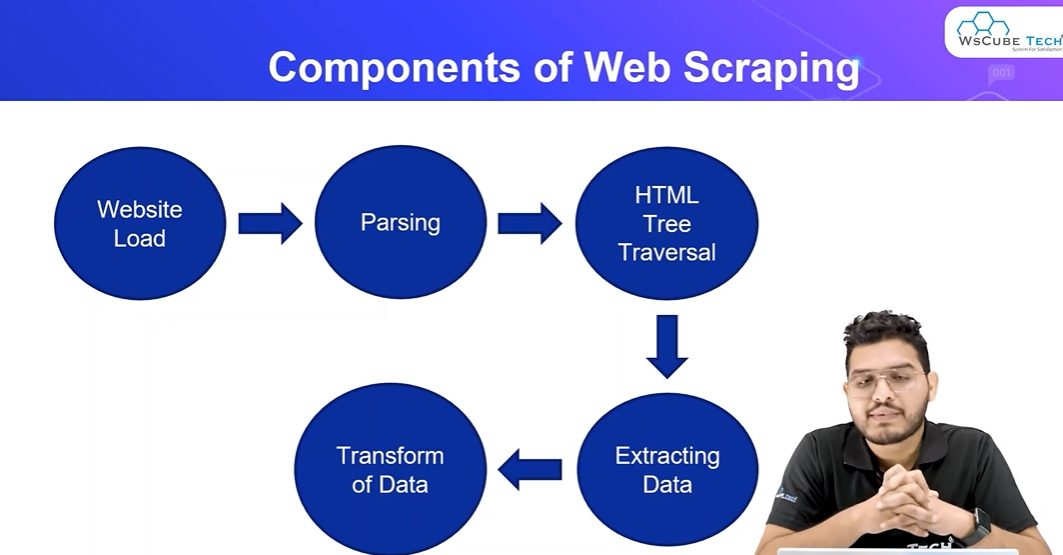
* The HTML content is parsed to find specific data points. This might involve looking for specific tags (like <div>, <p>, <a>), attributes (like class, id), or even text content.

1. **Extracting Data:**

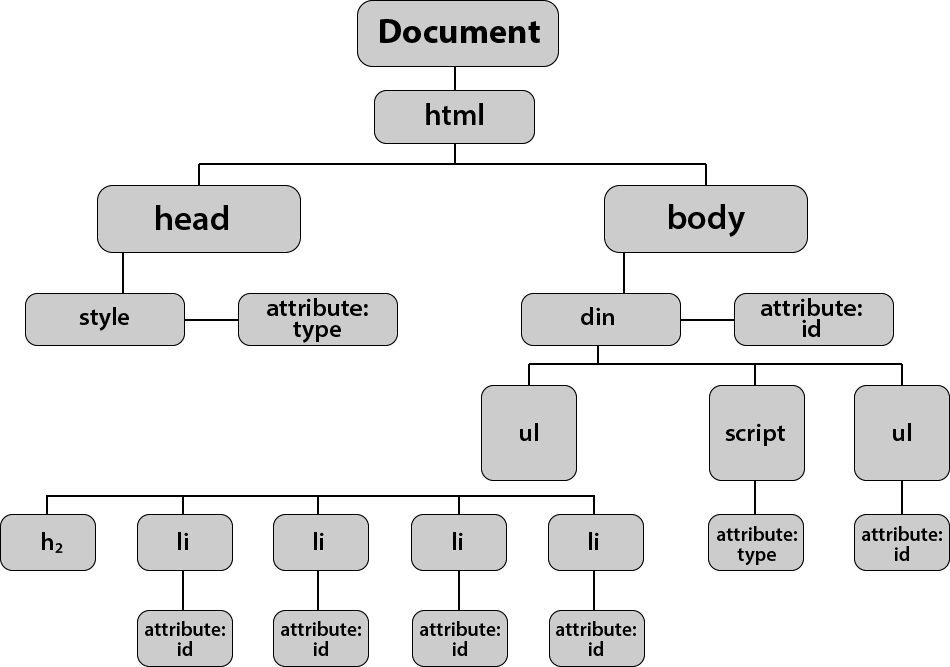
* The desired data is extracted from the parsed HTML. For example, you might extract product prices, headlines, or links to other pages.

1. **Storing Data:**

* The extracted data is stored in a structured format, such as a CSV file, database, or spreadsheet, for further analysis or processing.



