In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
Customer = pd.read_csv("Customers.csv")
```

In [3]:
```
Product = pd.read_csv("Products.csv")
```

In [4]:
```
Transaction = pd.read_csv("Transactions.csv")
```

In [5]:
```
Customer
```

Out[5]:

	CustomerID	CustomerName	Region	SignupDate
0	C0001	Lawrence Carroll	South America	2022-07-10
1	C0002	Elizabeth Lutz	Asia	2022-02-13
2	C0003	Michael Rivera	South America	2024-03-07
3	C0004	Kathleen Rodriguez	South America	2022-10-09
4	C0005	Laura Weber	Asia	2022-08-15
...	...	...	...	...
195	C0196	Laura Watts	Europe	2022-06-07
196	C0197	Christina Harvey	Europe	2023-03-21
197	C0198	Rebecca Ray	Europe	2022-02-27
198	C0199	Andrea Jenkins	Europe	2022-12-03
199	C0200	Kelly Cross	Asia	2023-06-11

200 rows × 4 columns

In [6]:
```
Customer.shape
```

Out[6]:
```
(200, 4)
```

In [7]:
```
Product
```

Out[7]:

	ProductID	ProductName	Category	Price
0	P001	ActiveWear Biography	Books	169.30
1	P002	ActiveWear Smartwatch	Electronics	346.30
2	P003	ComfortLiving Biography	Books	44.12
3	P004	BookWorld Rug	Home Decor	95.69
4	P005	TechPro T-Shirt	Clothing	429.31
...	...	...	...	...
95	P096	SoundWave Headphones	Electronics	307.47
96	P097	BookWorld Cookbook	Books	319.34
