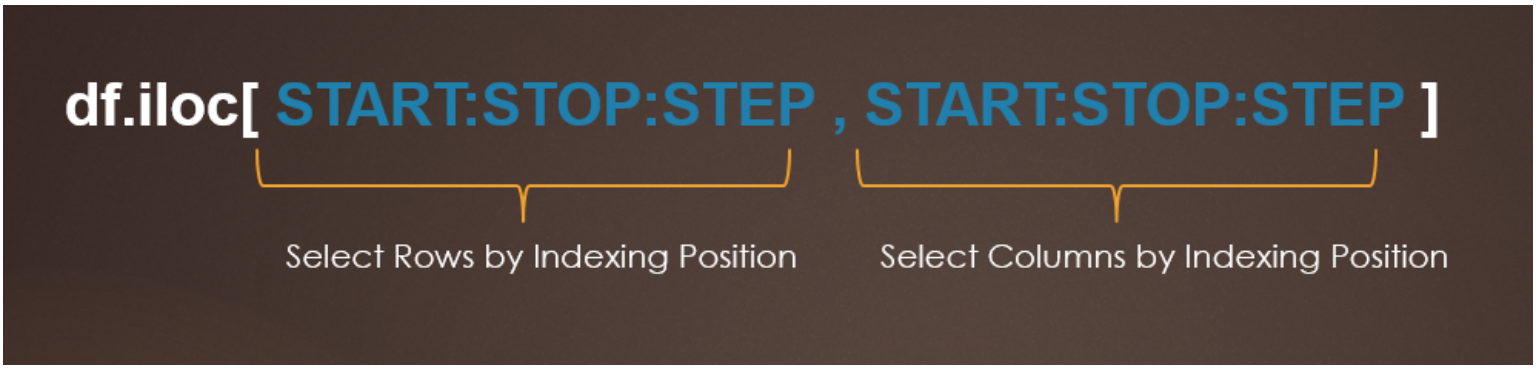
DataFrame.iloc[]

- **It is an index-based method to select rows and/or columns in pandas.**
- **iloc[] operator can accept single index, multiple indexes from the list, indexes by a range, and many more.**

Syntax:

> **Dataframe.iloc[row_segment , column_segment]**



- START is the integer index of the row/column.
- STOP is the integer index of the last row/column where you wanted to stop the selection, and
- STEP as the number of indices to advance after each extraction.

Key points:

By not providing a start index, iloc[] selects from the first row/column.

By not providing stop, iloc[] selects all rows/columns from the start index.

Providing both start and stop, selects all rows/columns in between.

Note:

The column_segment argument is optional. Therefore, if column_segment is not provided, iloc[ ] will select the subset of Dataframe based on row_segment argument only.

Let's create a DataFrame and run some examples of pandas iloc.

```
import pandas as pd
technologies = {
    'Courses':["Spark","PySpark","Hadoop","Python","pandas"],
    'Fee' :[20000,25000,26000,22000,24000],
    'Duration':['30day','40days','35days','40days','60days'],
    'Discount':[1000,2300,1200,2500,2000]
              }
index_labels=['r1','r2','r3','r4','r5']
df = pd.DataFrame(technologies,index=index_labels)
print(df)
```

```
       Courses    Fee Duration  Discount
r1       Spark  20000    30day      1000
r2     PySpark  25000   40days      2300
r3      Hadoop  26000   35days      1200
r4      Python  22000   40days      2500
r5      pandas  24000   60days      2000
```

Select Single Row & Column By Index

```
# Select Single Row by Index
print(df.iloc[1])
```

```
Courses     PySpark
Fee           25000
Duration     40days
Discount       2300
Name: r2, dtype: object
```

select column by Index

```
# Select Single Column by Index
print(df.iloc[:, 0])
```

```
r1      Spark
r2    PySpark
r3     Hadoop
r4     Python
r5     pandas
Name: Courses, dtype: object
```

Select Multiple Rows by Index

```
# Select Multiple Rows by Index
print(df.iloc[[1,2]])
```

```
     Courses    Fee Duration  Discount
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
```

select multiple columns from pandas DataFrame.

```
# Select Multiple Columns by Index
print(df.iloc[:, [0,1,3]])
```

```
     Courses    Fee  Discount
r1    Spark  20000      1000
r2  PySpark  25000      2300
r3   Hadoop  26000      1200
r4   Python  22000      2500
r5   pandas  24000      2000
```