**Components of Web Scraping**



**Tools for Web Scraping**

1. **Programming Languages:**

* **Python** is a popular choice for web scraping due to its libraries like BeautifulSoup, Scrapy, and Selenium.
* **JavaScript** and **Node.js** can also be used for scraping, especially for websites that heavily use JavaScript.

1. **Libraries:**

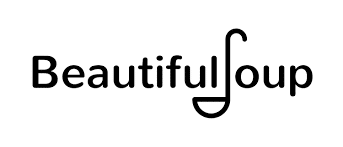
* **BeautifulSoup**: A Python library for parsing HTML and XML documents. It is often used in combination with requests to scrape web data.
* **Scrapy**: A powerful Python framework specifically designed for large-scale web scraping projects.
* **Selenium**: A tool that automates web browsers, often used for scraping dynamic websites that require JavaScript execution.

**Legal and Ethical Considerations**

* **Terms of Service**: Many websites have terms of service that prohibit or restrict scraping. It’s important to review and adhere to these terms.
* **Robots.txt**: Websites often have a robots.txt file that specifies which parts of the site can or cannot be scraped. Ethical scrapers respect these directives.
* **Rate Limiting**: Sending too many requests in a short period can overload a server. Ethical scrapers implement rate limiting to avoid being blocked and to minimize impact on the website.

**Use Cases of Web Scraping**

1. **Price Monitoring**: Automatically checking prices across different e-commerce sites.
2. **Content Aggregation**: Gathering news, articles, or blog posts from various sources.
3. **Market Research**: Collecting data from competitors' websites for analysis.
4. **Data Analysis**: Extracting data for analysis in research, business intelligence, or machine learning projects.



**BeautifulSoup** is a Python library used for web scraping purposes to pull data out of HTML and XML files. It provides Pythonic idioms for iterating, searching, and modifying the parse tree.

The **requests** module is a powerful and easy-to-use Python library for making HTTP requests. It allows you to send HTTP requests to interact with web services, retrieve data, and even submit forms or upload files.

**Install Modules**

1. pip install requests
2. pip install beautifulsoup4

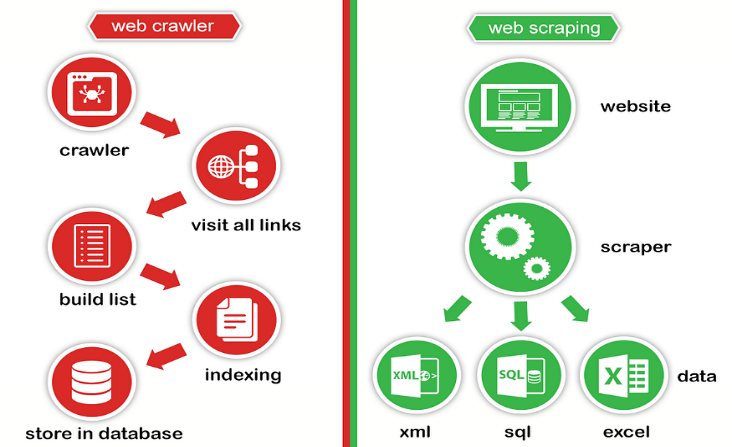
**How many way to perform Web Scraping?**

**1. Crawler (Web Crawler)**

**Definition**: A web crawler, also known as a spider or bot, is an automated program that systematically browses the web, moving from one web page to another by following links. The primary purpose of a crawler is to discover and index content across the internet.

**2. Scraper (Web Scraper)**

**Definition**: A web scraper is a tool designed to extract specific data from web pages. Unlike crawlers, which focus on discovering and indexing web pages, scrapers focus on extracting useful information from those pages.