97	P098	SoundWave Laptop	Electronics	299.93
98	P099	SoundWave Mystery Book	Books	354.29
99	P100	HomeSense Sweater	Clothing	126.34

100 rows × 4 columns

In [8]:
```
Product.shape
```

Out[8]:
```
(100, 4)
```

In [9]:
```
Transaction
```

Out[9]:

	TransactionID	CustomerID	ProductID	TransactionDate	Quantity	TotalValue	Price
0	T00001	C0199	P067	2024-08-25 12:38:23	1	300.68	300.68
1	T00112	C0146	P067	2024-05-27 22:23:54	1	300.68	300.68
2	T00166	C0127	P067	2024-04-25 07:38:55	1	300.68	300.68
3	T00272	C0087	P067	2024-03-26 22:55:37	2	601.36	300.68
4	T00363	C0070	P067	2024-03-21 15:10:10	3	902.04	300.68
...	...	...	...	...	...	...	...
995	T00496	C0118	P037	2024-10-24 08:30:27	1	459.86	459.86
996	T00759	C0059	P037	2024-06-04 02:15:24	3	1379.58	459.86
997	T00922	C0018	P037	2024-04-05 13:05:32	4	1839.44	459.86
998	T00959	C0115	P037	2024-09-29 10:16:02	2	919.72	459.86
999	T00992	C0024	P037	2024-04-21 10:52:24	1	459.86	459.86

1000 rows × 7 columns

In [10]:
```
Transaction.shape
```

Out[10]:
```
(1000, 7)
```

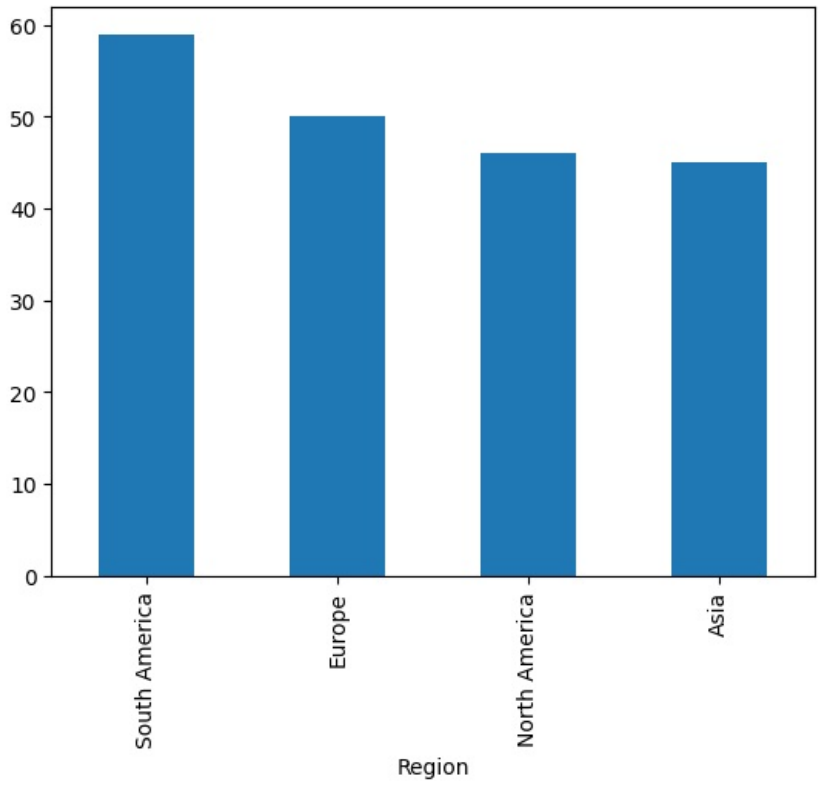
Region (Univariate Analysis)

In [11]:
```
Customer["Region"].value_counts()
```

Out[11]:
```
Region
South America    59
Europe           50
North America    46
Asia             45
