

 alifazahra729 / Workshop--python- Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)[master](#) Workshop--python- / teori / minggu-04 / ...

alifazahra000 minggu-04 ...

2 minutes ago [History](#)

..



README.md

2 minutes ago

README.md



Ringkasan :

- Bab 6 Jika Anda keluar dari juru bahasa Python dan memasukkannya lagi, definisi yang Anda buat (fungsi dan variabel) akan hilang. Oleh karena itu, jika Anda ingin menulis program yang agak panjang, lebih baik Anda menggunakan editor teks untuk menyiapkan input untuk juru bahasa dan menjalankannya dengan file tersebut sebagai input. Saat program Anda semakin panjang, Anda mungkin ingin membaginya menjadi beberapa file untuk pemeliharaan yang lebih mudah. Untuk mendukung ini, Python memiliki cara untuk meletakkan definisi dalam file dan menggunakannya dalam skrip atau dalam contoh interaktif dari juru bahasa. Modul adalah file yang berisi definisi dan pernyataan Python.

Ini tidak menambahkan nama fungsi yang didefinisikan di fibo langsung ke namespace saat ini (lihat Python Scopes dan Namespaces untuk detail lebih lanjut); itu hanya menambahkan nama modul fibo di sana.

Jika Anda keluar dari juru bahasa Python dan memasukkannya lagi, definisi yang Anda buat (fungsi dan variabel) akan hilang. Oleh karena itu, jika Anda ingin menulis program yang agak panjang, lebih baik Anda menggunakan editor teks untuk menyiapkan input untuk juru bahasa dan menjalankannya dengan file tersebut sebagai input. Saat program Anda semakin panjang, Anda mungkin ingin membaginya menjadi beberapa file untuk pemeliharaan yang lebih mudah. Untuk mendukung ini, Python memiliki cara untuk meletakkan definisi dalam file dan menggunakannya dalam skrip atau dalam contoh interaktif dari juru bahasa. Modul adalah file yang berisi definisi dan pernyataan Python.

Ini tidak menambahkan nama fungsi yang didefinisikan di fibo langsung ke namespace saat ini (lihat Python Scopes dan Namespaces untuk detail lebih lanjut); itu hanya menambahkan nama modul fibo di sana.

Modul dapat berisi pernyataan yang dapat dieksekusi serta definisi fungsi. Mereka dieksekusi hanya saat pertama kali nama modul ditemukan dalam pernyataan impor.) Setiap modul memiliki ruang nama pribadinya sendiri, yang digunakan sebagai ruang nama global oleh semua fungsi yang ditentukan dalam modul. Merupakan kebiasaan tetapi tidak diharuskan untuk menempatkan semua pernyataan impor di awal modul (atau skrip, dalam hal ini). Nama modul yang diimpor, jika ditempatkan di tingkat atas modul (di luar fungsi atau kelas apa pun), ditambahkan ke ruang nama global modul.

Ini tidak memperkenalkan nama modul dari mana impor diambil di namespace lokal (jadi dalam contoh, fibo tidak ditentukan).

Ini mengimpor semua nama kecuali yang dimulai dengan garis bawah (_). Dalam banyak kasus, pemrogram Python tidak menggunakan fasilitas ini karena fasilitas ini memperkenalkan serangkaian nama yang tidak diketahui ke dalam juru bahasa, mungkin menyembunyikan beberapa hal yang telah Anda tetapkan.

Ini secara efektif mengimpor modul dengan cara yang sama seperti impor fibo, dengan satu-satunya perbedaan adalah tersedia sebagai fib.

Catatan : Untuk alasan efisiensi, setiap modul hanya diimpor satu kali per sesi juru bahasa. muat ulang (nama modul).

kode dalam modul akan dieksekusi, sama seperti jika Anda mengimpornya, tetapi dengan **name** disetel ke "**main**".

Ini sering digunakan baik untuk menyediakan antarmuka pengguna yang mudah digunakan ke modul, atau untuk tujuan pengujian (menjalankan modul saat skrip menjalankan rangkaian pengujian).

Saat modul bernama spam diimpor, interpreter pertama-tama akan mencari modul bawaan dengan nama tersebut. Nama modul ini tercantum dalam sys. Jika tidak ditemukan, maka akan mencari file bernama spam. py dalam daftar direktori yang diberikan oleh variabel sys. path diinisialisasi dari lokasi berikut:

- Direktori yang berisi skrip input (atau direktori saat ini ketika tidak ada file yang ditentukan).
 - PYTHONPATH (daftar nama direktori, dengan sintaks yang sama dengan variabel shell PATH).
- Catatan : Pada sistem file yang mendukung symlink, direktori yang berisi skrip input dihitung setelah symlink diikuti. Dengan kata lain direktori yang berisi symlink tidak ditambahkan ke jalur pencarian modul. Direktori yang berisi skrip yang sedang dijalankan ditempatkan di awal jalur pencarian, di depan jalur pustaka standar. Ini berarti skrip di direktori itu akan dimuat alih-alih modul dengan nama yang sama di direktori perpustakaan.

