

**bar runes: core strength** *build your core*

**cen runes: just do it** *fire your core*

**lus runes: arm yourself** *stock your battery with arms*


































































































































## Catch Your Typos

**buc runes: shape sifters**

how the type system defines sets, the point of it is for you, the programmer, to declare your intentions of what you're trying to do and specifically what kind of data you're trying to generate and pass around. this helps you because then if you make some mistake or type in your program, that will happen *exactly* at the type of the data and the type system will catch that and fail to compile. so these two types are still the same, but if you use a mold it will have specific kind of gates and a very strong type and it takes in as general any noun and as long as that noun is within its *spec*, it outputs that same noun (except for *and* and *or* and *not*). passing *spec* generally means the noun is the right shape of type topologically, though sometimes specific atoms can be required at specific nodes. think of a shape-sorting toy with holes of different shapes and blocks to try to pass through them – a mold is like that. a few basic molds have been shortened to a single character:

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 already knows, obviously; so this should make sense. so the terminology "cast to x" means "cast away all info except x".

																																																																																																																																	
---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

**Nock**

- IS INITIAL SUBJECT, A NOUN (A TREE OF NUMBERS)
- P, Q, AND R ARE NOUNS
- [ ] IS A CELL (A NON-SINGLE-NODE TREE)
- $p\%(q)$  IS KNOCK THE NOCK FORMULA Q WITH THE SUBJECT P

- [illegible]

## Mod Your Tree

# tis runes: bonzai forestry

prune off, graft on, and swap out. a couple of off-used fas runes, too. 7 replaces the subject, 8 prepends to subject, 10 changes the subject.

NOCK 7	NOCK 8	NOCK 10
=>	=+	=.
compose 2 fhoons erases old subject	add a new noun to the subject	change a leg
=>	=-	=.
compose 2 fhoons in reverse order	=- in reverse order	change multiple legs
[sub 7 4] . ]	=/	=/ (= - pin a noun - change leg
+2 1 ] [ (add 7 4) . ] -> 27	pin add fhooc noun to subject	=/ (= - pin a noun - change leg
=~	=/	=?
compose many fhoons	=/ in reverse order	? change leg if true
=~	=1	=*
[catc:yo ] [catc 124] ] [dnu - 100] ] [dnu 100] ] [mul - 4] ] [sub - 1] ]	a typed tint to the subject	* alas define alias
	+/	=.
	import from /tis directory	define a bridge / expose a namespace
	/*	
	import converted to a mark	

**Do Tricks** *with enough logic, you can do anything logical*

[illegible]