(continued from previous page)

```
dotted_as_name:
    | dotted_name ['as' NAME ]
dotted_name:
    | dotted_name '.' NAME
    | NAME

if_stmt:
    | 'if' named_expression ':' block elif_stmt
    | 'if' named_expression ':' block [else_block]
elif_stmt:
    | 'elif' named_expression ':' block elif_stmt
    | 'elif' named_expression ':' block [else_block]
else_block: 'else' ':' block

while_stmt:
    | 'while' named_expression ':' block [else_block]

for_stmt:
    | 'for' star_targets 'in' ~ star_expressions ':' [TYPE_COMMENT] block [else_
↪block]
    | ASYNC 'for' star_targets 'in' ~ star_expressions ':' [TYPE_COMMENT] block␣
↪[else_block]
with_stmt:
    | 'with' '(' ','.with_item+ ','? ')' ':' block
    | 'with' ','.with_item+ ':' [TYPE_COMMENT] block
    | ASYNC 'with' '(' ','.with_item+ ','? ')' ':' block
    | ASYNC 'with' ','.with_item+ ':' [TYPE_COMMENT] block
with_item:
    | expression 'as' star_target &(',' | ')' | ':')
    | expression

try_stmt:
    | 'try' ':' block finally_block
    | 'try' ':' block except_block+ [else_block] [finally_block]
except_block:
    | 'except' expression ['as' NAME ] ':' block
    | 'except' ':' block
finally_block: 'finally' ':' block

return_stmt:
    | 'return' [star_expressions]

raise_stmt:
    | 'raise' expression ['from' expression ]
    | 'raise'

function_def:
    | decorators function_def_raw
    | function_def_raw

function_def_raw:
    | 'def' NAME '(' [params] ')' ['->' expression ] ':' [func_type_comment] block
    | ASYNC 'def' NAME '(' [params] ')' ['->' expression ] ':' [func_type_comment]␣
↪block
func_type_comment:
    | NEWLINE TYPE_COMMENT &(NEWLINE INDENT)   # Must be followed by indented block
    | TYPE_COMMENT

params:
    | parameters
```

(continues on next page)