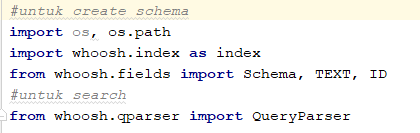
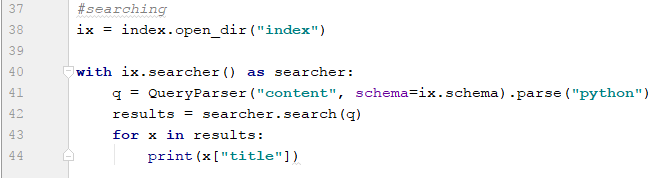
Library python yang digunakan untuk membuat sebuah inverted index adalah whoosh. Hingga saat ini versi library whoosh yang terbaru adalah whoosh 2.7.4. Setelah whoosh didownload dan diinstall pada python, library whoosh dapat langsung digunakan dengan melakukan import seperti berikut



Untuk library whoosh.fields akan digunakan untuk membuat schema dari sebuah inverted index. Sedangkan whoosh.index dan whoosh.qparser merupakan library yang digunakan untuk melakukan pencarian kata.

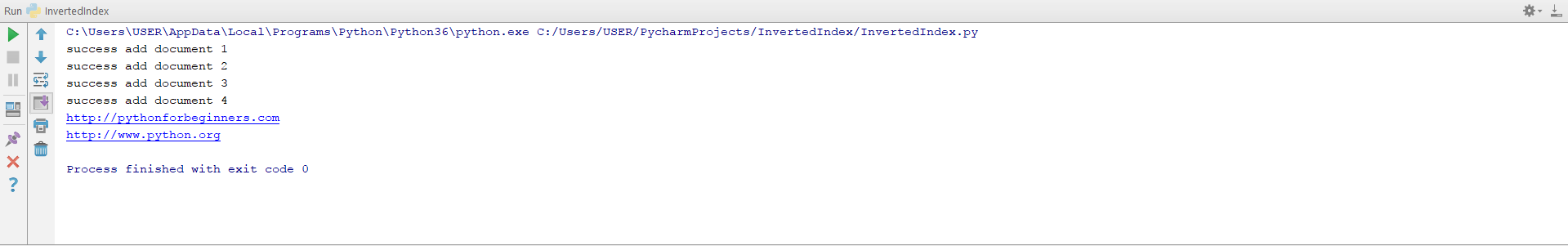
Pada awal program harus dibuat schema terlebih dahulu untuk menentukan struktur penyimpanan inverted index. Sebagai contoh pada program kali ini akan disediakan title yang berisi link html dan content yang berisi hasil fetching dari link html tersebut. Pada awal program dibuat schema dengan struktur title sebagai id dan content sebagai text(baris ke-13). Pada potongan code program ini akan dibuat sebuah folder bernama index dengan menggunakan library os. Kemudian untuk menambahkan sebuah document dapat digunakan method *writer()* seperti yang ditunjukan pada baris ke-21. Pada program ini akan menyimpan 4 dokumen dari link-link yang ditunjukan pada baris 24-27. Untuk menambahkan document dapat digunakan method *writer().add\_document()* dimana akan diminta 2 parameter(sesuai dengan struktur schema yang dibuat) yaitu title dan content(baris ke-31). Setelah selesai menambahkan document dilakukan commit seperti yang ditunjukan pada baris ke-35.





Selain membuat sebuah inverted index, library whoosh juga dapat melakukan search pada inverted index yang telah dibuat sebelumnya dengan menggunakan method *searcher().* Sebelum melakukan search , langkah awal yang harus dilakukan adalah membuka schema yang telah kita buat sebelumnya(baris ke-38). Setelah itu kita dapat membuat query dengan menggunakan QueryParser yang terdapat 2 parameter. Parameter pertama merupakan content yang berisi isi document. Parameter kedua berisi schema yang telah dibuat. Kemudian akan dilanjutkan dengan method parser dimana parser memiliki satu parameter yaitu kata-kata yang akan dicari. Setelah selesai membuat query, maka dapat dilakukan search dengan method *searcher().search()* dimana dibutuhkan 1 parameter yaitu query yang telah dibuat sebelumnya. Kemudian untuk mencetak kata “python” muncul di dokumen mana saja dapat dilakukan print results[i][“title”] dimana I adalah dokumen keberapa yang mengandung kata “python”atau seperti yang ditunjukan pada baris ke-43-44.

Hasil eksekusi program



Pada gambar di atas dapat dilihat bahwa kata-kata “python” terdapat pada link <http://pythonforbeginners.com> dan http://www.python.org.