***PRACTICAL***

**1. Sending a Request**



**2. Parsing the HTML**



**Extracting Data**

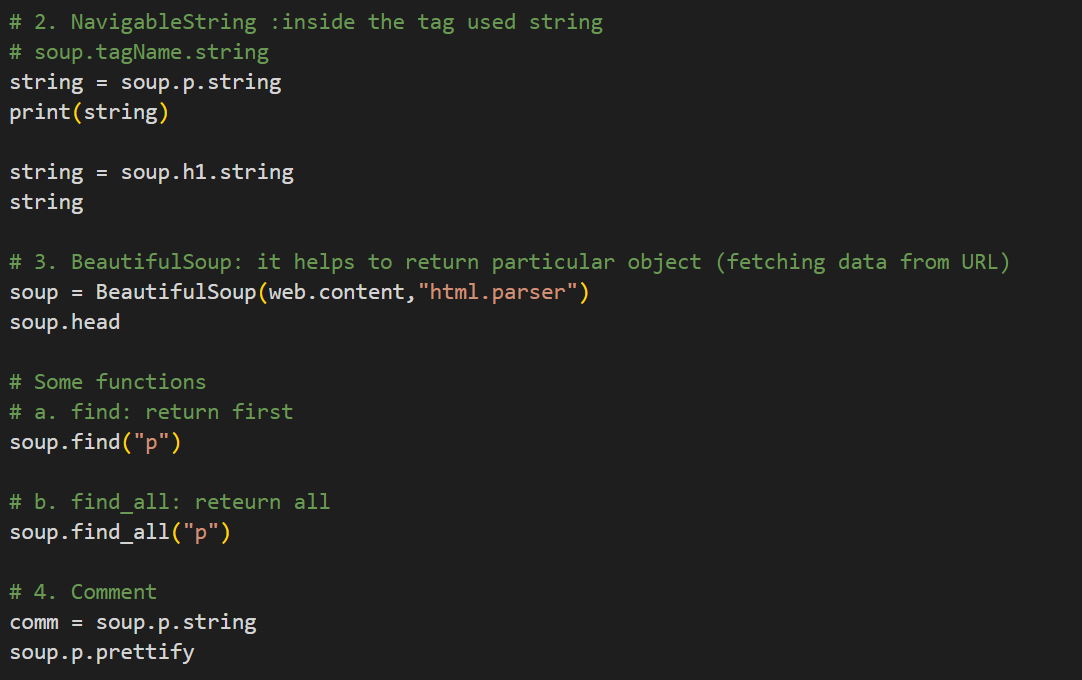
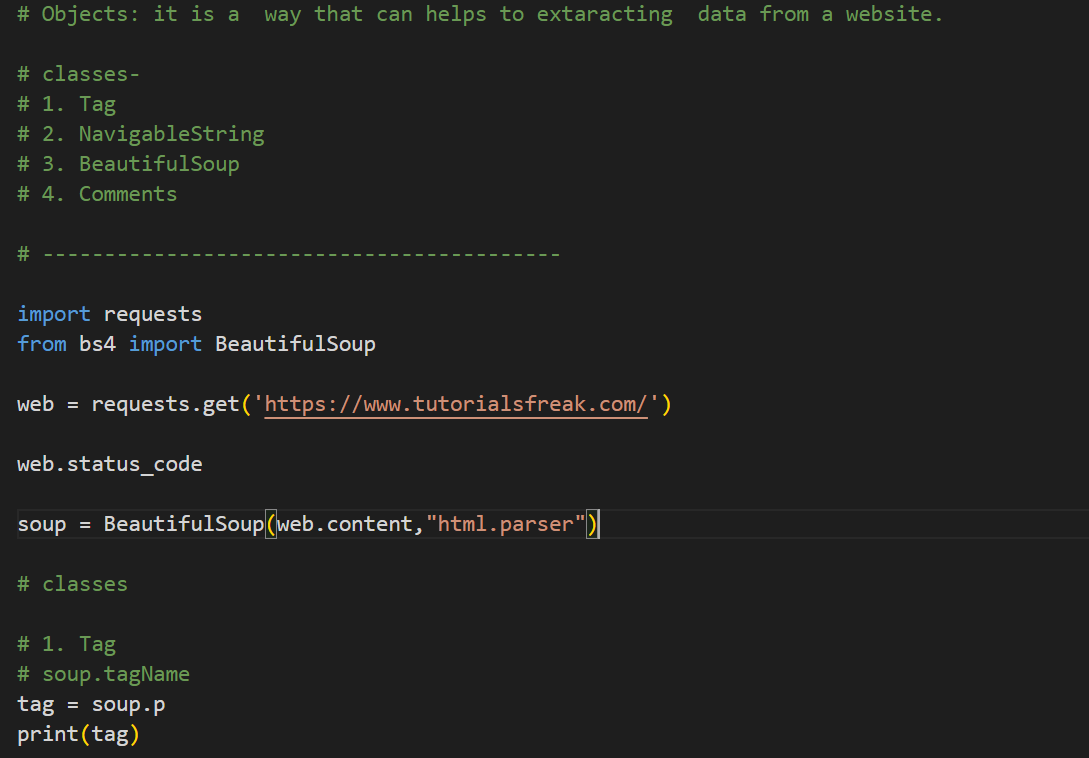
****

