

Chapter 13 - Advanced Python 2

Virtual Environment

An environment which is same as the system interpreter but is isolated from the other python environments on the system.

Installation

To use virtual environments, we write

`pip install virtualenv` → Install the package

We create a new environment using:

`virtualenv myprojectenv` → Creates a new venv

The next step after creating the virtual environment is to activate it.

We can now use this virtual environment as a separate python installation.

pip freeze Command

`pip freeze` returns all the packages installed in a given python environment along with the versions

"`pip freeze > requirements.txt`"

The above command creates a file named `requirements.txt` in the same directory containing the output of `pip freeze`.

We can distribute this file to other users and they can recreate the same environment using:

pip install -r requirements.txt

Lambda functions

functions created using an expression using lambda keyword

Syntax:

lambda arguments : expressions

↳ Can be used as a normal function

Example:

Square = lambda x: x*x

Square(6) → returns 36

Sum = lambda a, b, c: a+b+c

Sum(1, 2, 3) → returns 6

join method (Strings)

Creates a string from iterable objects

l = ["apple", "mango", "banana"]

","and,".join(l)

The above line will return "apple,and,mango,and,banana"

format method (Strings)

Formats the values inside the string into a desired output

template.format(p₁, p₂ ...)

↳ arguments

Syntax for format looks like:

"{ } is a good { }".format("Harry", "boy") — ①

"{1} is a good {0}".format("Harry", "boy") — ②

Output for ①

Harry is a good boy



Output for ②

boy is a good Harry

Map, Filter & Reduce

Map applies a function to all the items in an input_list

Syntax:

  map(function, input_list) ↳ Can be lambda function

Filter creates a list of items for which the function returns true.

list(filter(function))

↳ Can be a lambda function

Reduce applies a rolling computation to sequential pairs of elements

from functools import reduce

val = reduce(function, list1)

↳ Can be a lambda function

If the function computes sum of two numbers and the

list is [1, 2, 3, 4]

1 2 3 4
└───┘

3 3 4
└───┘

6 4
└───┘
10

\Rightarrow Sequential Computation