The Atmark

Release 0.7.2

Sep 27, 2017

Contents

CHAPTER 1

The Atmark

User's Guide

This part of the documentation will show you how to get started in using The Atmark.

Contents

Requirements

- python >= 2.6

Installation

The Atmark should be installed using pip:

```
pip install atmark
```

Bash completion

Atmark supports bash completion. Just add this lines to your *.bashrc*:

```
_atmark_complete() {
    COMPREPLY=( $( COMP_WORDS="${COMP_WORDS[*]}" \\
                COMP_CWORD=$COMP_CWORD \\
                _ATMARK_COMPLETE=complete $1 ) )
    return 0
}

complete -F _atmark_complete -o default @ @@;
```

You can easy do it with command:

```
@ -bs >> ~/.bashrc
```

Quickstart

Contents

The Atmark – Awk+sed for humans.

Do this:

```
# Rename a files in current directory (file-name -> file_name.jpg)
$ ls | @ sub - _ "mv # @.jpg" | sh
```

Except this:

```
# Rename a files in current directory (file-name -> file_name.jpg)
$ ls | awk '{print $1 $1}' | sed s/"-"/"_"/ | awk '{print "mv", $2, $1, ".jpg"}' | sh
```

More deep:

```
$ ls | @  sub  -  _  "mv # @.jpg" | sh
          ---  |  |  ------------
           |   |  |      \_ format string (# - link on first state,
           |   |  |                        @ - link on current state (after replace))
           |   |  |
           |   |   \_ second replace param (to replace)
           |   |
           |    \_ first replace param (what replace)
           |
            \_ function name (substitute)
```

More examples:

Change file's extension .html > .php

```
# Atmark
$ ls | @ split . head "mv # @.php"
```

```
# Awk/Sed
$ ls | awk '{printf "mv "$0; sub(/html/,"php"); print " "$0}' | sh
```

Print all but the first three columns

```
# Atmark (\t means tab)
$ ls -la | @ split_ drop 3 join \t

# Awk/Sed
$ ls -la | awk '{for(i=1;i<4;i++) $i="";print}'
```

Kill process by name

```
# Atmark
$ ps aux | @ grep sysmond$ index 2 "kill @" | sh

# Awk/Sed
$ ps aux | grep [s]ysmond | awk '{print "kill "$2}' | sh
```

And more, more, more.

Usage

Get help

```
$ @ -h

Atmark (@) -- is a command line utility for parsing text input and generating output.

You can pipe data within a Atmark (@) statement using standard unix style pipes ("|").
Provide for Atmark function composition and let them work for you.

Example. Replace "_" with "-" in files in current dir and change the files extensions␣
↪to jpg:

    $ ls | @ replace _ -  split . "mv # @.jpg"

It is mean:

    $ ls > replace($LINE, "_", "-") > split($RESULT, ".") > format($RESULT, "mv $LINE
↪$RESULT.jpg")

You can use "@ --debug ARGS" for debug Armark commands.

===================================================================================
LIST OF THE BUILT IN FUNCTIONS


format FORMAT_STRING         -- format and print a string.

    Symbol '@' in FORMAT_STRING represents the current value in process of␣
↪composition of fuctions.
    Symbol '#' in FORMAT_STRING represents the history state.
        Where   # or #0 -- first state, #<n> (#1, #2) -- state with number n

    Synonyms: You can drop `format` function name. This lines are equalent:
        $ ls | @ upper format "@.BAK"
        $ ls | @ upper "@.BAK"

capitalize/cap       -- capitalize the string.

drop N               -- drop N elements from list/string.

equal/== PATTERN     -- return None if arg is not equal to PATTERN.

filter/if            -- filter results by value has length

head/h               -- extract the first element/character of a list/string

index/ix N           -- get the N-th element/character from list/string.

join/j SEPARATOR     -- concatenate a list/string with intervening occurrences of␣
↪SEPARATOR

join_/j_             -- same as join but SEPARATOR set as ' '

kill REGEXP          -- replace in a string/list REGEXP to ''.

last                 -- get last element/character of incoming list/string.

length/len           -- return length of list/string.

lower/l              -- make the string is lowercase

map FUNCTION         -- apply the following function to each element/character in␣
↪list/string.

nocolor/nc           -- Remove ansi colors from string.

notequal/!= PATTERN  -- return None if arg is equal to PATTERN.

notgrep/ng REGEXP    -- filter results by REGEXP. Leave ungrepped

replace/r/sub FROM TO -- replace in a string/list FROM to TO.

reverse/rev          -- reverse list/string.

rstrip/rs/rtrim PATTERN -- return the string with trailing PATTERN removed.

grep/g REGEXP        -- filter results by REGEXP

sort                 -- sort list/string.

split/sp SEPARATOR   -- return a list of the substrings of the string splited by␣
↪SEPARATOR

split_/sp_           -- same as split by splited a string by whitespace characters

strip/s/trim PATTERN -- return the string with leading and trailing PATTERN removed.

strip_/s_/trim_      -- same as strip but trims a whitespaces.

tail/t               -- extract the elements after the head of a list
```

```
take N               -- take N elements from list/string.
upper/u              -- make the string is uppercase.
```

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

API

Contents

This part of the documentation documents all the public classes and functions in The Atmark.

Note: In progress

CHAPTER 2

Bug tracker

If you have any suggestions, bug reports or annoyances please report them to the issue tracker at https://github.com/klen/atmark/issues