**Kinds of Objects Beautiful Soup in web Scraping**

It have 4 types-

1. Tag
2. NavigableString
3. BeauftifulSoup
4. Comments

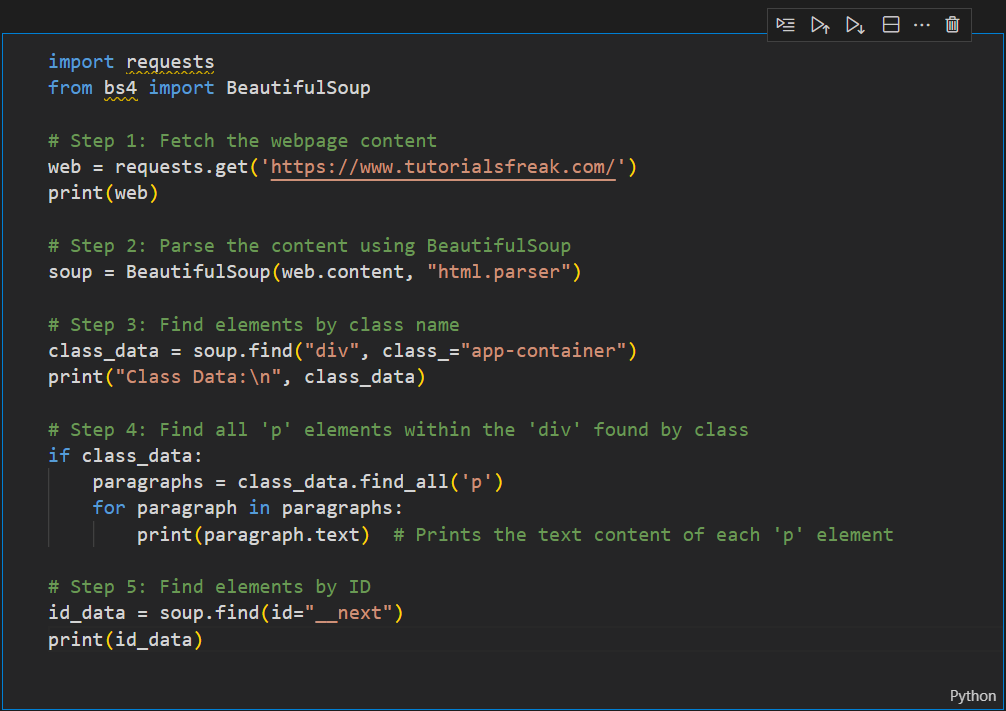
**BeautifulSoup Objects and Classes**

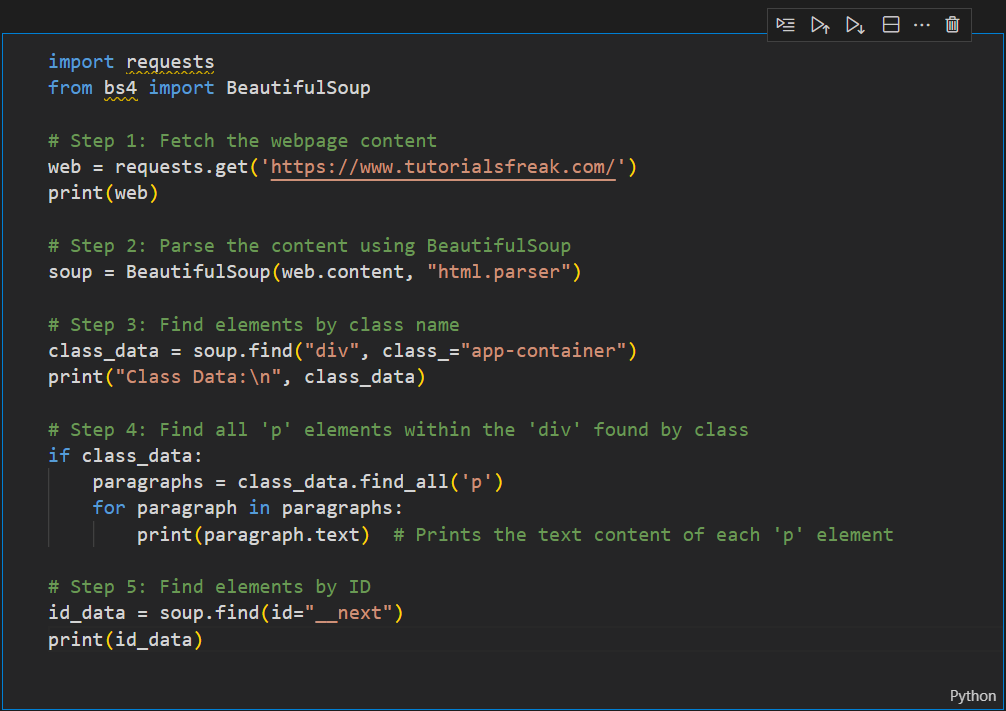


