

Import a module.

Require Import ModuleName

What is a *term*?

What are its subdivisions?

It's a well-formed expression.

*Expressions* are terms that can be thought of as programs.  
*Types* are terms that determine if a type is well-formed and obeys accompanying constraints.

Coq commands all end with what?

A period.

In Coq interpretation scopes are explicitly ...

... opened and closed, with the most recently opened scopes taking precedence.



Open a scope.

Close a scope.

Open Scope scope.

Close Scope scope.

Determine which interpretations are available  
for a notation.

Locate "something or other".

Use underscores for identifiers in the string.  
The bottom interpretation takes precedence.

What is a delimiting key?

Give its syntax.

A name associated with a scope used to explicitly provide the interpretation scope for a term.

`(term) %key`

Get everything Coq knows about an interpretation scope.

Print Scope scope



What is the `Check` command used for?

Give its syntax.

Determining if a term is well-formed and what its type is.

Check term.

Where can decimal notation be used for natural numbers?

In `nat_scope`, whose delimiting key is `nat`.

Where can decimal notation be used for all integers?

In `Z_scope` whose delimiting key is `Z`.  
It is provided in `ZArith`.

Why can't a `nat` be used as a `z`.

How might this be simulated?

Coq has no type coercion.

There is however the possibility of defining such a conversion using implicit coercions.



What are the two forms of simple types?

- *Atomic types*, represented by single identifiers.
- *Arrow types*, of the form  $\text{Type1} \rightarrow \text{Type2}$ .

In the context of Coq, "scope" can mean ...

... either interpretation scope (a collection of notations) or a scope in the usual sense (the textual area where a binding is active).

Give the syntax of a declaration.

Give the syntax of a definition.

(ident : SomeType)

(ident := term : someType)

Contrast *global* and *local* scopes.

Global scopes are active for the remainder of development.  
Local scopes are active only in the current *section*.



Contrast *environment* and *context*.

The environment contains global bindings while the context contains the local ones.

How can you return to the initial bindings?

This is a special case of what?

Reset Initial

It brings you to an empty context and the initial environment.

Reset identifier

Brings you back to the state just after the given identifier was introduced.