alifazahra729 / **Workshop--python-** Public

Code Issues Pull requests Actions Projects Wiki Security ...

master Workshop--python- / teori / **minggu-03** /

alifazahra000 minggu-03 ... 1 minute ago History

..

README.md 1 minute ago

README.md

Ringkasan :

- Bab 5 Lebih lanjut tentang Daftar Jenis data daftar memiliki beberapa metode lainnya. append(x) Tambahkan item ke akhir daftar. Setara dengan a[len(a):] = [x]. extend(iterable) Perpanjang daftar dengan menambahkan semua item dari iterable. Setara dengan a[len(a):] = iterable. insert(i, x) Masukkan item pada posisi tertentu. Argumen pertama adalah indeks elemen sebelum disisipkan, jadi a. insert(len(a), x) setara dengan a. hapus(x) Hapus item pertama dari daftar yang nilainya sama dengan x. pop([i]) Hapus item pada posisi yang diberikan dalam daftar, dan kembalikan. pop() menghapus dan mengembalikan item terakhir dalam daftar. (Kurung siku di sekitar i di tanda tangan metode menunjukkan bahwa parameter adalah opsional, bukan berarti Anda harus mengetikkan tanda kurung siku di posisi itu. clear() Hapus semua item dari daftar. index(x[, start[, end]]) Mengembalikan indeks berbasis nol dalam daftar item pertama yang nilainya sama dengan x. Argumen opsional awal dan akhir diinterpretasikan seperti dalam notasi irisan dan digunakan untuk membatasi pencarian ke urutan tertentu dari daftar. Indeks yang dikembalikan dihitung relatif terhadap awal urutan penuh daripada argumen awal. count(x) Mengembalikan berapa kali x muncul dalam daftar. sort(*, key=None, reverse=False) Sortir item daftar pada tempatnya (argumen dapat digunakan untuk penyesuaian sortir, lihat sortir() untuk penjelasannya). reverse() Balikkan elemen daftar pada tempatnya. copy() Return a shallow copy of the list. An example that uses most of the list methods: >>> fruits = [`orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana'] fruits. index('banana', 4) # Find next banana starting at position 4 6 fruits. membalikkan () buah ['pisang', 'apel', 'kiwi', 'pisang', 'pir', 'apel', 'jeruk'] buah-buahan. append('anggur') buah-buahan ['pisang', 'apel', 'kiwi', 'pisang', 'pir', 'apel', 'jeruk', 'anggur'] buah-buahan. sort() buah-buahan ['apel', 'apel', 'pisang', 'pisang', 'anggur', 'kiwi', 'jeruk', 'pir']. pop() 'pear' Anda mungkin telah memperhatikan bahwa metode seperti menyisipkan, menghapus, atau mengurutkan yang hanya mengubah daftar tidak memiliki nilai kembalian yang dicetak – mereka mengembalikan default Tidak ada. tambahkan(6) tumpukan. tambahkan(7) tumpukan [3, 4, 5, 6, 7] tumpukan. Sementara menambahkan dan muncul dari akhir daftar cepat, melakukan sisipan atau muncul dari awal daftar lambat (karena semua elemen lain harus digeser satu). deque yang dirancang untuk menambahkan dan muncul dengan cepat dari kedua ujungnya. append(''Terry'') # Terry tiba

antrian. append(''Graham'') # Graham tiba antrian. Aplikasi umum adalah membuat daftar baru di mana setiap elemen adalah hasil dari beberapa operasi yang diterapkan ke setiap anggota urutan lain atau iterable, atau untuk membuat urutan elemen yang memenuhi kondisi tertentu. Sebagai contoh, misalkan kita ingin membuat daftar kotak, seperti: >>> kotak = [] untuk x dalam rentang(10): kotak. Kita dapat menghitung daftar kotak tanpa efek samping menggunakan: kotak = daftar(peta(lambda x: x**2, range(10))) atau, setara: kotak = [x**2 untuk x dalam rentang(10)] yang lebih ringkas dan mudah dibaca. Hasilnya akan berupa daftar baru yang dihasilkan dari evaluasi ekspresi dalam konteks klausa for dan if yang mengikutinya. Misalnya, listcomp ini menggabungkan elemen dari dua daftar jika tidak sama: >>> [(x, y) for x in [1,2,3] for y in [3,1,4] if x != y] [(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)] dan setara ke: >>> sisir = [] untuk x di [1,2,3]: untuk y di [3,1,4]: jika x != y: sisir. tambahkan((x, y)) sisir [(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)] Perhatikan bagaimana urutan pernyataan for dan if

Give feedback