|  |  |
| --- | --- |
| TEA PRIMITIVE | SEMANTICS |
| G: | |  |  | | --- | --- | | NAME | Glue | | PURPOSE | Bind elements in AI using something | | SYNTAX  & SEMANTICS | |  | | --- | | g: | | Reduce AI to a string with all whitespace characters removed. Essentially binds any word in AI to all the rest. | | g:GLUE  g:GLUE:REGEX | | Set the IO to the result of reducing AI to a string where all instances of REGEX have already been replaced by GLUE. Without the second parameter – REGEX, g: merely replaces all whitespace in AI with GLUE. | | g!: | | INERT | | g.: | | Reduce AI to a string with all NEW LINE characters removed. Essentially helps reverse h!: | | g.:GLUE | | Reduce AI to a string with all NEW LINE characters replaced by GLUE. Essentially helps reverse h!:REGEX | | g!:GLUE | | Reduce AI to a string where all standard sentence punctuation marks, in addition to all whitespace, have been replaced with GLUE. GLUE is expected to be a string. The result of this transform then becomes IO. | | g\*:GLUE:v1:v2  g\*:GLUE:v1:v2:v3:…:vN | | Set IO to the result of joining the string stored in the vaults with the given names, using the specified EXPLICIT GLUE | | g\*!:vGLUE:v1:v2  g\*!:vGLUE:v1:v2:v3:…:vN | | Set IO to the result of joining the string stored in the vaults with the given names, using the glue stored in vault vGLUE | |  | | NOTES | G: is the most magical primitive in TEA. g: for example automagically sucks all space out of things and sets them aside for later use! Let’s look at an example:  i!:{BC CB BA AB}| g:  #(=BCCBBAAB)  i!:{BC CB BA AB}| g:{\_\*\_}  #(=BC\_\*\_CB\_\*\_BA\_\*\_AB)  That’s not without power. On the other hand, g!: can best be appreciated with examples processing regular human-readable text---which, is for example expected to have natural use of both whitespace and punctuation marks in it. Thus, the program  i!:{Which of this,  that or both do you want?  None} | g!:{\*}  Should return “Which\*of\*this\*\*that\*or\*both\*do\*you\*want\*\*None”  Other examples with their expected outputs shown in the comments:  i!:{BC CB BA AB}| g:{\_\*\_}:.[BC] #= \_\*\_\_\*\_B\_\*\_A \_\*\_  i!:{BC CB BA AB}|v:vIN|v:vP:---[|v:vS:]--|v:vG:{\_}|g\*!:vG:vP:vIN:vS #=---[\_BC CB BA AB\_]-- |  | |