

# Some problem solving and how it's easier with python

admin · Monday, March 23rd, 2009

There's a project I work on that required me to make an import utility for a CRM. The import should get a comma separated values file of clients and information about clients, and save it to the database. The database is split across several tables, so in the `clients` table I normally don't keep the name of the company, but just a foreign key. Now, our client is not very good with numbers and she needed to import files in which she could enter the name of the company instead of the database ID. A spreadsheet row representing a client looks like this:

```
FirstName | LastName | Email           | Company
John      | Doe      | johndoe@example.com | Coca Cola
```

But the database row in the `clients` table looks like this:

```
first_name | last_name | email           | company_id
John       | Doe       | john@example.com | 2
```

What I need to do is search for the company named 'Coca Cola' in the `companies` table and replace the name with it's ID. This is all fine except for one problem - typos. Moreover, the user could write "Apple Computer Inc." instead of "Apple Inc.". So I needed a way to compare the input strings with the ones in the database.

After poking around I found out about the [Levenshtein distance](#) between strings, but that solved only half of my problems - the typo part. The distance would be very small between "Apple" and "Aple" but very big between "ACME International Inc." and "International ACME Inc.", and the latter two are obviously the same.

I devised the following method to compare entries:

- Split up the terms by words and eliminate blanks
- Get the Levenshtein distance between each word from the first term and each word from the second term. Comparing "Apple Computer Inc." with "Apple Inc." for example, will give a matrix of 6 distances. ☒
- Get the shortest term (one with less words, not the one with less characters). It has 2 words in this case. Then choose the smallest values from each **row**. When you pick the smallest **row** value, you cannot pick anymore values from that **column**. This means that the word in the **column** is the best match for some word in the **rows**.
- Add these values up and add the difference between the word count of the 2 terms - and you have a score for the similarity of the terms. If the score is zero, they are the same. We are adding +1 for each extra word, but this can be weighted if needed. The point is that we don't care much for extra words since company names can have many words in them, but they are often called by one or two words.

But there is a problem with step 3. If, for example, a column has the lowest values for more than one row, we always choose the first, and this practice is not always the best answer. For instance, matching "Fast Cats" with "Fats Cats" (notice the typo) gets a total score of 3 - matching **Cats** to **Fats** and **Fast** to **Cats**, which is wrong - it will be 2 if we match **Fast** to **Fats** and **Cats** to **Cats**, which is the intended solution.



So to be sure we have the best match, we need to always have the lowest sum that is unique across rows and columns. One solution is to make all permutations of the words in the **columns** and join them to a single permutation of the words **in the rows** then see which one has the lowest score. If the words in the rows are fewer then we need to get all permutations  $P(n,k)$  of the words in **the columns**, where **n** is the number of columns and **k** is the number of rows. This is a  $O(n!)$  algorithm but it's the best that I could think of - practically the same problem as finding every possible way to place 8 rooks on a chess table without making them attack each other.

And finally, here is the part where we get to write some code. I need a function that can calculate all permutations consisted of **k** elements out of a larger set consisted of **n** elements ( $k \leq n$ ).

I decided first to write the algorithm in Python because it's cleaner and easier to think, and then to rewrite it in PHP. The first attempt was really, really sucky and I won't talk about it because I'm a bit embarrassed. But I wasn't aware of a neat thing that Python has: the **yield** statement. The darn thing can be written in 6 lines with it:

```
def permutations(the_set, n):
    if n==0:
        yield []
```

```

else:
    for i in xrange( len( the_set ) ):    for x in permutations
( the_set[0:i] + the_set[i+1:], n-1 ):
    yield [the_set[i]]+x

```

I will go into the yield statement later, maybe I will extend this post, but for now, I'll say that it allows you to make a function that will calculate the combinations on the fly, without storing them in a huge list and then returning the list. It sort of lazy-loads the list of combinations when needed. There is no such thing in PHP (as far as I know). So here's my best shot at the function in PHP:

```

function permutations( $array, $size )
{
    $result = array();
    $x = count($array);
    for( $i=0; $i<$x; $i++ ) {
        $copy = $array; // copy: array_splice gets the arg by reference
        $item = array_splice( $copy, $i, 1 );
        if( $size == 1 )
            $result[] = $item;
        else {
            $rest = permutations( $copy , $size - 1 );
            foreach( $rest as $r )
                $result[] = array_merge($item, $r);
        }
    }
    return $result;
}

```

There really are excessive parts of the PHP code like storing the final result, but more importantly copying the array each time because array\_splice takes the array argument by reference and modifies it ( talking about orthogonality ), plus its twice as long as the python code and half as readable.

