

# دوره دیتا ساینس کاربردی

## پایتون و کتابخانه پنداس



—● dataroadmap ●—

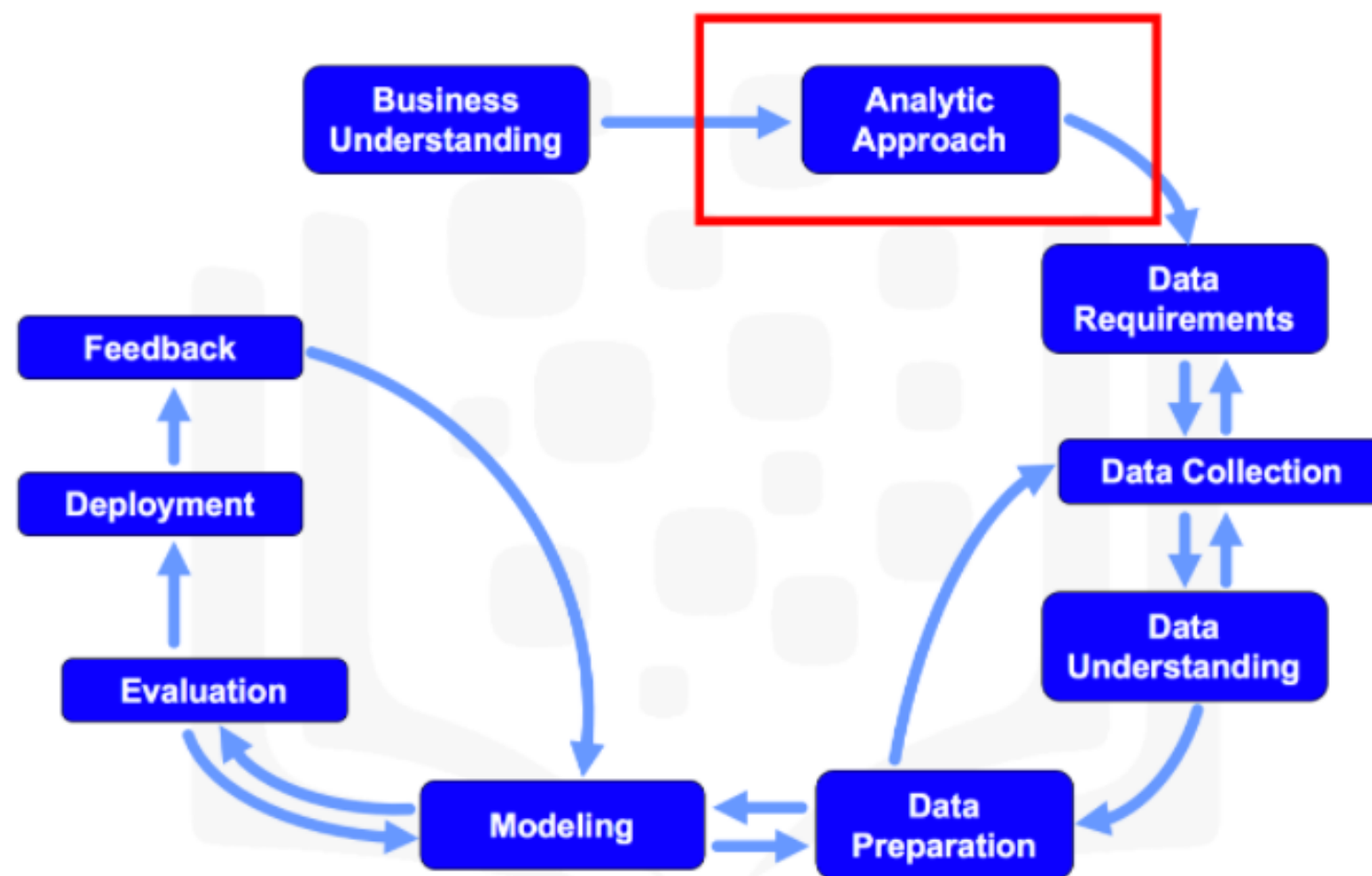
مدرس: مونا حاتمی



جلسه سوم

# مرور مساله

- پیش بینی اینکه آیا اسپیس ایکس مجدد از فاز اول استفاده می‌کند یا نه؟



**Descriptive**  
Explains what happened.

**Diagnostic**  
Explains why it happened.

**Predictive**  
Forecasts what might happen.

**Prescriptive**  
Recommends an action based on the forecast.

# Data Type in Python

- Numbers
- Strings
- Lists
- Dictionary
- Set
- Tuple
- Boolean

# Dictionary

دیکشنری

key



value



```
▶ d = {'key1': 'item1', 'key2': 'item2'}
```

```
▶ d
```

```
2]: {'key1': 'item1', 'key2': 'item2'}
```

```
▶ d['key1']
```

```
3]: 'item1'
```