Untuk mempercepat pemuatan modul, Python meng-cache versi terkompilasi dari setiap modul di direktori **pycache** di bawah nama module. pyc, di mana versi menyandikan format file yang dikompilasi; umumnya berisi nomor versi Python. 3 versi spam yang dikompilasi. Konvensi penamaan ini memungkinkan modul yang dikompilasi dari rilis yang berbeda dan versi Python yang berbeda untuk hidup berdampingan. Python memeriksa tanggal modifikasi sumber terhadap versi yang dikompilasi untuk melihat apakah sudah kedaluwarsa dan perlu dikompilasi ulang. Ini adalah proses yang sepenuhnya otomatis. Python tidak memeriksa cache dalam dua keadaan. Pertama, selalu mengkompilasi ulang dan tidak menyimpan hasil modul yang dimuat langsung dari baris perintah. Kedua, tidak memeriksa cache jika tidak ada modul sumber. Beberapa tip untuk para ahli: - Anda dapat menggunakan sakelar -O atau -OO pada perintah Python untuk mengurangi ukuran modul yang dikompilasi. pyc daripada saat dibaca dari file . pyc adalah kecepatan pemuatannya. pyc file untuk semua modul dalam direktori.

- Ada detail lebih lanjut tentang proses ini, termasuk diagram alir keputusan, di PEP 3147.

Beberapa modul dibangun ke dalam juru bahasa; ini memberikan akses ke operasi yang bukan bagian dari inti bahasa tetapi tetap dibangun, baik untuk efisiensi atau untuk menyediakan akses ke primitif sistem operasi seperti panggilan sistem. Satu modul tertentu patut mendapat perhatian: sys, yang dibangun di setiap juru bahasa Python.

Kedua variabel ini hanya ditentukan jika penafsir berada dalam mode interaktif. Variabel sys.

Fungsi built-in dir() digunakan untuk mengetahui nama mana yang didefinisikan oleh modul.

Perhatikan bahwa ini mencantumkan semua jenis nama: variabel, modul, fungsi, dll.

dir() tidak mencantumkan nama fungsi dan variabel bawaan.

B menunjuk submodule bernama B dalam sebuah paket bernama A. Misalkan Anda ingin merancang kumpulan modul ("paket") untuk penanganan file suara dan data suara yang seragam. Ada banyak format file suara yang berbeda (biasanya dikenal dengan ekstensinya, misalnya: . au), jadi Anda mungkin perlu membuat dan memelihara koleksi modul yang terus bertambah untuk konversi antara berbagai format file. Ada juga banyak operasi berbeda yang mungkin ingin Anda lakukan pada data suara (seperti mencampur, menambahkan gema, menerapkan fungsi equalizer, membuat efek stereo buatan), jadi selain itu Anda akan menulis aliran modul tanpa henti untuk dilakukan. operasi ini.

Saat mengimpor paket, Python mencari melalui direktori di sys. File py diperlukan untuk membuat Python memperlakukan direktori yang berisi file sebagai paket. Ini mencegah direktori dengan nama umum, seperti string, menyembunyikan modul valid secara tidak sengaja yang muncul kemudian di jalur pencarian modul.

File py diperlukan untuk membuat Python memperlakukan direktori yang berisi file sebagai paket.

Ini bisa memakan waktu lama dan mengimpor sub-modul mungkin memiliki efek samping yang tidak diinginkan yang seharusnya hanya terjadi ketika sub-modul diimpor secara eksplisit. Pernyataan impor menggunakan konvensi berikut: jika sebuah paket ``s`` **init**. Terserah pembuat paket untuk tetap memperbarui daftar ini ketika versi baru dari paket dirilis. Pembuat paket juga dapat memutuskan untuk tidak mendukungnya, jika mereka tidak melihat kegunaan untuk mengimpor `*` dari paket mereka.

`effects import *` akan mengimpor tiga submodul suara yang disebutkan. efek telah diimpor (mungkin menjalankan kode inisialisasi apa pun di **init**.py) lalu mengimpor nama apa pun yang ditentukan dalam paket. Ini termasuk nama apa pun yang ditentukan (dan submodul dimuat secara eksplisit) oleh **init**. Ini juga mencakup submodul paket apa pun yang dimuat secara eksplisit oleh pernyataan impor sebelumnya.

Dalam contoh ini, modul `echo` dan `surround` diimpor dalam namespace saat ini karena ditentukan dalam suara. Ingat, tidak ada salahnya menggunakan `from package import specific_submodule`! Faktanya, ini adalah notasi yang disarankan kecuali modul pengimpor perlu menggunakan submodul dengan nama yang sama dari paket yang berbeda.

Ketika paket disusun menjadi sub-paket (seperti paket suara dalam contoh), Anda dapat menggunakan impor absolut untuk merujuk ke submodul dari paket saudara. Misalnya, jika modul berbunyi. `vocoder` perlu menggunakan modul `gema` dalam suara. efek mengimpor `gema`.

Perhatikan bahwa impor relatif didasarkan pada nama modul saat ini. Karena nama modul utama selalu **"main"**, modul yang dimaksudkan untuk digunakan sebagai modul utama aplikasi Python harus selalu menggunakan impor absolut.

Variabel ini dapat dimodifikasi; hal itu memengaruhi pencarian modul dan subpaket di masa mendatang yang terdapat dalam paket. Meskipun fitur ini tidak sering dibutuhkan, fitur ini dapat digunakan untuk memperluas kumpulan modul yang ditemukan dalam sebuah paket.

Give feedback