

✓ 7. Solution: Variables and Assignm...

✓ 8. Integers and Floats

✓ 9. Quiz: Integers and Floats

✓ 10. Booleans, Comparison Operato...

✓ 11. Quiz: Booleans, Comparison Op...

✓ 12. Solution: Booleans, Comparison...

✓ 13. Strings

✓ 14. Quiz: Strings

✓ 15. Solution: Strings

✓ 16. Type and Type Conversion

✓ 17. Quiz: Type and Type Conversion

✓ 18. Solution: Type and Type Conver...

✓ 19. String Methods

✓ 20. Quiz: String Methods

Solution: Variables and Assignment Operators

Solution: Assign and Modify Variables

This is our solution to the previous quiz:

```
# The current volume of a water reservoir (in cubic metres)
reservoir_volume = 4.445e8
# The amount of rainfall from a storm (in cubic metres)
rainfall = 5e6

# decrease the rainfall variable by 10% to account for runoff
rainfall *= .9

# add the rainfall variable to the reservoir_volume variable
reservoir_volume += rainfall

# increase reservoir_volume by 5% to account for stormwater that flows
# into the reservoir in the days following the storm
reservoir_volume *= 1.05

# decrease reservoir_volume by 5% to account for evaporation
reservoir_volume *= 0.95

# subtract 2.5e5 cubic metres from reservoir_volume to account for water
# that's piped to arid regions.
reservoir_volume -= 2.5e5

# print the new value of the reservoir_volume variable
print(reservoir_volume)
```



Solution: Variables and Assignment Operators

has not changed. That is, it is still 3.0.

This is because when a variable is assigned, it is assigned to the *value of the expression* on the right-hand-side, not to the expression itself. In the line:

```
>>> crs_per_rab = carrots/rabbits
```

Python actually did the calculation to evaluate the expression on the right-hand-side, `carrots/rabbits`, and then assigned the variable `crs_per_rab` to be the value of that expression. It promptly forgot the formula, only saving the result in the variable.

In order to update the value of `crs_per_rab` to take into account the change in `rabbits`, we need to run this line again:

```
>>> crs_per_rab = carrots/rabbits
>>> print(crs_per_rab)
2.0
```

That's the new number of carrots per rabbit after the increase in the number of rabbits. All of our variables have been updated to take this into account.

[NEXT](#)

