Todoian Documentation

Release 1.0

IFinners

Oct 17, 2018

Contents

CHAPTER 1

Introduction

Todoian lacks a GUI but doesn't lack much in terms of features with simple, intuitive commands allowing you to take control of not only your Tasks but your longer term Goals as well. Some of the more notable features include:

- Chronological organisation of Tasks means a quick glance is enough to ascertain what you should have already done, what you need to do today and what awaits you in the future
- Display as much or as little as you want with commands ranging from viewing a single Goal to viewing all overdue, due today and future Tasks
- Break down Tasks and Goals into more managable chunks or clearer steps with Subitems
- Track your progress towards a Goal's completion with a progress bar that can be set manually or calculated automatically from your Subgoal completion
- Add highly customisable repeats to your Tasks
- Tag Tasks and Goals with an unlimited number of keywords

CHAPTER 2

Getting Started

Simply download the Todoian Repository to any folder you like and run the todoian.py file from the Command Line using

```
python3 todoian.py
```

CHAPTER 3

Usage Instructions

3.1 General Usage Information

3.1.1 Synonyms

All commands have shorter synonyms, for example ‘a’ can be used instead or ‘add’, and once a synonym is mentioned one can assume it works for any given example even if not explicitly stated.

3.1.2 The Two Types of Optional Arguments

Optional arguments are shown by square brackets ([]) or curly brackets ({}) throughout this command guide. Arguments in square brackets, if not given, will be asked for through a seperate prompt before the operation can be completed whereas ones in curly brackets will be automatically set to a default - for example:

```
add ["New Task Description"] {Date} {Repeat}
```

The user will be prompted for a task description if one isn’t provided but both the Date and the Repeat will be set to their default values - today’s date and no repeat respectively.

3.2 Task Commands

Ordered by their due dates and customisable with repeats, tags and Subitems(see *Subitem Commands* for details) Tasks are what Task Managers… well… manage and here is a breakdown of how to do just that.

3.2.1 Adding

(‘a’, ‘t’, ‘add’, ‘task’)

```
add New Task Description
```

Optional keywords of date, repeats and tags can be added using the following format:

```
keyword=value
```

To set the date add the Task with the keyword ‘date’ or ‘d’:

```
add Task date=2018-01-01
```

- Note that he date must be in the format detailed in the *Changing a Due Date* section.

A repeat can be added with the keyword ‘repeat’ or ‘r’:

```
add Task repeat=mon, wed, sat
```

- Note that the repeat must be one of the options detailed in the *Adding a Repeat* section and spaces are optional.

A tag, or tags, can be added in the same way with the keyword ‘tag’ or ‘t’:

```
add Task tags= tag1, tag2, tag3
```

- Note that the list of tags, like the list of repeat days above it, can have spaces between the items or between the = and the first item or not.

All of these options can be combined into one addition by combining them:

```
add Task with Everything d=2018-01-01 t=tag1, tag2 r=7
```

- Note that the keywords can be in any order but the Task title must come first

3.2.2 Completing

(c, comp, complete)

Marking a Task as complete moves it from the active Task list to an unseen cache, unless it has a repeat in which case it is added back onto the Task list with the appropriate new information.

```
complete task-number
```

All Tasks with due dates equal to the current date are marked as complete by writing ‘t’ or ‘today’ instead of a Task number:

```
complete today
```

All Tasks with due dates earlier than the current date are marked as complete by writing ‘o’ or ‘overdue’ instead of a Task number:

```
complete overdue
```

Note that this completes overdue tasks with repeats until their due date is no longer overdue.

3.2.3 Deleting

(d, del, delete)

Deleting a Task is similar to completing one in that it moves the Task from the active Task list to an unseen cache. However, with deletion this is done regardless of any repeat flags.

```
delete task-number
```

All Tasks can be deleted by writing ‘a’ or ‘all’ instead of a Task number - a prompt will ask for confirmation:

```
delete all
```

3.2.4 Undoing a Completion or Deletion

As long as the program hasn’t been exited since a completion or deletion was made, the items can be restored to their previous state using the commands:

undo-comp, uc - for undoing a completion

undo-del, ud - for undoing a deletion

Both of these commands will work for the restoration of multiple items if done repeatedly

3.2.5 Moving

(m, move)

Tasks are sorted by their due dates, and whilst this cannot be changed, the order of Tasks within their date brackets can with the following command:

```
move task-number move-number
```

Where the move number is the position to move the Task to in the list.

