PYTHON

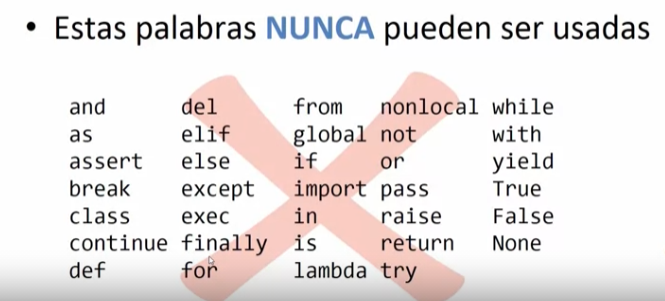
FUNCIONA EN WINDOWS MAC O LINUX

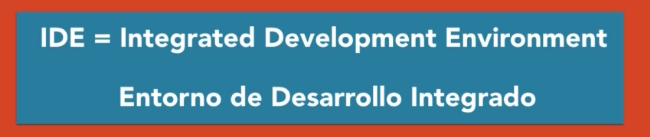
SINTAXIS LIMPIA

SIN COMPILADORES

VARIABLE ES UN CONTENEDOR PARA UN VALOR

EXISTEN PALABRAS RESERVADAS

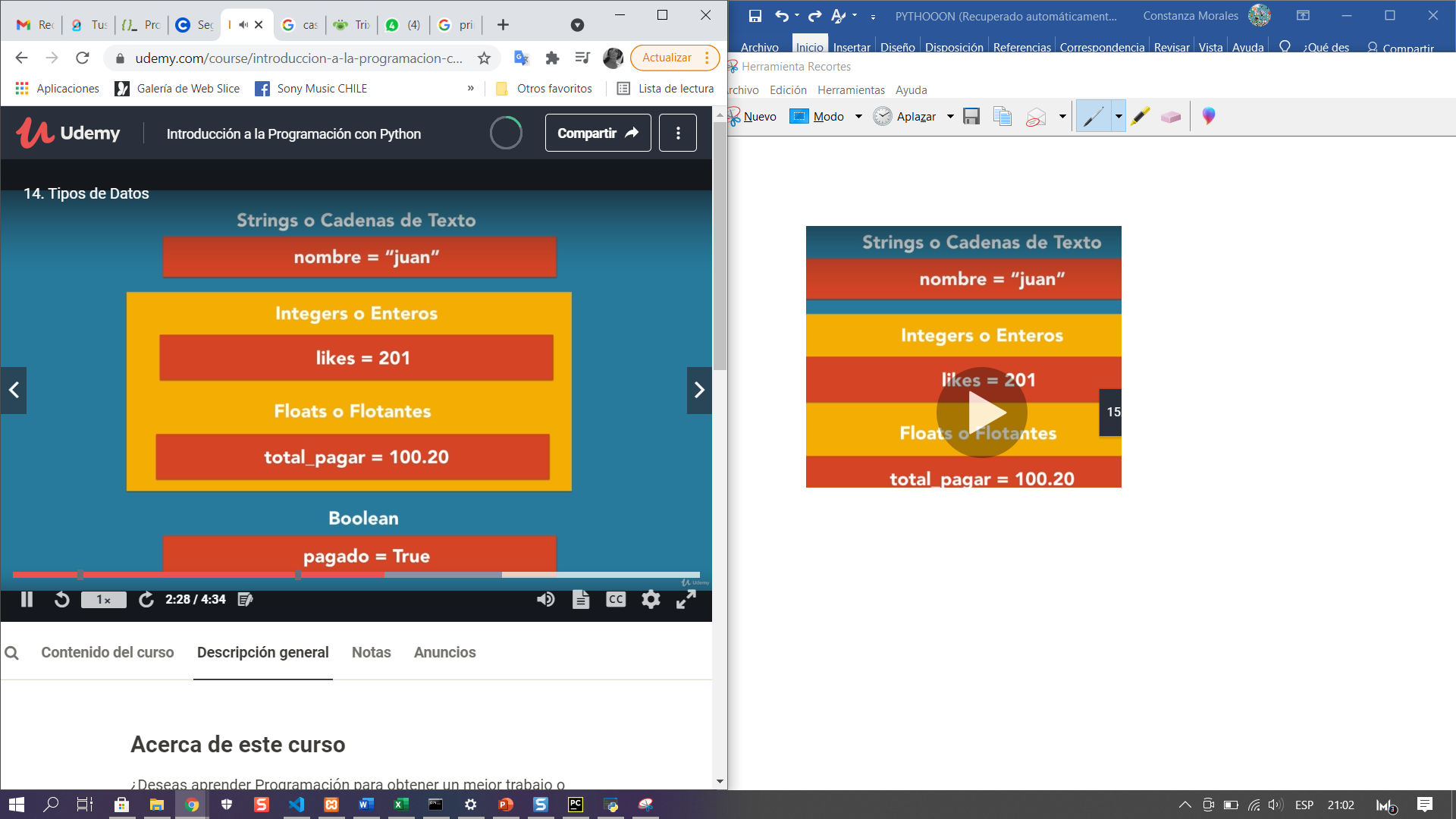




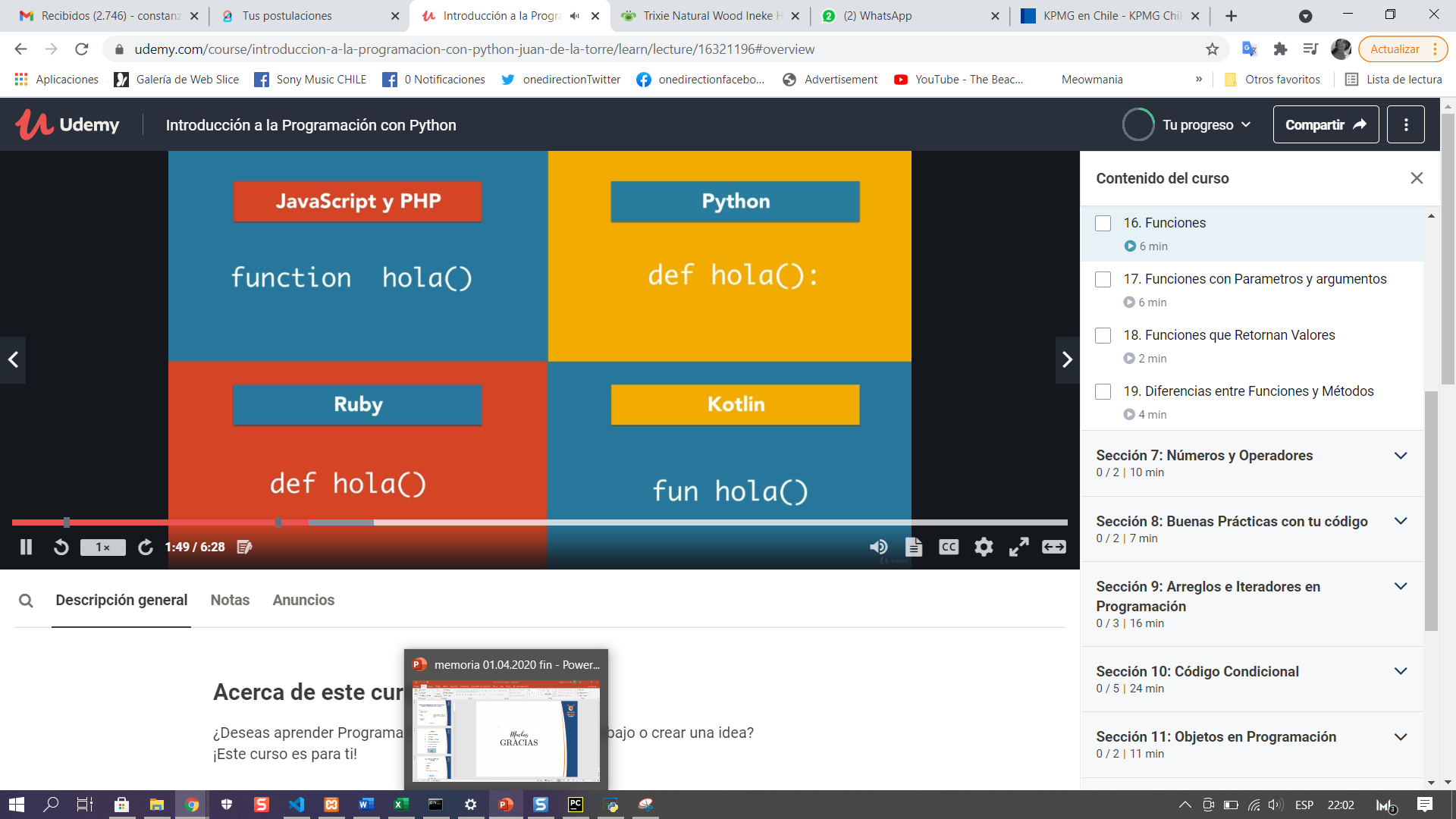
* Tipos de datos:

