

# Lección 4

taller

# Constantes

Valores arreglados, pueden ser números, letras, y palabras llamadas constantes por que sus valores no cambian

Valores números son constantes eso es lo que esperamos

Palabras (String) al usar comilla simple (') o doble (")

```
C:\Users\acer>python
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print(123)
123
>>> print(98.6)
98.6
>>> print("hola mundo")
hola mundo
>>>
```

# Constantes

# Variables

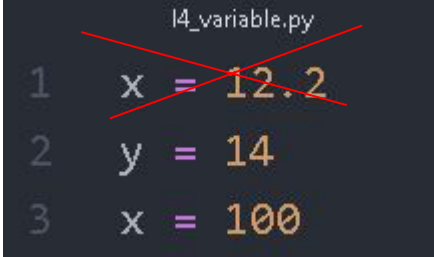
Variable es la forma en la cual guardamos información en el programa

```
l4_variable.py
1  x = 12.2
2
3  y = 14
```

# Variables

Regla de almacenamiento

```
l4_variable.py
1  x = 12.2
2  y = 14
3  x = 100
```

A code snippet from a file named l4\_variable.py. The code consists of three lines: 1 x = 12.2, 2 y = 14, and 3 x = 100. The first line is crossed out with a red 'X'.

# Variables

Regla de nombres a las variables

deben comenzar con una letra o guión bajo \_

pueden ser letras, números y guión bajo \_

Correcto:

spam

eggs

spam23

\_bot23

Mal:

23spam

#eggs

var.12

casa hola

Diferentes:

spam

Spam

SPAM

RECAP.PY

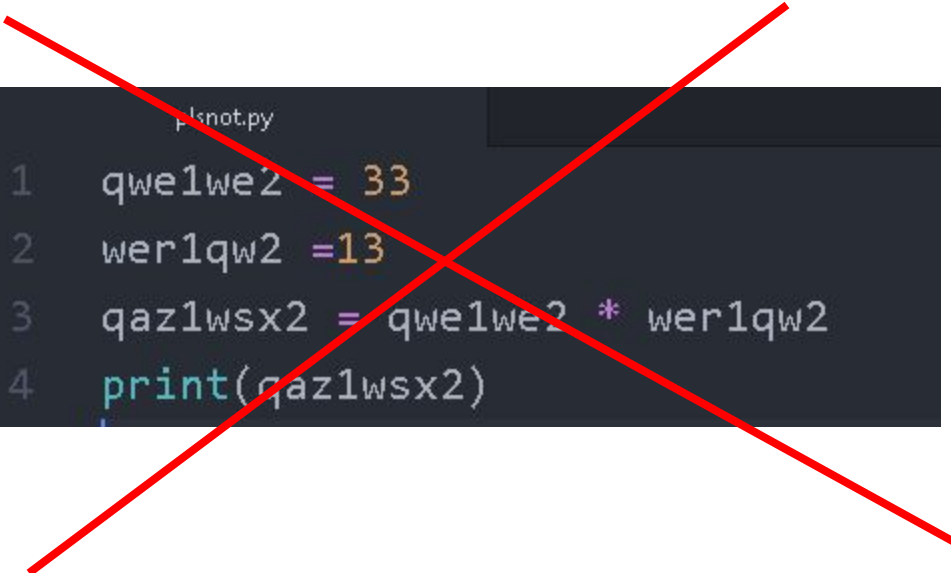
```
1 x = 2
2 x = x + 2
3 print(x)
4 print("x")
5 print(2)
6 print("2")
```

# Variables

```
plsnot.py
1  qwe1we2 = 33
2  wer1qw2 =13
3  qaz1wsx2 = qwe1we2 * wer1qw2
4  print(qaz1wsx2)
```



# Variables



```
plsnnot.py
1 qwe1we2 = 33
2 wer1qw2 = 13
3 qaz1wsx2 = qwe1we2 * wer1qw2
4 print(qaz1wsx2)
```

```
now_y.py
1 a = 33
2 b = 13
3 c = a * b
4 print(c)
```


# Variables

h\_study.py

```
1  Horas = 1.5
2  Cantidad_Clases = 13
3  Estudio_semanal = Horas * Cantidad_Clases
4  print("cantidad de horas destinadas a estudiar son",
5  Estudio_semanal,"a la semana")
```

# Declaración de Asignaciones

usamos el símbolo = para asignar valor a las variables, el valor asignado es el que está a la derecha del signo igual y la variable almacena el resultado



```
2 x = 3.9 * x * ( 1 - x )
```

x debe ser una variable con un valor asignado previamente para así resolver la expresión

The diagram shows the equation  $2x = 3.9 * x * (1 - x)$  on a dark background. Two red arrows point down to the variable  $x$  in the term  $3.9 * x$  and the variable  $x$  in the term  $(1 - x)$ . Four grey arrows point from the equation to a point below: one from the coefficient 3.9, one from the  $x$  in  $3.9 * x$ , one from the minus sign in  $(1 - x)$ , and one from the  $x$  in  $(1 - x)$ . Two more grey arrows point from that point towards the bottom of the slide.

$$2x = 3.9 * x * (1 - x)$$

expression\_as.py

```
1 x = 2
2 x = 3.9 * x * ( 1 - x )
3 print(x)
```