

A few tricks

Jannusch Bigge

12.12.2023

A few tricks

Lists

Lists are a very important data structure in Python.

- you can increase it by multiplying
- access elements from behind with negative indices
- ...

But sometimes you need more than that.

- luckily one can easily convert it to

Lists

Lists are a very important data structure in Python.

- you can increase it by multiplying
- access elements from behind with negative indices
- ...

But sometimes you need more than that.

- luckily one can easily convert it to
 - sets

Lists

Lists are a very important data structure in Python.

- you can increase it by multiplying
- access elements from behind with negative indices
- ...

But sometimes you need more than that.

- luckily one can easily convert it to
 - sets
 - numpy arrays
- and often its easier to use numpy arrays

Example

Until now we used a file handle and `readlines()` to load data.

Until now we used a file handle and `readlines()` to load data. More complex files like csv or excel is possible as well.

- csv `csv.reader(filehandle)`
- pandas `pd.read_csv(filename)`
- data `np.array(panda_data)`

Numpy can store and load data as well with `np.save()` and `np.load()`

Example

Dicts and Sets

- dicts are like lists but with keys instead of indices
- sets are like lists but without duplicates

Sets are very useful for counting unique elements

- `set(list)` gives you a set of all elements in list
- `len(set(list))` gives you the number of unique elements

Example

You can store the handle to a function in a variable.

- `func = np.mean`
- `func(data)`

This is useful if you want to use a function as an argument for another function. Or if you want a abstract function.

Lambda functions are a way to create functions on the fly.

- `func = lambda x: x**2`
- `func(2)`

This is useful if you want to use a function as an argument for another function. Or if you want a abstract function.

Task
