Member-only story

Python Pandas: loc and iloc

How to Select and Filter Data in Python

Python Fundamentals · Following
Published in Towards Dev · 5 min read · Mar 16, 2024

Python pandas library provides several methods for selecting and filtering data, such as **loc, iloc, [ ] bracket operator, query, isin, between.**

Image from pexels.com

This article will guide you through the essential techniques and functions for data selection and filtering using pandas. Whether you need to extract specific rows or columns or apply conditional filtering, pandas has got you covered. Let's dive in!

Table of Contents

1. Selecting Columns

- **loc[ ] :** This accessor selects *rows and columns by labels.*
  Example: `df.loc[row_label, column_label]`

*** You can also use `loc` for ***slicing operations:***

```
df.loc['row1_label':'row2_label' , 'column1_label':'column2_label']
```

```
# Using loc for label-based selection
df.loc[:, 'Customer Country':'Customer State']
```

	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State
0	Puerto Rico	Sean	568	Consumer	PR
1	Puerto Rico	Carol	3341	Consumer	PR
2	Puerto Rico	Mary	7459	Consumer	PR
3	Puerto Rico	Mary	7459	Consumer	PR
4	Puerto Rico	Mary	10740	Consumer	PR
...	...	...	...	...	...
115741	United States	Mary	2666	Corporate	CA
115742	United States	Victoria	8100	Corporate	CA
115743	United States	Mary	2666	Corporate	CA
115744	United States	Mary	9547	Corporate	CA
115745	United States	Mary	11747	Corporate	NV

115746 rows × 5 columns

all the rows & columns from Customer Country to Customer State

```
# Using loc for label-based selection
df.loc[[0,1,2], 'Customer Country':'Customer State']
```

	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State
0	Puerto Rico	Sean	568	Consumer	PR
1	Puerto Rico	Carol	3341	Consumer	PR
2	Puerto Rico	Mary	7459	Consumer	PR

rows 0,1,2 & columns from Customer Country to Customer State

- **iloc[ ]** : This accessor selects *rows and columns by integer location.*
  Example: `df.iloc[row_position, column_position]`

*** You can also use `iloc` for ***slicing operations:***

`df.iloc['row1_position':'row2_position','col1_position':'col2_position']`

```
# Using iloc for index-based selection
df.iloc[[0,1,2,3] , [3,4,5,6,7,8]]

# or
df.iloc[[0,1,2,3] , 3:9]
```

	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode
0	Puerto Rico	Sean	568	Consumer	PR	725.0
1	Puerto Rico	Carol	3341	Consumer	PR	725.0
2	Puerto Rico	Mary	7459	Consumer	PR	680.0
3	Puerto Rico	Mary	7459	Consumer	PR	680.0

rows 0,1,2,3 & columns 3 to 8

```
# Using iloc for index-based selection
df.iloc[:, 3:8]
```

	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State
0	Puerto Rico	Sean	568	Consumer	PR
1	Puerto Rico	Carol	3341	Consumer	PR
2	Puerto Rico	Mary	7459	Consumer	PR
3	Puerto Rico	Mary	7459	Consumer	PR
4	Puerto Rico	Mary	10740	Consumer	PR
...	...	...	...	...	...
115741	United States	Mary	2666	Corporate	CA
115742	United States	Victoria	8100	Corporate	CA
115743	United States	Mary	2666	Corporate	CA
115744	United States	Mary	9547	Corporate	CA
115745	United States	Mary	11747	Corporate	NV

115746 rows × 5 columns

all the rows & columns from 3 to 8

- **[ ] Bracket operator :** It allows to *select one or multiple columns*.
  Example: `df[['column_label']]` or `df[['column1', 'column2']]`

```
# Selecting a single column
df[['Customer Country']]
```

	Customer Country
0	Puerto Rico
1	Puerto Rico
2	Puerto Rico
3	Puerto Rico
4	Puerto Rico
...	...
115741	United States
115742	United States
115743	United States
115744	United States
115745	United States

115746 rows × 1 columns

a single column

```
# Selecting multiple columns
df[['Customer Country', 'Customer State']]
```

	Customer Country	Customer State
0	Puerto Rico	PR
1	Puerto Rico	PR
2	Puerto Rico	PR
3	Puerto Rico	PR
4	Puerto Rico	PR
...	...	...
115741	United States	CA
115742	United States	CA
115743	United States	CA
115744	United States	CA
115745	United States	NV

