Joshua Pollock

CS126 Section 2

Class ID: 2-159

3.2. How is the assignment operation different from other Python expressions?

Assignment operations in python are quite different from their counterparts. They allow for simplifying of code while also doing many basic operations that would take longer to write out. An example of these are +=. Here is some example coding:  
x=10  
y=20  
x+=y  
This will set x equal to x+y. It is the equivalent of x=x+y. x+=y is much faster to type and can simplify many lines of code down into one operation.

3.4. Which is not a literal value? (22.0, red, " blue ", or False)

To solve this problem, we need to know what a literal value actually is. A literal value is a value that is represented textually. Plugging these four different things into Python IDLE can easily point out which is not a literal value. The only one of these options to give an error when imputed was red. Inputting red gave this as the response:

>>>x= red

Traceback (most recent call last):

File "<pyshell#5>", line 1, in <module>

x= red

NameError: name 'red' is not defined

3.5. Give an example where a Python variable takes on different values over time.

Variables can change at any moment in time while coding. They offer a way to set a value into memory to be used or shown later. An example where a python variable would change its value over time would be something along the lines of:

X=2

X=X+500

In this example X begins at 2 but then adds 500 to itself making X now equal to 502.

X=’race’

X=x+’car’  
This is another example using strings. X begins as the string race but will add the string car to it and now display as ‘racecar’.