

A **syntax error** is a case where Python is unable to understand what you are asking it to do. It means that your program is ill-formatted in some way. This is distinct from a runtime error, which happens while Python is actually running your code. Syntax errors occur before your code begins to run, and are thus reported quickly.

Syntax errors can be very frustrating to beginning programmers, but as you gain experience you will find them much easier to deal with than runtime errors. A syntax error is the equivalent of a misspelling when writing English, and doesn't mean there is anything wrong with what you *meant* to write, it just means that you haven't written it properly.

Common syntax errors

Unmatched parentheses Probably the number one most common mistake, and Python seldom identifies it as such, so the syntax error message often refers to a different (later) line of code.

Missing colon In an `if` statement or `for` loop, you omitted a colon.

Poor indentation Python relies on indentation to identify “blocks” of code for `if` statements or `for` loops. These blocks must have identical indentation in order for Python to feel confident it knows when the block ends.

Tab characters A special case of poor indentation that can be immensely frustrating is when you get a tab character into your code. A tab character (which can come from either copying and pasting, or from a poorly configured editor) counts to Python as 8 spaces, but your editor may display it as a different number of spaces, making a block *look* properly aligned when Python sees it as unaligned.