~~~toml
[[questions]]
type = "MultipleChoice"
prompt.prompt = """
Which of the following statements best describes the function of the `use` keyword?
"""
prompt.distractors = [
  "`use` copies the contents of an item from one module to another",
  "`use` indicates to the compiler that an item will be used and should be optimized",
  "`use` allows access to items that circumvents normal privacy rules",
]
answer.answer = "`use` reduces the verbosity of referring to items in the used path"
context = """
`use` allows references to items in the used path that are shorter than without `use`,
reducing the verbosity of references.
"""
id = "446098fa-3695-43e9-8d70-a160188b1c30"
[[questions]]
type = "ShortAnswer"
prompt.prompt = """
Consider this module and `use` statement:
```
pub mod parent {
    pub fn a() {}
    fn b() {}
    pub mod child {
        pub fn c() {}
    }
}
use parent::{a, child as alias};
fn main() {
    // here
}
```
Inside `main`, what is the total number of paths that can refer to `a`, `b`, or `c` (not
including those that use `self` and `super`)?
Write your answer as a digit such as 0 or 1.
For example, if the only two valid paths were `a` and `parent::b`, then the answer would
be 2.
"""
answer.answer = "4"
context = """
There are two paths to `a`: `parent::a` and `a`.
There is no path to `b`, because it is private.
There are two paths to `c`: `parent::child::c`, and `alias::c`.
"""
id = "21fd3314-cf90-4b32-9223-d5d773f484dd"
~~~