

Indenting Code

Python functions have no explicit `begin` or `end`, and no curly braces to mark where the function code starts and stops. The only delimiter is a colon (:) and the indentation of the code itself.

```
def approximate_size(size, a_kilobyte_is_1024_bytes=True): #①
    if size < 0: #②
        raise ValueError('number must be non-negative') #③
    #④

    multiple = 1024 if a_kilobyte_is_1024_bytes else 1000
    for suffix in SUFFIXES[multiple]: #⑤
        size /= multiple
        if size < multiple:
            return '{0:.1f} {1}'.format(size, suffix)

    raise ValueError('number too large')
```

① Code blocks are defined by their indentation. By “code block,” I mean functions, `if` statements, `for` loops, `while` loops, and so forth. Indenting starts a block and unindenting ends it. There are no explicit braces, brackets, or keywords. This means that whitespace is significant, and must be consistent. In this example, the function code is indented four spaces. It doesn’t need to be four spaces, it just needs to be consistent. The first line that is not indented marks the end of the function.

② In Python, an `if` statement is followed by a code block. If the `if` expression evaluates to true, the indented block is executed, otherwise it falls to the `else` block (if any). Note the lack of parentheses around the expression.

③ This line is inside the `if` code block. This `raise` statement will raise an exception (of type `ValueError`), but only if `size < 0`.

④ This is `not` the end of the function. Completely blank lines don’t count. They can make the code more readable, but they don’t count as code block delimiters. The function continues on the next line.

⑤ The `for` loop also marks the start of a code block. Code blocks can contain multiple lines, as long as they are all indented the same amount. This `for` loop has three lines of code in it. There is no other special syntax for multi-line code blocks. Just indent and get on with your life.

After some initial protests and several snide analogies to Fortran, you will make peace with this and start seeing its benefits. One major benefit is that all Python programs look similar, since indentation is a language requirement and not a matter of style. This makes it easier to read and understand other people's Python code.

Python uses carriage returns to separate statements and a colon and indentation to separate code blocks. c++ and Java use semicolons to separate statements and curly braces to separate code blocks.