

7 temp slave

i have no idea how to start articulating an answer to the question of the previous section, but the zine can't end here. a good way of carrying on may be to look into a text from the zine 'temp slave'.

it may be said that one of the premises of that zine was a tactical approach to temporary, odd jobs. from it, i will take the cue for the tactical approach to coding of the next sections. the text at hand is called 'temporary insanity.' it proposes detailed, cold, detached classification as a way to not go crazy inside a white cubicle:

emotionless action and emotionless reaction is the solitary buffer between your sanity and your soul [...] if you forget this principle, you begin to revel in the act and the outcome of the organizational process.

post-its get stacked according to 'the laws of diffracted light as exhibited by the rainbow'; productivity is increased by meticulously re-organizing the file cabinets; the filesystem of the computer is cleaned and the temporary worker hides behind the monitor playing the version of tetris that was found; etc.

the text ends on a sad note. the temporary worker forgets the principle of detachment s/he spoke about, and tries to see some sense in the inventory. it's fitting that the arbitrariness of the encoding starts to manifest in front of the computer:

you despise the random pattern of stars and have done away with the frivolity of screen savers. the screen of your computer terminal is clear and black.

a few lines after the temporary worker has a temporary crash. likewise, this would be a good point for my computer to crash.

9 print function

the idea of print that i'm trying to push feels very natural, at least to me. this is the reason why i'm surprised to see how much in common it has with the usual print functions of the computer languages.

the perlfunc man page says that 'print prints a string or a list of strings.' that is, print and strings are intimately tied in perl. with regards to strings, the introductory book 'learning perl' says that strings are sequences of characters, typically plucked from the ascii 32 to ascii 126 range. this means that printing in perl is very often made up of the same thing that text is made of.

but then there are things that we would have never thought of printing had it not been for the print function. browsing through the index we get at least two ideas, printing time and printing databases:

3.8. printing a date 11.11. printing data structures print function, 29 databases records and, 227

perhaps more interesting is the possibility for a text to be empty, and for there to still be printing. printing and text can be null. null is as natural as whatever: trees, squirrels, bees, and null: 'the shortest possible string has no characters' (learning perl, page 22).

but basically the idea of print in perl is to show the world your stuff, much like print matter is used. in print culture, this happens on paper, most of the time. likewise, in perl there's a standard medium for print namely the so-called 'standard output'. this often means an aggregation of linebreaks: a terminal.

something else should be written about text processing: reading a file backwards by line or paragraph, trailing a growing file, randomizing all lines.