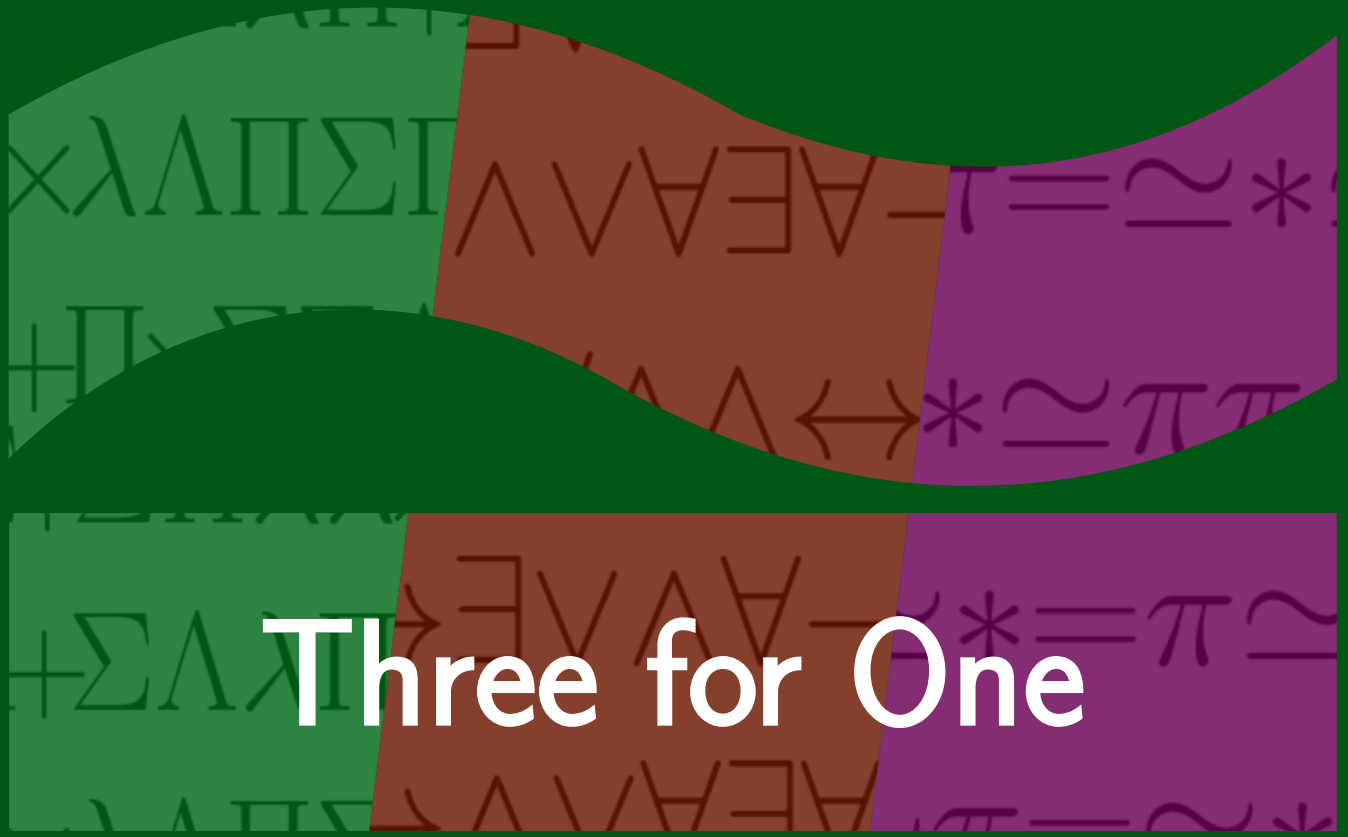

Three for One

Type Theory & Functional Programming

Intro to Homotopy Type Theory, No. 1

1 Type Theory & Functional Programming

```
if n>13 then "2" else n
```

string

number

Python isn't *completely* untyped. It does some amount of typechecking at runtime, and crashes if you try to execute certain kinds of ill-typed code. This is a typing discipline known as *dynamic typing*, whereas the kinds of type systems I'm referring to as "typed" are more precisely known as *statically typed*. There are further distinctions, like *strong* versus *weak* typing (when I say "typed", I mean "strongly typed"), but this video series isn't about programming language theory, so I'm sweeping a lot of this under the rug. Also, the programming languages I'll be talking about from here on out are *functional* programming languages, which Python is not. This is also an important distinction that I'm not getting into.

Typed Programming

Programming Interpretation of HoTT:

Types are types

Terms are expressions

$t : T$ means $t : T$

Programming with `unit`

What is `unit` good for?