Name: count, dtype: int64
```

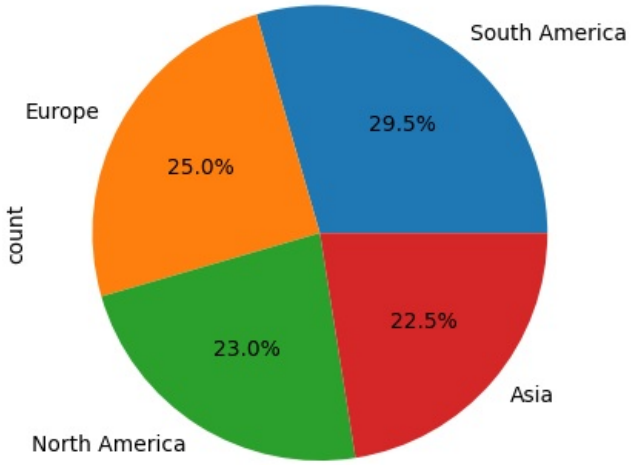
In [12]:
```
Customer["Region"].value_counts().plot(kind="bar")
```

Out[12]:
```
<Axes: xlabel='Region'>
```



In [14]:
```
Customer["Region"].value_counts().plot(kind="pie",autopct="%0.1f%%")
```

Out[14]:
```
<Axes: ylabel='count'>
```



In [15]:
```
Customer["Region"].isnull().sum()
```

Out[15]:
```
0
```

In [16]:
```
Customer["CustomerID"].value_counts().sum()
```

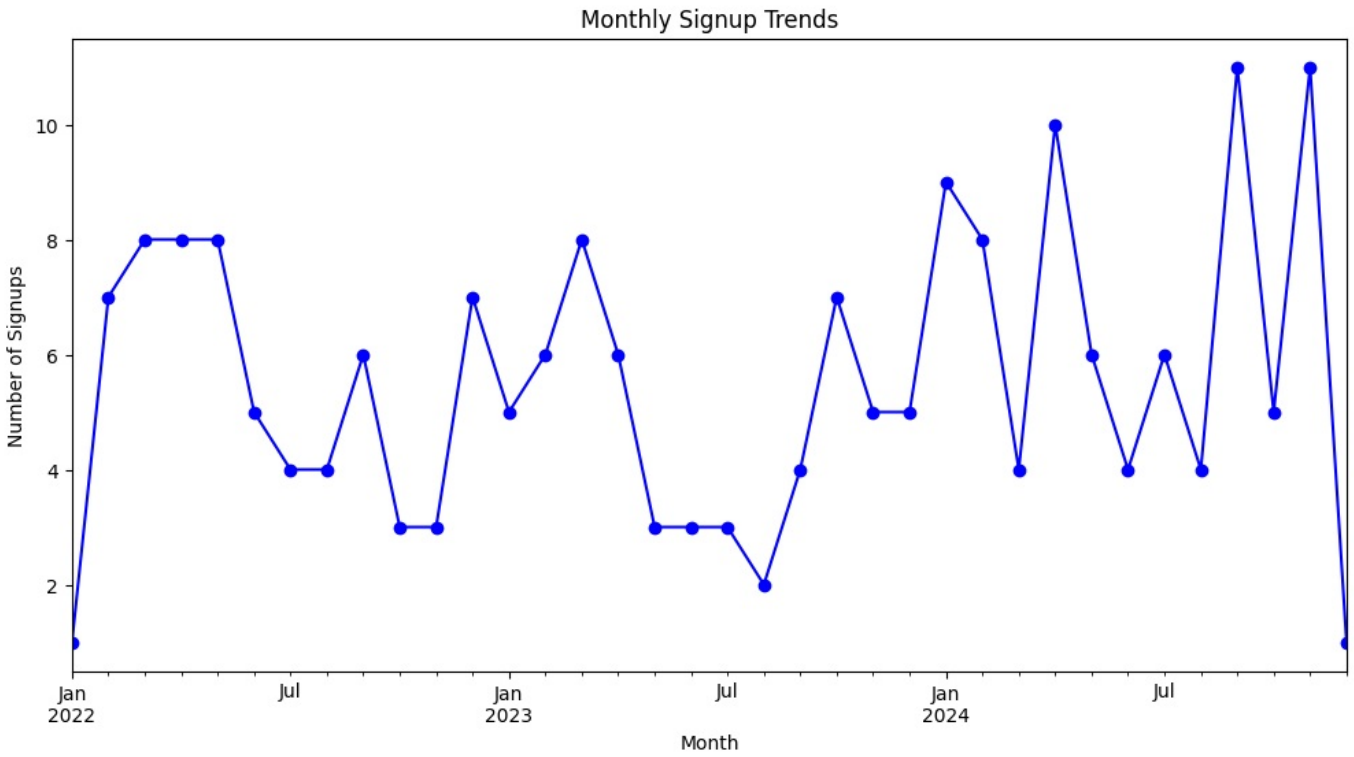
Out[16]:
```
200
```

In [17]:
```
# Ensure the column is properly converted to datetime
Customer['SignupDate'] = pd.to_datetime(Customer['SignupDate'], errors='coerce')

# Check for any conversion errors
if Customer['SignupDate'].isnull().sum() > 0:
    print("There are invalid dates in the SignupDate column.")

# Extract monthly signup trends
signup_trends = Customer['SignupDate'].dt.to_period('M').value_counts().sort_index()

# Plot monthly signup trends
plt.figure(figsize=(12, 6))
signup_trends.plot(kind='line', marker='o', color='blue')
plt.title("Monthly Signup Trends")
plt.xlabel("Month")
plt.ylabel("Number of Signups")
plt.show()
```



Conclusions

There are no missing values in Region column.

There are no Outliers.

