

Indexing/Manipulating/Assigning/





Indexing



In Pandas, indexing retrieves data using labels or positions with .loc[] for labels and .iloc[] for integer positions.

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'City': ['New York', 'Los Angeles', 'Chicago']
}

df = pd.DataFrame(data)
```

Name Age City
O Alice 25 New York
Bob 30 Los Angeles
Charlie 35 Chicago

```
df.loc[1, 'City']

df.iloc[1, 2]
```

Label-based Indexing.loc[]



```
Name
                       City
           Age
    Alice
            25
                   New York
0
            30 Los Angeles
      Bob
  Charlie
            35
                    Chicago
```

print(df.loc[1, 'City'])

Los Angeles

Integer-based Indexing.iloc[]



	Name	Age	City	
0	Alice	25	New York	
1	Bob	30	Los Angeles	
2	Charlie	35	Chicago	

print(df.iloc[1])

Name Bob

Age 30

City Los Angeles

Name: 1, dtype: object

print(df.iloc[1, 2])

Los Angeles

Manipulating



Manipulate Pandas DataFrames using methods like .drop(), .rename(), .sort_values(), .apply(), and .groupby() for data transformation.

```
df_dropped = df.drop('City', axis=1)

df_renamed = df.rename(columns={'Name': 'Full Name'})

df_sorted = df.sort_values(by='Age')
```

```
Name Age City
O Alice 25 New York
Bob 30 Los Angeles
Charlie 35 Chicago
```

```
df_dropped = df.drop('City', axis=1)
print(df_dropped)
```

```
Name Age

0 Alice 25

1 Bob 30

2 Charlie 35
```



```
df_renamed = df.rename(columns={'Name': 'Full Name'})
print(df_renamed)
```

```
Full Name Age City

0 Alice 25 New York

1 Bob 30 Los Angeles

2 Charlie 35 Chicago
```

```
df_sorted = df.sort_values(by='Age')
print(df_sorted)
```

City	Name Age		
New York	25	Alice	0
Los Angeles	30	Bob	1
Chicago	35	Charlie	2

df_grouped = df.groupby('City').mean()
print(df_grouped)

pandas

df['Age in Months'] = df['Age'].apply(lambda x: x * 12)
print(df)

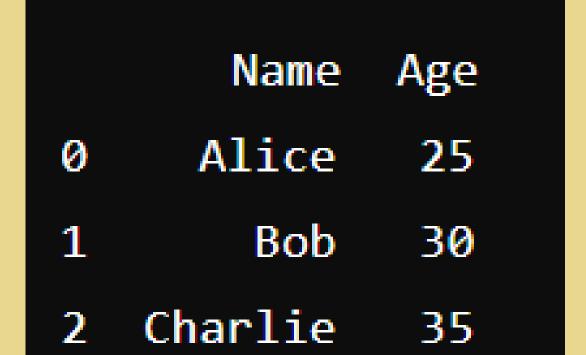
Name Age			City Age in Months	
0	Alice	25	New York	300
1	Bob	30	Los Angeles	360
2	Charlie	35	Chicago	420

City
Chicago 35.0
Los Angeles 30.0
New York 25.0



Assigning

Assign values in Pandas DataFrames using column assignment, like df['column'] = value, to update or create columns.



```
| pandas
```

```
df['Age'] = [26, 31, 36]
print(df)
```

```
Name Age City

0 Alice 26 New York

1 Bob 31 Los Angeles

2 Charlie 36 Chicago
```

```
df['City'] = ['New York', 'Los Angeles', 'Chicago']
print(df)
```

```
Name Age City
O Alice 25 New York
Bob 30 Los Angeles
Charlie 35 Chicago
```

```
df['Senior'] = df['Age'] > 30
print(df)
```

```
Name Age City Senior
Alice 26 New York False
Bob 31 Los Angeles True
Charlie 36 Chicago True
```





Thankyou