2.13. Updating Variables

Activity: 1 -- Video: (updatevid)

One of the most common forms of reassignment is an **update** where the new value of the variable depends on the old. For example,

```
x = x + 1
```

This means get the current value of x, add one, and then update x with the new value. The new value of x is the old value of x plus 1. Although this assignment statement may look a bit strange, remember that executing assignment is a two-step process. First, evaluate the right-hand side expression. Second, let the variable name on the left-hand side refer to this new resulting object. The fact that `x` appears on both sides does not matter. The semantics of the assignment statement makes sure that there is no confusion as to the result. The visualizer makes this very clear.

Next Step Reset

```
x = 6
x = x + 1
```

```
x = x + 1
x = 6 + 1
x = 7
```

Activity: 2 -- ShowEval (se_ac2_14_1)

Save & Run 4/17/2021, 1:16:37 PM - 2 of 2 Show in CodeLens

```
x = 6          # initialize x
print(x)
x = x + 1      # update x
print(x)

```

```
6
7
```

Activity: 3 -- ActiveCode (ac2_14_1)

If you try to update a variable that doesn't exist, you get an error because Python evaluates the expression on the right side of the assignment operator before it assigns the resulting value to the name on the left. Before you can update a variable, you have to **initialize** it, usually with a simple assignment. In the above example, `x` was initialized to 6.

Updating a variable by adding something to it is called an **increment**; subtracting is called a **decrement**. Sometimes programmers talk about incrementing or decrementing without specifying by how much; when they do they usually mean by 1. Sometimes programmers also talk about **bumping** a variable, which means the same as incrementing it by 1.

Incrementing and decrementing are such common operations that programming languages often include special syntax for it. In Python `+=` is used for incrementing, and `-=` for decrementing. In some other languages, there is even a special syntax `++` and `--` for incrementing or decrementing by 1. Python does not have such a special syntax. To increment x by 1 you have to write `x += 1` or `x = x + 1`.

Save & Run 4/17/2021, 1:16:47 PM - 2 of 2 Show in CodeLens

```
x = 6          # initialize x
print(x)
x += 3         # increment x by 3; same as x = x + 3
print(x)
x -= 1         # decrement x by 1
print(x)

```

```
6
9
8
```

Activity: 4 -- ActiveCode (ac2_14_2)

Imagine that we wanted to not increment by one each time but instead add together the numbers one through ten, but only one at a time.

Save & Run 4/17/2021, 1:17:14 PM - 2 of 2 Show in CodeLens

```python
s = 1
print(s)
s = s + 2
print(s)
s = s + 3
print(s)
s = s + 4
print(s)
s = s + 5
print(s)
s = s + 6
print(s)
s = s + 7
print(s)
s = s + 8
```

```
1
3
6
10
15
21
28
36
45
55
```

Activity: 5 -- ActiveCode (ac2_14_3)

After the initial statement, where we assign `s` to 1, we can add the current value of `s` and the next number that we want to add (2 all the way up to 10) and then finally reassign that that value to `s` so that the variable is updated after each line in the code.

This will be tedious when we have many things to add together. Later you'll read about an easier way to do this kind of task.

Check your understanding

data-14-1: What is printed when the following statements execute?

```python
x = 12
x = x - 1
print(x)
```

○ A. 12

○ B. -1

● C. 11

○ D. Nothing. An error occurs because x can never be equal to x - 1.

Check me Compare me

✓ Yes, this statement sets the value of x equal to the current value minus 1.

Activity: 6 -- Multiple Choice (question2_14_1)

data-14-2: What is printed when the following statements execute?

```python
x = 12
x = x - 3
x = x + 5
x = x + 1
print(x)
```

○ A. 12

○ B. 9

● C. 15

○ D. Nothing. An error occurs because x cannot be used that many times in assignment statements.

Check me Compare me

✓ Yes, starting with 12, subtract 3, than add 5, and finally add 1.

Activity: 7 -- Multiple Choice (question2_14_2)

data-14-3: Construct the code that will result in the value 134 being printed.

Drag from here

Drop blocks here

```
mybankbalance = 100
mybankbalance = mybankbalance + 34
print(mybankbalance)
```

Check Reset

Perfect! It took you only one try to solve this. Great job!

Activity: 8 -- Parsons (pp2_14_1)

data-14-4: Which of the following statements are equivalent?

☑ A. x = x + y

□ B. y += x

□ C. x += x + y

☑ D. x += y

□ E. x++ y

Check me Compare me

✓ Correct.

A. x is updated to be the old value of x plus the value of y.

D. x is updated to be the old value of x plus the value of y.

Activity: 9 -- Multiple Choice (question2_14_3)

You have attempted 9 of 9 activities on this page

✓ Completed. Well Done!

2.12. Reassignment">

eassignment">

2.14. Hard-Coding">

| Back to top