Where does Coq define booleans and numbers?

In the standard library.

What's a "type" in Coq?

A set of data values.

Define the boolean type

```
Inductive bool : Type :=
 | true : bool
 | false : bool.
```

Define a boolean negation function.

```
Definition negb (b : bool) : bool :=
match b with
| true => false
| false => true
```

end.

Define a function for boolean conjunction.

```
Definition andb (b1 : bool) (b2 : bool) : bool :=
match b1 with
| true => b2
| false => false
end.
```

Coq can infer which parts of a function prototype?

Argument and result types.

Make a named assertion that ~true is false, then prove it.

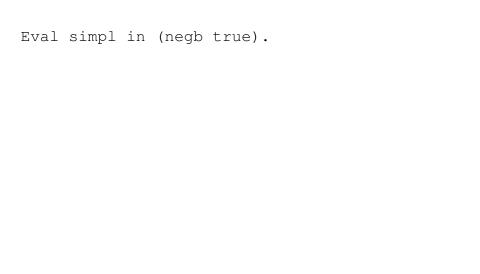
Example test_negation:
(negb true) = false.

Proof. simpl. reflexivity. Qed.

Name three ways to check that a function works

- Use Eval on a test case and observe the result.
- Use Example to record expected result, then as Coq to verify.
- "extract" function Definition to OCaml, Scheme, or Haskell.

Apply negation to the boolean true.



How do you delimit Coq fragments in comments for the benefit of coqdoc?

Surround the fragments with square brackets.

Define and explain the purpose of the admit value.

Definition admit {T : Type} : T. Admitted.

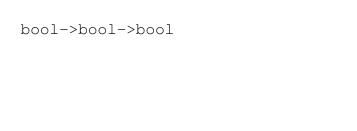
It fills in the hole in an incomplete Definitions.

How do you fill in a hole in a Definition? In an Example?

admit fills in holes in Definitions.

Admitted fills in holes in Examples.

How does Coq write the type of a boolean conjunction function?



What does the Check command do?

It causes Coq to print the type of an expression.