

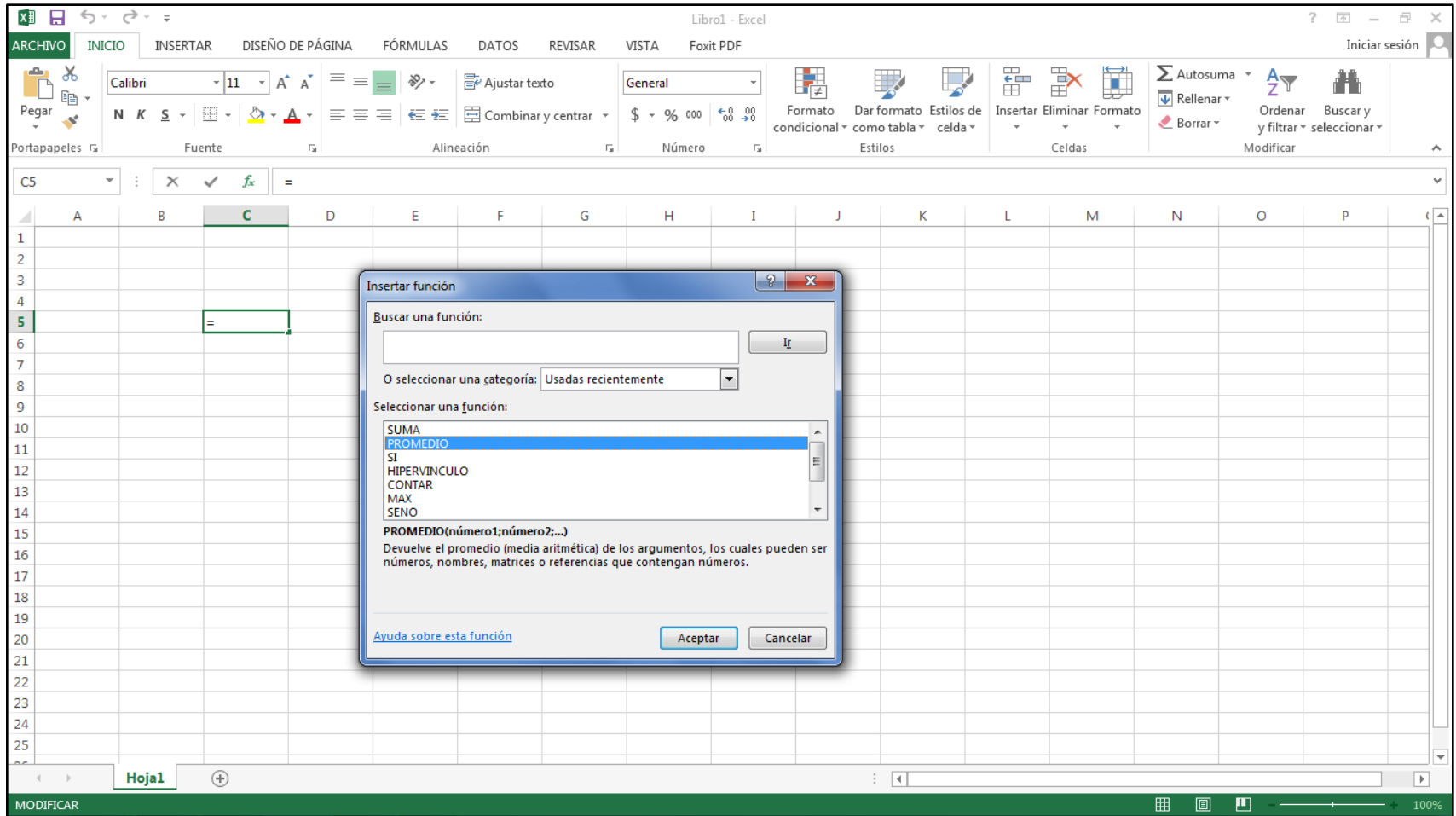
# Algoritmia y Programación

Oscar Bedoya

`oscar.bedoya@correounivalle.edu.co`

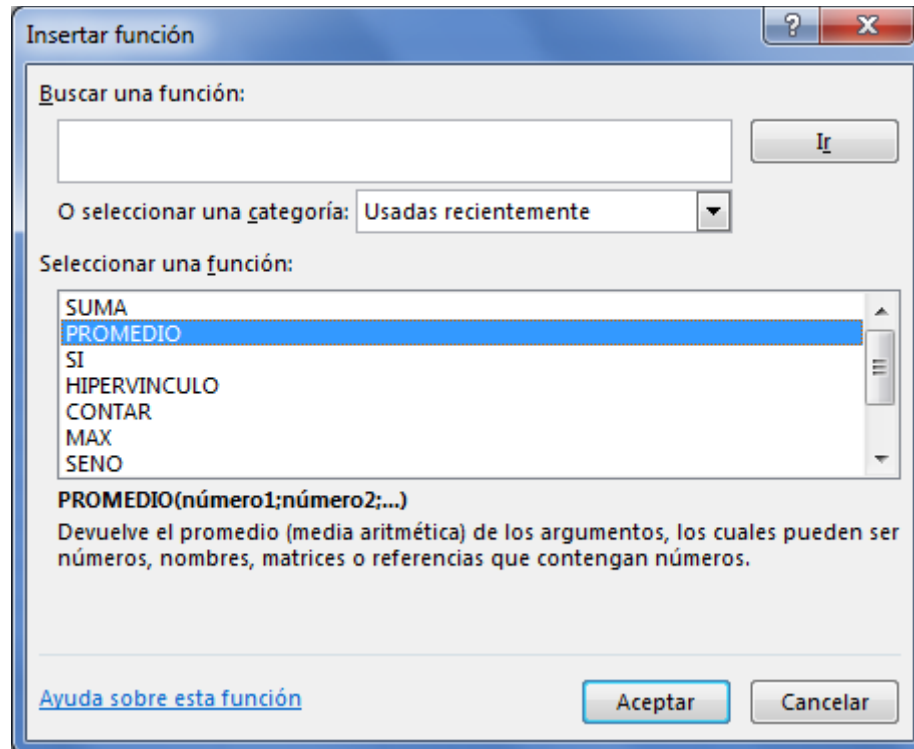
# Funciones

# Funciones en Python



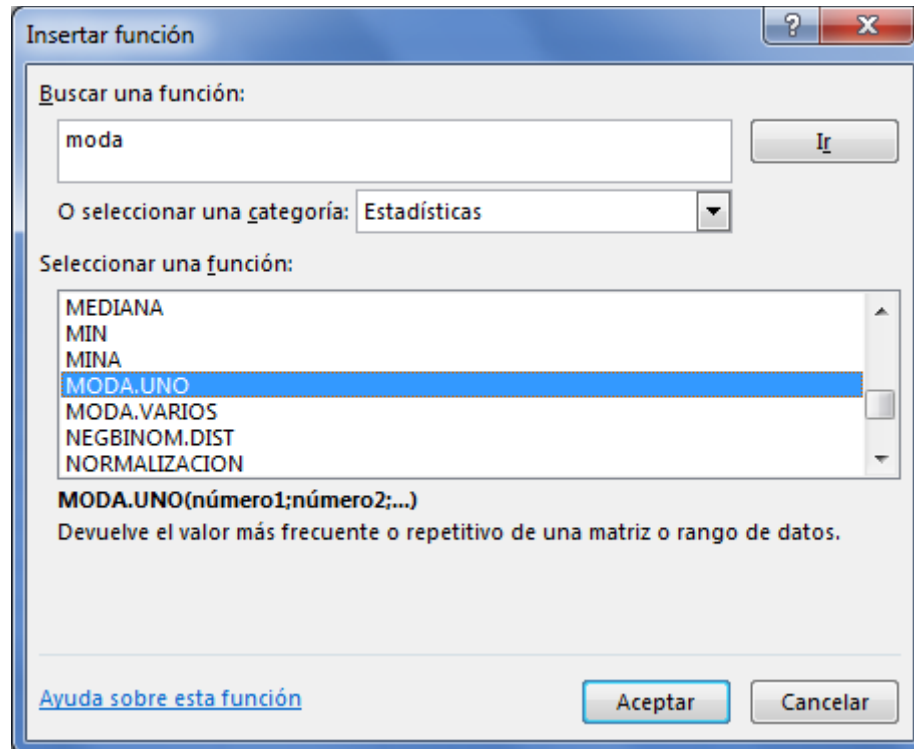
# Funciones en Python

---



