

Introduction to Python for Data Science

Session 1: Python refresher and NumPy

Gianluca Campanella

Today's class

Prerequisites

- Knowledge of Python syntax
- Lots of slides with `actual` code

Goals

- Review the basics of Python
- Introduce you to the scientific computing library NumPy

Python refresher

Variables and operators

Variables

```
1 session = 'Introduction to Python'  # Or use "..."  
2 day = 26  
3 temperature = 21.2  
4 pressure = 7.6e2  # Same as 7.6 * 10**2 = 760
```

Operators

- Comparison: **==**, **!=**, **<**, **<=**, **>**, **>=**
- Mathematical: **+**, **-**, *****, **/**, **//**, ******, **%**
- Logical: **and**, **or**, **not**

Containers

Lists

```
1 a = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
2 b = ['I', 'love', 'Python!']
3 a + b
```

Dictionaries

```
1 john = {
2     'first_name': 'John',
3     'age': 32
4 }
5 john['age']
```

Flow control

'If' statements

```
1  if condition:
2      ...
3  elif other_condition:
4      ...
5  else:
6      ...
```

Loops

```
1  for x in container:
2      ...
```

```
1  while condition:
2      ...
```

Functions

Defining a function

```
1 def sum_even(numbers):  
2     total = 0  
3     for number in numbers:  
4         if number % 2 == 0:  
5             total += number  
6     return total
```

Calling a function

```
1 sum_even([0, 1, 1, 2, 3, 5, 8, 13, 21, 34])  
2 sum_even(range(101))
```

Introduction to NumPy

NumPy provides...

- A new type `ndarray` representing multidimensional arrays
- Operators and functions to work with these arrays

```
1 import numpy as np
2
3 a = np.array([0, 1, 1, 2, 3, 5, 8, 13, 21, 34])
4 np.mean(a**2)
```

NumPy is...

- (Somewhat) similar to MATLAB® and R
- High-performance
- The basis of the data analysis library `pandas`
- Part of a bigger ecosystem of open-source scientific computing software