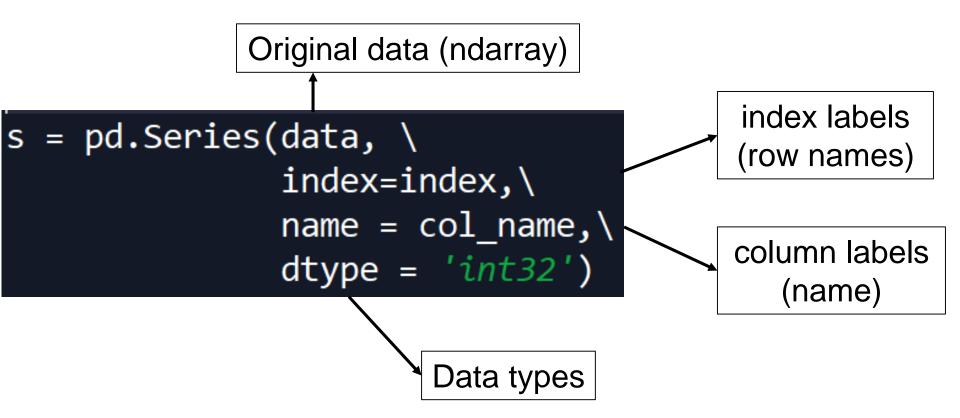
Algorithms and Data Structures with Python

Pandas - intro

Pandas Series – how to

Assign it all together into a pandas Series

(don't forget to define data type)



More about Pandas Series

Pandas series are data structures that hold retrievable index and values of certain type



Index	val
0	5
1	1
2	3
23456	8
4	0
5	4
6	2
7	9
8	6
9	7

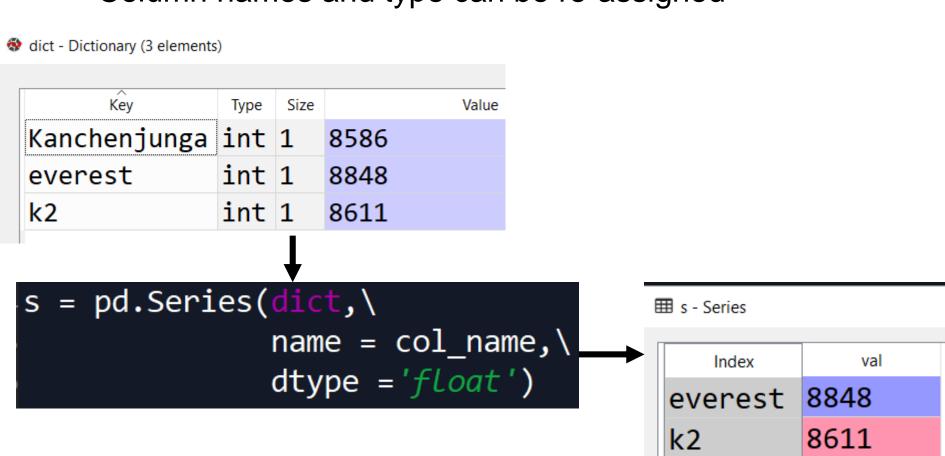
```
s.values
array([5, 1, 3, 8, 0, 4, 2, 9, 6, 7])
```

```
s.index
RangeIndex(start=0, stop=10, step=1)
```

```
s.dtype
dtype('int32')
```

Pandas Series from Dictionaries

- Data series can be directly assigned from dictionaries
- Index labels are defined by dictionary keys
- Column names and type can be re-assigned



Kanche...

Pandas Series from Dictionaries

- What if there are more rows than dictionary data inserts?
- pd.Series leaves 'nan' vacancies
- dict Dictionary (3 elements)

Key	Туре	Size		Value
Kanchenjunga	int	1	8586	
everest	int	1	8848	
k2	int	1	8611	

Kanchenjunga

mount Elbrus nan

Basic Pandas Series Operations

pd.Series slicing is similar to values retrieval from dictionary

```
In [31]: s[[3,5,9]]
Out[31]:
d     2
f     8
j     5
Name: val, dtype: int32
Name: val, dtype: int32
```

```
Index
                      val
а
                6
                4
h
                5
```

s - Series

pd.Series enables dictionary-like slicing

```
In [18]: s['j']
Out[18]: 5
```

```
In [19]: 'e' in s
Out[19]: True
```

Basic Pandas Series Operations

pd.Series enables basic math operations

Series Mean

```
In [22]: s.mean()
Out[22]: 4.5
```

Series standard deviation

```
In [33]: s.std()
Out[33]: 3.0276503540974917
```

Scalar / cross product multiplication

```
In [39]: s*5
Out[39]:
a 5
b 0
c 45
d 10
e 15
f 40
g 30
```

```
In [43]: s*s
Out[43]:
a 1
b 0
c 81
d 4
e 9
f 64
g 36
```

