```toml
[[questions]]
id = "fa63d18f-fee3-4923-962b-bdeabbf61fbd"
type = "ShortAnswer"
prompt.prompt = "The keyword for declaring a new function in Cairo is:"
answer.answer = "fn"
[[questions]]
id = "6343598f-380f-4a59-b6fb-7dfadc397f74"
type = "Tracing"
prompt.program = """
fn f(x) {
  println!("{x}");
}
fn main() {
  f(0);
}
"""
answer.doesCompile = false
answer.lineNumber = 1
context = """
A function must declare the types of its parameters. In this case, function `f` could be corrected by adding `u8` type to the `x` parameter like this: `fn f(x:u8)`.
"""
[[questions]]
id = "695f318f-539b-4ce8-9c00-92e29f7f1bb3"
type = "MultipleChoice"
prompt.prompt = """
In Cairo, a curly-brace block like `{ /* ... */ }` is:
1. An expression
2. A statement
3. A syntactic scope
"""
answer.answer = "1 and 3"
prompt.distractors = [ "1 only", "2 only", "2 and 3" ]
context = """
A block is an expression (#1) that is allowed to contain statements. It also defines a syntactic scope for let-bindings inside it (#3).
"""
[[questions]]
id = "7bbc0afb-2ad7-4d8d-97fd-610a65a85e38"
type = "Tracing"
prompt.program = """
fn f(x: usize) -> usize { x + 1 }
fn main() {
  println!("{}", f({
    let y = 1;
    y + 1
  }));
}
"""
answer.doesCompile = true
answer.stdout = "3"
```