115746 rows × 2 columns

multiple columns

2. Filtering Rows

- loc[ ] : It filters *rows by labels.*

  Example: `df.loc[condition]`

```
# Using loc for filtering rows
condition = df['Order Quantity'] > 3
df.loc[condition]

# or
df.loc[df['Order Quantity'] > 3]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
45	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Tammy	6398	Consumer	PR	725.0	LATAM	6398	02-07-2017	52640	South America	189.949997	5
46	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	8917	Consumer	PR	725.0	LATAM	8917	13-04-2017	57106	South America	236.250000	5
47	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Christina	1425	Consumer	PR	725.0	LATAM	1425	31-01-2017	52166	Caribbean	136.500000	5
48	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Mary	2737	Consumer	PR	725.0	LATAM	2737	30-03-2017	56172	Caribbean	181.949997	5
49	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	6428	Consumer	PR	725.0	LATAM	6428	25-03-2017	55829	Caribbean	225.000000	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115364	Indoor/Outdoor Games	Indoor/Outdoor Games	Fort Lauderdale	United States	Matthew	4915	Corporate	FL	33324.0	Pacific Asia	4915	17-03-2016	30275	Southeast Asia	244.899994	5
115365	Cleats	Cleats	Winter Park	United States	Nancy	8694	Corporate	FL	32792.0	Pacific Asia	8694	26-03-2016	30860	Oceania	287.950012	5
115366	Cleats	Cleats	West Covina	United States	Mary	6187	Corporate	CA	91790.0	Pacific Asia	6187	03-03-2016	29270	South Asia	283.450012	5
115367	Indoor/Outdoor Games	Indoor/Outdoor Games	Milford	United States	Brenda	5362	Corporate	CT	6460.0	Pacific Asia	5362	18-01-2016	26235	Southeast Asia	236.160004	5
115368	Girls' Apparel	Girls' Apparel	New Orleans	United States	Mary	11730	Corporate	LA	70117.0	Pacific Asia	11730	28-03-2016	31000	Oceania	325.500000	5

39716 rows × 20 columns

dataframe with the Order Quantity > 3

```
# Using loc for filtering rows
df.loc[df['Customer Country'] == 'United States']
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State
117	Electronics	Electronics	Freeport	United States	Patricia	1509	Consumer	NY
118	Cardio Equipment	Cardio Equipment	Fort Washington	United States	Julie	1636	Consumer	MD
119	Cardio Equipment	Cardio Equipment	Bakersfield	United States	Lisa	2784	Consumer	CA
120	Cardio Equipment	Cardio Equipment	Fort Washington	United States	Julie	1636	Consumer	MD
121	Cleats	Cleats	Corona	United States	Tyler	9174	Consumer	NY

dataframe with the Customer Country = United States

- **iloc() :** It filters *rows by integer positions.*

```
# Using iloc for filtering rows
df.iloc[[0, 2, 4]]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	C
0	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Sean	568	Consumer	PR	725.0	LATAM	
2	Camping & Hiking	Camping & Hiking	Mayaguez	Puerto Rico	Mary	7459	Consumer	PR	680.0	LATAM	
4	Hunting & Shooting	Hunting & Shooting	Caguas	Puerto Rico	Mary	10740	Consumer	PR	725.0	LATAM	

rows with integer index 0, 2 and 4

```
# Using iloc for filtering rows
df.iloc[:3, :2]
```

	Additional Order items	Category Name
0	Camping & Hiking	Camping & Hiking
1	Camping & Hiking	Camping & Hiking
2	Camping & Hiking	Camping & Hiking

- **[ ] Bracket operator :** It allows *filtering rows based on a condition.*

  Example: `df[condition]`

```
# Using [] bracket operator for filtering rows# Using [] bracket operator for fi
condition = df['Order Quantity'] > 3
df[condition]
```