# Funciones en Python

---



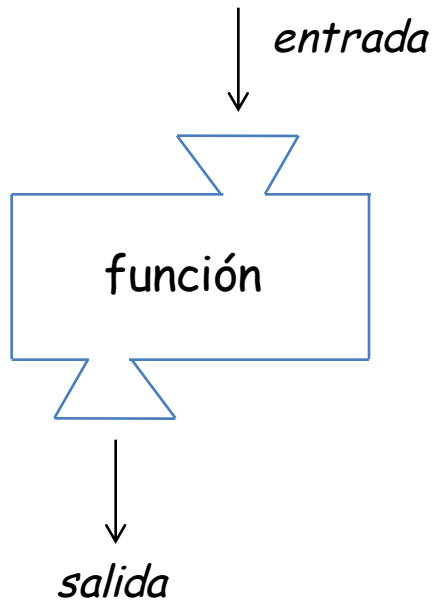
# Funciones en Python

---

Función	Descripción
<code>promedio(x,y,z)</code>	Devuelve el promedio de tres valores
<code>mayor(x,y,z)</code>	Devuelve el mayor valor entre x, y, z
<code>cos(x)</code>	Devuelve el coseno del ángulo x
<code>raíz(x)</code>	Devuelve la raíz cuadrada de x

# Funciones en Python

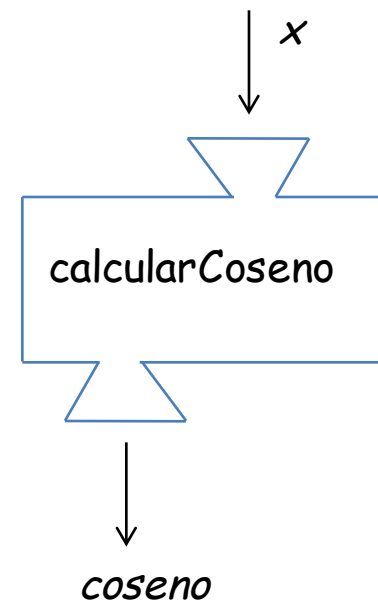
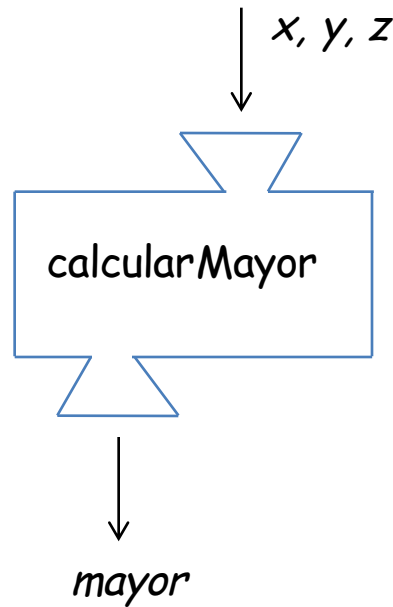
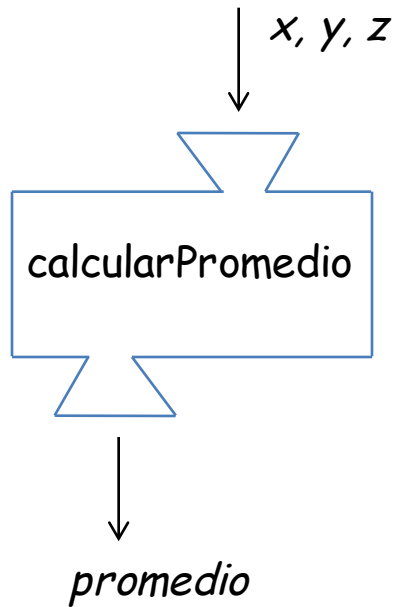
---



