🧠 BEAUTIFULSOUP DATA EXTRACTION CHEATSHEET

(Save this. Print this. Ye tera scraper ka हथियार hai.)

Soch:

> **Soup = DOM tree Tag = node Class / attribute = filter**

◍ 1️⃣ Basic find / find_all

```
# Pehla matching tag
soup.find("article")

# Saare matching tags (list)
soup.find_all("article")
```

◍ 2️⃣ Tag + class

```
# Single
soup.find("p", class_="price_color")

# Multiple
soup.find_all("article", class_="product_pod")
```

⚠️ `class` keyword hai → **class_**

◍ 3️⃣ Nested search (MOST COMMON)

```
article = soup.find("article", class_="product_pod")

# Article ke andar <a>
article.find("a")

# Article ke andar <p class="price_color">
article.find("p", class_="price_color")
```

Rule:

> Parent pe kaam karo → child nikaalo Never soup se direct sab nikaalne ki aadat daal ❌

🟢 4️⃣ Text nikalna

```
tag.text           # raw text (with newlines)
tag.text.strip()   # clean text (ALWAYS use strip)
```

Example:

```
price = article.find("p", class_="price_color").text.strip()
```

🟢 5️⃣ Attribute nikalna (VERY IMPORTANT)

⭐ Title (attribute me hota hai)

```
title = article.find("a")["title"]
```

⭐ href (link)

```
link = article.find("a")["href"]
```

Safe way (no crash):

```
link = article.find("a").get("href")
```

🟢 6️⃣ Class list se data nikalna (Star rating)

HTML:

```html
<p class="star-rating Three"></p>
```

Code:

```python
rating_tag = article.find("p", class_="star-rating")

classes = rating_tag["class"]
# ['star-rating', 'Three']

rating_word = classes[1]
```

Convert:

```python
rating_map = {
    "One": 1,
    "Two": 2,
    "Three": 3,
    "Four": 4,
    "Five": 5
}

rating = rating_map.get(rating_word, 0)
```

◍ 7️⃣ Boolean data (availability)

```python
availability_text = article.find(
    "p", class_="instock availability"
).text.strip()

availability = "In stock" in availability_text
```

Pattern:

> Text → condition → True/False

◍ 8️⃣ Multiple classes (space-separated)

```
soup.find("p", class_="instock availability")
```

BeautifulSoup **space ko samajhta hai**.

▨ 9️⃣ Optional tag handling (PRO TIP)

```
tag = article.find("p", class_="price_color")

if tag:
    price = tag.text.strip()
else:
    price = None
```

Isse scraper crash nahi hota.

▨ 🔟 find vs select (CSS selectors)

```
# CSS selector
article.select_one("p.price_color")

# Multiple
article.select("article.product_pod")
```

Rule:

- `find` = simple, readable ☑
- `select` = complex CSS, fragile ❌ (mostly)

▨ 1️⃣1️⃣ Parent → Child chaining

```
article.find("h3").find("a")["title"]
```

Readable + safe (jab structure stable ho).

▨ 1️⃣2️⃣ Debugging Soup (LIFESAVER)

```
print(article.prettify())
```

Isse **actual HTML** dikhega jo scraper dekh raha hai.

🧠 MASTER RULES (YAAD RAKH)

1️⃣ **Soup se direct data mat nikaal** 2️⃣ **Parent pe kaam kar, child nikaal** 3️⃣ *extract_one_ = no loops** 4️⃣ *extract_all_ = sirf loop + delegation** 5️⃣ **Selectors sirf extract methods me**

🔚 TL;DR CHEAT MAP

Kaam	Code
Single tag	`find()`
Multiple tags	`find_all()`
Text	`.text.strip()`
Attribute	`["href"]` / `["title"]`
Class list	`tag["class"]`
Boolean	`"text" in string`
Nested	`parent.find(child)`