Cada lenguaje tendrá tipos de datos

Clasificación de los valores



Funciones en diferentes lenguajes:



**Árbol de navidad**

import turtle

screen = turtle.Screen()

screen.setup(800,600)

circle = turtle.Turtle()

circle.shape('circle')

circle.color('red')

circle.speed('fastest')

circle.up()

square = turtle.Turtle()

square.shape('square')

square.color('green')

square.speed('fastest')

square.up()

circle.goto(0,280)

circle.stamp()

k = 0

for i in range(1, 17):

    y = 30\*i

    for j in range(i-k):

        x = 30\*j

        square.goto(x,-y+280)

        square.stamp()

        square.goto(-x,-y+280)

        square.stamp()

    if i % 4 == 0:

        x =  30\*(j+1)

        circle.color('red')

        circle.goto(-x,-y+280)

        circle.stamp()

        circle.goto(x,-y+280)

        circle.stamp()

        k += 2

    if i % 4 == 3:

        x =  30\*(j+1)

        circle.color('yellow')

        circle.goto(-x,-y+280)

        circle.stamp()

        circle.goto(x,-y+280)

        circle.stamp()

square.color('brown')

for i in range(17,20):

    y = 30\*i

    for j in range(3):

        x = 30\*j

        square.goto(x,-y+280)

        square.stamp()

        square.goto(-x,-y+280)

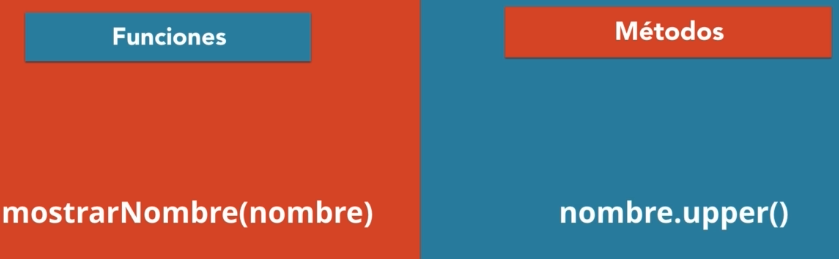
        square.stamp()

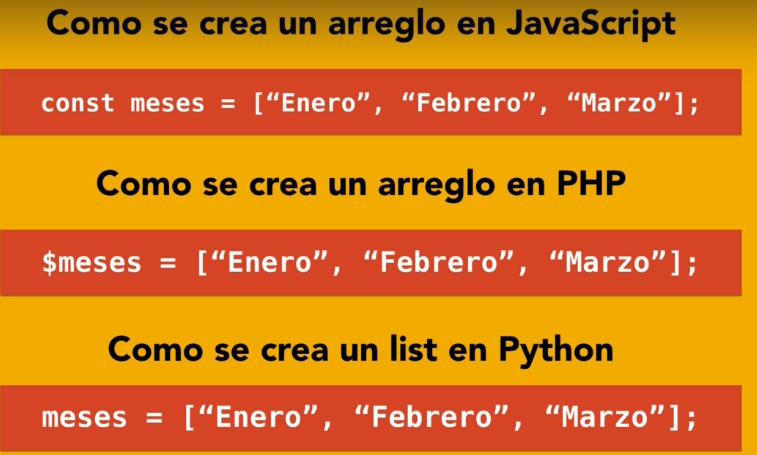
turtle.exitonclick()