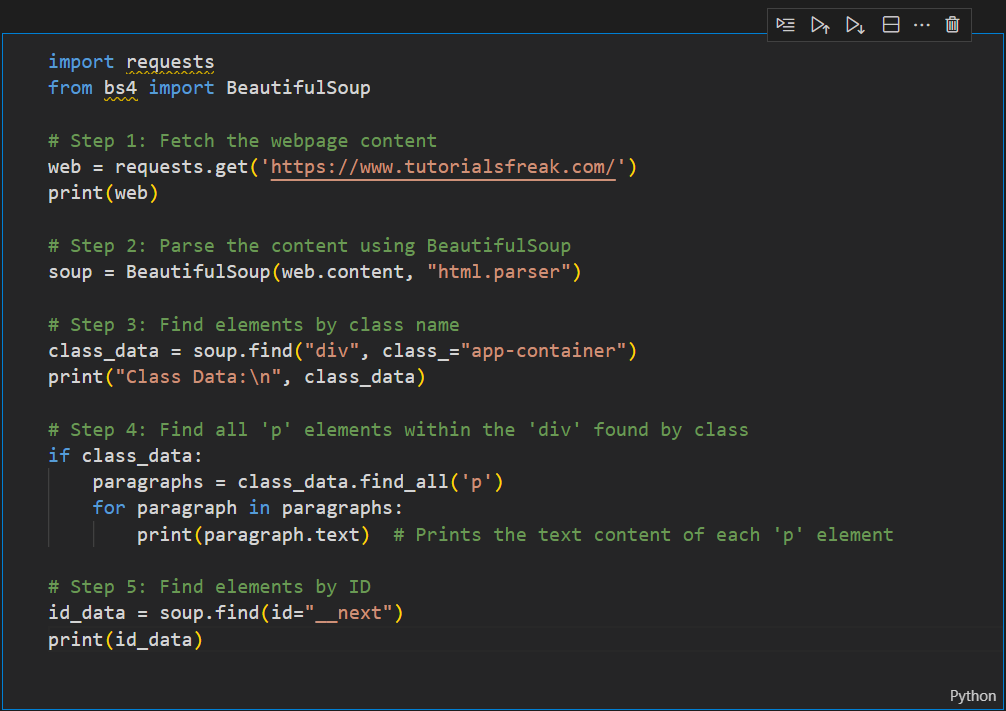
**Finding Elements**

1. **Finding Element using Class**
2. **Finding Element using Id**

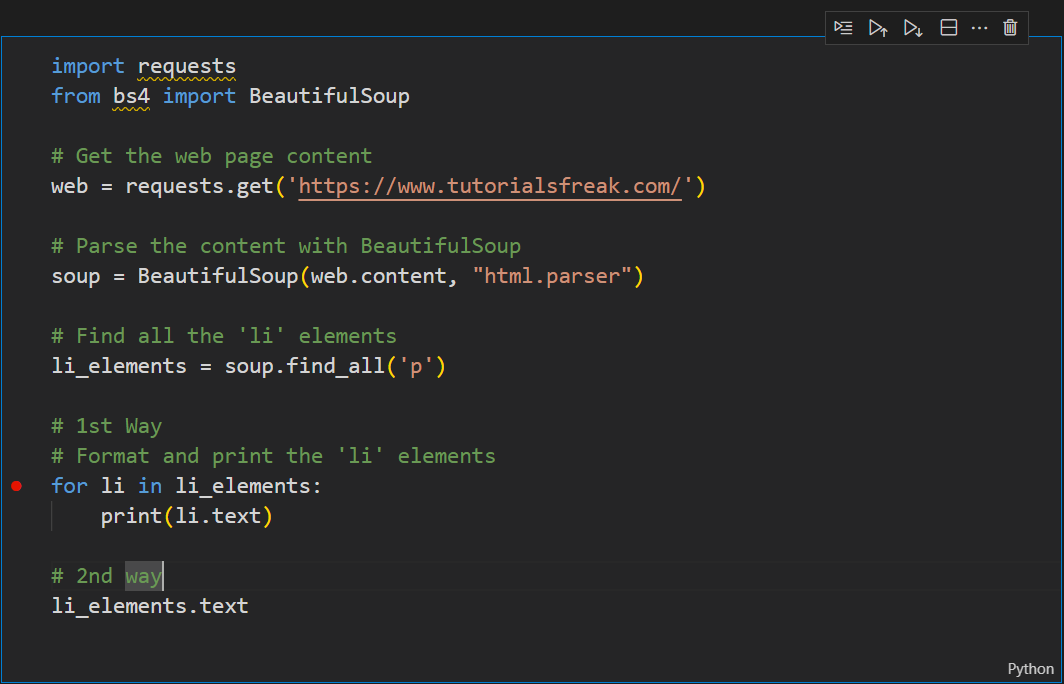
**1. Finding Element using Class**

****

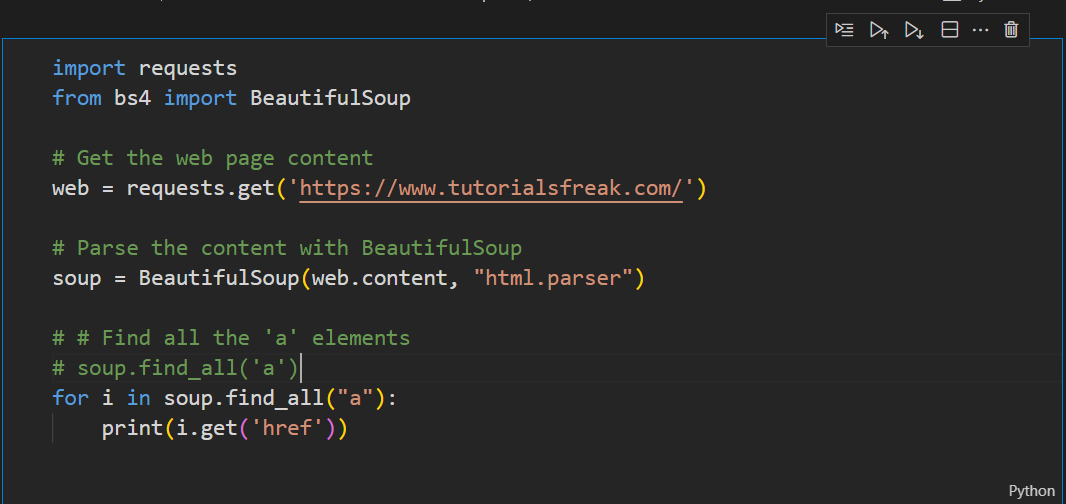
