Language Guide (proto 3)

Skimming

→ What are options.
  ↳ what options are available.
→ Use of package keyword.
→ Basic datatypes. (C++) language.
→ what is the number (1,2..) of a variable name.

⇒ Describes how to use the protocol buffer language to structure your protocol buffer data.

Syntax = "proto3";
  ↳ Specifies that you're using proto3 syntax.
  ↳ If you don't do this, the protocol buffer compiler will assume you are using proto2.
  ↳ This must be the first non-empty, non-commented line of the file.

★ Field Number

⇒ Each field in the message definition has a unique number.

⇒ These field numbers are used to identify your field in the message binary format.
  ↳ Should not be changed once your message type is in use.

Field Number	Details
1 — 15	One byte to encode
16 — 2047	two byte to encode
19000 - 19999	Reserved for the Protocol buffer implementation
$2^{29}-1$	largest field number

⇒ For CH, the compiler generates .h & .cc file for each .proto.
  ↳ Class for each message type described in your file.

⇒ Types: bool, String, int32, float, double, int64

⇒ Message fields can be one of the following:
  - → Singular
    - ↳ default field rule for proto3 Syntax
  - → repeated
    - ↳ This field can be repeated any number of times.

⇒ Messages can be nested together.

★ Package

⇒ You can add an optional package specifier to a .proto file to prevent name clashes between Protocol message types.

```
Package foo.bar
message Open { ... }
```

⇒ On C++ the generated class are wrapped inside a C++ namespace.

↳ Exaple Open would be in the namespace foo::bar.

★ Options

- File-level Options
  - → java_package
  - → java_multiple_files
  - → java_outer_classname
- Message level Options
- Field-level Options