

```
--:! {planner: []: 😊} <- Functions Library to Build a Plan -> muse/docs/lib/planner.md
--😞 planner: Given a plan, create a table of operations to be performed by worker.execute. ->
planner, plan, moves, steps

--:> plan: How to do work -> {name: plan.name, path: plan.path, work: plan.work,
fixtures: plan.fixtures, mark: plan.mark}

--:> plan.name: for status and error reporting -> ":"

--:> plan.path: table of space separated character sequence strings describing path -> ":"[]

--:> plan.work: for execution at every step iteration in plan movement direction -> (:plan:,
direction: ":"): ":"?

--:> plan.fixtures: for placement as specified by path elements beginning with a digit -> ":"[]

--:> plan.mark: for execution as specified by plan.path markers -> (:plan:, :marking:):
markerName: ":", label: ":", report: ":"

--:> marking: tuple table of marker parts -> :[prefix: ":", base: ":", label: ":"]

--:> markings: dictionary of markings keyed by a label -> [label: ""]: marking

--:# Plan elements beginning with a letter indicate stepped movement in one of six
directions: u, d, n, e, s, or w.
--:+ Optionally, the letter may be followed by a number of blocks for the movement. The element ends
with a space.
--:+ Elements beginning with a digit (indexing plan's fixtures table) are followed by a letter specifying
direction.
--:+ Plan elements beginning with a colon indicate placement of a mark, a named and labelled
place.
--:+ The plan.mark function might replace each colon in a marker string prefix with, for example, a
shaft name and level.
--:+ Characters between the last colon and a "|" vertical bar character are the base argument for
plan.mark.
--:+ Characters following the "|" are the marker label for plan.mark.

--:: planner.load(planFileName: "😊" -> Instantiates what is returned from a plan file. -> plan &!

--:: planner.make(plan:plan) -> Create path operations table for plan. -> pathElements, fuelOK:
^:, pathDistance: #:

--:> pathElements: Used by worker.execute to run plan ->
(stepElement|putElement|markElement)[]

--:> stepElement: Iterate steps function in direction for distance -> :[op: "step", :stepping:,
direction: ":", distance: #:]

--:> putElement: Put fixture in specified direction -> `[op: "put", direction: ":", fixture: ":"]`
```

2025-11-25

--:> markElement: *Current situation in named places* -> :[op: "mark", :marking:]

--:# **Marker parsing utility function**

--:: planner.mark(markerName: "😊" -> *Parse marker name into parts.* -> shaft: ":"?", level: ":"?", tag: ":"?)"