3.2.6 Editing a Description

(e, edit)

Editing a Task’s description can be done through the following command:

```
edit task-number [new Task description]
```

3.2.7 Changing a Due Date

(ed, edit-date)

Changing a Task’s due date can be done through the prompt that follows the following command:

```
change-date task-number [due date]
```

The date must be entered as:

- A date formatted like YYYY-MM-DD e.g. 2018-01-25:

3.2.8 Adding a Repeat

(r, repeat)

Repeats allow Tasks to be automatically re-added to the Task list upon completion. The repeat can be set with the following command:

```
repeat task-number [repeat]
```

There are two types of repeat that can be set. The simplest of these is the number of days repeat - for example setting the repeat to the value 7 will result in a Task that repeats weekly.

Another way to specify a repeat is through a three letter day name or a list of day names (of any length) seperated by a comma:

```
repeat task-number [mon,wed,fri]
```

This Task would repeat every Monday, Wednesday and Friday.

To remove a repeat, simply do the above but set the repeat to ‘none’:

```
repeat task-number [none]
```

3.2.9 Adding a Tag

(tg, tag)

Tagging a Task with a keyword means it can be displayed with other Tasks and Goals (see the Display Command section of this guide) that share that tag. To add tag(s) to a Task, enter the following command:

```
tag task-number [tag,tag2,tag3]
```

3.2.10 Deleting a Tag

(dt, delete-tag)

A specific tag can be deleted from a Task by using it as the keyword in the command to follow, or all tags for that Task can be deleted by using the keyword ‘all’:

```
delete-tag task-number [keyword]
```

3.3 Goal Commands

Goals have many of the commands that Tasks do and are invoked in the same way with the command slightly modified with either the letter ‘g’ or the hyphened word ‘goal’.

3.3.1 Adding

(‘ga’, ‘g’, ‘goal’, ‘add-goal’)

```
goal New Goal Description
```

Optional keywords of date and tags can be added with keywords using the following format:

```
keyword=value
```

To set the date add the Goal with the keyword ‘date’ or ‘d’:

```
ga Goal date= Sometime Next Month
```

- Note that the date can be anything and the space between the = and the start of it is optional.

A tag, or tags, can be added in the same way with the keyword ‘tag’ or ‘t’:

```
ga Goal tags=tag1, tag2, tag3
```

- Note that the spaces in the list of tags, like the list of repeat days above it, are optional

All of these options can be combined into one addition simply by having a space between the keywords:

```
add Task with Everything d=This Month t=tag1, tag2
```

- Note that the keywords can be in any order but the Goal title must come first.

3.3.2 Completing

(‘gc’, ‘goal-comp’, ‘goal-complete’)

Marking a Goal as complete moves it from the active Goal list to an unseen cache.

```
complete goal-number
```

3.3.3 Deleting

(‘gd’, ‘goal-delete’)

Deleting a Goal is identical to completing one in that it moves the Goal from the active Goal list to an unseen cache. The only real difference comes when retrieving the Goal from the cache and the lack of accomplishment one feels with a deletion.

```
delete goal-number
```

All Goals can be deleted by writing ‘a’ or ‘all’ instead of a Goal number - a prompt will ask for confirmation:

```
delete all
```

3.3.4 Undoing a Completion or Deletion

As long as the program hasn’t been exited since a completion or deletion was made, the items can be restored to their previous state using the commands:

guc, goal-undo-comp - for undoing a completion

gud, goal-undo-del - for undoing a deletion

Both of these commands will work for the restoration of multiple items if done repeatedly

3.3.5 Moving

(gm, goal-move)

By default goals are sorted by when they were added but they can be moved at will with the following command:

```
goal-move goal-number move-number
```

Where the move number is the position to move the Task to in the list.

