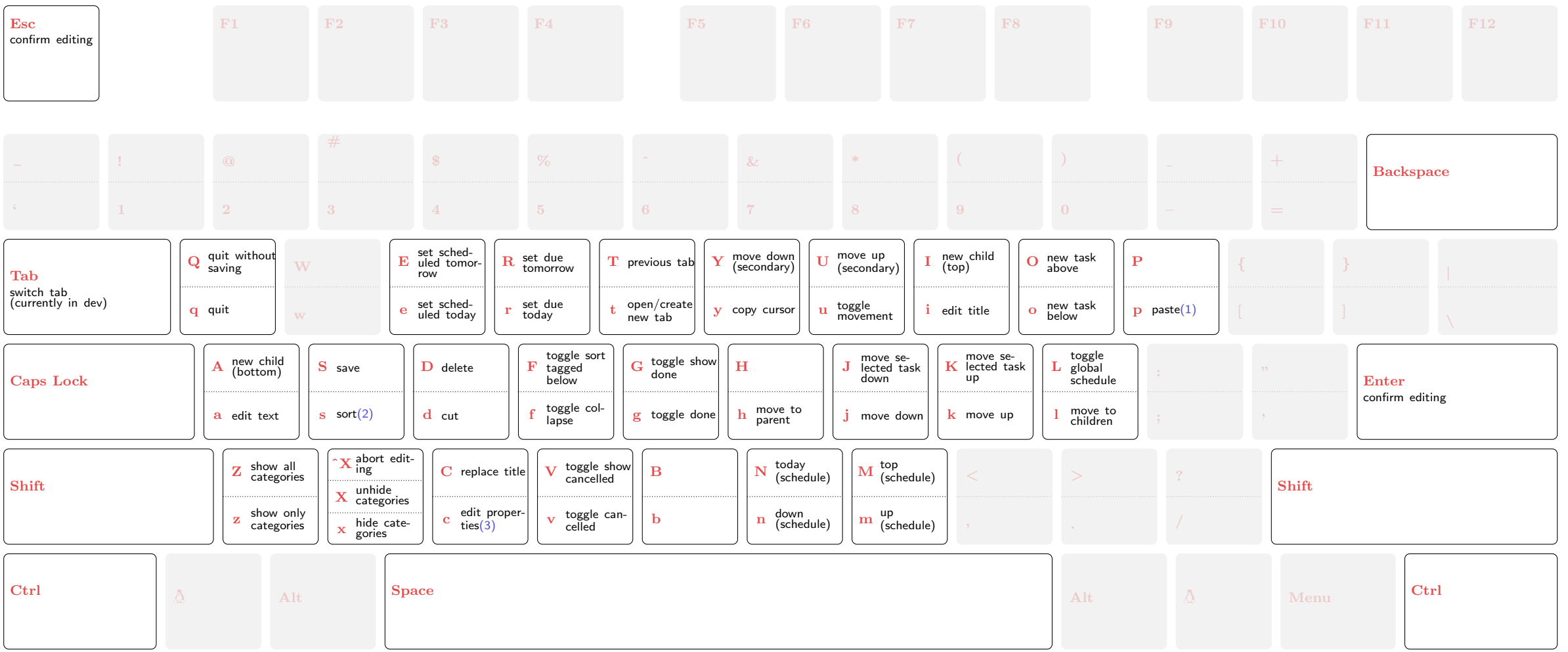
Treetasks cheatsheet



Paste (1)
Keychains for pasting:

- `pp` Paste after cursor
- `pP` Paste in front of cursor
- `po` Paste as child, after other children
- `pO` Paste as child, in front of other children

Sort (2)
Keychains for reordering the tree. The "natural" order defined by moving around tasks is always saved. Replace the second letter of the keychain by its corresponding capital to order in reverse.

- `ss` Restore natural order
- `st` Sort by title
- `sp` Sort by priority
- `sc` Sort by category
- `sr` Sort by due date
- `se` Sort by scheduled date
- `sd` Sort by whichever date is earlier (like the schedule)

Edit (3)
Keychains for editing tasks.

- `cp` Edit priority
- `cc` Edit category
- `cr` Edit due date
- `ce` Edit scheduled date

Without using keychains, the rest of the properties can be edited:

- `i` Edit title
- `C` Replace title
- `a` Edit description text

Movement
You can traverse the tree in two ways:

- **Flat:** The next/previous tasks are the tasks on the next/previous line, regardless of hierarchy
- **Hierarchically:** Traversing only traverses the siblings, visiting children/parents need explicit operations.

Treetasks always keeps both ways at your hands: there is a primary and a secondary mode of movement, the primary is mapped on the well-known `j` and `k` keys per default, and the secondary is mapped on `Y` and `U` keys per default. By pressing `u` (default) the primary and secondary modes are swapped. The default order can be set by the config variable `behaviour.primary_movement_hierarchic`. When `True` the primary mode is hierarchical movement, when `False` the primary mode is flat movement.
For making hierarchical movement more practical there is the option `behaviour.auto_move_up` which when set to `True` (default) moves the cursor as in flat movement for one step when reaching the end of a set of children.
When `behaviour.roundtrip` is set the cursor can go around the ends of the list. If `behaviour.auto_move_up` is not set, children are also traversed in roundtrips.

Schedule
The schedule shows tasks that are not marked as done or cancelled and have a scheduled date or due date assigned. They are sorted by the earlier date of those two. Tasks whose scheduled date or due date lies in the past are printed in red, tasks that are scheduled or due today are printed in yellow.
The schedule has its own independent cursor which is controlled by `n` and `m` by default. You can directly skip to the top using `M` and to today using `N`.

- `behaviour.follow_schedule`: If set to `True` cursor movements in the schedule result in the tree cursor jumping to the task selected in the schedule (only if the task is currently shown in the tree). Default: `True`.
- `behaviour.filter_categories_schedule`: If set to `True`, the schedule is filtered by the same category filters that apply to the tree. Default: `False`.
- `behaviour.global_schedule`: If set to `True`, the schedule is generated from all tasks from all open tabs. If set to `False`, every tab has its own schedule. If a global schedule is used, a + sign denotes tasks that are not in the currently shown tab. If the cursor follows the schedule cursor, tabs are automatically switched, too. Default: `True`.