Index	val
a	1
b	0
С	9
d	2
e	3
f	8
g	6
h	4
i	7
i	5

m s - Series

Basic Pandas Series Operations

data series filtration
enables slicing by criterion (condition) using
a boolean vector as series index "slicer"

```
In [28]: s[s>s.mean()]
Out[28]:
c    9
f    8
g    6
i    7
j    5
Name: val, dtype: int32
```

Ⅲ s - Series	_
Index	val
a	1
b	0
С	9
d	2
e	3
f	8
g	6
h	4
i	7
j	5

Pandas Dataframes

DataFrame is a **2-dimensional labeled data structure** with columns of potentially different types

- Most commonly used data structure in pandas
- Has index and column labels
- Accepts the following inputs:
- Dict of 1D ndarrays, lists, dicts, or Series
- 2-D numpy.ndarray
- Series
- Another DataFrame

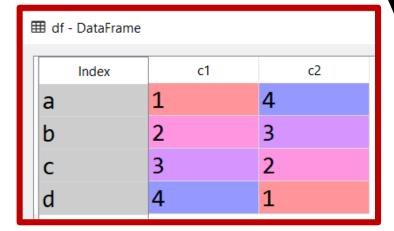
Pandas Dataframes – how to

1. Create dictionary with 2D list entries + column labels

Key	Туре	Size			Value	
c1	list	4	[1.0,	2.0,	3.0,	4.0]
c2	list	4	[4.0,	3.0,	2.0,	1.0]

2. Plug it into pd.DataFrame() function

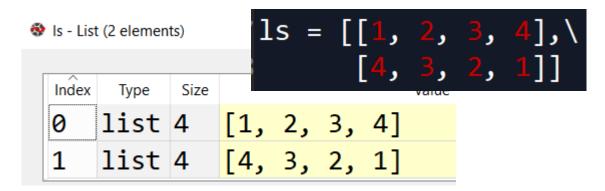




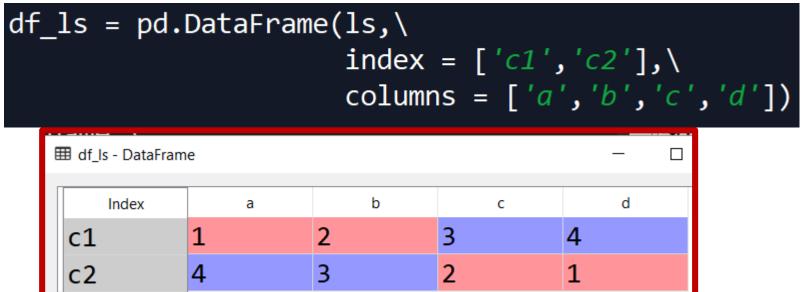
> Don't forget row labels!

Pandas Dataframes – how to

1. Create 2D list



2. Plug it into pd.DataFrame() function



Basic operations with Pandas Dataframes



Column Assignment with Pandas Dataframes



Index	c1	c2
a	1	4
b	2	3
С	3	2
d	4	1

 $df_ls['c3'] = df_ls['c1']+df_ls['c2']$

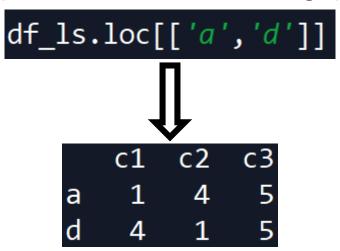


■ df_ls - DataFrame

c1	c2	c3
1	4	5
2	3	5
3	2	5
4	1	5
	_	

Pandas Dataframes Selection by label

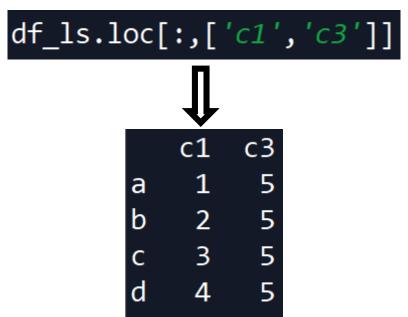




Pandas Dataframes Selection by label

Index	c1	c2	c3
a	1	4	5
b	2	3	5
С	3	2	5
d	4	1	5

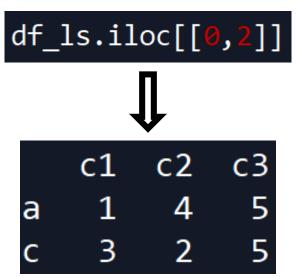
df.loc[:,[]] enables column slicing by label (similar to list slicing, column label is list)



Pandas Dataframes selection by position

Index	c1	c2	с3
a	1	4	5
b	2	3	5
С	3	2	5
d	4	1	5

df.iloc[] enables row selection by position
 (index is a list of integers)



Pandas Dataframes selection by position

	1		
Index	c1	c2	c3
a	1	4	5
b	2	3	5
С	3	2	5
d	4	1	5

