Workflow Reference Sheet

Why?

A new way to work, a new habit, is difficult to form without reminder. Whence this brief cheat sheet to get me going on this GTD process.

Overview of the day

Start of the day: ~1 hour

- ◇ Look at mail with `M-x gnus` and capture tasks indiscriminately *unless* marked urgent.
- ◇ Quickly review upcoming week in the agenda, ensuring you're working on the right stuff –shuffle if need be.
- ◇ If there are many `C` tasks, reschedule without a second thought. Aim for 15 tasks.
- ◇ Progress towards `A` tasks completion by documenting work completed.

A *good day* is doing *at least* all of the most important and urgent tasks –the `A` tasks.

End of the day:

- ◇ Journal about what was completed and what was not.
- ◇ Process, schedule, and archive tasks.

`C-c c`	Capture an idea quickly
`C-c a`	Open agenda to see tasks
`C-c a t`	Compile global todo list
`C-c a u`	See unprocessed tasks
`C-c C-s`	Schedule task
`C-c a c`	See completed tasks
`C-c C-x C-s`	Archive task
`(org-search-view)`	Search todo's and archive

Throughout the day: When new tasks come up, enter them impulsively and indiscriminately. Do not make any further scheduling or processing considerations.

Progressing Towards Completion

Can't manage what you don't know!
Where does the time go?
Log it & document it!

Clock in on a heading with `I`, or in the subtree with `C-c C-x C-i`.

- ◇ An org-timer for 25 minutes starts, in case the task is dreadful.

Clock out of a heading with `O`, or in the subtree with `C-c C-x C-o`.

- ◇ When we clock-out, a note of what's accomplished is added to the task.
- ◇ Even if little, the note motivates progress.
- ◇ Continuous journaling is akin to literate programming —no one wants to do it after the fact.

Other ideas to assist in completing a task:

- ◇ Add an arbitrary remark to a task with `C-c C-z`.
- ◇ Move the state of a task with `C-c C-t`.
- ◇ Focus on a particular region when working to avoid being distracted. `C-x n n/e/s/w` for narrowing to region, org element or subtree, or widen.
- ◇ From anywhere, `C-u C-c C-x C-i` yields a pop-up for recently clocked in tasks.

Reflection; Journaling

`C-c j`	Discuss what's accomplished in journal
`C-c C-j`	New entry into journal
`C-c C-s`	Search the journal

- ◇ What was accomplished today?
  - ○ See clocked times with `C-c C-x C-r`.
- ◇ What are some notably bad habits? Good habits?
- ◇ What are some future steps?

Processing Tasks

Within the list of captured items, for each item we do as follows.

1. Not actionable? Refile to trash, someday/maybe, or reference sections.
2. Multiple steps to completion? Make it into a project.
3. Less than 2 minutes? Do it now.
4. Is someone else supposed to do this? Delegate!
5. If it has a specific date, schedule it; otherwise tag it with context needed to get it done.

How to 'Schedule' a Task

1. Provide an estimated time effort.
2. On heading, press `,` then one of `A, B, C` to set its priority.

A	High urgency & important; rare.
B	Moderate urgency & importance; most tasks.
C	Pretty much optional, or very quick or fun to do

3. Pick a date to get it done with `C-c C-s`, then `+n` to have it be started `n` days from now.
   - ◇ Any time no time.
   - ◇ Always schedule *then* reschedule, if needed.

Working with 'Projects'

If a task needs less than 2 minutes, do it immediately since the processing overhead is not worth it; otherwise, it's a 'project' and should be split up into multiple tasks.

1. Append the project's name with `[/]` or `[%]` to get progress statistics.
2. Produce a list of sub-tasks, in the form `- [ ] subtask1`, that progress the project to completion.
3. Test toggle a task into completion with `C-c C-c` on it, now the statistics are updated. If you manually toggle a task, use `C-c #` on the statistics to update them.

At the top, for the project, add a single sentence description of a successful outcome, defining 'done' and clarifying the desired outcome.