**Syntax:** (‘tagName’,class\_ = className)

**1. Finding Element using Id**

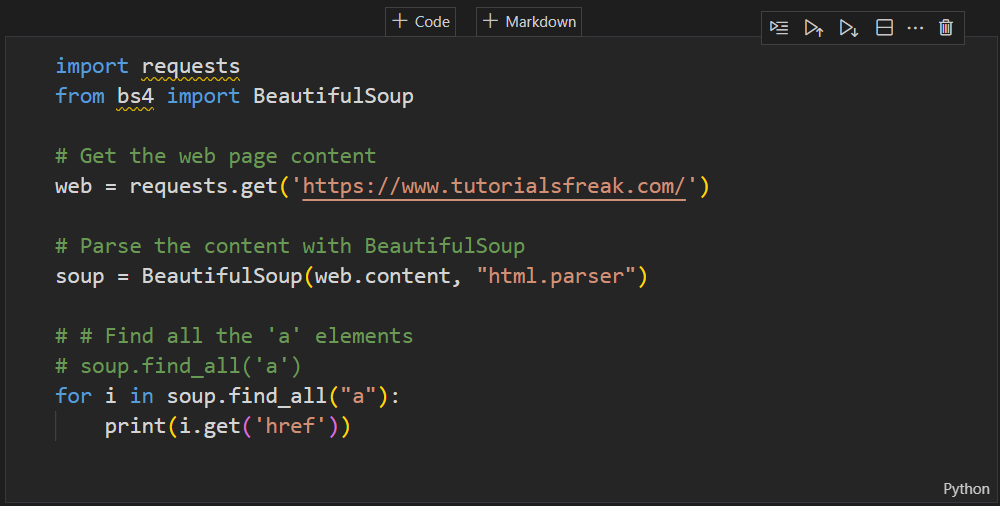
**Extracting Text From Tags -**

****

**Extracting Links From Tags -**



**Extracting Images From Tags -**

****