CSP1150/CSP5110: Programming Principles
Reading 2.1: Indenting Code Blocks

One of the most commonly recognised differences between Python and most other languages is that it uses indentation to express blocks of code, rather than curly brackets. A "block" is a section of code statements that are grouped together for some reason – usually that reason relates to control structures such as "if" statements or loops, which run a block of code if a condition is met.

Code Blocks in Most Languages

As covered in the lecture, most languages use curly brackets to denote a block, and they are optional if the block consists of just one statement.

Most Languages

```
if (value > 0) // curly brackets needed for block
{
    total += value;
    print('Value added.');
}

if (error == True) // curly brackets optional for single statement
    print('Error!');
```

In these languages, the indentation of the block is purely for readability and has no impact upon the code. This means that all of the following examples are perfectly valid and equivalent:

Most Languages

```
if (value > 0)
{
    total += value;
 print('Value added. ');
}

if (value > 0) {
total += value;
print('Value added.'); }

if (value > 0) { total += value; print('Value added.'); }
```

While some of these examples are more concise, they are generally less readable than they would be if indentation was used consistently to denote a block.

It is common practise and convention in these languages to consistently indent code blocks, even though it isn't necessary or enforced by the language, in order to make code more readable.

Code Blocks in Python

Python uses indentation to denote blocks of code, instead of curly brackets. This is known as the "off-side rule" (a soccer/football reference), and the main reason for it is to enforce some level of consistency and readability in the code. Python aims to be a very readable programming language.

> *Because blocks are denoted by indentation in Python, indentation is uniform in Python programs. And indentation is meaningful to us as readers. So because we have consistent code formatting, I can read somebody else's code and I'm not constantly tripping over, "Oh, I see. They're putting their curly braces here or there." I don't have to think about that.*
>
> - Bruce Eckel (author of numerous programming books)

"Indentation" simply means some amount of whitespace to the left of a statement. It can be one or more spaces or one or more tabs. The recommended standard is 4 spaces, and using tabs is discouraged as different operating systems and code editors treat them differently. Statements in a block must have the same amount of indentation, and it must be more than the indentation of the statement directly before the start of the block.

Python

```
if value > 0: # indentation used to denote block
    total += value
    print('Value added.')

if (error == True): # indentation used to denote block
    print('Error!')
```

The off-side rule is not hard to follow, and matches the way that a *good* programmer indents their code regardless of which language they are using.

As always, there are a few things you should look out for:

Python

```
if value > 0: # logic error - print statement not in block
    total += value
print('Value added.')

if value > 0: # syntax error - unmatched indent
    total += value
  print('Value added.')

if value > 0: # syntax error - unexpected indent
    total += value
        print('Value added.')
```

In the first example the print statement is not indented, so it is not considered part of the block – hence it will always be run, regardless of the result of the if statement.

In the second example, a syntax error is caused because the print statement is indented less than the previous statement, and the indentation does not match any previously indented block that would be relevant.

In the third example, the print statement is indented more, but there is no reason to do define a new block at this point in the code.