**Tipos de datos**

* ¿Qué valores puede representar el computador?
* 1 y 0 lo que entiende el computador, datos binarios.

|  |  |  |
| --- | --- | --- |
| Tipos numéricos | Tipos texto | Tipos lógicos |
| * Int   Negativos y positivos   * Float | * Str   Secuencias o cadenas de caracteres | * Bool   Verdadero o falso  Bool(0) = false  Bool(“”)= false |





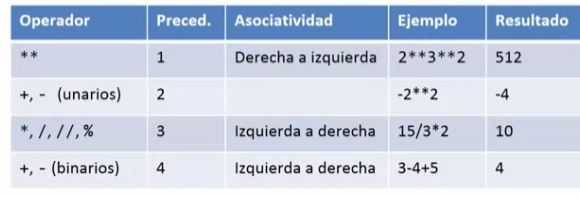


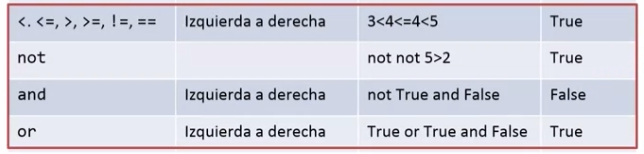
**Calculando valores operadores y expresiones**

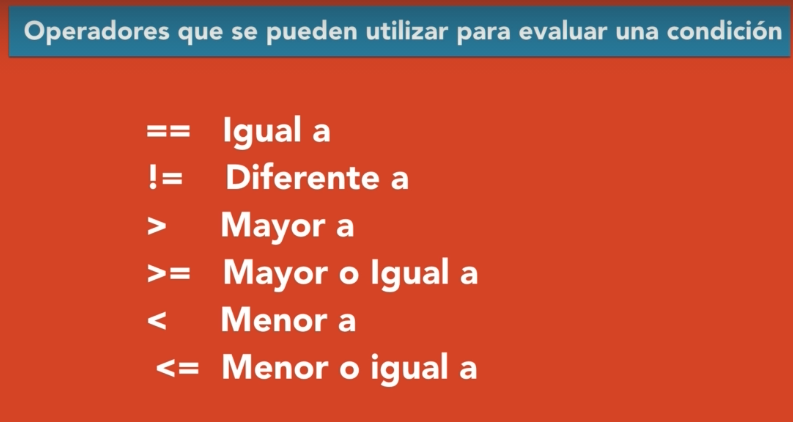
// resto

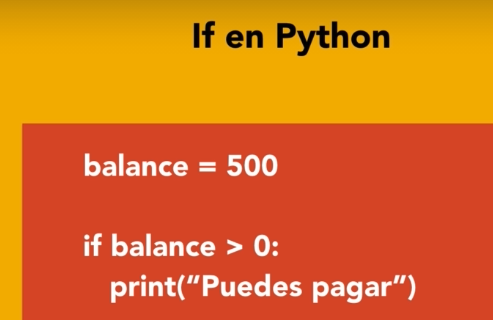
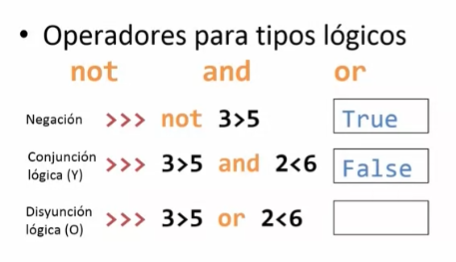
% módulo

