Assignment operator:-

The assignment operator will work from right to left.

L = R Must be variable only value

Variable

Expression

Function calling

Object.

Example:-

a=10

b=a

c=a+b

x=len(“bunny”)

e=Employee()

In Python we can declare multiple variables in one line.

Example:-

a,b,c = 10,20,25

Print(a) #10

10

Print(b) #20

20

Print(c) #25

25

(Or)

We can call all the variables in one line like

Print(a,b,c) #10,20,25

Example 2:-

a,b,c,d,e=10,”Bunny”,10.25,True,None

Print(a) #10

Print(type(a)) # < class ‘int’>

Print(b) #Bunny

Print(type(b)) #< class ‘string’>

Print(c) #10.25

Print(type(c)) #<class ‘float’>

Print(d) #True

Print(type(d)) #<class ‘bool’>

Print(e) #None

Print(type(e)) #<class ‘None type’>

In Python variables are immutable:-

which means Everything in **Python** is an object. ... Simple put, a mutable object can be changed after it is created, and an **immutable** object can't. Objects of built-in types like (int, float, bool, str, tuple, unicode) are **immutable**. Objects of built-in types like (list, set, dict) are mutable. Custom classes are generally mutable.

Example:-

a=10

Print(a) # 10 Print(id(a)) # 546821328442

a=a+5

Print(a) # 15

Print(id(a)) # 8545895454545

|  |
| --- |
| In Python variable can hold any Big Value. |
|  |

Example:-

X= 5765454655468468434684384484

Print(x+6) #5765454655468468434684384490

In Python if two variables are holding same values so the variable will share same memory instead of creating new memory. For suppose if have any doubt we can check by id’s of a variable.

Example:-

X=10

Print(x) #10

Print(x+20) #30

a,b,c = x,45,86

print(a,b,c) #10,45,86