**Artikel Library Python**

Library yang digunakan :

1. **Import os**

Modul OS dengan Python menyediakan cara menggunakan ketergantungan sistem operasi fungsionalitas. Fungsi yang disediakan modul OS memungkinkan untuk berinteraksi dengan sistem operasi yang mendasari bahwa Python sedang berjalan - baik itu Windows, Mac atau Linux.

Berikut beberapa fungsi **Import os** :

Import os

Menjalankan perintah shell

os.system ()

Dapatkan lingkungan pengguna

os.environ ()

#Kembalikan direktori kerja saat ini.

os.getcwd ()

Kembalikan id grup sebenarnya dari proses saat ini.

os.getgid ()

Kembalikan id pengguna proses saat ini.

os.getuid ()

Mengembalikan ID proses nyata dari proses saat ini.

os.getpid ()

Atur umask angka saat ini dan kembalikan umask sebelumnya.

os.umask (topeng)

Kembalikan informasi yang mengidentifikasi sistem operasi saat ini.

os.uname ()

Ubah direktori root dari proses saat ini ke path.

os.chroot (jalan)

Kembalikan daftar entri dalam direktori yang diberikan oleh path.

os.listdir (jalur)

Buat direktori bernama jalur dengan mode mode numerik.

os.mkdir (jalan)

Fungsi pembuatan direktori rekursif.

os.makedirs (jalan)

Hapus (hapus) path file.

os.menghapus (jalan)

Hapus direktori secara rekursif.

os.removedirs (path)

Ganti nama file atau direktori src menjadi dst.

os.rename (src, dst)

Hapus (hapus) jalur direktori.

os.rmdir (jalan)

1. **Import time**

Modul waktu memperlihatkan fungsi pustaka C untuk memanipulasi tanggal dan waktu. Karena ini terkait dengan implementasi C yang mendasarinya, beberapa detail (seperti awal zaman dan nilai tanggal maksimum yang didukung) bersifat spesifik platform.

Salah satu fungsi inti dari modul waktu adalah waktu (), yang mengembalikan jumlah detik sejak dimulainya zaman sebagai nilai titik mengambang.

import time

print 'The time is:', time.time()

Meskipun nilainya selalu float, presisi sebenarnya bergantung pada platform.

$ python time\_time.py

The time is: 1205079300.54

Representasi float berguna ketika menyimpan atau membandingkan tanggal, tetapi tidak berguna untuk menghasilkan representasi yang dapat dibaca manusia. Untuk logging atau mencetak waktu ctime () dapat lebih bermanfaat.

import time

print 'The time is :', time.ctime()

later = time.time() + 15

print '15 secs from now :', time.ctime(later)

Di sini garis keluaran kedua menunjukkan cara menggunakan ctime () untuk memformat nilai waktu selain waktu saat ini.

$ python time\_ctime.py

The time is : Sun Mar 9 12:18:02 2008

15 secs from now : Sun Mar 9 12:18:17 2008

1. **Import threading**

Threading dalam python digunakan untuk menjalankan banyak utas (tugas, panggilan fungsi) secara bersamaan. Perhatikan bahwa ini tidak berarti bahwa mereka dijalankan pada CPU yang berbeda. Utas Python TIDAK akan membuat program lebih cepat jika sudah menggunakan waktu CPU 100%.

Utas Python digunakan dalam kasus di mana pelaksanaan tugas melibatkan beberapa menunggu. Salah satu contohnya adalah interaksi dengan layanan yang dihosting di komputer lain, seperti server web. Threading memungkinkan python untuk mengeksekusi kode lain sambil menunggu; ini mudah disimulasikan dengan fungsi tidur.

1. **Import multiprocessing**

Multiprocessing adalah paket yang mendukung proses pemijahan menggunakan API yang mirip dengan modul threading. Paket multi-pemrosesan menawarkan konkurensi lokal dan jarak jauh, secara efektif menuntun Global Interpreter Lock dengan menggunakan subproses alih-alih utas. Karena hal ini, modul multiprosesing memungkinkan programmer untuk memanfaatkan sepenuhnya beberapa prosesor pada mesin yang diberikan. Ini berjalan di kedua Unix dan Windows.

Modul multiprosesing juga memperkenalkan API yang tidak memiliki analog dalam modul threading. Contoh utama dari ini adalah objek Pool yang menawarkan cara yang nyaman untuk memparalelkan pelaksanaan fungsi di beberapa nilai input, mendistribusikan data input di seluruh proses (paralelisme data). Contoh berikut menunjukkan praktik umum mendefinisikan fungsi tersebut dalam modul sehingga proses anak dapat mengimpor modul itu dengan sukses. Contoh dasar paralelisme data ini menggunakan Pool,

From multiprocessing import Pool

def f (x):

return x \* x

if \_\_name\_\_ == '\_\_main\_\_':

p = Pool (5)

print (p.map (f, [1, 2, 3]))

akan mencetak ke output standar

[1, 4, 9]