Optimizations:

* **Porting local shell program interface to website.**

**File vs. Database Approach**

* **The problem with testing local paths**
  + Some examples won't run if you open them as local files. This can be due to a variety of reasons, the most likely being that users have different directory hierarchies.
* **They feature asynchronous requests.**
  + Some browsers (including Chrome) will not run async requests if you just run the example from a local file. This is because of security restrictions (for more on web security, read Website security).
  + In the database approach, a single repository maintains data that is defined once and then accessed by various users.
    - The names or labels of data are defined once, and used repeatedly by queries, transactions, and applications.
    - Support of multiple views of the data
    - Sharing of data and multiuser transaction processing
    - Self-describing nature of a database system
    - Insulation between programs and data, and data abstraction
    - Controlling redundancy
    - Enforcing constraints (uniqueness)
* **Decoupling python code from html code**
  + Any change to the design of a webpage requires a change to the Python code.
    - The design of a site tends to change far more frequently than the underlying Python code, so it would be convenient if the design could change without needing to modify the Python code.
  + Writing Python code and designing HTML are two different disciplines, and most professional Web development environments split these responsibilities between separate people (or even separate departments).
    - Designers and HTML/CSS coders shouldn’t be required to edit Python code to get their job done.
  + It’s most efficient if programmers can work on Python code and designers can work on templates at the same time, rather than one person waiting for the other to finish editing a single file that contains both Python and HTML.
  + For these reasons, it’s much cleaner and more maintainable to separate the design of the page from the Python code itself. We can do this with Django’s template system.