# Python

## List vs Set

## پایتون

مقایسه لیست و ست

```
In [1]: my_list = [1,2,3]
```

```
In [2]: my_list.append(4)
```

```
In [3]: my_list
```

```
Out[3]: [1, 2, 3, 4]
```

```
In [4]: my_list.append(4)  
my_list
```

```
Out[4]: [1, 2, 3, 4, 4]
```

```
In [5]: my_set = {1,2,3}
```

```
In [6]: my_set.add(4)
```

```
In [7]: my_set
```

```
Out[7]: {1, 2, 3, 4}
```

```
In [8]: my_set.add(4)  
my_set
```

```
Out[8]: {1, 2, 3, 4}
```

# Python

## List vs Set

## پایتون

مقایسه لیست و ست

```
In [9]: new_list=[2,2,4,5,3,3,3]
```

```
In [10]: new_set=set(new_list)  
new_set
```

```
Out[10]: {2, 3, 4, 5}
```

```
In [11]: new_list[2]
```

```
Out[11]: 4
```

```
In [12]: new_set[2]
```

```
-----  
TypeError                                Traceback (most recent call  
~\AppData\Local\Temp\ipykernel_62416\1309137323.py in <module>  
----> 1 new_set[2]
```

```
TypeError: 'set' object is not subscriptable
```

# Python

## List vs Tuple

## پایتون

مقایسه لیست و تاپل

```
In [13]: my_list = [1,2,3]
```

```
In [14]: my_list.append('4')  
my_list
```

```
Out[14]: [1, 2, 3, '4']
```

```
In [15]: my_tuple = (1,2,3)
```

```
In [16]: my_tuple.append('4')
```

```
-----  
AttributeError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_62416\1812677532.py in <module>  
----> 1 my_tuple.append('4')
```

```
AttributeError: 'tuple' object has no attribute 'append'
```

The key difference between tuples and lists is that while tuples are immutable objects, lists are mutable.

```
my_tuple.
```

count

index

# Python Boolean

```
# Value false
```

```
False
```

```
: False
```

```
# Type of True
```

```
type(True)
```

```
: bool
```

```
# Type of False
```

```
type(False)
```



# Python

# Help

## پایتون

## کمک

1

```
In [30]: my_tuple.count?
```

```
In [ ]:
```

**Signature:** `my_tuple.count(value, /)`

**Docstring:** Return number of occurrences of value.

**Type:** `builtin_function_or_method`

2

```
In [31]: help(my_tuple.count)
```

Help on built-in function count:

`count(value, /)` method of `builtins.tuple` instance  
Return number of occurrences of value.

