

In [271... `import pandas as pd`

Inspect the code under this section, then add a cell to:

1. Create a new dictionary, name it your firstname where `firstname_fruits` is your first name.
2. Add four items to the dictionary with names of your favorite fruits as keys and the respective color as values.
3. Convert the dictionary into a pandas series named `firstname_f`.
4. Print out the second and third items.
5. Create a sub series named `firstname_f2` containing the second and third items.
6. Printout from the sub series the last item using `iloc`.

```
In [272... #Create a new dictionary called renee_fruits.
renee_fruits = {}
#Add four items to the dictionary with names of your favorite fruits as keys and th
renee_fruits['apples'] = 'red'
renee_fruits['bananas'] = 'yellow'
renee_fruits['oranges'] = 'orange'
renee_fruits['pears'] = 'green'
print(renee_fruits)
#Convert the dictionary into a pandas series named firstname_f.
renee_f = pd.Series(renee_fruits)
renee_f
# Print out the second and third items.
print(renee_f[1:3])
# Create a sub series named firstname_f2 containing the second and third items.
renee_f2 = renee_f[1:3]
renee_f2
# Printout from the sub series the last item using iloc.
print(renee_f2.iloc[1])

{'apples': 'red', 'bananas': 'yellow', 'oranges': 'orange', 'pears': 'green'}
bananas    yellow
oranges    orange
dtype: object
orange
```

Add a cell with the following logic:

1. Create a list containing four rainfall amounts of values 10, 23,24,30 name the list `firstname_amounts`.
2. Using pandas create a `date_range` for todays date/time (you can set any time) with four time intervals.
3. Create a series that combines both the list and date range name it `firstname_rainfall amounts_today`.
4. Plot as bar chart.

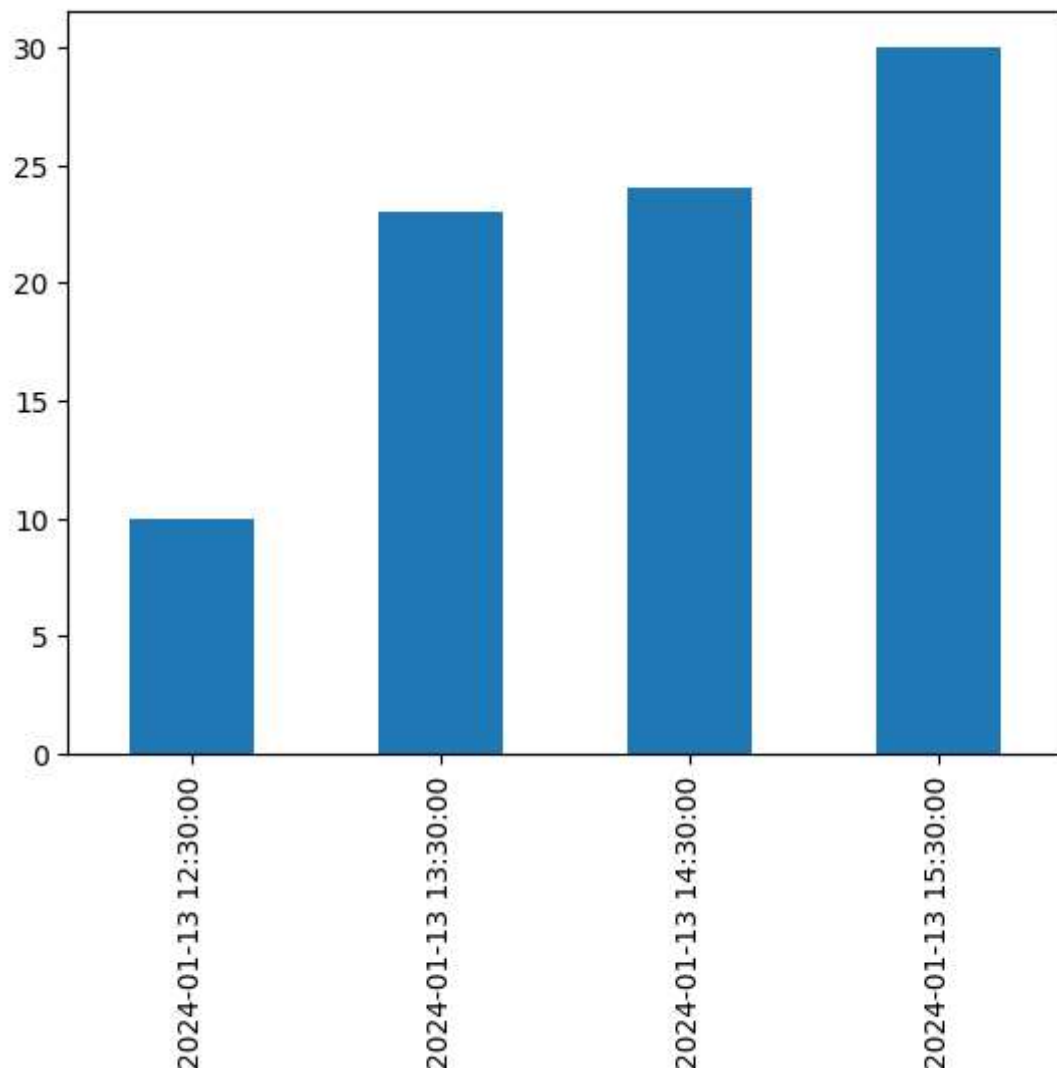
```
In [273... # Create a list containing four rainfall amounts of values 10, 23,24,30 name the
renee_amounts = [10, 23, 24, 30]
# Using pandas create a date_range for todays date/time (you can set any time) with
renee_dates = pd.date_range('2024-01-13 12:30pm', periods=4, freq='H')
renee_dates
# Create a series that combines both the list and date range name it firstname_rain
```

```

renee_rainfall = pd.Series(renee_amounts, renee_dates)
renee_rainfall
# Plot as bar chart.
renee_rainfall.plot(kind='bar')

