🌈 BeautifulSoup (bs4) in Python – Colorful Notes

BeautifulSoup is a Python library used for **web scraping** → extracting data from **HTML** and **XML** documents.
✨It works perfectly with `requests` (or any HTML source) to **parse, search, and navigate** web content.

Why use BeautifulSoup?

🦜Find elements by **tag, class, id, or attributes**.
🦜Extract **text, links, images, tables** easily.
🦜Handles even **messy HTML** with a clean structure.

Installation

```
pip install beautifulsoup4
```

🌳 `beautifulsoup4` = package name
🌳 `BeautifulSoup` = class you import

Basic Example

```python
from bs4 import BeautifulSoup

html = """
<html>
  <head><title>My Page</title></head>
  <body>
    <h1>Hello World</h1>
    <p class="info">This is a paragraph.</p>
    <a href="https://example.com">Click me</a>
  </body>
</html>
"""

soup = BeautifulSoup(html, "html.parser")

print(soup.title)         # <title>My Page</title>
print(soup.title.text)    # My Page
```

```
print(soup.h1.text)       # Hello World
print(soup.a["href"])     # https://example.com
```

🙈**Note:** `html.parser` is built-in, but you can also use `lxml` or `html5lib` for faster parsing.

Real Web Example (with requests)

```
import requests
from bs4 import BeautifulSoup

url = "https://quotes.toscrape.com/"
response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

# Get all quotes on the page
quotes = soup.find_all("span", class_="text")

for q in quotes:
    print(q.text)
```

Targeting by Tag, Class, ID & Attributes

Here's an HTML snippet:

```
<html>
  <head>
    <title>Example Page</title>
  </head>
  <body>
    <h1 id="main-title">Welcome to My Website</h1>
    <p class="description">This is a short description.</p>
    <p class="description">Another paragraph with same class.</p>

    <a href="https://example.com" target="_blank">Visit Example</a>
    <a href="https://openai.com" target="_self">Visit OpenAI</a>

    <div data-info="123" class="box">Box with data attribute</div>
  </body>
</html>
```

🌳How to extract with **BeautifulSoup**:

```
from bs4 import BeautifulSoup

html = """ (HTML above) """
soup = BeautifulSoup(html, "html.parser")

# By TAG
print(soup.h1.text)              # Welcome to My Website

# By ID
print(soup.find(id="main-title").text)   # Welcome to My Website

# By CLASS
for p in soup.find_all("p", class_="description"):
    print(p.text)
# → This is a short description.
# → Another paragraph with same class.

# By ATTRIBUTE (target)
link = soup.find("a", {"target": "_blank"})
print(link["href"])   # https://example.com

# Custom attribute (data-info)
box = soup.find("div", {"data-info": "123"})
print(box.text)       # Box with data attribute
```

🙈Breakdown:

- **Tag** → `<h1>`
- **ID** → `id="main-title"`
- **Class** → `class="description"`
- **Attribute** → `target="_blank"`, `data-info="123"`

Common Methods

- 🦘 `soup.find(tag, attrs={...})` → first match
- 🦘 `soup.find_all(tag, attrs={...})` → all matches
- 🦆 `soup.get_text()` → get all text inside a tag
- 🦞 `soup.select("css-selector")` → select using CSS selectors

Quick Recap

- ⚡ `requests` → fetch web page
- ⭐ `BeautifulSoup` → parse & extract data

🙊Together = **powerful web scraping combo!** 🌳

Would you like me to add a **mini-project** example (like scraping news titles 🦊) to make this guide more practical?