Anyway, to get back at my original problem - the solution worked in terms of accuracy (at least for the first few test cases), but I fear it's going to be slow for large datasets. I have around 7 fields to compare with each respective table of the database, each table having 100 records on average; each record is 3 words long on average which gives 6 permutations per comparison. Importing a list of 1000 clients would require  $1000*7*100*6 = 4,200,000$  comparisons, plus 700,000 calls to the permutations function (not counting the recursive calls :). I still think that it's better than hammering the database with 7000 fulltext serach queries, not to mention moving the database tables to MyISAM and indexing a bunch of fields. After all, it's an import. I could put one of those useless progress indicators like when you're starting Windows.

Posted in [PHP](#), [Programming](#), [Python](#) | [No Comments](#) »

## Fun with pythonchallenge.com

admin · Tuesday, March 17th, 2009

This site is ultra cool. I should be studying for my exams, but this is bloody addictive: <http://pythonchallenge.com>

Anyway, I made it to the 10th level where you can see a bull saying:  $\text{len}(a[30]) = ?$  so obviously i need to calculate the length of the 31st member of a. Now, we don't know what a is, but fortunately, the kaw does. Clicking on it will give you this:

```
a = [1, 11, 21, 1211, 111221,
```

A sequence of some sort ... i spent like half an hour trying to decipher it and it didn't make any sense, so I did what every sane person would do: consult the [On-line encyclopedia of integer sequences](#) and it turns out it's the look-and-say sequence. How was I supposed to know that? It goes like this: first we have **ONE** one (1), so we say it: ONE ONE. That's 11. Now we have **TWO** ones so that's 21. Then we have **ONE** Two and **ONE** One (1211). Aham... So to cut things short, here's what I wrote to calculate the length of the 30th member:

```

def make_look_and_say( members = 1 ):
    seq = ["1"]

```

```

for x in range(0,members):
    new_member = ""
    char = seq[-1][0]
    streak = 1
    for c in seq[-1][1:]:
        if c == char:
            streak += 1
        else:
            new_member += str(streak) + char
            streak = 1
            char = c
    else:
        new_member += str(streak) + char
    seq.append( new_member )
return seq
print len(make_look_and_say(31)[30])

```

Now I'm stuck at level 11 and it seems that there is some image that has pixels interpolated or extrapolated, but I really cannot guess what it is. I suppose I'll read up on image libraries some other time.

Anyway, the challenge is awesome, you should try it. There really is only one python specific question (involving pickles) and everything else can be solved (up to level 10 of course :) with any other programming language .

Posted in [Python](#) | [No Comments](#) »

## Configuring Django to work with your OSX X (Leopard) apache

admin · Friday, February 27th, 2009

I hope that I finally got it right, since I can see the admin interface and the media files are being served by the same development server as the site. The machine is an Intel MacBook running OS X 10.5.6 and python 2.6.1 I suggest reading the official [Django documentation](#) on setting it with up mod\_python first. I hope that this article can fill in the gaps. Remember to change the paths and names to the ones that you use.

### Configure the virtual hosts

In this case 'mysite' is the name of the virtual host and 'my\_site' is the name of the project, and server root directory. The server root was in my /Users/discodancer/Dev/my\_site directory

```

<VirtualHost *:80>
    ServerAdmin jordanovskid@gmail.com
    DocumentRoot "/Users/discodancer/Dev/my_site"
    ServerName mysite
    ServerAlias mysite
    ErrorLog "/private/var/log/apache2/my_site-error_log" CustomLog "/private/var/log/apache2/my_site
-access_log" common
    <Directory "/Users/discodancer/Dev/my_site
">
        Options FollowSymLinks MultiViews Includes
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
    <Location "/">
        SetHandler mod_python SetEnv DJANGO_SETTINGS_MODULE my_site
.settings
        PythonHandler django.core.handlers.modpython
        PythonPath sys.path+['/Users/discodancer/Dev/']
    </Location>

