|  |  |
| --- | --- |
| TEA PRIMITIVE | SEMANTICS |
| T: | |  |  | | --- | --- | | NAME | Transform | | PURPOSE | Transform strings using certain methods | | SYNTAX  & SEMANTICS | |  | | --- | | t: | | Apply the rightmost triangulation transform to AI | | t:STR | | Same as t:, but operating on the given string STR | | t!: | | Apply the leftmost triangulation transform to AI | | t!:STR | | Same as t!:, but operating on the given string STR | | t\*: vNAME | | Same as t:, but operating on the string stored in the vault with the name vNAME | | t\*!: vNAME | | Same as t!:, but operating on the string stored in the vault with the name vNAME | |  | | NOTES | Some string transformations are so important in TEA, it was decided to make them primitives in the language so as to utilize them as first-level citizens of the language. Currently, the T-command space mostly implements the so-called “Cetamol Triangulation”, “Cetamol Vertical Inversion Transformer” [1] transform on AI or strings held in vaults. Perhaps other transforms might also get this privilege in future TEA, or not. Time will tell, however, the inventor of the language chose to implement T: this way, and he has respectable reasons why it must be that way.  A basic demonstration of the T-transform is demonstrated below using a command line invocation of TTTT:  # implements the PARACETAMOL T-transform  tttt -i PARACETAMOL -c t:  PARACETAMOL  ARACETAMOL  RACETAMOL  ACETAMOL  CETAMOL  ETAMOL  TAMOL  AMOL  MOL  OL  L  Talking of different ways to transform inputs using TEA APIs, note that currently, TEA is on the WEB! Yes, so, the above example can also be run on any device, any operating system, without installing anything, simply by visiting the following URL in a modern web browser:  **https://tea.nuchwezi.com**?i=PARACETAMOL&c=t:&run  That shall effectively run the TEA program:  i:PARACETAMOL | t:  And produce the same output as above. Checkout **Section 5** for more on this.  It should be interesting to note that this transform has potential roots in the literature and methods of Western Esotericism, in particular, in the occult science literature on crafting magic spells, such as in the work of Janet & Farrar [8], we find several string transforms classified as “pyramid-pattern” spells, some of which reduce a word one or two characters at a time, sometimes from the left, other times from the right. These ancient ideas have had use in creating potent incantations used in healing, blessing and protective magic in Europe and the Middle-East [8], and in TEA, the T-command space preserves this ancient, but useful class of word or phrase transformations, if for no other purpose, at least to curate this ancient wisdom for the modern explorer.  Further, in an age where computer generated art such as with artificial intelligence is common-place, it feels that allowing for some means to create art even for a mere text processing language, using bare TEA primitives and no external libraries or power, would be worth exploring. The following example minimalist program generates some interesting art that looks like statistical graphs – bar charts, rendered without any special graphics facility but the text transformations possible with commands like t: and r:, and it bases on the same interesting sample input as the previous example---just the word “PARACETAMOL”  # an art-generator twist of the PARACETAMOL Transform example  i!:PARACETAMOL| t: | h: | a!: | r!:  And the sample output shown in the following screenshot:  **CLITTTT|< 13:48:30 $>\*** tttt -c "i!:PARACETAMOL| t: | h: | a!: | **r!:"**  **.**  **..... ... ... .. . . . .. .. .. . . . .**  **....**  **. . . .. . ... . . . . .... .. . . . ... ..**  **.**  **.**  **. . ... ... .**  **.. .**  **.. .** |  |  | |