Plagiarism Scan Report

Summary	
Report Genrated Date	26 Oct, 2017
Plagiarism Status	**100% Unique**
Total Words	1000
Total Characters	6347
Any Ignore Url Used	

Content Checked For Plagiarism:

Python's * untuk * dan * in * constructs sangat berguna, dan penggunaan pertama dari yang akan kita lihat adalah dengan da□tar. * Untuk * membangun - untuk da□tar var - adalah cara mudah untuk melihat setiap elemen dalam da□tar (atau koleksi lainnya). Jangan menambah atau menghapus dari da□tar selama iterasi.

~ kotak = [1, 4, 9, 16]
~ Jumlah = 0
~ Untuk num dalam kotak:
~~~ Jumlah + = num
~ Jumlah cetak

Jika Anda tahu hal macam apa yang ada dalam da□tar, gunakan nama variabel dalam lingkaran yang menangkap in□ormasi seperti "num", atau "name", atau "url". Karena kode python tidak memiliki sintaks lain untuk mengingatkan Anda tentang tipe, nama variabel Anda adalah cara kunci bagi Anda untuk tetap mempertahankan apa yang sedang terjadi.

* Dalam * membangun sendiri adalah cara mudah untuk menguji apakah sebuah elemen muncul dalam da□tar (atau koleksi lainnya) - nilai dalam koleksi - tes jika nilainya ada dalam koleksi, mengembalikan True / False.

~ Da□tar = ['larry', 'curly', 'moe']
~ jika 'keriting' dalam da□tar:
~~~ Cetak 'yay'

The □or / in constructs sangat umum digunakan pada kode Python dan bekerja pada tipe data selain list, jadi sebaiknya ha□alkan sintaksnya. Anda mungkin memiliki kebiasaan dari bahasa lain di mana Anda memulai pengulangan manual melalui koleksi, dengan Python yang seharusnya Anda gunakan untuk / in.

Anda juga dapat menggunakannya untuk / dalam mengerjakan sebuah string. String bertindak seperti da□tar karakternya, jadi untuk ch di s: print ch mencetak semua karakter dalam sebuah string.

Jarak

Fungsi range (n) menghasilkan angka 0, 1, ... n-1, dan range (a, b) mengembalikan a, a + 1, ... b-1 - sampai tapi tidak termasuk angka terakhir . Kombinasi □ungsi □or-loop dan range () memungkinkan Anda membuat numerik tradisional untuk loop:

print the numbers □rom 0 through 99
~ □or i in range(100):
~~~ print i

Ada varian xrange () yang menghindari biaya membangun keseluruhan da□tar untuk kasus sensiti□ kinerja (dalam Python 3000, range () akan memiliki perilaku kinerja yang baik dan Anda dapat melupakan xrange ()).
Sementara Loop

Python juga memiliki standar while-loop, dan * break * dan * continue * statements bekerja seperti di C ++ dan Java, mengubah jalannya loop terdalam. Di atas untuk / dalam loop memecahkan kasus umum iterasi pada setiap elemen dalam da□tar, namun loop sementara memberi Anda kontrol penuh atas angka indeks. Berikut adalah loop sementara yang mengakses setiap elemen ke-3 dalam da□tar:

~ Mengakses setiap elemen ke-3 dalam da□tar
~ I = 0
~ sementara i
~~~