```

Out[273]: <Axes: >



Make a copy of the dataframe d5 and name it fristname_d5, carryout the following:

1. print out a dataframe containing all “private” columns
2. Swap the columns and rows (hint: look at transpose)

```

In [274... # Make a copy of the dataframe d5 and name it fristname_d5, carryout the following:
# 1.    print out a dataframe containing all “private” columns
# 2.    Swap the columns and rows (hint: Look at transpose)

import numpy as np
renee_d5 = pd.DataFrame(
    {
        ("public", "birthyear"):
            {("Paris","alice"):1985, ("Paris","bob"): 1984, ("London","charles"): 1992},
        ("public", "hobby"):
            {("Paris","alice"): "Biking", ("Paris","bob"): "Dancing"},
        ("private", "weight"):
            {("Paris","alice"):68, ("Paris","bob"): 83, ("London","charles"): 112},
        ("private", "children"):
            {("Paris", "alice"):np.nan, ("Paris","bob"): 3, ("London","charles"): 0}
    }
)

```

```

renee_d5["private"]

# Swap the columns and rows (hint: Look at transpose)
renee_d5.T

```

Out[274]:

		Paris	London	
		alice	bob	charles
public	birthyear	1985	1984	1992
	hobby	Biking	Dancing	NaN
private	weight	68	83	112
	children	NaN	3.0	0.0

Use the query() to query the people dataframe you created earlier and retrieve everything related to alice.

In [275...

```

# Use the query() to query the people dataframe you created earlier and retrieve e
alice_data = renee_d5.loc[renee_d5.index.get_level_values(1) == 'alice']
alice_data

```

Out[275]:

		public		private	
		birthyear	hobby	weight	children
Paris	alice	1985	Biking	68	NaN

Add a cell to create a dataframe containing grade for four students choose the name of the students and use the names as index. For columns create four columns to reflect the months April, May, June, July. Set grade items for each student for each month to be between 0 and 100. Name the dataframe fristname_grades. Carry out the following using pandas operations:

1. Print out the average for the month of April
2. Adjust all the grades by 2% (i.e. increase)
3. Printout the grades for the month of may that are higher than 50%
4. Group the failing students i.e. the students with average over four month below 50%

In [276...

```

# Add a cell to create a dataframe containing grade for four students choose the no
grades = {
    'April': [95, 92, 76, 99],
    'May': [99, 99, 83, 88],
    'June': [100, 95, 90, 82],
    'July': [100, 100, 90, 82]
}
students = ['Alice', 'Bob', 'Charlie', 'Duan']
renee_grades = pd.DataFrame(grades, students)
# Print out the average for the month of April
renee_grades["April"].mean()
# Adjust all the grades by 2% (i.e. increase)
renee_grades = renee_grades + 2
# printout the grades for the month of may that are higher than 50%
renee_grades["May"][renee_grades["May"] > 50]
(renee_grades > 50).all(axis=1)
#Group the failing students i.e. the students with average over four month below 50

```

```
renee_grades.mean(axis=1)[renee_grades.mean(axis=1) < 50]  
(renee_grades < 50).all(axis=1)
```

```
Out[276]: Alice      True  
          Bob       True  
          Charlie   False  
          Duan      False  
          dtype: bool
```