



```

> (setq jds-search-regexp ".")
> ;; This is what JDS had.
> ;; (setq jds-search-regexp (concat "^" (regexp-quote str)))
> (setq jds-search-regexp (regexp-quote str)))
> (add-hook 'pre-command-hook 'jds-search-pre-command-hook nil t))
> (comint-previous-matching-input jds-search-regexp dir))
>
> (defun jds-history-search-up ()
> (interactive)
> (jds-history-search 1))
>
> (defun jds-history-search-down ()
> (interactive)
> (jds-history-search -1))

```

This is just a subset of the command line searching already built into IDLWAVE (courtesy of COMINT mode). I use M-r to search back using any regular expression, C-c C-l to show a selectable list of prior commands, middle click on any old input line in the shell to paste it to the current, etc.

As for Matthew's customization, recent versions of Emacs have a new command C-c M-r to do just what this .emacs stuff does (so it's no longer needed). For an easier binding, I'd probably stay away from the up arrow for this (multi-line editing is built-in to the coming IDLWAVE release), but might try:

```

(add-hook 'idlwave-shell-mode-hook
  (lambda ()
    (local-set-key [M-up] 'comint-previous-matching-input-from-input)
    (local-set-key [M-down] 'comint-next-matching-input-from-input)))

```

to have Meta-up and Meta-down do this. Note that it works on input up to the cursor, so position your cursor on a long command to cap the section you want to search, and fire. Successive searches will do the right thing.

JD

P.S. IDLWAVE is now working with the upcoming IDL help system, with direct linking to keyword help sections, etc.