Python 101

Python STL - Recommended by Guido Von Rossum

Contents

Regex - Regular Expressions

Regular Expressions in python are a way to match strings to a pattern.

eg. Lets say I want to define a pattern to match all of the strings that have any nymber of lowercase characters in them then I could write something like [a-z]*.

Regular Expressions are represented using strings as well.

Regex - Regular Expressions

```
import re
```

`re.findall('regex', 'string to search regex in ')` ← It searches for occurence of regex in string to search regex in and returns all of the sequences that match the regex.

`re.sub('regex', 'sub', 'string to search regex in ')` ← It is like a replace with regex.

Regex - Regular Expressions

Defining a regular expression

As we know its a string, But its a string with special characters to define the patterns. Few of the important special characters are these.

1. . ← Anything but a new line
2. ^ ← Represents start of a string
3. $ ← End of a string ie. just before the newline
4. * ← 0 or more occurences. ‘a’* means ‘’, ‘a’, ‘aa’, ‘aaa’, ‘aaaa’....
5. + ← 1 or more occurences ‘a’+ means a* - ‘’
6. \s ← match whitespace.
7. \t ← match tab.
8. \w ← match words in any language supported by unicode

Regex - Regular Expressions

Defining a regular expression

[ ] ← Define a set of characters

eg. [a-z] ← all characters from a to z

[abc] ← ‘a’, ‘b’, ‘c’

[a-zA-Z0-9] ← all characters from a to z A to Z and 0 to 9

character**{m, n}** ← match a character minimum m and maximum n times.

Omit m or n to give lower bound 0 and upper bound inf.

character**{m}** ← match a character exactly m times.

Regex - Regular Expressions

Writing a little complex Regex

- **Write a regex for matching indian cellphone numbers.**

It must start with +91 and must contain exactly 10 digits after it.

''\+91[0-9]{10}' ← Using escape is necessary here because + is a special character.

- **Write a regex for matching indian and israelie cellphone numbers.**

It must start with either +91 or +972 and must have 10 digits if starts with +91 and must have 8 to 9 digits if starts with +972

See example /Day10/regex.py

OS tasks using Python

We can use these three modules in python for OS tasks, like finding files, creating dirs, deleting dirs, invoking commands etc.

1. os
2. shutil
3. glob

See example /Day10/os_tasks.py

Fiddling with sys module

Changing prompt signs on interpreter -

```
import sys

sys.ps1 = 'XYZ '

sys.ps2 = 'ABC '
```

Getting commandline arguments-

sys.argv ← Contains all of the command line arguments in a list.

Fiddling with sys module

sys.version_info ← version of python being used.

sys.exit(exit_code) ← Immediately exit with the status_code 0: success non zero error.

import errno ← This module lists all of the error codes.

Doing Random things

Random module is used to perform these tasks.

These are often useful while making **fair** and **unbiased** choices like in games.

See example /Day10/random_things.py

Doing Math with Python

math and statistics modules are used to perform general mathematics tasks with inbuilt functions in python.

See example /Day10/do_math.py

Date and Time

datetime module is used for performing datetime manipulation tasks in python.

See example /Day10/do_datetime.py

Downloading Files Using Python

urllib can be used in python3 for making http requests. In order to download a file we need to make a http requests to the url and write it’s response into a file.

See example /Day10/downloader.py

Compression & Bundling Using Python

We can create tars, zips and even compress raw strings using pythons gzip, zipfile and tarfile modules.

See example /Day10/compression.py

QA using embedded tests

We can effortlessly embed tests in python docstrings. Which can later be tested using testmod function of doctest module.

See example /Day10/do_QA.py

Writing simple unittests

We can effortlessly embed tests in python docstrings. Which can later be tested using testmod function of doctest module.

See example /Day10/do_unittest.py