```
# or
df[df['Order Quantity'] > 3]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
45	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Tammy	6398	Consumer	PR	725.0	LATAM	6398	02-07-2017	52640	South America	189.949997	5
46	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	8917	Consumer	PR	725.0	LATAM	8917	13-04-2017	57106	South America	236.250000	5
47	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Christina	1425	Consumer	PR	725.0	LATAM	1425	31-01-2017	52166	Caribbean	136.500000	5
48	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Mary	2737	Consumer	PR	725.0	LATAM	2737	30-03-2017	56172	Caribbean	181.949997	5
49	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	6428	Consumer	PR	725.0	LATAM	6428	25-03-2017	55829	Caribbean	225.000000	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115364	Indoor/Outdoor Games	Indoor/Outdoor Games	Fort Lauderdale	United States	Matthew	4915	Corporate	FL	33324.0	Pacific Asia	4915	17-03-2016	30275	Southeast Asia	244.899994	5
115365	Cleats	Cleats	Winter Park	United States	Nancy	8694	Corporate	FL	32792.0	Pacific Asia	8694	26-03-2016	30860	Oceania	287.950012	5
115366	Cleats	Cleats	West Covina	United States	Mary	6187	Corporate	CA	91790.0	Pacific Asia	6187	03-03-2016	29270	South Asia	283.450012	5
115367	Indoor/Outdoor Games	Indoor/Outdoor Games	Milford	United States	Brenda	5362	Corporate	CT	6460.0	Pacific Asia	5362	18-01-2016	26235	Southeast Asia	236.160004	5
115368	Girls' Apparel	Girls' Apparel	New Orleans	United States	Mary	11730	Corporate	LA	70117.0	Pacific Asia	11730	28-03-2016	31000	Oceania	325.500000	5

39716 rows × 20 columns

dataframe with the order quantity > 3

- **isin([ ])** : It is used to filter data based on a list.

  Example: `df[df['column_name'].isin(['value1', 'value2'])]`

```
# Using isin for filtering rows
df[df['Customer Country'].isin(['United States', 'Puerto Rico'])]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market
0	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Sean	568	Consumer	PR	725.0	LATAM
1	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Carol	3341	Consumer	PR	725.0	LATAM
2	Camping & Hiking	Camping & Hiking	Mayaguez	Puerto Rico	Mary	7459	Consumer	PR	680.0	LATAM
3	Camping & Hiking	Camping & Hiking	Mayaguez	Puerto Rico	Mary	7459	Consumer	PR	680.0	LATAM
4	Hunting & Shooting	Hunting & Shooting	Caguas	Puerto Rico	Mary	10740	Consumer	PR	725.0	LATAM
...	...	...	...	...	...	...	...	...	...	...
115741	Water Sports	Water Sports	Los Angeles	United States	Mary	2666	Corporate	CA	90033.0	USCA
115742	Shop By Sport	Shop By Sport	Upland	United States	Victoria	8100	Corporate	CA	91786.0	USCA
115743	Fishing	Fishing	Los Angeles	United States	Mary	2666	Corporate	CA	90033.0	USCA
115744	Fishing	Fishing	Los Angeles	United States	Mary	9547	Corporate	CA	90027.0	USCA
115745	Golf Balls	Golf Balls	Las Vegas	United States	Mary	11747	Corporate	NV	89123.0	USCA

115746 rows × 20 columns

rows where column ‘Customer Country’ is ‘United States’ or ‘Puerto Rico’

```
# Filter rows based on values in a list and select spesific columns
df[["Customer Id", "Order Region"]][df['Order Region'].isin(['Central America',
```

	Customer Id	Order Region
2	7459	Central America
3	7459	Central America
4	10740	Central America
6	2092	Caribbean
7	9645	Central America
...	...	...
112429	6671	Central America
112431	908	Central America
112432	908	Central America
112433	3830	Caribbean
112434	622	Central America

18489 rows × 2 columns

it selected Customer Id and Order Region columns, where Order Region is Central America or Caribbean

```
# Using NOT isin for filtering rows
df[~df['Customer Country'].isin(['United States'])]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market
0	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Sean	568	Consumer	PR	725.0	LATAM
1	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Carol	3341	Consumer	PR	725.0	LATAM
2	Camping & Hiking	Camping & Hiking	Mayaguez	Puerto Rico	Mary	7459	Consumer	PR	680.0	LATAM
3	Camping & Hiking	Camping & Hiking	Mayaguez	Puerto Rico	Mary	7459	Consumer	PR	680.0	LATAM
4	Hunting & Shooting	Hunting & Shooting	Caguas	Puerto Rico	Mary	10740	Consumer	PR	725.0	LATAM
...	...	...	...	...	...	...	...	...	...	...
112430	Cleats	Cleats	Caguas	Puerto Rico	Mary	4517	Corporate	PR	725.0	LATAM
112431	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Mary	908	Corporate	PR	725.0	LATAM
112432	Fishing	Fishing	Caguas	Puerto Rico	Mary	908	Corporate	PR	725.0	LATAM
112433	Men's Footwear	Men's Footwear	Caguas	Puerto Rico	William	3830	Corporate	PR	725.0	LATAM
112434	Water Sports	Water Sports	Caguas	Puerto Rico	Karen	622	Corporate	PR	725.0	LATAM