Una función en Python se **especializa** en realizar un cálculo determinado

# Funciones en Python

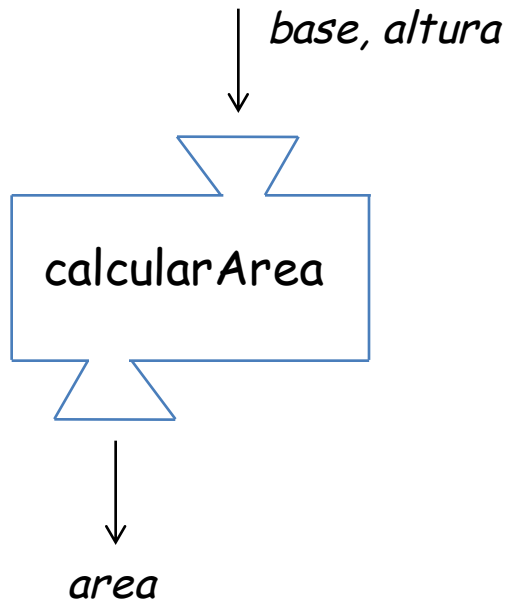
---





# Funciones en Python

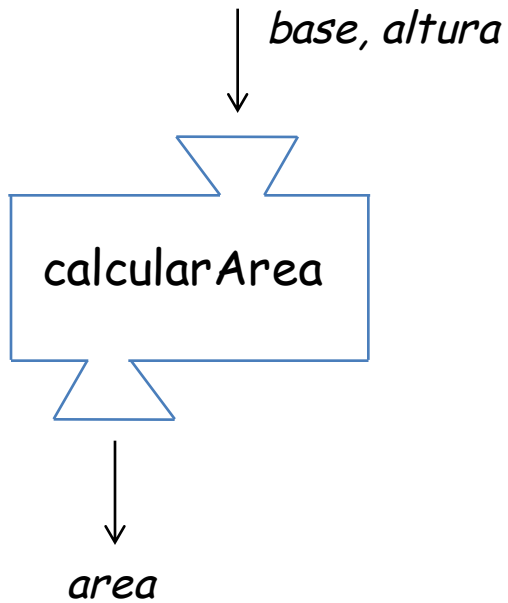
---



