Augmented Assignment

Often we want to modify and updata a variable. The most common are things like:

```
index = index + 1
k = k - 1
```

Here is a shorthand:

```
index += 1
k -= 1
```

A more complete test would be:

```
index = 2
k = 5
index += 1
k -= 1
print(index, k) # 3 4
```

The operators here, `+` and `-`, must be directly before the equal sign, with *no space in between*. If you get that wrong, at least you get reminded of it with a syntax error.

Be very careful not to reverse the `=` and operator accidentally, since this gives a different result (a regular assignment), and no syntax error to warn you:

```
index = 2
k = 5
index =+ 1
k =- 1
print(index, k) # 1 -1
```

This same idea works with other operators `*`, `/`, `//`, `%`, `**`. If we let op stand for a binary operator, including any of those above, then, in general

```
variableName op= expression
```

means

```
variableName = variableName op (expression)
```

for whatever types and operators allow the longer version.

Think and check: What is printed?

```
x = 3
x += 7
x *= 2
x -= 6
print(x)
```

Java, C++, C# and other languages have similar syntax for binary operands that they allow.
