In [1]:

```
from bs4 import BeautifulSoup
import requests
```

In [2]:

```
url = 'https://www.amazon.com/dp/B0BYV51VBN/ref=sspa_dk_detail_3?pd_rd_i=B0BYV51VBN&pd_rd_w=V7rwe&content-id=amzn1.sym.386c274b-4bfe-4421-

page = requests.get(url)

soup = BeautifulSoup(page.text, 'html')
```

In [3]:

```
print(soup)
```

```
<a href="/ref=cs_503_logo"><img alt="Amazon.com" id="b" src="https://images-na.ssl-images-amazon.com/images/G/01/error/l
ogo._TTD_.png"/></a>
<form accept-charset="utf-8" action="/s" id="a" method="GET" role="search">
<div id="c">
<input id="e" name="field-keywords" placeholder="Search"/>
<input name="ref" type="hidden" value="cs_503_search"/>
<input id="f" type="submit" value="Go"/>
</div>
</form>
<div id="g">
<div><a href="/ref=cs_503_link"><img alt="Sorry! Something went wrong on our end. Please go back and try again or go to
Amazon's home page." src="https://images-na.ssl-images-amazon.com/images/G/01/error/500_503.png"/></a>
</div>
<a href="/dogsofamazon/ref=cs_503_d" rel="noopener noreferrer" target="_blank"><img alt="Dogs of Amazon" id="d"/></a>
<script>document.getElementById("d").src = "https://images-na.ssl-images-amazon.com/images/G/01/error/" + (Math.floor(Ma
th.random() * 43) + 1) + "._TTD_.jpg";</script>
</div>
</body>
</html>
```

In [4]:

```
HEADERS = ({'User-Agent':
            'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.157 Safari/537.36',
            'Accept-Language': 'en-US, en;q=0.5'})
```

In [5]:

```
URL = "https://www.amazon.com/dp/B0BYV51VBN/ref=sspa_dk_detail_3?pd_rd_i=B0BYV51VBN&pd_rd_w=V7rwe&content-id=amzn1.sym.386c274b-4bfe-4421-
webpage = requests.get(URL, headers=HEADERS)
```

In [6]:

```
soup = BeautifulSoup(webpage.content, "lxml")
```

In [7]:

```
# Outer Tag Object
title = soup.find("span", attrs={"id":'productTitle'})
```

In [8]:

```
# Inner NavigableString Object
title_value = title.string
```

In [9]:

```
# Title as a string value
title_string = title_value.strip()
```

In [10]:

```
# Printing types of values for efficient understanding
print(type(title))
print(type(title_value))
print(type(title_string))
print()

# Printing Product Title
print("Product Title = ", title_string)
```

```
<class 'bs4.element.Tag'>
<class 'bs4.element.NavigableString'>
<class 'str'>

Product Title =  3 in 1 Magnetic Wireless Travel Foldable Charging Station with Light for Apple Multiple Devices, Mag-Safe
Charger Portable Stand, Compatible with iPhone 14/13/12 Series | Apple Watch | Airpods
```

In [12]:

```
from bs4 import BeautifulSoup
import requests

# Function to extract Product Title
def get_title(soup):

    try:
        # Outer Tag Object
        title = soup.find("span", attrs={"id":'productTitle'})

        # Inner NavigableString Object
        title_value = title.string

        # Title as a string value
        title_string = title_value.strip()

        # # Printing types of values for efficient understanding
        # print(type(title))
        # print(type(title_value))
        # print(type(title_string))
        # print()

    except AttributeError:
        title_string = ""

    return title_string

# Function to extract Product Price
def get_price(soup):

    try:
        price = soup.find("span", attrs={'id':'priceblock_ourprice'}).string.strip()

    except AttributeError:
        price = ""

    return price

# Function to extract Product Rating
def get_rating(soup):

    try:
        rating = soup.find("i", attrs={'class':'a-icon a-icon-star a-star-4-5'}).string.strip()

    except AttributeError:

        try:
            rating = soup.find("span", attrs={'class':'a-icon-alt'}).string.strip()
        except:
            rating = ""

    return rating

# Function to extract Number of User Reviews
def get_review_count(soup):
    try:
        review_count = soup.find("span", attrs={'id':'acrCustomerReviewText'}).string.strip()

    except AttributeError:
        review_count = ""

    return review_count

# Function to extract Availability Status
def get_availability(soup):
    try:
        available = soup.find("div", attrs={'id':'availability'})
        available = available.find("span").string.strip()

    except AttributeError:
        available = ""

    return available

if __name__ == '__main__':

    # Headers for request
    HEADERS = ({'User-Agent':
                'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.157 Safari/537.36',
                'Accept-Language': 'en-US, en;q=0.5'})

    # The webpage URL
    URL = "https://www.amazon.com/dp/B0BYV51VBN/ref=sspa_dk_detail_3?pd_rd_i=B0BYV51VBN&pd_rd_w=V7rwe&content-id=amzn1.sym.386c274b-4bfe-4

    # HTTP Request
    webpage = requests.get(URL, headers=HEADERS)

    # Soup Object containing all data
    soup = BeautifulSoup(webpage.content, "lxml")
```