

STRUCTURING WITH INDENTATION

BLOCKS

A block is a group of statements in a program or script. Usually it consists of at least one statement and of declarations for the block, depending on the programming or scripting language. A language, which allows grouping with blocks, is called a block structured language. Generally, blocks can contain blocks as well, so we get a nested block structure. A block in a script or program functions as a mean to group statements to be treated as if they were one statement. In many cases, it also serves as a way to limit the lexical scope of variables and functions.

Initially, in simple languages like Basic and Fortran, there was no way of explicitly using block structures. Programmers had to rely on "go to" structures, nowadays frowned upon, because "Go to programs" turn easily into spaghetti code, i.e. tangled and inscrutable control structures.



The first time, block structures had been formalized was in ALGOL, called a compound statement.

Programming languages usually use certain methods to group statements into blocks:

- begin ... end
ALGOL, Pascal and others
An code snippet in Pascal to show this usage of blocks:

```
with ptoNode^ do
begin
  x := 42;
  y := 'X';
end;
```

- do ... done
e.g. Bourne and Bash shell
- Braces (also called curly brackets): { ... }
By far the most common approach, used by C, C++, Perl, Java, and many other programming languages are braces.
The following examples shows a conditional statement in C:

```

if (x==42) {
    printf("The Answer to the Ultimate Question of Life, the
Universe, and Everything\n");
} else {
    printf("Just a number!\n");
}

```

The indentations in this code fragment are not necessary. So the code could be written - offending common decency - as

```

if (x==42) {printf("The Answer to the Ultimate Question of
Life, the Universe, and Everything\n");} else {printf("Just a
number!\n");}

```

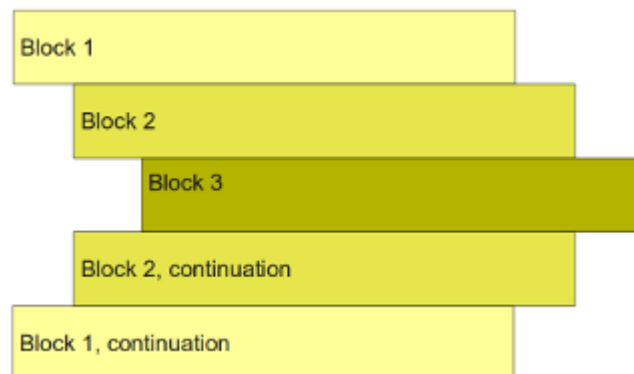
Please, keep this in mind to understand the advantages of Python!

- if ... fi
e.g. Bourne and Bash shell

INDENTING CODE

Python uses a different principle. Python programs get structured through indentation, i.e. code blocks are defined by their indentation. Okay that's what we expect from any program code, isn't it? Yes, but in the case of Python it's a language requirement not a matter of style. This principle makes it easier to read and understand other people's Python code.

So, how does it work? All statements with the same distance to the right belong to the same block of code, i.e. the statements within a block line up vertically. The block ends at a line less indented or the end of the file. If a block has to be more deeply nested, it is simply indented further to the right.



Beginners are not supposed to understand the following example, because we haven't introduced most of the used structures, like conditional statements and loops. Please confer the following chapters about loops and conditional statements for explanations.

The program implements an algorithm to calculate Pythagorean triples. You will find an explanation of the Pythagorean numbers in our chapter on [for loops](#).

```

from math import sqrt
n = input("Maximum Number? ")
n = int(n)+1

```

```
for a in range(1,n):  
    for b in range(a,n):  
        c_square = a**2 + b**2  
        c = int(sqrt(c_square))  
        if ((c_square - c**2) == 0):  
            print(a, b, c)
```

There is another aspect of structuring in Python, which we haven't mentioned so far, which you can see in the example. Loops and Conditional statements end with a colon ":" - the same is true for functions and other structures introducing blocks. So, we should have said Python structures by colons and indentation.

© 2011 - 2018, Bernd Klein, Bodenseo; Design by Denise Mitchinson adapted for python-course.eu by Bernd Klein