* IsInitialized(): checks if all the required fields have been set.
* \_\_str\_\_(): returns a human-readable representation of the message, particularly useful for debugging. (Usually invoked as str(message) or print message.)
* CopyFrom(other\_msg): overwrites the message with the given message’s values.
* Clear(): clears all the elements back to the empty state.
* SerializeToString(): serializes the message and returns it as a string. Note that the bytes are binary, not text; we only use the str type as a convenient container.
* ParseFromString(data): parses a message from the given string.
* Protocol buffer classes are basically data holders (like structs in C) that don’t provide additional functionality

## Extending a Protocol Buffer

* you must not change the tag numbers of any existing fields.
* you must not add or delete any required fields.
* you may delete optional or repeated fields.
* you may add new optional or repeated fields but you must use fresh tag numbers (that is, tag numbers that were never used in this protocol buffer, not even by deleted fields).

old code will happily read new messages and simply ignore any new fields

To the old code,

deleted optional fields will simply have their default value,

deleted repeated fields will be empty.

**New code will also transparently read old messages.**

new optional fields will not be present in old messages,

so you will need to either check explicitly whether they’re set with has\_, or provide a reasonable default value in your .proto

for strings, the default value is the empty string.

For booleans, the default value is false.

For numeric types, the default value is zero

if you added a new repeated field,:

your new code will not be able to tell whether it was left empty (by new code) or never set at all (by old code) since there is no has\_ flag for it

protoc -I**=/home/ameri/PycharmProjects/pythonProject** --python\_out**=/home/ameri/PycharmProjects/pythonProject** **/home/ameri/PycharmProjects/pythonProject**/addressbook.proto

Fatal Python error: none\_dealloc: deallocating None

Python runtime state: finalizing (tstate=0x00005570ca02dae0)

Current thread 0x00007f8776194000 (most recent call first):

Garbage-collecting

<no Python frame>

Process finished with exit code 134 (interrupted by signal 6: SIGABRT)