We have few mp3 players which no longer work, but are still under warranty. So idea was to pick another device (which will hopefully work longer). However, on-line shops leave a lot to be desired if you want to just do quick filtering of data.

As a very fortunate incident, I stumbled upon Exhibit from SMILE project at MIT which brought us such nice tools as Timeline and Potluck.

So, I scraped web, converted it to CSV and tried to do something with it. In the process I again re-visited the problem of semi-structured data: while data is separated in columns, one column has generic description, player name and all characteristics in it.

So, what did I do? Well, I started with CPAN and few hours later I had a script which is rather good in parsing semi-structured CSV files. It supports following:

- guess CSV delimiter on it's own (using `Text::CSV::Separator`)
- recognize 10 Kb and similar sizes and normalize them (using `Number::Bytes::Human`)
- splitting of comma (`,`) separated values within single field
- strip common prefix from all values in one column
- group values and produce additional properties in data
- generate specified number of groups for numeric data, useful for price ranges
- produce JSON output for Exhibit using `JSON::Syck`

"
So how does it look?"http://blog.rot13.org/demo/links/links.html

In the end, it is very similar to the way Dabble DB parses your input. But, I never actually had any luck importing data into Dabble DB, so this one works better for me `:-)`

This will probably evolve to universal munger from CSV to arbitrary hash structure. What would be good name? `Text::CSV::Mungler`?

This is a first post in series of posts which will cover one hack a week on my blog. This will (hopefully) force me to write at least one post a week on one side, and provide some historic trace about my work for later.