Una función en Python se **especializa** en realizar un cálculo determinado

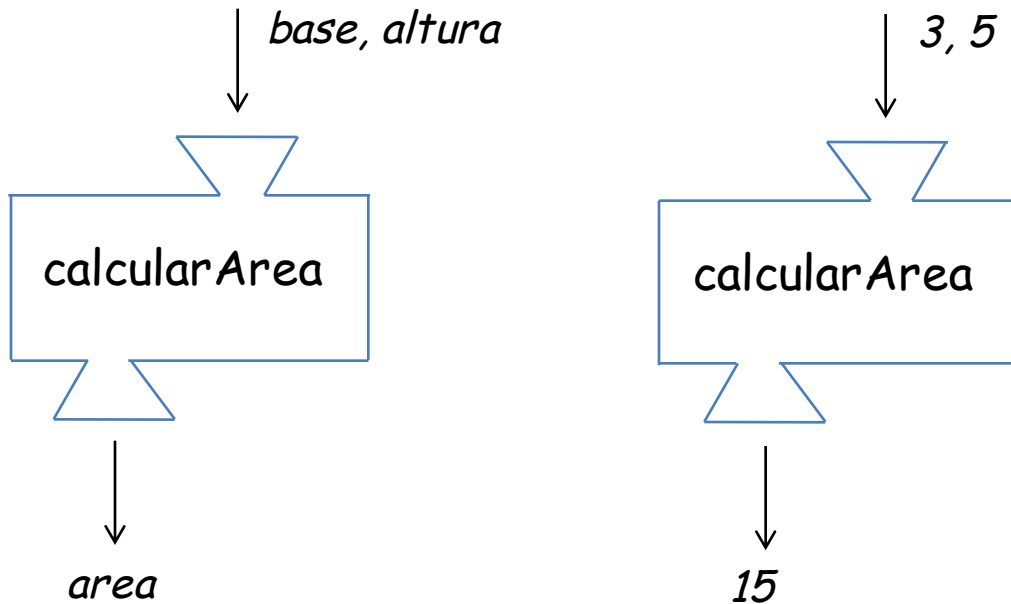
# Funciones en Python

---



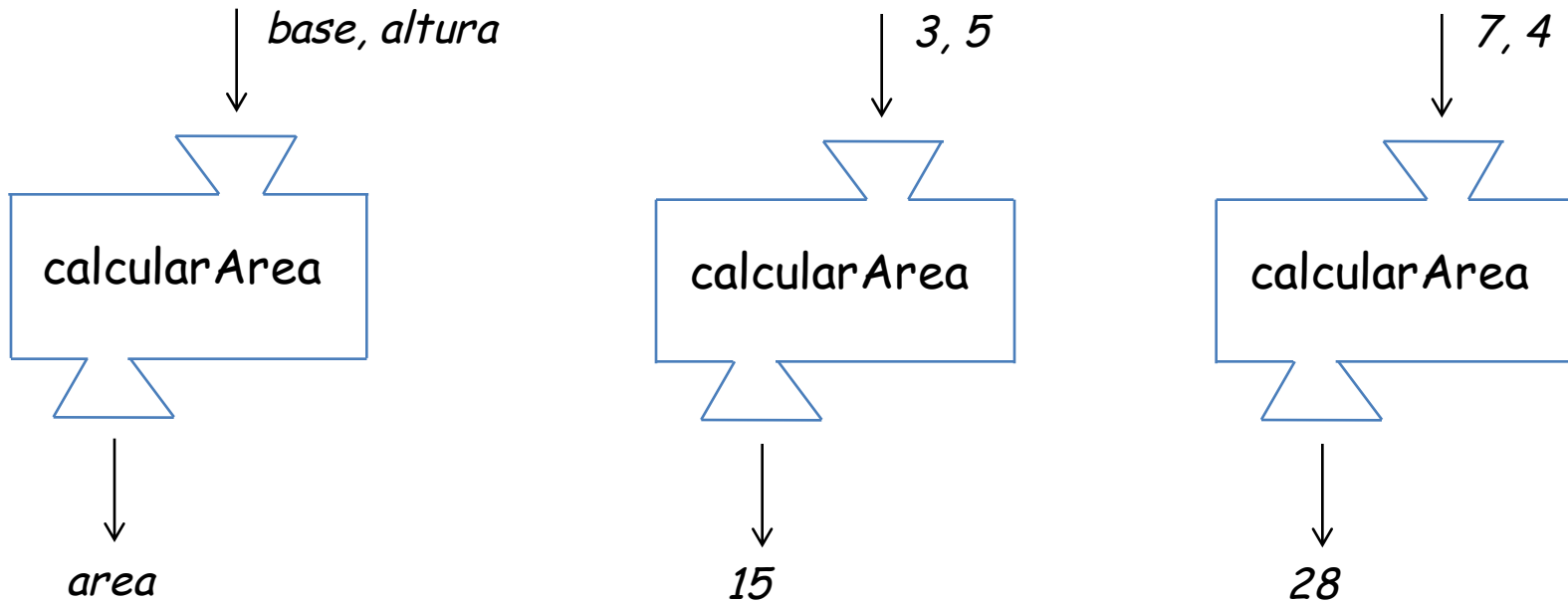
# Funciones en Python

---



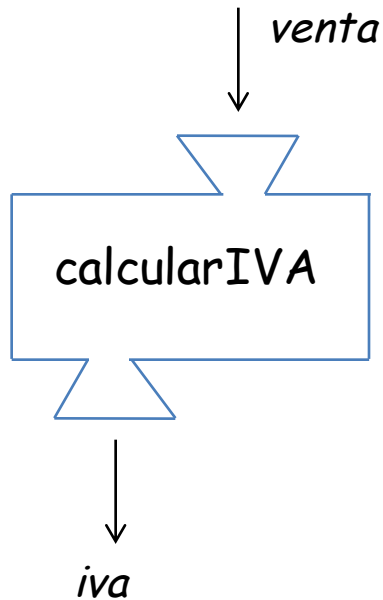
# Funciones en Python

---



