The Curry-Howard Isomorphism

It can be discussed as a specific isomorphism between a specific type theory and a specific proof theory.

Or it is a general relationship between intuitionistic logic and type theory.

It is one of the fundamental underpinnings of this course – we can use the Curry-Howard Isomorphism to turn advances in Automated Theorem Proving into advances in Program Synthesis. I poke at STLC and intuitionistic propositional logic.
From a high level:
Theorems correspond to Types
Proofs correspond to (well-typed) programs
Provable theorems correspond to non-empty types

So along these lines:
Proof search is program search
Proof generation is program generation
Type checking is theorem induction

Let's ask ourselves: along those lines, what does System $\text{F}$ correspond to?
STLC corresponds to Propositional logic
System F is STLC along with constructs that generalize over types

second order logic
all the rules of FOL, and
What about other relationships?

First-Order Logic → Dependent Types

Linear Logic → Linear... types

The rest of this course explores this relationship and shows how to use it for code reasoning...