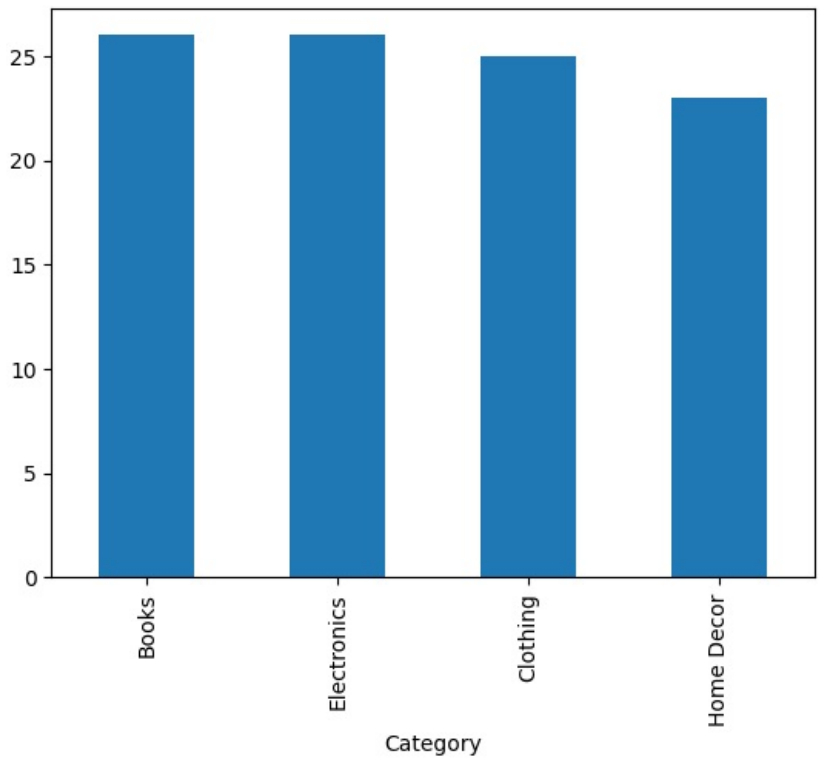
Category(Univariate Analysis)

In [20]:
```
Product["Category"].value_counts()
```

Out[20]:
```
Category
Books          26
Electronics    26
Clothing       25
Home Decor     23
Name: count, dtype: int64
```

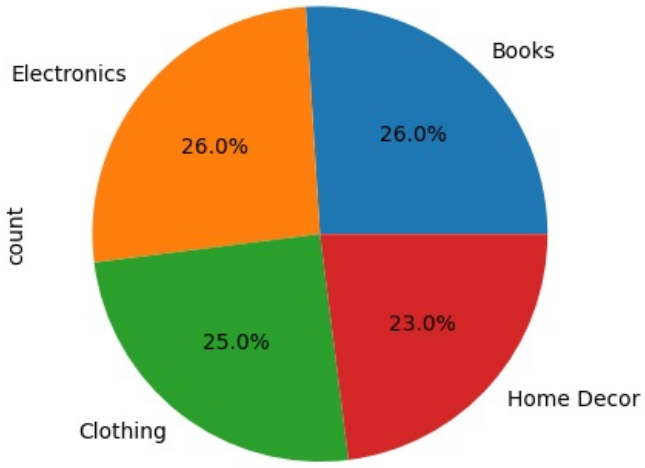
In [21]:
```
Product["Category"].value_counts().plot(kind = "bar")
```

Out[21]:
```
<Axes: xlabel='Category'>
```



In [22]:
```
Product["Category"].value_counts().plot(kind="pie",autopct="%0.1f%%")
```

Out[22]:
```
<Axes: ylabel='count'>
```



In [23]:
```
Product["Category"].isnull().sum()
```

Out[23]:
```
0
```

Conclusions

There are no missing values.