# Funciones en Python

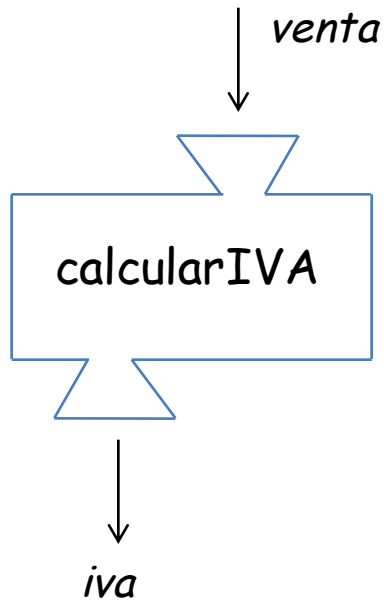
---



Una función en Python se **especializa** en realizar un cálculo determinado

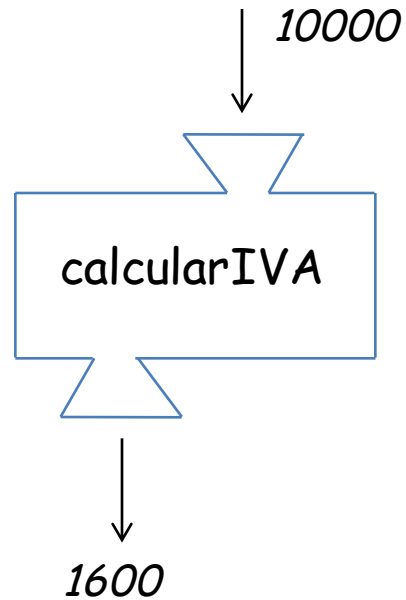
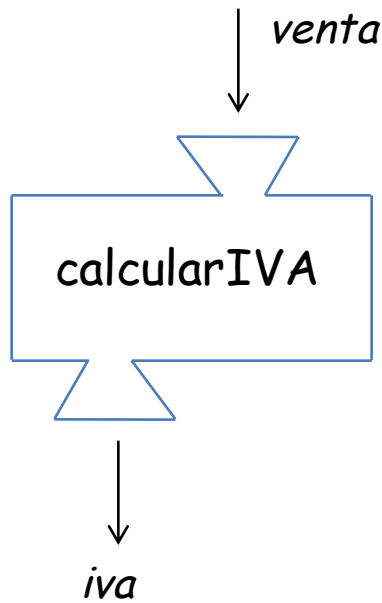
# Funciones en Python