Select Rows by Index Range

```
# Select Rows Between two Indexs
# Includes Index 0 & Execludes 4
print(df.iloc[0:4])
```

```
     Courses    Fee Duration  Discount
r1    Spark  20000    30day      1000
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
r4   Python  22000   40days      2500
```

Select Columns by Index Range

```
# Select Columns between two Indexes
# Includes Index 1 & Execludes 4
print(df.iloc[:,1:4])
```

```
      Fee Duration  Discount
r1  20000    30day      1000
r2  25000   40days      2300
r3  26000   35days      1200
r4  22000   40days      2500
r5  24000   60days      2000
```

Select Alternate Rows or Columns

```
# Select Alternate rows By Index
print(df.iloc[0:4:2])
```

```
    Courses    Fee Duration  Discount
r1   Spark  20000    30day      1000
r3  Hadoop  26000   35days      1200
```

```
# Select Alternate Columns between two Indexes
print(df.iloc[:,1:4:2])
```

```
      Fee  Discount
r1  20000      1000
r2  25000      2300
r3  26000      1200
r4  22000      2500
r5  24000      2000
```

Using Conditions with iloc[]

```
# By Condition
print(df.iloc[list(df['Fee'] >= 24000)])
```

```
     Courses    Fee Duration  Discount
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
r5   pandas  24000   60days      2000
```

Pandas iloc[] Complete Example

```
import pandas as pd
technologies = {
    'Courses':["Spark","PySpark","Hadoop","Python","pandas"],
    'Fee' :[20000,25000,26000,22000,24000],
    'Duration':['30day','40days','35days','40days','60days'],
    'Discount':[1000,2300,1200,2500,2000]
              }
index_labels=['r1','r2','r3','r4','r5']
df = pd.DataFrame(technologies,index=index_labels)
print(df)

# Select Single Row by Index
print(df.iloc[1])

# Select Single Column by Index
print(df.iloc[:, 0])

# Select Multiple Rows by Index
print(df.iloc[[1,2]])

# Select Multiple Columns by Index
print(df.iloc[:, [0,1,3]])

# Includes Index 0 & Execludes 4
print(df.iloc[0:4])
```

```
# Includes Index 1 & Execludes 4
print(df.iloc[:,1:4])

# Select Alternate rows By Index
print(df.iloc[0:4:2])

# Select Alternate Columns between two Indexes
print(df.iloc[:,1:4:2])

print(df.iloc[list(df['Fee'] >= 24000)])
```

```
        Courses    Fee Duration  Discount
r1    Spark  20000    30day      1000
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
r4   Python  22000   40days      2500
r5   pandas  24000   60days      2000
Courses     PySpark
Fee           25000
Duration     40days
Discount       2300
Name: r2, dtype: object
r1      Spark
r2    PySpark
r3     Hadoop
r4     Python
r5     pandas
Name: Courses, dtype: object
    Courses    Fee Duration  Discount
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
    Courses    Fee  Discount
r1    Spark  20000      1000
r2  PySpark  25000      2300
r3   Hadoop  26000      1200
r4   Python  22000      2500
r5   pandas  24000      2000
    Courses    Fee Duration  Discount
r1    Spark  20000    30day      1000
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
r4   Python  22000   40days      2500
      Fee Duration  Discount
r1  20000    30day      1000
r2  25000   40days      2300
r3  26000   35days      1200
r4  22000   40days      2500
r5  24000   60days      2000
   Courses    Fee Duration  Discount
r1   Spark  20000    30day      1000
r3  Hadoop  26000   35days      1200
      Fee  Discount
r1  20000      1000
r2  25000      2300
r3  26000      1200
r4  22000      2500
r5  24000      2000
    Courses    Fee Duration  Discount
r2  PySpark  25000   40days      2300
r3   Hadoop  26000   35days      1200
r5   pandas  24000   60days      2000
```

