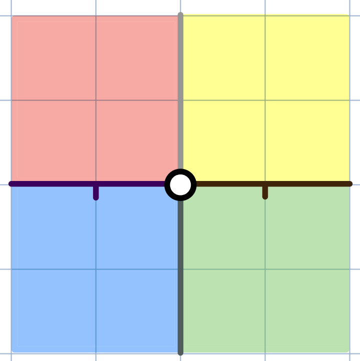
logic in color

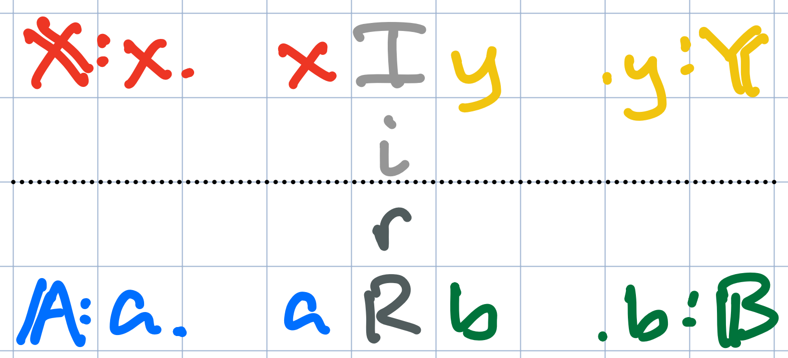
Binary Logic

Binary logic is the logic we learn in school: the logic of <u>true</u> or <u>false</u>, 1 or 0.

In binary logic, there are <u>collections</u> of things, <u>relations</u> of collections, and <u>implications</u> of relations. These form $\mathbb{R}\mathrm{el}$, a structure called a "2-category".

We develop Logic as a language in the complementary forms of <u>imagery</u> + <u>syntax</u>.





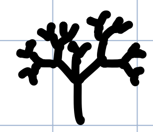
The first provides a generic, conceptual depiction of Logic in two dimensions; while the second is a specific, complete language in which to reason.

Each logic is formed from a <u>base</u> logic Ω ("om"). The base of binary logic is simply $0 \vdash 1$, "false entails true".

1.1 Type, Judgement, Inference

- Type

Things are determined by <u>type</u>.

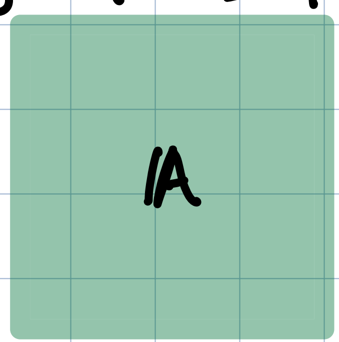
"This is a Tree". 

Every thing is some type of thing.
A "thing" of a type is a <u>term</u>.
We can write

$$a : \mathbb{A}$$

to express that a is an $\mathbb{A}$.
(Let's use small letters for terms + big letters for types.)

We can draw a type as a colored area:
imagine this green is the type Organism.



building Ω

Simplest type is (no color)

— "everything" is a ...

In predicate logic, a type is a <u>set</u>:
a collection of elements.
But this abstracts from life —
Think of the interconnection of Organisms!
To understand logic as <u>concrete</u>,
our notion of "type" ought to
encompass all kinds of (structure).
This is our aim.