---



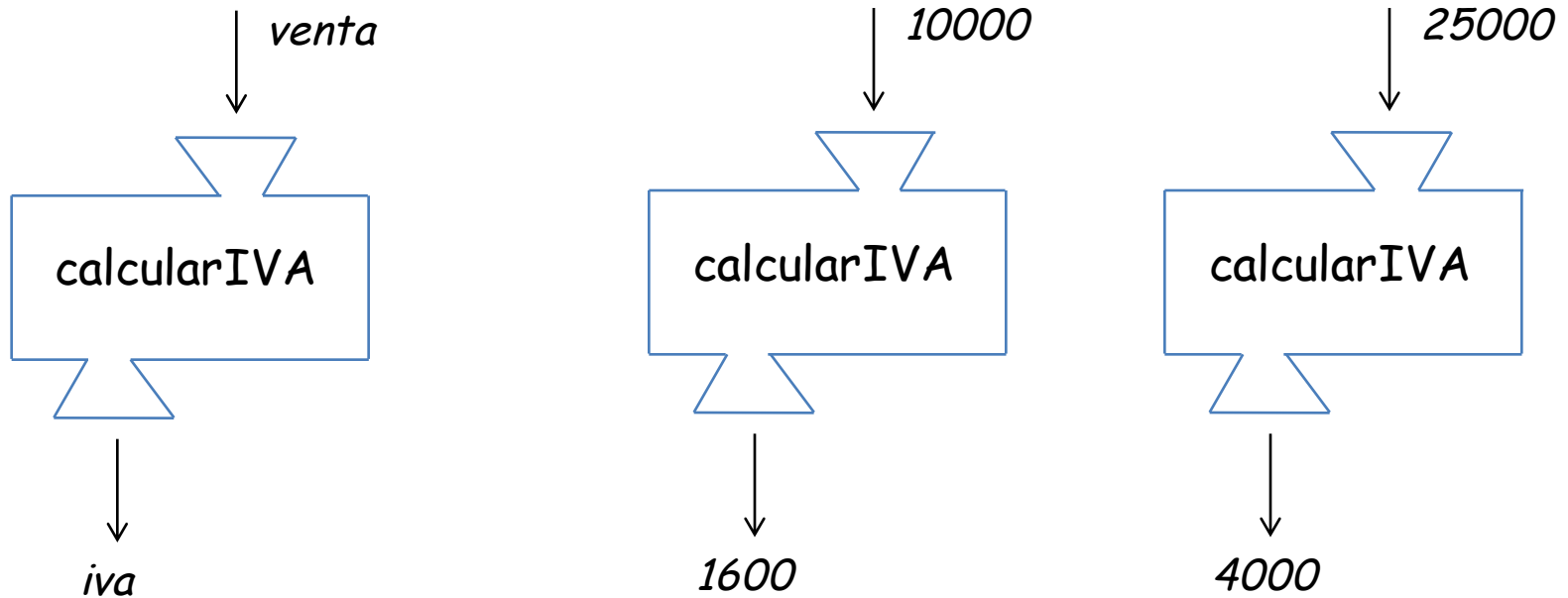
# Funciones en Python

---



# Funciones en Python

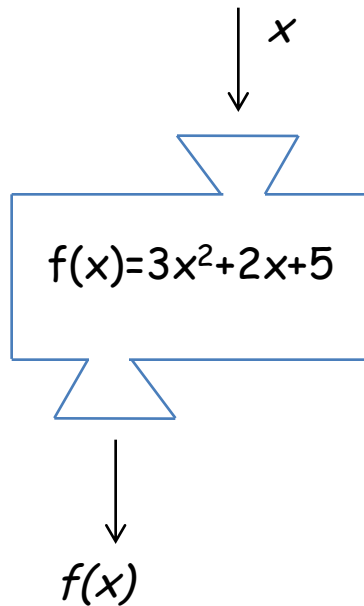
---





# Funciones en Python

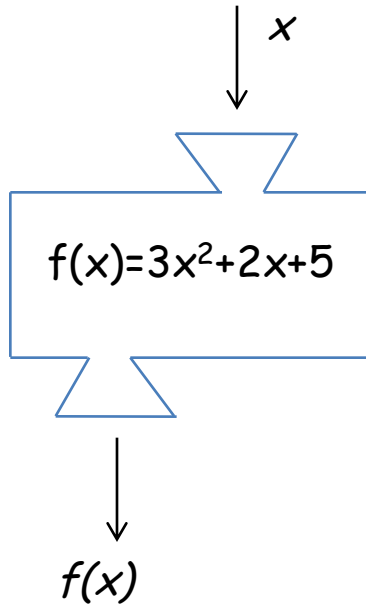
---



