What are Coq's formal foundations?

Gallina, being an extension of the Calculus of Inductive Constructions (CIC), being an extension of the Calculus of Constructions (CoC), being an extension of the typed lambda calculus.

The strong normalization property guarantees what?

Every CIC program and proof term terminates.

The relative consistency property guarantees what?

CIC is *relatively consistent* with ZF so CIC claims are as believable as ZF ones.

What does the annotation `Set` mean on an `Inductive` definition?

It means that what's being defined is part of the object language.

What are Coq's constituent languages?

- Gallina
- Ltac - for queries and requests of Coq
- The Vernacular - a DSL for writing proofs and decision procedures

Interactively test a function by applying it.

```
Eval simpl in ...
```