**Arithmetic Assignment Operators**

In Python you can combine the assignment operator with an arithmetic operator, such as +=, to perform an operation.

Let us say y = 4, and to increase y by 1, your program might have two statements like this:  
y = 4  
y += 1  
  
Which Python interprets as   
y = y + 1   
and substitutes the value for y on the right side giving the result,  
y = 5

Or you may want to triple x, you might have two statements like this:  
x = 8  
x \*= 3  
  
Which Python interprets as   
x = x \* 3   
and substitutes the value for x on the right side giving the result,  
x = 24

|  |  |
| --- | --- |
| Symbol | Operator name |
| += | Increment assignment |
| -= | Decrement assignment |
| \*= | Multiplication assignment |
| /= | Division assignment |
| %= | Modulus (Remainder) assignment |
| //= | Floor division (quotient) assignment |
| \*\*= | Exponent (Power) assignment |

Here are Python arithmetic assignment operators:

**Python primary prompt**   
You can use the Python arithmetic assignment operators in computation at the Python primary prompt.

Example:  
Decrease x by 3, given the initial value of 2  
  
At the Python primary prompt, type  
>>>x = 2  
>>>x -= 3  
>>>x  
Python displays:  
-1  
>>>

**Assignment:**1.  
Use the Python prompt to display the result for each of the following

A.  
x = 2  
x += 3  
B.  
x = 15  
x -= 3  
C.  
x = 6  
x \*= 2  
D.  
x = 17  
x /= 5  
D.  
x = 19  
x %= 6  
E.  
x = 20  
x //= 7  
x = x // 7  
F.  
x = 2  
x \*\*= 4

2.  
Use the Python primary prompt to complete each of the following

Given:

y = 4.2  
z = 5  
k = 2.1  
m = 7  
n = 6  
p = 10

A.  
y += 3

B.  
z -= 3

C.  
z -= 3

D.  
k \*= 2

E.  
m /= 5

F.  
n %= 6

G.  
p //= 7

H.  
p %= 20 - p

I.  
p %= 20 % p

J.   
m += z\*\*2 + y \* p - p

K.

m -= z\*\*2 + y - p % p