Applied Data Analytics

Basic Python

Lists and tuples

Hans-Martin von Gaudecker and Aapo Stenhammar

Contents

- Unlabelled, ordered containers
    - Lists
    - Tuples
- Selecting elements

Lists

```
>>> a = [1, 2, 3]
>>> type(a)
<class 'list'>

>>> a.append(4)
>>> a
[1, 2, 3, 4]

>>> a[0] = "bla"
>>> a
['bla', 2, 3, 4]

>>> len(a)
4
```

- Created with square brackets
- Definition: Mutable sequence of objects
    - **mutable**: Can change it after creation
    - **sequence**: An ordered collection
    - **of objects**: Items can consist of anything
- Lists are used a lot!
- `len` works for all collections

Tuples

```
>>> a = (1, 2, 3)
>>> type(a)
<class 'tuple'>

>>> b = (1)
>>> type(b)
<class 'int'>

>>> c = (1,)
>>> type(c)
<class 'tuple'>

>>> d = 2,
>>> type(d)
<class 'tuple'>
```

- Created with round brackets
- Definition: Immutable sequence of objects
  - **immutable**: Cannot change after creation
- Single element tuples need a comma
- But sometimes you don’t need the brackets!
- Less flexible than lists, less common
- Somewhat unfair

Selecting elements

```
>>> a = [1, 2, 3, 4, 5]
>>> a[1]
2

>>> a[1: 2]
[2]

>>> a[:2]
[1, 2]

>>> a[2:]
[3, 4, 5]

>>> a[-1]
[5]
```

- Selecting elements is the same for lists, tuples, and other sequences
- Indexing starts at 0
- Upper index of slices is not included
- lower and upper index can be left implicit
- negative indices start from the end