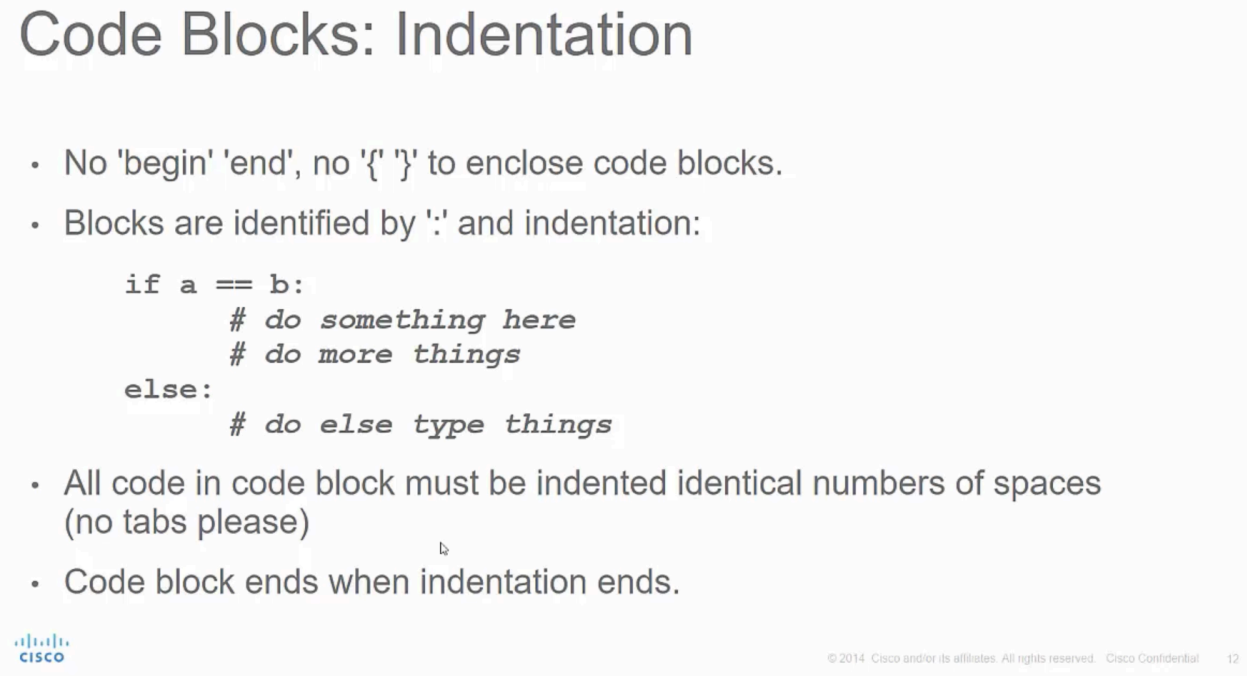

Programming languages have different ways of defining code blocks, which are segments of code that get executed as a result of loop iterations, conditional statements, and functions. In contrast to some other languages, in Python code blocks do not have 'begin' and 'end' delimiters such as curly brackets ('{}'). Instead, Python has two components that combine to define a code block:

- **Colon (':').** The statement preceding the block will end with a colon (':').

```
if len(my_string) > 10:
```

- **Indentation.** Python uses indentation to identify statements in a code block. Indentation must be uniform and consistent; and when the indentation ends, the code block ends.

```
if len(my_string) > 10:

    print my_string
```

This indentation scheme may seem strange at first, but consider that other more flexible schemes for identifying code blocks are well-suited to computers, but less suited to humans. You can easily create mis-matched curly brackets which are syntactically correct, but semantically flawed, due to misplaced brackets. This is not an issue for Python.

Note
In Python, it is highly suggested you use spaces for indentation. While tabs will work, they can cause readability problems between different platforms and editors. The key thing to remember is being consistent about your indent choice so that your program is more readable.