

hiki 編集作業を容易にするツールの開発

情報科学科 西谷研究室 3554 山根 亮太

1 目的

もう少し広い範囲の開発目標の中で、具体的な作業として、コマンドの命名規則を見直す。

本研究の目的は、hiki のコマンド名を shell 風にすることで瞬時にコマンドの機能を判断し使用することができるようにすることである。現在 hiki でコマンドを入力する際、ユーザーは瞬間的にそのコマンドの機能や意味を判断することが難しく迷いが生じてしまうことがある。その解決策として、hiki のコマンド名を shell 風に書き換えることでこの問題を解決したい。

2 手法

2.1 hiki について

hiki とは wiki エンジンの 1 つとされており、Ruby を用いられることで作られた wiki クローンである。主な hiki の特徴として

- オリジナル wiki に似たシンプルな書式
- プラグインによる機能拡張
- 出力する HTML を柔軟に変更可能
- ページにカテゴリ付けできる
- CSS を使ったテーマ機能
- 携帯端末可能
- InterWiki のサポート
- HikiFarm に対応
- ページの追加、編集がしやすい

等がある [1]。

gem について何を書く？

hikiutils の動作内容を解説。hikiutils は gem で提供されている。CLI(character line interface) で提供されているので、そのコマンドを標準化するライブラリ optparse を利用する。

CLI の option の parse を簡単に扱うために Ruby にはいくつかの library が用意されている。

評価。観点。optparse

3 開発要件

Todo アプリケーションの構造は lib/todo/command/options.rb にサブコマンド用の定義があり、入力されたコマンドは lib/todo/command.rb で判断されることで lib/todo/command/options.rb 内で実行される。hikiutils の構造は Todo アプリケーションの構造と

同じで hiki の編集を容易にするためのツール群のため、この部分の変更を行うことでコマンド名を書き換える。

4 考察

現在は Todo アプリケーションの作成を行うことでプログラムの構造の理解をしている。しかし、まだ hikiutils のコマンド名の変更までには達していない。今後の課題としては、誰が見てもコマンドの機能が判断できるコマンド名を考え、実際に hikiutils 内でそのコマンドを実装する必要がある。

参考文献

[1] hiki サイト。