Python's Fake Increment and Decrement Operators

In Python, you can increase the value of a variable by 1 or reduce it by 1 using the augmented assignment operators. The code spam += 1 and spam -= 1 increments and decrements the numeric values in spam by 1, respectively.

Other languages such as C++ and Java have the ++ and -- operators for incrementing and decrementing variables. (The name of C++ itself reflects this; the name is a tongue-in-cheek joke that indicates it's an enhanced form of the C language.) Code in C++ and Java could have ++spam or spam++. Python wisely doesn't include these operators; they are notoriously susceptible to subtle bugs.

However, it is perfectly legal to have the following Python code:

```
1    >>> spam = 42
2    >>> spam = ++spam
3    >>> spam
4    42
5    >>> spam = --spam
6    >>> spam
7    42
```

The first thing you notice is that the ++ and -- "operators" in Python don't actually increment or decrement the value in spam. Rather, the leading - is Python's unary negation operator. It allows you to have code like this:

```
1    >>> spam = 42
2    >>> print(-spam)
3    -42
```

It's legal to have multiple unary negative operators in front of a value. With two of them, you'd get the negative of the negative of the value, which for integer values just evaluates to the original value:

```
1    >>> print(--spam)
2    42
3    >>> --42
4    42
```

This is a quite silly thing to do, and you won't ever see a unary negation operator used twice in real-world code. (Though if you did, it's probably because the programmer learned to program in another language has just written buggy Python code!) There is also a + unary operator. It evaluates an integer value to the same sign as the original value, which is to say, it does absolutely nothing:

```
1    >>> +42
2    42
3    >>> spam = -42
4    >>> print(+spam) # +spam is not the same as abs(spam)
5    -42
```

Being able to write +42 (or ++42) seems just as silly as −42, so why does Python even have this unary operator? It exists only to complement the − operator if you need to overload these operators for your own classes. (See the documentation for the ___pos___ special method.)

The + and − unary operators are only valid when in front of a Python value, not after it. While spam++ and spam− might be legal code in C++ or Java, they produce syntax errors in Python:

```
1    >>> spam++
2     File "<stdin>", line 1
3       spam++
4           ^
5    SyntaxError: invalid syntax
```

Python doesn't have increment and decrement operators, it's only a quirk of the langauge syntax that can make it seem like it does.

For more Information and to build website/System using Python, Hire Python Developer from us as we give you high quality product by utilizing all the latest tools and advanced technology. E-mail us any clock at – hello@hkinfosoft.com or Skype us: "hkinfosoft".

To develop custom web app using Python, please visit our technology page.

Content Source