Data Types & Libraries

Jannusch Bigge

21.11.2023

Libraries

Libraries

Libraries are collections of functions.

They are not all included in Python by default.

Libraries

Libraries are collections of functions.

They are not all included in Python by default.

$\rightarrow$ We have to install them.

Libraries - Background

Python as the interpreter needs to know were to find the code.

Libraries - Background

Python as the interpreter needs to know were to find the code.

```
echo $PATH
whereis python3
python3 -v
ls /usr/lib/python3.11/
```

Libraries - Background

Python as the interpreter needs to know were to find the code.

```
echo $PATH
whereis python3
python3 -v
ls /usr/lib/python3.11/
```

$\rightarrow$ installing by hand is not a good idea.

Libraries - Background

Python as the interpreter needs to know were to find the code.

```
echo $PATH
whereis python3
python3 -v
ls /usr/lib/python3.11/
```

→ installing by hand is not a good idea.

→ installing in general not the best idea.

Virtual env

Python module that creates "fake environments"

- Isolated from the rest of the system
- Can be deleted without any problems
- Can be shared with others

Virtual env - activating

First we have to create a virtual environment. In this case called venv.

```
python3 -m venv venv
```

Next we have to activate it.

```
source venv/bin/activate
```

Now we can do our python stuff.

```
...
```

And finally if we are done we can deactivate it.

```
deactivate
```

pip

To install a python library we can use pip.

pip

To install a python library we can use pip.

```
pip install numpy
```

Data Types

Data Types

Python has built in data types which allow us to store data.

- list $\rightarrow$ [1,2,3]
- tuple $\rightarrow$ (1,2,3)
- dict $\rightarrow$ {1:2, 3:4}

List

Lists are the most common data type.

List

Lists are the most common data type.

In contrast to tuples they are mutable.

List

Lists are the most common data type.

In contrast to tuples they are mutable.

Lists have a lot of helpful functions:

```
my_list = [1,2,3]
my_list.append(4)
my_list[0] = 5
my_list.remove(2)
my_list.pop(0)
my_list.insert(0, 1)
my_list.sort()
my_list.reverse()
```

Dictonary

Dictonaries are a bit more complex.

Dictonary

Dictonaries are a bit more complex.

They are a collection of key-value pairs.

Dictonary

Dictonaries are a bit more complex.

They are a collection of key-value pairs.

They are mutable.

Dictonary

Dictonaries are a bit more complex.

They are a collection of key-value pairs.

They are mutable.

```
my_dict = {1:2, "three":4}
my_dict[1] = 5
my_dict["three"] = 6
my_dict.pop(1)
my_dict.popitem()
my_dict.clear()
```