Let’s create another DataFrame and explore how to use pandas iloc[].

```
import pandas as pd
# List of Tuples
students = [('jack',  34, 'Sydeny',    'Australia'),
            ('Riti',  30, 'Delhi',     'India'),
            ('Vikas', 31, 'Mumbai',    'India'),
            ('Neelu', 32, 'Bangalore', 'India'),
```

```
           ('John',  16, 'New York',   'US'),
           ('Mike',  17, 'las vegas',  'US')]
# Create a DataFrame from list of tuples
df = pd.DataFrame( students,
                   columns=['Name', 'Age', 'City', 'Country'],
                   index=['a', 'b', 'c', 'd', 'e', 'f'])
print(df)
```

```
        Name  Age       City    Country
    a   jack   34     Sydeny  Australia
    b   Riti   30      Delhi      India
    c  Vikas   31     Mumbai      India
    d  Neelu   32  Bangalore      India
    e   John   16   New York         US
    f   Mike   17  las vegas         US
```

```
# Select row at index position 2 i.e. the 3rd row of Dataframe
row = df.iloc[2]
print(row)
```

```
    Name        Vikas
    Age            31
    City       Mumbai
    Country     India
    Name: c, dtype: object
```

```
# Select rows of Dataframe based on row indices in list
subsetDf = df.iloc[ [2,4,1] ]
print(subsetDf)
```

```
        Name  Age      City Country
    c  Vikas   31    Mumbai   India
    e   John   16  New York      US
    b   Riti   30     Delhi   India
```

```
# Select rows of Dataframe based on row index range
subsetDf = df.iloc[ 1:4 ]
print(subsetDf)
```

```
        Name  Age       City Country
    b   Riti   30      Delhi   India
    c  Vikas   31     Mumbai   India
    d  Neelu   32  Bangalore   India
```

```
# Select rows of Dataframe based on bool array
subsetDf = df.iloc[ [True, False, True, False, True, False] ]
print(subsetDf)
```

```
        Name  Age      City    Country
    a   jack   34    Sydeny  Australia
    c  Vikas   31    Mumbai      India
    e   John   16  New York         US
```

```
# Select single column by index position
column = df.iloc[:, 2]
print(column)
```

```
    a       Sydeny
    b        Delhi
    c       Mumbai
    d    Bangalore
    e     New York
    f    las vegas
    Name: City, dtype: object
```

```
# Select multiple columns by indices
subsetDf = df.iloc[:, [2, 3, 1]]
```

```
print(subsetDf)
```

```
          City    Country  Age
a     Sydeny  Australia   34
b      Delhi      India   30
c     Mumbai      India   31
d  Bangalore      India   32
e   New York         US   16
f  las vegas         US   17
```

```
# Select multiple columns by index range
subsetDf = df.iloc[:, 1 : 4]
print(subsetDf)
```

```
        Age       City    Country
a   34     Sydeny  Australia
b   30      Delhi      India
c   31     Mumbai      India
d   32  Bangalore      India
e   16   New York         US
f   17  las vegas         US
```

```
# Select columns of Dataframe based on bool array
subsetDf = df.iloc[ : , [True, True, False, False] ]
print(subsetDf)
```

```
         Name  Age
a   jack   34
b   Riti   30
c  Vikas   31
d  Neelu   32
e   John   16
f   Mike   17
```

```
# Select a Cell value from Dataframe
cellValue = df.iloc[3,2]
print(cellValue)
```

```
     Bangalore
```

```
# Select sub set of Dataframe based on row/column indices in list
subsetDf = df.iloc[[1,3],[2,1]]
print(subsetDf)
```

```
          City  Age
b      Delhi   30
d  Bangalore   32
```

```
# Select subset of Dataframe based on row and column index range.
subsetDf = df.iloc[1:4, 1:4]
print(subsetDf)
```

```
        Age       City Country
b   30      Delhi   India
c   31     Mumbai   India
d   32  Bangalore   India
```

```
# change the value of 3rd row of Dataframe
df.iloc[2] = 0
print(df)
```

```
         Name  Age       City    Country
a   jack   34     Sydeny  Australia
b   Riti   30      Delhi      India
c      0    0          0          0
d  Neelu   32  Bangalore      India
e   John   16   New York         US
f   Mike   17  las vegas         US
```