44458 rows × 20 columns

rows where column 'Customer Country' is NOT 'United States'

- **query()** : This method is used to select data based on a SQL-like expression.
  Example: `df.query('condition')`

In case your column names contain spaces or special characters, first you should use the rename() function to rename them.

```
# Rename the columns before performing the query
df.rename(columns={'Order Quantity' : 'Order_Quantity', "Customer Fname" : "Cust
```

```
# Using query for filtering rows with a single condition
df.query('Order_Quantity > 3')
```

	Additional Order Items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
45	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Tammy	6398	Consumer	PR	725.0	LATAM	6398	02-07-2017	52640	South America	189.949997	5
46	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	8917	Consumer	PR	725.0	LATAM	8917	13-04-2017	57106	South America	236.250000	5
47	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Christina	1425	Consumer	PR	725.0	LATAM	1425	31-01-2017	52186	Caribbean	136.500000	5
48	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Mary	2737	Consumer	PR	725.0	LATAM	2737	30-03-2017	56172	Caribbean	181.949997	5
49	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	6428	Consumer	PR	725.0	LATAM	6428	25-03-2017	55829	Caribbean	225.000000	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115364	Indoor/Outdoor Games	Indoor/Outdoor Games	Fort Lauderdale	United States	Matthew	4915	Corporate	FL	33324.0	Pacific Asia	4915	17-03-2016	30275	Southeast Asia	244.899994	5
115365	Cleats	Cleats	Winter Park	United States	Nancy	8694	Corporate	FL	32792.0	Pacific Asia	8694	26-03-2016	30860	Oceania	287.950012	5
115366	Cleats	Cleats	West Covina	United States	Mary	6187	Corporate	CA	91790.0	Pacific Asia	6187	03-03-2016	29270	South Asia	283.450012	5
115367	Indoor/Outdoor Games	Indoor/Outdoor Games	Milford	United States	Brenda	5362	Corporate	CT	6460.0	Pacific Asia	5362	18-01-2016	26235	Southeast Asia	236.160004	5
115368	Girls' Apparel	Girls' Apparel	New Orleans	United States	Mary	11730	Corporate	LA	70117.0	Pacific Asia	11730	28-03-2016	31000	Oceania	325.500000	5

39716 rows × 20 columns

dataframe with the order quantity > 3

```
# Using query for filtering rows with multiple conditions
df.query('Order_Quantity > 3 and Customer_Fname == "Mary"')
```

	Additional Order Items	Category Name	Customer City	Customer Country	Customer_Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order_Quantity
46	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	8917	Consumer	PR	725.0	LATAM	8917	13-04-2017	57106	South America	236.250000	5
48	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Mary	2737	Consumer	PR	725.0	LATAM	2737	30-03-2017	56172	Caribbean	181.949997	5
49	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	6428	Consumer	PR	725.0	LATAM	6428	25-03-2017	55829	Caribbean	225.000000	5
52	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	9204	Consumer	PR	725.0	LATAM	9204	27-05-2017	60127	South America	217.500000	5
53	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	9554	Consumer	PR	725.0	LATAM	9554	31-05-2017	60386	South America	212.500000	5