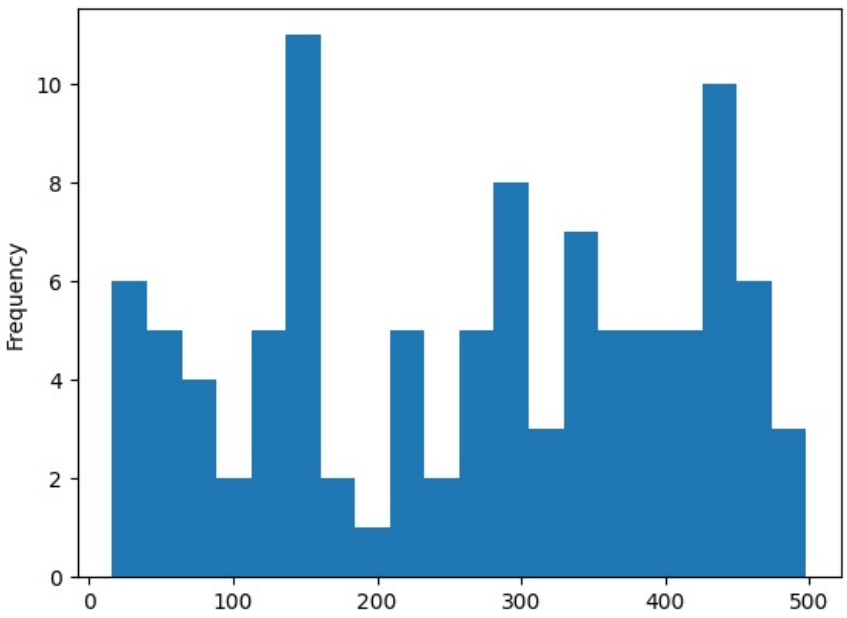
Price

In [25]:
```
Product["Price"].describe()
```

Out[25]:
```
count    100.000000
mean     267.551700
std      143.219383
min       16.080000
25%      147.767500
50%      292.875000
75%      397.090000
max      497.760000
Name: Price, dtype: float64
```

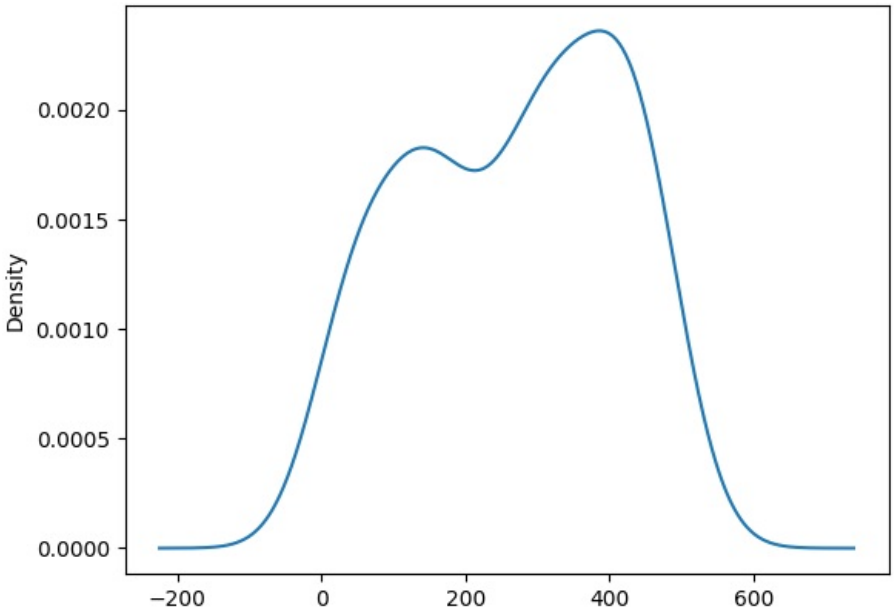
In [26]:
```
Product["Price"].plot(kind="hist",bins = 20)
```

Out[26]:
```
<Axes: ylabel='Frequency'>
```



In [27]:
```
Product["Price"].plot(kind="kde")
```

Out[27]: <Axes: ylabel='Density'>

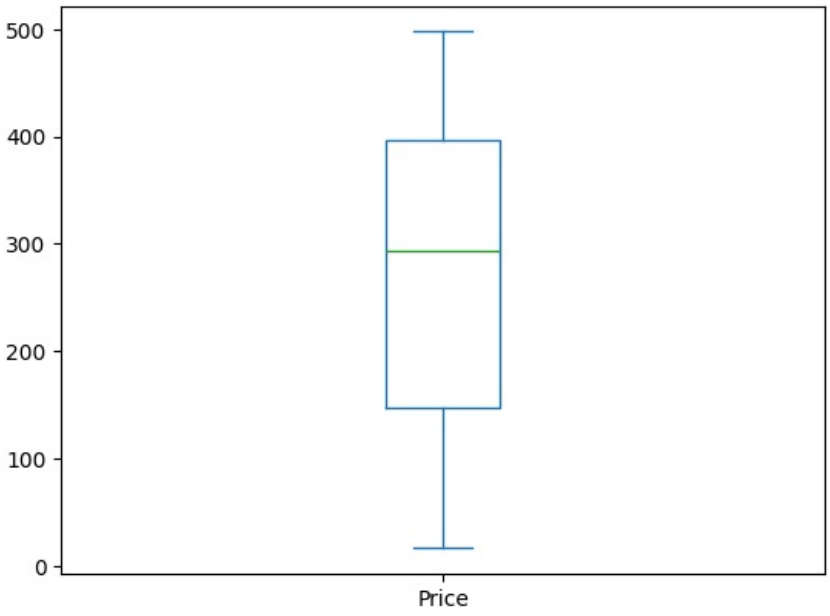


In [28]:
```
Product["Price"].skew()
```

Out[28]: -0.2076196737826489

In [29]:
```
Product["Price"].plot(kind="box")
```

Out[29]: <Axes: >