```

```
# Do not use python interpreter for /media
<Location "/media">
  SetHandler none
</Location>
# Do not use python interpreter for images
<LocationMatch ".(jpg|gif|png)$">
  SetHandler None
</LocationMatch></VirtualHost>
```

Then, to allow serving of media files, you need to make a symlink from django's contrib/admin/media directory to your project. The apache user normally does not have privileges to the django installation, so you need to do this.

```
In -s /Library/Frameworks/Python.framework/Versions/2.6/lib/python2.6/site-packages/django/
contrib/admin/media/Users/discodancer/Dev/my_site/media
```

(the path is too long, try not to paste the line breaks in your terminal :) Then make a file `apache_settings.py` in your project directory/server root and paste these lines in it:

```
import osos.environ['PYTHON_EGG_CACHE'] = '/Users/discodancer/Temp'
```

The path in my case is writable by the webserver (anyone for that matter). Finally add these 2 lines in the apache `httpd.conf` file. They will tell apache to load the settings from the file you just created.

```
PythonInterpreter my_sitePythonImport /Users/discodancer/Dev/my_site/apache_settings.py my_site
```

Restart the web server. I suppose you already know, but the apache `httpd.conf` file can be found in `/etc/apache2/httpd.conf` and the virtual hosts file can be found in `/etc/apache2/extra/httpd-vhosts.conf`. This should work :) at least it did for me. One more note: at the moment of writing there is no current MySQLdb module for python 2.6. I am using the one that works with python 2.5 and each time I import it it throws a warning that the sets module is deprecated. Just ignore this, it didn't cause any trouble to me. If someone can explain what it really means, i'd be grateful.

Posted in [Django](#) | [No Comments](#) »

## Install mod\_python on Mac OS X

admin · Thursday, February 26th, 2009

This is not my article, i copied it from [here](#) for safe keeping.

First, you'll need to [grab the source](#) to `mod_python`. I recommend version 3.3.1, which is what I've worked with. Then, you'll need to unpack it:

```
$ tar xvzf mod_python-3.3.1.tar
$ cd mod_python-3.3.1$ ./configure --with-apxs=/usr/sbin/apxs
```

At that point, the configuration script will spit out a lot of things that you shouldn't really care much about. Just make sure at the end it spits out a bunch of things about creating Makefiles. From there comes the normal sequence of events with most open source software:

```
$ make$ sudo make install
```

The last requires the `sudo` command because it installs a bunch of pieces in privileged areas. Never run as root; always use `sudo` for your administrative needs. Finally, you need to add the module to your `httpd.conf` file, which is located in the `/etc/apache2/` directory, after making a back-up of course.

```
$ cd /etc/apache2
$ sudo cp httpd.conf httpd.conf.orig$ sudo vi /etc/apache2/httpd.conf
```

Then, scroll down to where you'll find all the `LoadModule` commands, and add another line:

```
LoadModule python_module /usr/libexec/apache2/mod_python.so
```

Now all that's left is to reload Apache to make sure it loads the module for you:

```
$ sudo /usr/sbin/apachectl restart
```

At that point, you're ready to proceed. The best place to start is the [mod\\_python documentation](#), specifically the [section on testing](#).

Posted in [Django](#) | [2 Comments](#) »

## Get your copy of bronze framework

admin · Wednesday, February 25th, 2009

Bronze framework is a [PHP MVC](#) framework for building web applications. It was inspired by many other frameworks that can be found around. It uses a system of internal redirects and a recursive front controller to allow the developer to reuse code as much as possible, and avoid repetition.

For now, the only thing you can get is the source and consult :) I need some time to write proper documentation.

Anyway, download [here](#), and please comment on it [here](#)

Posted in [Programming](#) | [No Comments](#) »

## Where to learn to program ?

admin · Wednesday, January 21st, 2009

This is not a blog post, more like personal notes. I'll jot these down so I don't forget them and structure them into a post later (maybe never :D).

1. Pirate sites for .pdf books
2. Colleagues (depends on where you're workin')
3. Apparently, iTunes podcasts (screencasts)
4. Coding forums ([stackoverflow](#))
5. K&R - The C Programming language :)

Now I only need to find a way to filter out and read the GIGANTIQUE amount of information I download every week. Most of it is probably stuff I already know of - for example the first few chapters of every book are about for loops and operator precedence - but the sheer amount of programming knowledge out there is simply indigestible. Maybe someday I'll try summing it all up. Or maybe that was an overly enthusiastic idea ...

