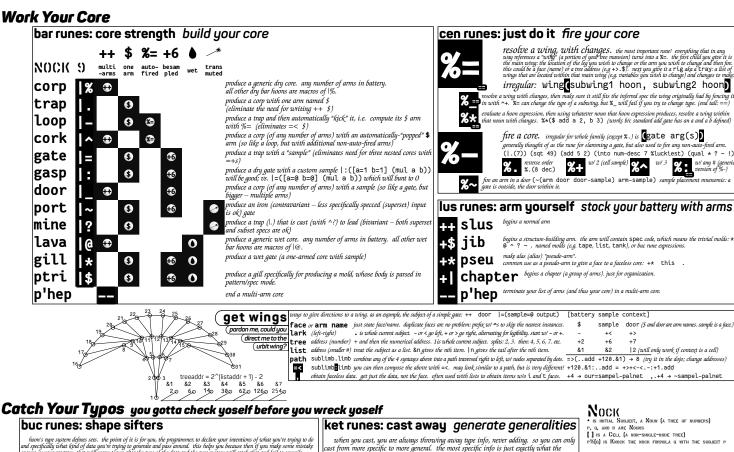
Work Your Core





buc runes: shape sifters

hoon's type system defines sets. the point of it is for you, the programmer, to declare your intentions of what you're trying to and specifically what kind of data you're trying to generate and pass around. this helps you because then if you make some mistal or type in your program, that will many times alter the upo of the data and the type system will calc that and fall to compile these but runes are the mod huiders, what is a mold, you ask? a mod is just a specific kind of gate, and a very simple one, takes in as a sumple on you must all configure to know the system of the programmer when the system is a which it is specific, it tought star are nown (expect for \$2\$ and \$3\$), passing spec generally means the noun is the right shape of tree topologically, though sometimes specific atoms can be required at opening modes, triant of a shape-sorting toy with holes of different shapes and blocks to try to pass through them—a modd is like that, a few lossic modds funce been shortened to a single character:

* noun ^ cell @ atom ? flag (an atom that's 0 or 1) ~ null (atom that's 0). also, ! represents not-a



%spam totally discards the sample it's given and just produces the noun produced by the \$'s child hoon no matter what.

[a=@ b= \star c= $^$ d= \star 27] create a cell mold (to pass through, the noun mu be a cell). the required shape of the cell further defined by specs within the \star :

?(%good %evil) the noun may fit any of the specs in the list (an "or" union use only for a union of atoms. more complex nouns, use \$@, $\* , and/or \$%.

\$\(\epsilon\) a pass-through, allowing all atoms and cells through the mold. \$\(\epsilon\) (null spec) allows only a particular kind of atom (a null) and cell (a spec).

\$%([%number @] [%fizz tape] [%buzz tape]) a whole list of cell molds, with the head a constant (cold atom), to be selected based on the head.

nequire the head of a cell mold to fit a mold. e.g., could be fit a mold. e.g. could be fit a mold fit a mold. e.g. fit or a \$% list. using above example:

\$\(\(\) \(\ \$|

[spec hoon] upgrades a structure from an old version to a new one.
\$&(a-\$%-mold-combining-old-and-new-version gate-normalizing-to-new)

take any spec and define a custom default value for it. \$~(%88 \$=(face @)) is face=@ but with a default of 88

input a noun futing one mold to get the bunt of another as output. $\$-(@ ^)$ takes an atom, outputs a cell (it will just bunt) $\$-(@ @) ^)$ will take a cell of two atoms and output a cell bunt. \$-(@ %0k) returns %ok when fed an atom

ket runes: cast away generate generalities

when you cast, you are always throwing away type info, never adding, so you can only cast from more specific to more general. the most specific info is just exactly what the particular noun is, which the compiler aready knows, obviously; so this should make sense, so the terminology "cast to x" means "cast away all info except x".



cast to a spec by simply explicitly writing the spec to which you wish to cast the noun. assign a face to a noun.

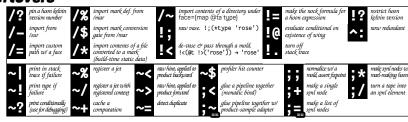
Fence as thy writing a noun and letting the system infer its spec. because you know what spec the system will infer, you know to what it will be cast. $Spec \quad \text{Amul} \rightarrow 1 \quad \text{Amul} \ irs \rightarrow 0 \ (a poorly-chosen bunt)$ [gate hoon] cast based on the noun produced by passing 4 to p — the hoon (any expression) to the gate (e.g. 1 timo which makes it a list).

transmute to zinc transmute to iron transmute to lead constant

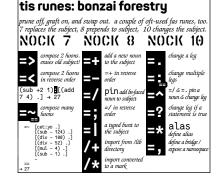
Nodock P and noder o, then ender the α -product bith the p-product as subject 3 is P a Cell? $^{9}(3|r) \rightarrow x$ or x 4 and One to P (Terrenewit'), $^{9}(4|r) \rightarrow r$ plus one 5 is P insertical to Q2: $^{9}(5|r) = x$ or x 6 if Thex-Elbest $^{9}(6|r) = \alpha$ if $r \rightarrow r$ produces x, r if x 7 P Marcs a Net Subject to Q (Gains) and (Gains of Courbose') $^{9}(7|r) = \alpha$ $\rightarrow ^{9}(p)^{9}(6)$ prest node r . Net Subject that as the nee subject indica 8 P And (as Head) to Subject to Q $^{9}(4|r) = \alpha$ $^{9}(4|r) = \alpha$. Perf $^{9}(4|r) = \alpha$. Page r as the case subject indica 8 P And (as Head) to Subject that as the nee subject indica 8 P And (as Head) to Subject that 3 The nee subject indica 3 P and (as Head) to Subject that 3 The nee subject indica 3 P and (as Head) to Subject that 3 The nee subject indica 3 P and (as Head) to Subject that 3 The need subject indicates 3 P and (as Head) to Subject that 3 The P and 3 P and 3 P and (as Head) to Subject that 3 P and (as Head) to Subject that 3 P and (as Head) to Subject that 3 P and (as Head) that 3 P

Auras add'l syntax -%\$ %za-9 url_eafe cuntay ∽scrmbl-dallup ······ D, .∼fipfes-dozzod ····· qы . ~~z.a~.9~~0-9~fa90.9 . ~~-fa90. . 0czmkaZPNJHA91 . 0i909999 . 0b1011.1111 .9.0.9.9.9.9.9 ···· s^{v6} **(** ~90.12.31..23.59.59..fa90 ···· r₈₁ d_{ata} -vy90.d90.h90.m90.s90..fa90 ···· a_{ba} d_{ata} -q_{uad} 0xfa90.ffff 0w~-ZAz.a90~~ 0vva90v.vvvvv gap can follow dot in @u a .~~9 A~9.0 · · · · · · d₅ī

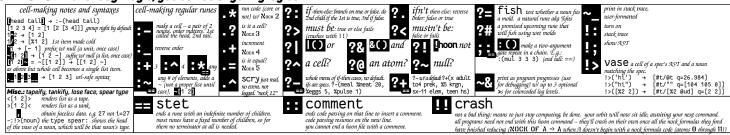
Leftovers



Mod Your Tree



Do Tricks with enough logic, you can do anything logical



not a bad thing: means to just stop computing, be done, your urbit will now sit idle, awaiting your next command.
all programs need not end with this hoon command – they'll crash on their own once all the nock formulas they fired have finished reducing (NOCK OF A → A when A doesn't begin with a nock formula code (atoms 0 through 11))