|  |  |
| --- | --- |
| TEA PRIMITIVE | SEMANTICS |
| H: | |  |  | | --- | --- | | NAME | Hew | | PURPOSE | Explode or split up AI using something | | SYNTAX  & SEMANTICS | |  | | --- | | h: | | Split up AI such that all its elements are separated by a single space. In effect, padding the inner contents of AI with space. That becomes IO. | | h:REGEX | | Same as h:, but the splitting happens only at the beginning of where the contents of AI match REGEX. The original contents of AI remain preserved by this operation. | | h!: | | Same as h: but instead using the New Line character as separator. | | h!:REGEX | | Same as h!:, but the splitting happens only at the beginning of where the contents of AI match REGEX. | | h\*: | | INERT | | h\*:vNAME:vREGEX  h\*!:vNAME:vREGEX | | The vault operating versions of h: and h!: perform corresponding transformations, but instead operate on the data stored in the vault with the specified vault name, vNAME. | |  |  | | NOTES | Unlike g:, h: is meant to help with fragmenting things. H: splits the AI by the Empty String, essentially seeming as though it has exploded, then rejoins each character to the next using a single space or new line (for the inverse). Some examples follow:  i!:{ABC}| h:  #(=“A B C”)  Note that for a string of length N, h: will return a string of length 2N-1. Some examples follow…  i!:{123}| h!:  #(=  “1  2  3”)  i!:{http://127.0.0.1/path}| h:[\/\.]  #(=“http: / /127 .0 .0 .1 /path”)  i!:{1234567890}|v:vIN|v:vHEW:[02468]|h\*:vIN:vHEW  #=1 23 45 67 89 0 |  | |