

## Pet Peeves

Pet Peeves is also important in Python. Pet Peeves means how to write whitespaces in expressions and statements correctly. We need whitespaces in expressions and statements, but do not always need them. For example, `spam(ham[1], {eggs: 2})`. We do not need more than one space after open parentheses, open brackets and, open braces, and before close parentheses, close brackets and, close braces. So, `spam( ham[ 1 ], { eggs: 2 } )` is wrong, but we need a whitespace after the `:` before element. There is another situation, when between a trailing comma and a following close parenthesis. For example, `foo = (0,)`. So, `foo = (0, )` is false, because there is a whitespace between a trailing comma and a following close parenthesis. The third situation where we do not need whitespace is immediately before a comma, semicolon, or colon. For example, `if x == 4: print(x, y); x, y = y, x`, but `if x == 4 : print(x , y) ; x , y = y , x` is wrong. However, in slices, the colon is represented as a binary operator and both sides must be set to the same number (treated as the lowest priority operator). In extended slices, both colons must have the same number of spaces. Exception: If a slice omits a parameter, the spaces are omitted. `ham[lower:upper]`, `ham[lower:upper:]`, `ham[lower::step]` is correct, but `ham[1: 9]`, `ham[1 :9]`, `ham[1:9 :3]`, `ham[lower : : upper]` is wrong.