

```
In [4]: 1 # Function to read CSV data into a DataFrame and
2 # returns the DataFrame object
3
4 filepath = 'DataFiles/Income.csv'
5 import pandas as pd
6 def readCSVdata(filepath):
7     return pd.read_csv(filepath)
8
9 readCSVdata(filepath)
```

Out[4]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	47215	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

```
In [5]: 1 incomedf = readCSVdata(filepath)
2
3 # Function to print all column names in a single line
4 # GEOID State 2005 2006 2007 2008 2009 2010 2011 2012 2013
5
6 def printDataFrameColumns(df):
7     columns = df.columns
8     for column in columns:
9         print(column, end=' ')
10    return
11
12 printDataFrameColumns(incomedf)
```

GEOID State 2005 2006 2007 2008 2009 2010 2011 2012 2013

```
In [6]: 1 # Function to access a row based on a unique column value
2
3 def accessDataFrameRow(df, key):
4
5     for row in df.values:
6         if key in row:
7             for item in row:
8                 print(item, end=' ')
9                 print('\n')
10    return
11
12 accessDataFrameRow(incomedf, '04000US06')
```

04000US06 California 51755 55319 55734 57014 56134 54283 53367 57020 57528

In [7]: 1 incomedef

Out[7]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	47215	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

```

In [8]: 1 # Accessing a unique value based on row, column info
2 # Income of a state in a given year
3 def getRowIndex(df, rowkey):
4     for i in range(len(df.values)):
5         if df.values[i][0] == rowkey or df.values[i][1] == rowkey:
6             rowindex = i
7     return rowindex
8
9 def getColumnIndex(df, columnkey):
10     for i in range(len(df.columns)):
11         if df.columns[i] == columnkey:
12             columnindex = i
13     return columnindex
14
15 def valueFromRowColumn(df, rowkey, columnkey):
16     rowindex = getRowIndex(df, rowkey)
17     columnindex = getColumnIndex(df, columnkey)
18     return df.values[rowindex][columnindex]
19
20 valueFromRowColumn(incomedef, 'California', '2009')

```

Out[8]: 56134

```

In [9]: 1 # Function to update data based on rowkey and columnkey
        2
        3 def updateDataCellFromRowColumn(df, rowkey, columnkey, newdata):
        4     rowindex = getRowIndex(df, rowkey)
        5     columnindex = getColumnIndex(df, columnkey)
        6     row = df.values[rowindex]
        7     row[columnindex] = newdata
        8     df.loc[rowindex] = row
        9     return
       10
       11 updateDataCellFromRowColumn(incomedf, 'Arizona', '2007', 62993)
       12 incomedf

```

Out[9]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	62993	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

```

In [10]: 1 accessDataFrameRow(incomedf, 62993)

04000US02 Alaska 55891 56418 62993 63989 61604 57848 57431 63648 61137

04000US04 Arizona 45245 46657 62993 46914 45739 46896 48621 47044 50602

```

```

In [12]: 1 # Function to write DataFrame to CSV
        2
        3 incomedf.to_csv(filepath, index=False)

```

```
In [13]: 1 # Function to add a new row of data to DataFrame
2
3 def addRowDataDataFrame(df, rowdata):
4     lastrowindex = len(df.values)-1
5     df.loc[lastrowindex+1] = rowdata
6     return
7
8 rowdata = [1,2,3,4,5,56,13,14,313,12,0]
9 addRowDataDataFrame(incomedf, rowdata)
10 incomedf
```

Out[13]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	62993	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528
5	1	2	3	4	5	56	13	14	313	12	0

```
In [14]: 1 # Function to delete a row in a DataFrame
2
3 def deleteRowDataDataFrame(df, rowkey):
4     rowindex = getRowIndex(df, rowkey)
5     return df.drop(rowindex)
6
7 incomedf = deleteRowDataDataFrame(incomedf, 1)
8 incomedf
```

Out[14]:

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	62993	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

In []:

```
1
```