dataframe with the order quantity > 3 and Customer Fname = Mary

- **between()** : Filters rows based on values that fall within a specified range.
  Example: `df[df['column_name'].between(start, end)]`

```
# Filter rows based on values within a range
df[df['Order Quantity'].between(3, 5)]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
26	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Brenda	5197	Home Office	PR	725.0	Africa	5197	10-04-2016	44046	West Africa	263.970001	3
27	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Frank	1535	Home Office	PR	725.0	Africa	1535	11-08-2016	46414	Central Africa	113.970001	3
28	Trade-In	Trade-In	Caguas	Puerto Rico	Johnny	6122	Home Office	PR	725.0	Africa	6122	11-11-2016	46599	West Africa	63.990002	3
29	Electronics	Electronics	Caguas	Puerto Rico	Mary	9451	Home Office	PR	725.0	Africa	9451	12-08-2016	48434	North Africa	79.660004	3
30	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Hannah	11329	Home Office	PR	725.0	Europe	11329	22-07-2017	63936	Western Europe	269.970001	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115364	Indoor/Outdoor Games	Indoor/Outdoor Games	Fort Lauderdale	United States	Matthew	4915	Corporate	FL	33324.0	Pacific Asia	4915	17-03-2016	30275	Southeast Asia	244.899994	5
115365	Cleats	Cleats	Winter Park	United States	Nancy	8694	Corporate	FL	32792.0	Pacific Asia	8694	26-03-2016	30860	Oceania	287.950012	5
115366	Cleats	Cleats	West Covina	United States	Mary	6187	Corporate	CA	91790.0	Pacific Asia	6187	03-03-2016	29270	South Asia	283.450012	5

dataframe with the order quantity between 3 and 5

- **String methods** : Filters rows based on string matching conditions.
  Example: `str.startswith()`, `str.endswith()`, `str.contains()`

```
# Using str.startswith() for filtering rows
df[df['Category Name'].str.startswith('Cardio')]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market
26	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Brenda	5197	Home Office	PR	725.0	Africa
30	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Hannah	11329	Home Office	PR	725.0	Europe
31	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Mary	3570	Home Office	PR	725.0	Europe
65	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Eugene	6277	Consumer	PR	725.0	LATAM
66	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	John	4126	Consumer	PR	725.0	LATAM

```
# Using str.contains() for filtering rows
df[df['Customer Segment'].str.contains('Office')]
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market
16	Cleats	Cleats	Bayamon	Puerto Rico	Mary	9083	Home Office	PR	957.0	Pacific Asia
17	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Mary	4741	Home Office	PR	725.0	Pacific Asia
18	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Elizabeth	639	Home Office	PR	725.0	Pacific Asia
19	Shop By Sport	Shop By Sport	Caguas	Puerto Rico	Katherine	9702	Home Office	PR	725.0	Pacific Asia
20	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	9114	Home Office	PR	725.0	Pacific Asia

3. Updating Values

- **loc[ ]** : This accessor selects specific rows and columns in the DataFrame and assigns new values.

```
# Update values in a column based on a condition
df.loc[df['Customer Country'] == 'United States', 'Customer Country'] = 'USA'
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	C
94936	Men's Footwear	Men's Footwear	Caguas	Puerto Rico	Elizabeth	2143	Consumer	PR	
64706	Cleats	Cleats	Meridian	USA	Mary	9371	Consumer	ID	
42102	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Pamela	2390	Corporate	PR	
78583	Women's Apparel	Women's Apparel	Far Rockaway	USA	Patricia	5543	Corporate	NY	
96145	Fishing	Fishing	Caguas	Puerto Rico	Ruth	3150	Home Office	PR	

It changed the United Stated to USA, in Customer Country column

- **iloc[ ]** : This accessor selects specific rows and columns in the DataFrame and assigns new values.

```
# Update values in a column based on a condition
df.iloc[df['Order Quantity'] > 3, 15] = 'greater than 3'
```

```
# or
condition = df['Order Quantity'] > 3
df.iloc[condition, 15] = 'greater than 3'
```

	Additional Order items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
60653	Cardio Equipment	Cardio Equipment	Caguas	Puerto Rico	Marie	3883	Corporate	PR	725.0	LATAM	3883	21-02-2017	53592	South America	359.959992	greater than 3
11544	Cleats	Cleats	Saint Paul	United States	Paul	2995	Corporate	MN	55106.0	LATAM	2995	16-04-2017	57300	South America	50.990002	greater than 3
109276	Consumer Electronics	Consumer Electronics	Caguas	Puerto Rico	Kellie	14444	Home Office	PR	725.0	Europe	14444	31-10-2017	70891	Southern Europe	238.970001	1
50286	Men's Footwear	Men's Footwear	Hayward	United States	Mary	8792	Consumer	CA	94541.0	Europe	8792	10-07-2016	44208	Eastern Europe	116.989998	1
14058	Cleats	Cleats	Endicott	United States	Mary	6701	Consumer	NY	13760.0	USCA	6701	21-04-2016	32633	West of USA	101.980003	2

It changed the values greater than 3 in Order Quantity column to “greater than 3” text

- **replace()** : It replaces specific values in a DataFrame with new values.