Posted in [Uncategorized](#) | [No Comments](#) »

## On complicated websites

admin · Sunday, January 18th, 2009

These days I am doing a little pre-purchase research on virtual machine hosting services. The company I work for needs a testing environment for the projects in development and I figured it would be best to get a server on which we could install SVN and make a shadow copy of the most recent version directly in the server root so we can avoid uploading files manually every time we make a change.

I checked the 'internets' for some hosting options and i found out about few. One of the first hosting providers I found was [Rackspace](#). Now, i want to mention that looking for hosting companies and checking thir prices isn't exactly my favorite thing to do, and these companies aren't making my life easier. If you visit Rackspace's website, the first thing your sight lands on is some guy saying "If you're looking for a partner, you can look no furter than Rackspace". Gee, thanks, that's just what I needed, now i just need to find the subscription form. Rackspace is one of the leading hosting companies on the market and the experience fron visiting their website is almost like visiting [Godaddy](#). I see a bunch of stuff I dont care about that are only obscuring the link I really need: Pricing information for their services.

It's not rocket science to find out why a visitor is really coming to your website. He needs either **support** or **info about your merchandise (read: prices)**. If you really want to respect your customer's time the only 2 things on your website would be: A large button saying 'SUPPORT', and a nice, formatted table showing the prices on your different hosting plans. That simple.

It's not just rackspace. After visiting a bunch of websites, it turns out that almost none of the hosting companies have the info I am looking for on their home page. After a while, i got tired of clicking links and if I didn't find the pricing tables on the homepage I just started to click the back button.

The only hosting company who's website I enjoyed using so far is [Slicehost](#). For now, they are on top of my list.

Update: I'm going with Slicehost. See how a nice and clean website **DOES** get you the cash? :)

Posted in [General](#) | [No Comments](#) »

## Apple vs. Microsoft vis-à-vis Keynote vs. Powerpoint

admin · Sunday, January 11th, 2009

I always thought that MS Office is \*THE\* best office software out there. I've been using it since ever. I only tried OpenOffice while I was using linux and I have to say that although is **does** give you the most bang for the buck (infinite :) it just sucks in comparison with MS office. I never tried out the iWork office suite and when I got a Mac i installed a MS office on it (its slow and not as good as the windows version), but today I ran into the homepage of Apple's Keynote (it's the Powerpoint's counterpart) and I just loved it (the homepage). The most important thing is right there in front of you and it looks good. And the most important thing about this kind of apps is ... making a cool presentation. On the other hand, if you visit the MS Powerpoint homepage, it looks cluttered with stuff and cheap. It has ribbons in the navigation (lol!). It even has a menu link called "Product Overview", which makes it look like "just another" product. I mean ... it looks like those \$4.99 website templates.

Suppose I never made a presentation in my life and now I'm in a need of such an app. The first place to go to in search for a tool would be the internet. And after taking a look at both of the websites I'm pretty sure I'd chose Keynote. As I'm doing right now. The first thing I learned on Keynote's homepage is that you can do very very cool animations with it like "magic move" and those perspective slide transitions. And this precious information was just ONE click away in a video.

You know how my search for a similar video describing the features of Powerpoint went? Well, I'll let the screenshot speak for itself:



After all, what else could I want besides that uber-cool perspective transition and flash-like animations. Why on earth would I want to buy Powerpoint. And this is not just my impressed half talking. The other, rational half is thinking it too. Marble graph bars?! Cool! Tinted glass pie charts?! Wow! As far as I'm concerned (a presentation noob) Powerpoint is just some application that everyone's using, but Keynote is fun fun fun. I have this big urge to make a presentation. About anything. Im going out to buy the Economist and redraw all it's charts in Keynote. Srsly.

So thoughts are welcome on why would I want to use Powerpoint after visiting Keynote's website.

The links ( Even the URL is prettier on apple's site ):

<http://www.apple.com/iwork/keynote/>

<http://office.microsoft.com/en-us/powerpoint/default.aspx>

Posted in [Uncategorized](#) | [No Comments](#) »

**Hi**

admin · Wednesday, January 7th, 2009

This is the homepage of Dusko Jordanovski. For info click [here](#).

Posted in [General](#) | [No Comments](#) »