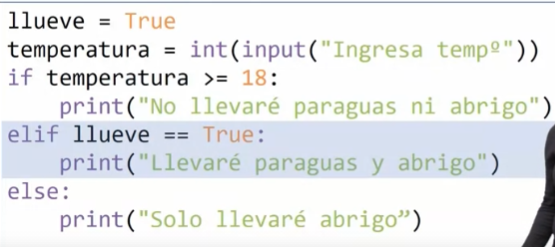
\*\* exponenciación

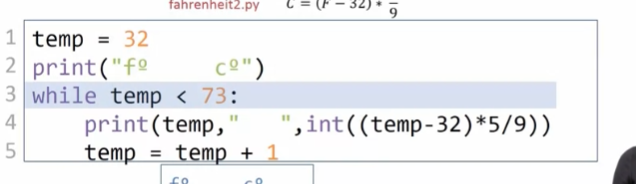


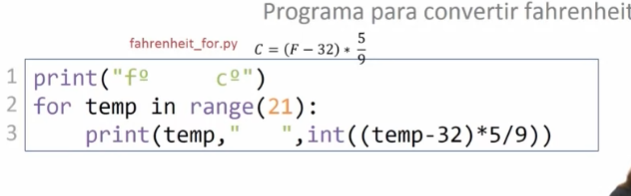


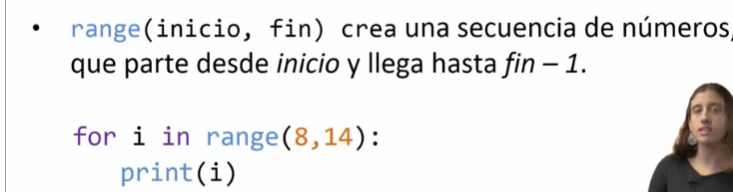


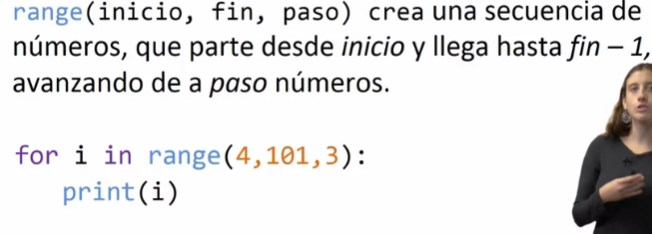


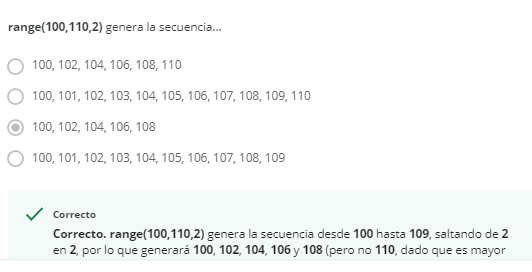


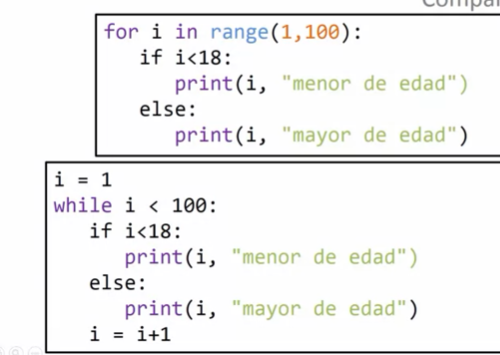












Es mejor while por que for es un rango

x = 1  
i = 0  
while i < 4:  
 x = x \* 2  
 i += 1 # es lo mismo que i = i + 1  
 print(x)  
print(x)

2, 4, 8, 16

WHILE CONSIDERA EL 0

a = 3

for i in range(2, 3):

    a = a \* i

3\*2

a = 2

for i in range(1, 4):

    a = i \*\* a

print(a)

1, 4, 9

Número primo

def es\_primo(numero):

  primo = True

  c = 0

  for i in range(1, numero + 1):

    if numero % i == 0:

      c = c + 1

      if c==2:

        primo = True

      else:

        primo = False

  return primo

Exponenciación

numero= int(input("ingrese un numero:"))  
if numero%2==0:  
 cuadrado= numero \*\* 2  
 print(cuadrado)  
else:  
 cubo= numero \*\* 3  
 print(cubo)