  Ex: `df.['column_name'].replace(old_value, new_value, inplace=True)`

```
# Replace specific values in a column
df['Order Quantity'].replace(5, 'equals 5', inplace=True)
```

	Additional Order Items	Category Name	Customer City	Customer Country	Customer Fname	Customer Id	Customer Segment	Customer State	Customer Zipcode	Market	Order Customer Id	Order Date	Order Id	Order Region	Order Item Total	Order Quantity
5144	Fishing	Fishing	Tonawanda	United States	Stephen	5268	Consumer	NY	14150.0	LATAM	5268	29-04-2017	58207	South America	339.980011	equals 5
18310	Shop By Sport	Shop By Sport	Bronx	United States	David	11787	Consumer	NY	10460.0	USCA	11787	19-08-2016	40852	East of USA	151.960007	4
46346	Women's Apparel	Women's Apparel	Caguas	Puerto Rico	Mary	3767	Consumer	PR	725.0	USCA	3767	15-05-2016	34308	West of USA	196.000000	4
5135	Fishing	Fishing	Escondido	United States	Tiffany	5503	Consumer	CA	92025.0	LATAM	5503	04-01-2017	56272	Caribbean	339.980011	equals 5
90250	Camping & Hiking	Camping & Hiking	Caguas	Puerto Rico	Mary	5498	Corporate	PR	725.0	LATAM	5498	26-02-2017	53979	South America	239.979996	1

It changed the “5” to “equals 5” in Order Quantity column

Conclusion

Python pandas provides several functions and techniques for selecting and filtering data within a DataFrame.

By mastering these techniques, you’ll be well-equipped to explore and analyze your data effectively.

Remember, practice makes perfect.
Thank you for reading!

Python Fundamentals

Thank you for your time and interest! 🚀
You can find even more content at ***Python Fundamentals*** 🪐

Python | Data Science | Data Analysis | Data Analytics | Data Scientist

Published in Towards Dev

6.3K Followers · Last published 17 hours ago

Follow

A publication for sharing projects, ideas, codes, and new theories.

Written by Python Fundamentals

2.5K Followers · 1.1K Following

Following

Learn Python with new contents every day

No responses yet

What are your thoughts?

Respond

More from Python Fundamentals and Towards Dev

In Python in Plain English by Python Fundamentals

Selecting and Filtering Data in Python Pandas: loc and iloc

Python pandas library provides several methods for selecting and filtering data, suc...

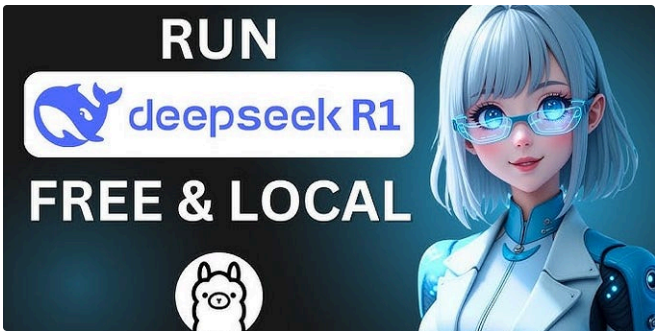
Jan 27 12

In Towards Dev by AshokReddy

Swagger UI is Gone in .NET 9: Here's What You Need to Do Next

Introduction:

Jan 27 66 2

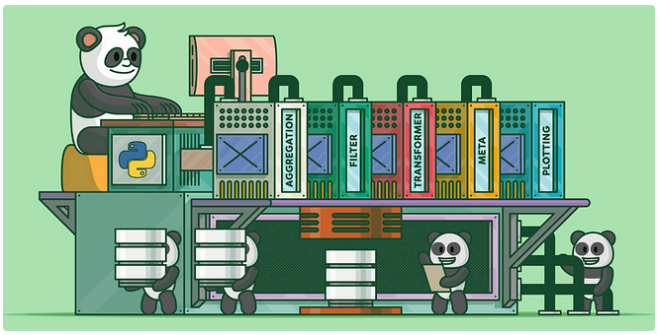


In Towards Dev by Bill WANG

Running DeepSeek R1 Locally on Mac in Two Simple Commands

Follow up on my DeepSeek blogs about

Jan 26 79 4



Python Fundamentals

10 Useful Python Pandas Functions for Data Exploration

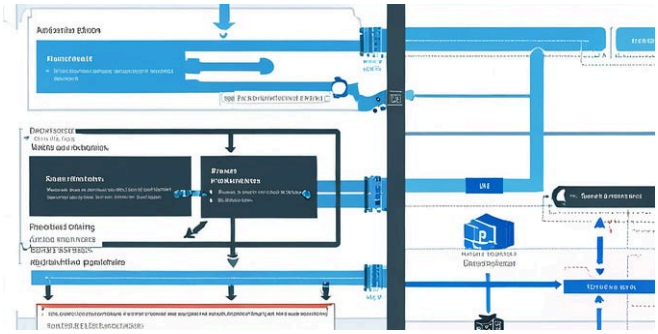
Data exploration and data understanding are crucial steps to take before diving into data...

May 4, 2024 20

See all from Python Fundamentals

See all from Towards Dev

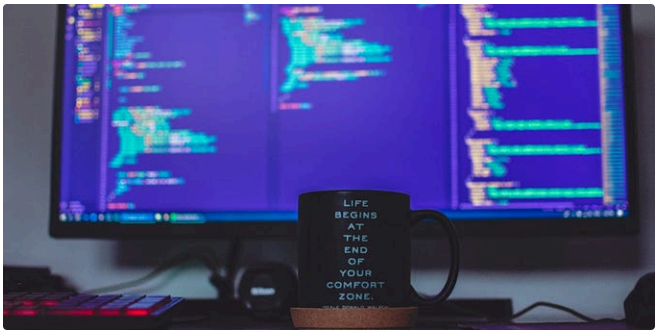
Recommended from Medium



J.

Building a Data Pipeline with Python: A Step-by-Step Guide to...

Nov 4, 2024 21



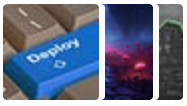
Aashish Kumar

25 Amazing Python Tricks That Will Instantly Improve Your Code

Photo by Tudor Baciu on Unsplash

Feb 17 1

Lists



Predictive Modeling w/ Python

20 stories · 1838 saves

Coding & Development

11 stories · 1016 saves

Practical Guides to Machine Learning

10 stories · 2214 saves

ChatGPT prompts

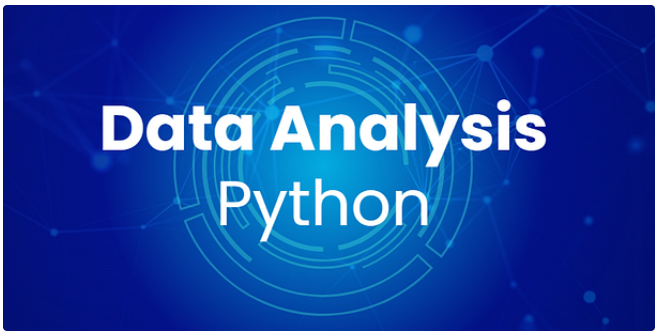
51 stories · 2599 saves

PURRFECT SOFTWARE LIMITED

10 Python Libraries for Web Scraping That Changed the Game...

Join Me as I Share the Essential Python Libraries That Made Web Scraping a Breeze

Feb 16 10

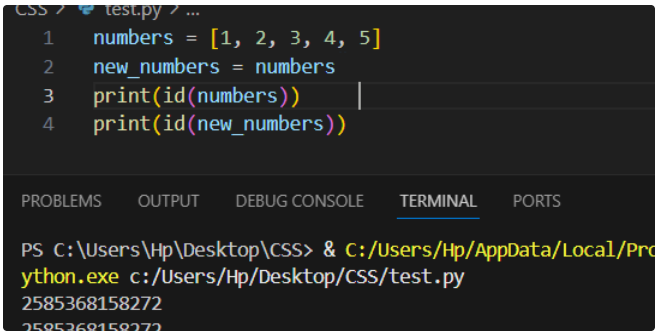


In Stackademic by Khouloud Haddad Amamou

Data Analysis with Python Pandas and Matplotlib

1. Introduction

Jan 30 150 8



In Python in Plain English by Kiran Maan

Don't Use This Way to Copy the Objects in Python

You won't regret learning this!!!

Feb 17 187

Ime Eti-mfon

Data Wrangling with Python

Merging, Joining, and Concatenating Data

Nov 7, 2024 10 1

See more recommendations