|  |  |
| --- | --- |
| TEA PRIMITIVE | SEMANTICS |
| X: | |  |  | | --- | --- | | NAME | Xenograft | | PURPOSE | Affix things to things | | SYNTAX  & SEMANTICS | |  | | --- | | x: | | Return AI with AI affixed to itself. Essentially, multiplying AI | | x:PREFIX | | Prefix PREFIX to AI. That becomes AI | | x!: | | Return Half of AI. Essentially, reducing AI | | x!:SUFFIX | | Affix SUFFIX to the end of AI | | x\*:vPREFIX  x\*:vPREFIX:vSTR | | Prefix the string in vault vPREFIX to the string in vault vSTR . That becomes AI. Without vSTR , operates on AI | | x\*!:vSUFFIX:vSTR | | Suffix the string in vault vSUFFIX to the string in vault vSTR. That becomes AI. Without vSTR, operates on AI | |  | | NOTES | Xenografting is the only correct way to affix strings to other strings in TEA. Of course, the Glue command already allows some kind of xenografting especially with its g\*: command space. For example, one could both prefix and suffix the string “---“ to any other string, to easily turn it into a title in some text environments. The following example TEA program does this using pure X-primitives  V:vHEADLINE:{Interoperability Is Possible} | v:vAFFIX:{---} | x\*:vAFFIX:vHEADLINE | v:vHEADLINE | x\*!:vAFFIX:vHEADLINE  While the following does the same, using pure G-primitives:  V:vHEADLINE:{Interoperability Is Possible} | v:vAFFIX:{---} | g\*:{}:vAFFIX:vHEADLINE:vAFFIX  Both programs should return the result  “---Interoperability is Possible---“  Otherwise, simple demonstrations of what X: is about include these simple TEA programs:  i!:World|x:{Hello } # (=Hello World, VAULTS:{})  i!:World|x!:{Hello } # (=WorldHello , VAULTS:{})  i!:Hello World|x: # (=Hello WorldHello World, VAULTS:{}) doubles the input  i!:Hello World|x!: #(=Hello, VAULTS:{}) halves the input  i!:TEST|v:vI:Hello World|x\*:vI #= Hello WorldTEST  i!:TEST|v:vI:Hello World|x\*!:vI #= TESTHello World |  |  | |