# Python

# Help

پایتون

کمک

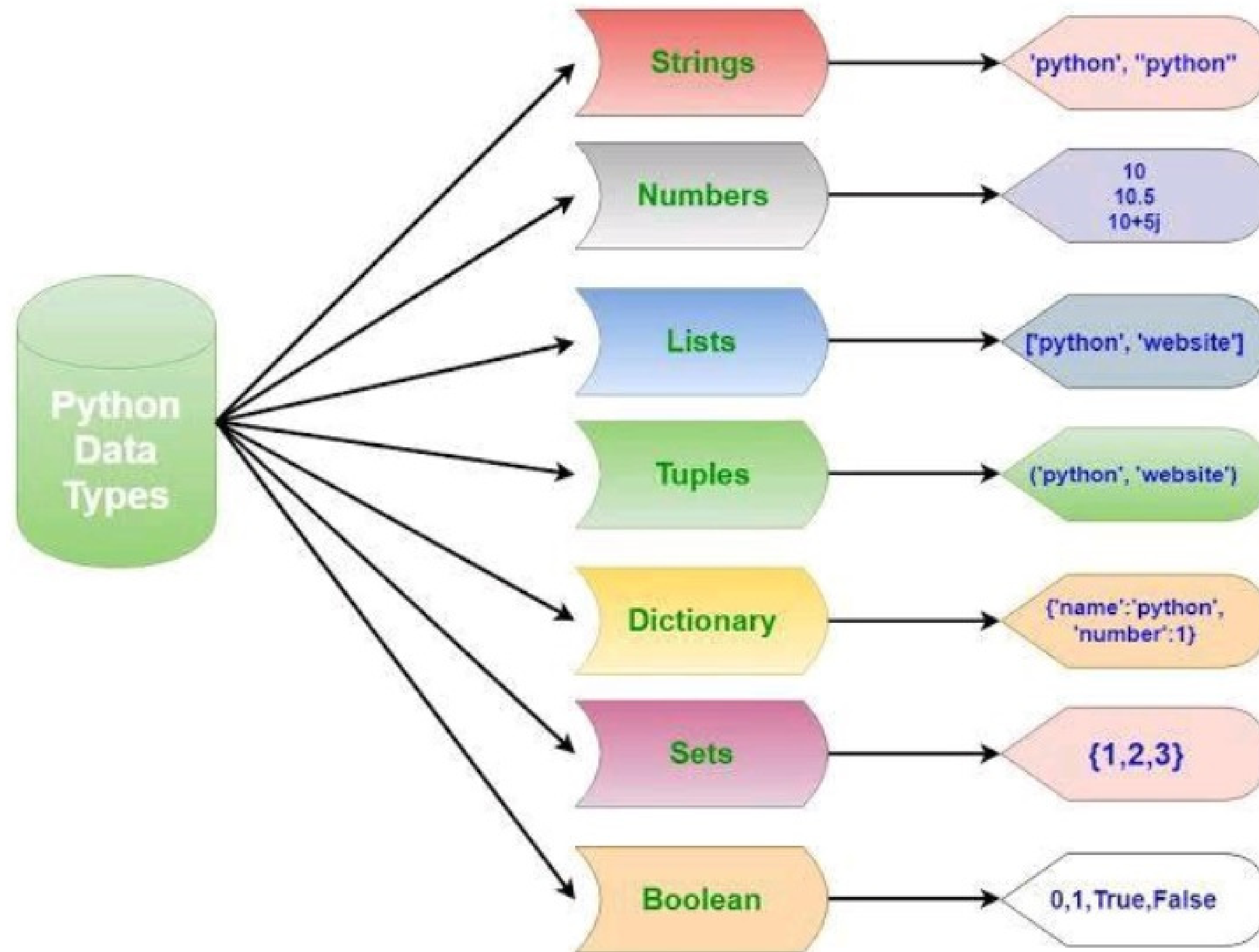
3

```
In [ ]: my_tuple.count()
```

**Signature:** my\_tuple.count(value, /)

**Docstring:** Return number of occurrences of value.

**Type:** builtin\_function\_or\_method



Python Data Types

# Python Libraries

## کتابخانه ها

کتابخانه ها مجموعه ای از توابع آماده جهت  
سهولت کار در زبانهای برنامه نویسی هستند.

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Scikit-learn

# Install Pandas

In [1]: `pip list`

Package	Version
absl-py	0.10.0
alabaster	0.7.12
anaconda-client	1.7.2
anaconda-navigator	1.9.12
anaconda-project	0.8.3
appdirs	1.4.4
argh	0.26.2
argon2-cffi	21.1.0
asn1crypto	1.3.0
astroid	2.4.2
astropy	4.0.1.post1
astunparse	1.6.3
atomicwrites	1.4.0
attrs	21.2.0
autopep8	1.5.3
Babel	2.8.0
backcall	0.2.0

In [2]: `!pip install pandas`

Defaulting to user installation because normal site-packages is not writeable

# Read file in Pandas

```
import pandas as pd
```

```
df=pd.read_csv('d')
```

dataset\_falcon9.csv

%ddir

%debug

%%debug

%dhist

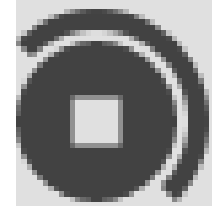
%dirs

%doctest\_mode

```
import os  
os.getcwd()
```

```
: 'C:\\dataroadmap\\Monogram2\\week-3'
```

# Read file in Google Colab



```
from google.colab import files  
uploaded = files.upload()
```

...

Choose Files

No file chosen

Cancel upload

# Read csv file in Pandas

```
df=pd.read_csv('dataset_falcon9.csv')
```

```
df
```

4]:

ass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs		LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
412	LEO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0	0	B0003	-80.577366	28.561857	0
000	LEO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0	0	B0005	-80.577366	28.561857	0
000	ISS	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0	0	B0007	-80.577366	28.561857	0
000	PO	VAFB SLC 4E	False Ocean	1	False	False	False		NaN	1.0	0	B1003	-120.610829	34.632093	0
000	GTO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0	0	B1004	-80.577366	28.561857	0
...	...	...	...	...	...	...	...		...	...	...	...	...	...	...
000	VLEO	KSC LC 39A	True Success	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	5.0	2	R1060	-80.603956	28.608058	1



# Info Method in DataFrame

```
: ▶ df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 90 entries, 0 to 89
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	FlightNumber	90 non-null	int64
1	Date	90 non-null	object
2	BoosterVersion	90 non-null	object
3	PayloadMass	90 non-null	float64
4	Orbit	90 non-null	object
5	LaunchSite	90 non-null	object
6	Outcome	90 non-null	object
7	Flights	90 non-null	int64
8	GridFins	90 non-null	bool
9	Reused	90 non-null	bool
10	Legs	90 non-null	bool
11	LandingPad	64 non-null	object
12	Block	90 non-null	float64
13	ReusedCount	90 non-null	int64
14	Serial	90 non-null	object
15	Longitude	90 non-null	float64
16	Latitude	90 non-null	float64
17	Class	90 non-null	int64

