

## 2.3.1. Keywords

The following identifiers are used as reserved words, or *keywords* of the language, and cannot be used as ordinary identifiers. They must be spelled exactly as written here:

|        |          |         |          |        |
|--------|----------|---------|----------|--------|
| False  | class    | finally | is       | return |
| None   | continue | for     | lambda   | try    |
| True   | def      | from    | nonlocal | while  |
| and    | del      | global  | not      | with   |
| as     | elif     | if      | or       | yield  |
| assert | else     | import  | pass     |        |
| break  | except   | in      | raise    |        |

## 2.5. Operators

The following tokens are operators:

|    |    |    |    |    |    |   |   |
|----|----|----|----|----|----|---|---|
| +  | -  | *  | ** | /  | // | % | @ |
| << | >> | &  |    | ^  | ~  |   |   |
| <  | >  | <= | >= | == | != |   |   |

## 2.6. Delimiters

The following tokens serve as delimiters in the grammar:

|    |    |    |     |     |     |    |
|----|----|----|-----|-----|-----|----|
| (  | )  | [  | ]   | {   | }   |    |
| ,  | :  | .  | ;   | @   | =   | -> |
| += | -= | *= | /=  | //= | %=  | @= |
| &= | =  | ^= | >>= | <<= | **= |    |

The period can also occur in floating-point and imaginary literals. A sequence of three periods has a special meaning as an ellipsis literal. The second half of the list, the augmented assignment operators, serve lexically as delimiters, but also perform an operation.

The following printing ASCII characters have special meaning as part of other tokens or are otherwise significant to the lexical analyzer:

|   |   |   |   |
|---|---|---|---|
| ' | " | # | \ |
|---|---|---|---|

The following printing ASCII characters are not used in Python. Their occurrence outside string literals and comments is an unconditional error:

|    |   |   |
|----|---|---|
| \$ | ? | ` |
|----|---|---|