3.3.6 Editing a Description

(ge, goal-edit)

Editing a Goal's description can be done through the following command:

```
edit goal-number [new Goal description]
```

3.3.7 Changing a Due Date

(ged, goal-edit-date)

Changing a Goal's date - a target for when you wish to complete it by so it can be more general than a specific day - can be done with the following command:

```
change-date goal-number [due date]
```

3.3.8 Adding a Tag

(gtg, goal-tag)

Tagging a Goal with a keyword means it can be displayed with other Tasks and Goals (see the Display Command section of this guide) that share that tag. To add tag(s) to a Goal, enter the following command:

```
goal-tag goal-number [tag, tag2]
```

3.3.9 Deleting a Tag

(gdt, goal-delete-tag)

A specific tag can be deleted from a Goal by using it as the keyword in the command to follow, or all tags for that Goal can be deleted by using the keyword 'all':

```
gdt goal-number [keyword]
```

3.3.10 Changing a Percentage

(gp, goal-percentage) Beneath a Goal there is the option to have a progress bar indicating how close the Goal is to being complete - this is where the percentage comes in. By default, percentages are set to 'auto', allowing the number of Subgoals completed and still to do determine the percentage completion, but custom percentages can be added using the following command:

```
goal-percentage goal-number [percentage]
```

Note that in order to restore the default 'auto' percentage setting one simply has to enter 'auto'(without the quotes) as the value either on the command line or at the prompt and to disable the progress bar simply enter 'none' as the value.

3.4 Subitem Commands

Both Tasks and Goals can be broken down into smaller parts through the use of Subitems. The functionality is identical for Tasks and Goals but the commands vary slightly as indicated beneath each section below.

3.4.1 Adding

(s, subtask) (gs, subgoal)

```
subtask task-number [subitem description]
```

3.4.2 Completing and Undoing Completion

(ts, toggle-sub) (gts, toggle-subgoal)

Marking a Subitem as complete strikesthrough the subitem when displayed and updates a Goal's progress bar if percentage is set to 'auto'. This is done and undone by the toggle command.

```
toggle-subgoal goal-number [subgoal-number]
```

All Subitems under a specified Task or Goal can be marked as complete ('done') or incomplete ('todo') using the toggle commands:

```
toggle-sub task-number ['done']
```

3.4.3 Deleting

(ds, delete-subtask) (gds, delete-subgoal)

Unlike the deletion of a Task or Goal, the deletion of a Subitem is permanent even prior to closing Todoian.

```
delete task-number [subtask-number]
```

3.4.4 Moving

(ms, move-sub) (gms, goal-move-sub)

Subitems can't be transferred to another item, but their order beneath the Task or Goal can be changed.

```
move-sub task-number subtask-number new-position
```

3.4.5 Editing a Description

(es, edit-sub) (ges, edit-subgoal)

```
edit-sub task-number subtask-number [new subtask description]
```

3.5 Display and Miscellaneous Commands

Todoian customises its display based upon the command it has just been given, but the various types of display can also be invoked through their own set of commands.

3.5.1 The List Command

The main display command is the list (ls, list) command which with no modifiers will print all Tasks organised by their Due Dates:

```
list
```

The following modifications to the list command, and their effects are… View all tasks with due dates of today with (t, today):

```
list today
```

All overdue tasks with (o, overdue):

```
list overdue
```

Task's due tomorrow with (tm, tomorrow):

```
list tomorrow
```

Task's due tomorrow and beyond with (f, future):

```
list future
```

All Goals with (g, goals):

```
list goals
```

By default, Goals are displayed without their Subgoals. Those with Subgoals are indicated by a trailing tilde ('~'). Goals, and all of their Subgoals, can be viewed with (gs, goals-subs):

```
list goals-subs
```

All Goals (without Subgoals) and Tasks can be viewied with (a, all):

```
list all
```

Any Task or Goal with a specific Tag can be viewed with (tg, tag):

```
list tag [tag]
```

All Goals and Tasks that have tags can be displayed with their tags listed beneath them with (tgs, tags):

```
list tags
```

3.5.2 The View Command

Long-term goals can often have a lot of Subgoals. A single goal with all of its Subgoals can be viewed with (vg, view_goal)

```
view-goal goal-number
```

3.5.3 Other Commands

A backup of the Task and Goal data can be created with the commands (bkup, backup):

```
backup
```

A link to this documentation can be displayed within the program with (h, help)

```
help
```