```
dtypes: bool(3), float64(4), int64(4), object(7)
```

```
memory usage: 10.9+ KB
```

# Head Method in DataFrame

```
df.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0

```
df.head(2)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0

# Tail Method in DataFrame

```
df[5:8]
```

```
]:
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
5	6	2014-01-06	Falcon 9	3325.0	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0
6	7	2014-04-18	Falcon 9	2296.0	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0
7	8	2014-07-14	Falcon 9	1316.0	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0	0

```
df.tail(2)
```

```
:
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount
88	89	2020-10-24	Falcon 9	15400.0	VLEO	CCAFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	0
89	90	2020-11-05	Falcon 9	3681.0	MEO	CCAFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	0

# Read Column in DataFrame

```
In[3]: df['FlightNumber']
```

```
Out[43]: 0      1  
         1      2  
         2      3  
         3      4  
         4      5  
         ..  
        85     86  
        86     87  
        87     88  
        88     89  
        89     90  
         Name: FlightNumber, Length: 90, dtype: int64
```

```
In[4]: df['FlightNumber'].head()
```

```
Out[44]: 0      1  
         1      2  
         2      3  
         3      4  
         4      5  
         Name: FlightNumber, dtype: int64
```

# Read Column in DataFrame

```
df[['FlightNumber', 'LaunchSite']].head(10)
```

5]:

	FlightNumber	LaunchSite
0	1	CCAFS SLC 40
1	2	CCAFS SLC 40
2	3	CCAFS SLC 40
3	4	VAFB SLC 4E
4	5	CCAFS SLC 40
5	6	CCAFS SLC 40
6	7	CCAFS SLC 40
7	8	CCAFS SLC 40
8	9	CCAFS SLC 40
9	10	CCAFS SLC 40

# Drop a Row in DataFrame

```
df.drop(2,axis=0)
```

```
5]:
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs		LandingPad	Block
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False		NaN	1.0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False		NaN	1.0

# Drop a Column in DataFrame

```
df.drop('Outcome', axis=1)
```

7]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Flights	GridFins	R
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	1	False	
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	1	False	
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	1	False	
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	1	False	
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	1	False	
...	...	...	...	...	...	...	...	...	
85	86	2020-09-03	Falcon 9	15400.000000	VLEO	KSC LC 39A	2	True	
86	87	2020-10-06	Falcon 9	15400.000000	VLEO	KSC LC 39A	3	True	
87	88	2020-10-18	Falcon 9	15400.000000	VLEO	KSC LC 39A	6	True	
88	89	2020-10-24	Falcon 9	15400.000000	VLEO	CCAFS SLC 40	3	True	
89	90	2020-11-05	Falcon 9	3681.000000	MEO	CCAFS SLC 40	1	True	

90 rows × 17 columns

# Add New Column in DataFrame

```
df['new']=df['FlightNumber']
```

```
df.columns
```

```
[ ]: Index(['FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit',  
         'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs',  
         'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude',  
         'Class', 'new'],  
      dtype='object')
```

```
df.shape
```

```
[ ]: (90, 19)
```



# Inplace= True

```
df.columns
```

```
|: Index(['FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit',  
        'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs',  
        'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude',  
        'Class', 'new'],  
        dtype='object')
```

```
df.shape
```

```
|: (90, 19)
```

```
df.drop('new', axis=1,inplace=True)
```

```
df.columns
```

```
|: Index(['FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit',  
        'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs',  
        'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude',  
        'Class'],  
        dtype='object')
```

```
df.shape
```

```
|: (90, 18)
```

# Add New Row in DataFrame

```
▶ new_row = {'FlightNumber':1, 'Date':2, 'BoosterVersion':3, 'PayloadMass':4, 'Orbit':5,
            'LaunchSite':6, 'Outcome':7, 'Flights':8, 'GridFins':9, 'Reused':10, 'Legs':11,
            'LandingPad':12, 'Block':13, 'ReusedCount':14, 'Serial':15, 'Longitude':16, 'Latitude':17,
            'Class':18, 'new':19}
#append row to the dataframe
df_marks = df.append(new_row, ignore_index=True)
```

```
▶ df_marks.tail(1)
```

```
]:
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	S
90	1	2	3	4.0	5	6	7	8	9	10	11	12	13.0	14	

# Assignment:

## تمرین:

کدهای ارائه شده در درس را در نوتبوک جدیدی انجام داده و در صورت نیاز از نوتبوک هفته سوم استفاده کنید.

با سرچ در اینترنت روش ساختن یک دیتافریم با استفاده از دیکشنری پیدا کرده و یک دیتافریم دلخواه بسازید.