Conclusions

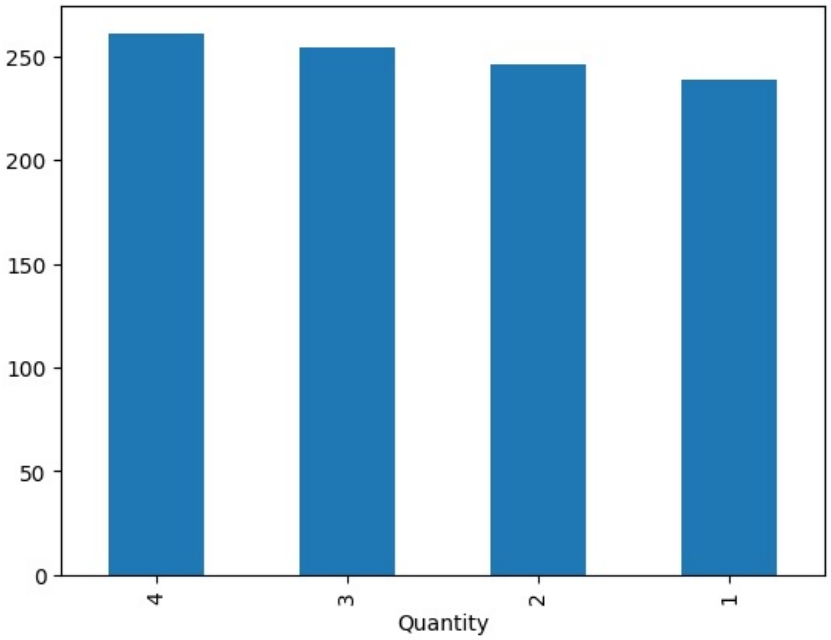
There are no Outliers.

It is left skewed.

Transactions

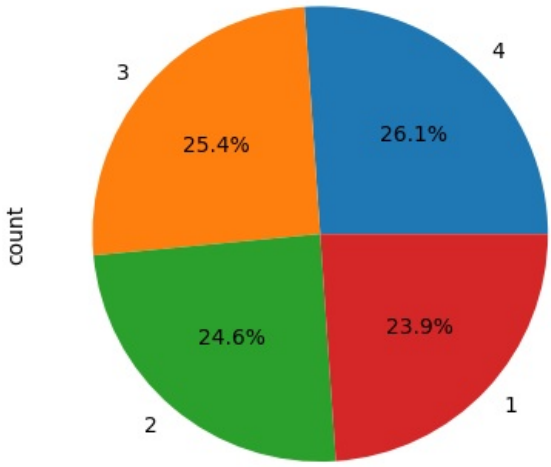
In [31]:
```
Transaction["Quantity"].value_counts().plot(kind="bar")
```

Out[31]: <Axes: xlabel='Quantity'>



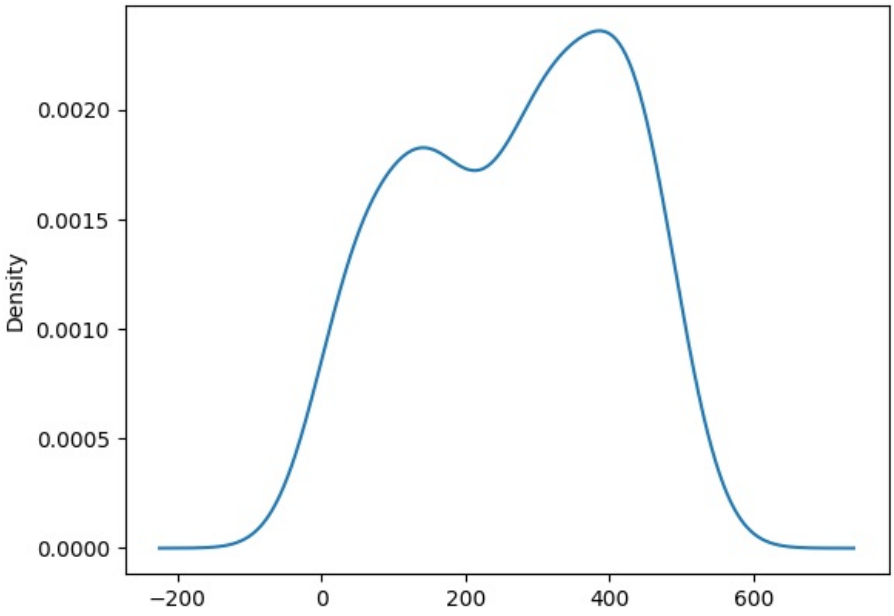
In [32]:
```
Transaction["Quantity"].value_counts().plot(kind="pie",autopct="%0.1f%%")
```

Out[32]: <Axes: ylabel='count'>



In [33]:
```
Product["Price"].plot(kind="kde")
```

Out[33]: <Axes: ylabel='Density'>

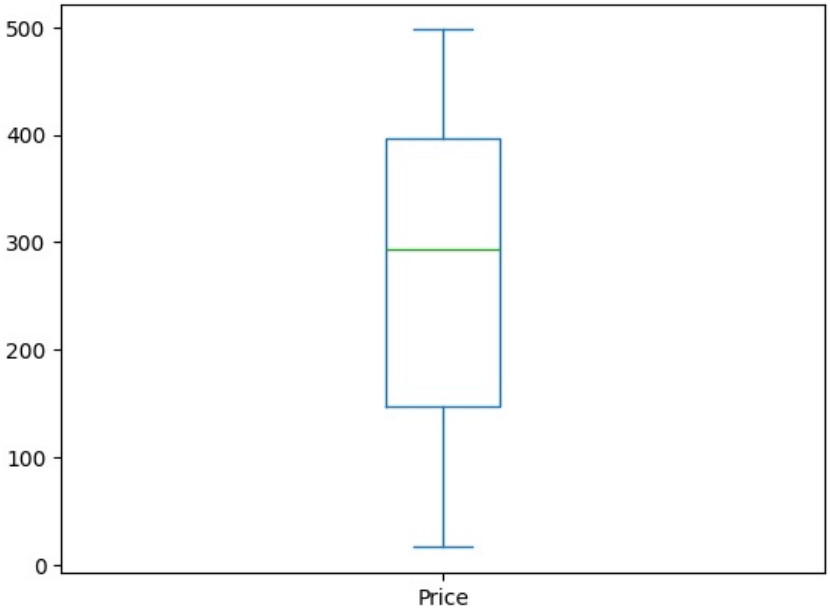


In [34]:
```
Product["Price"].skew()
```

Out[34]: -0.2076196737826489

In [35]:
```
Product["Price"].plot(kind="box")
```

Out[35]: <Axes: >



Conclusions

There are no Outliers.

It is left skewed.

5 Business Insights

1. Top-Selling Products: The top 10 products contribute significantly to revenue. These should be prioritized for inventory and promotions.
2. Books Dominate: The "Books" category has the most products, showing customer preference. Offering discounts in this category could boost sales.
3. Regional Sales Focus: South America leads in transactions. Focusing marketing efforts in this region could yield higher returns.
4. Repeat Customers: Customers with frequent purchases (e.g., C0109) should be targeted for loyalty programs or personalized offers.
5. New Customer Acquisition: Efforts should focus on regions like Asia, which show fewer transactions but high potential for growth.