Una función en Python se **especializa** en realizar un cálculo determinado

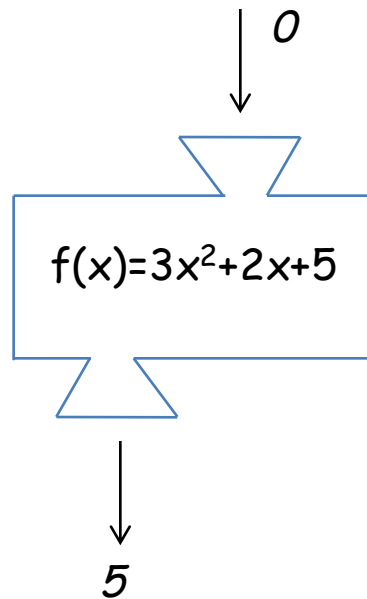
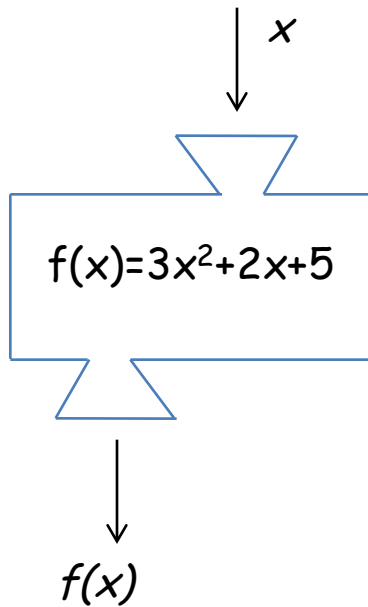
# Funciones en Python

---



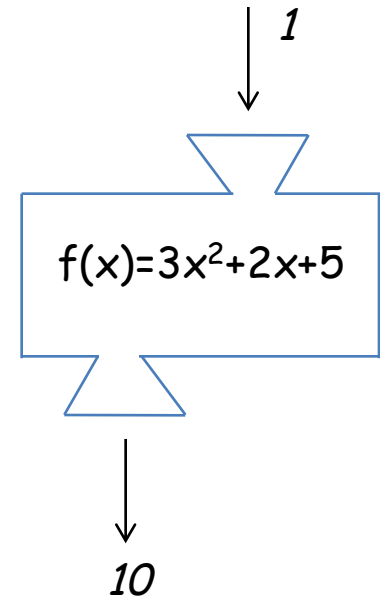
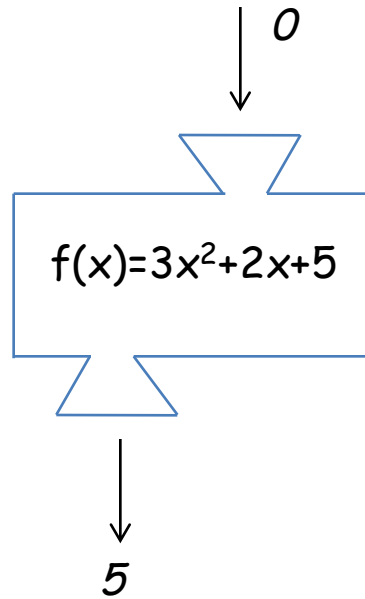
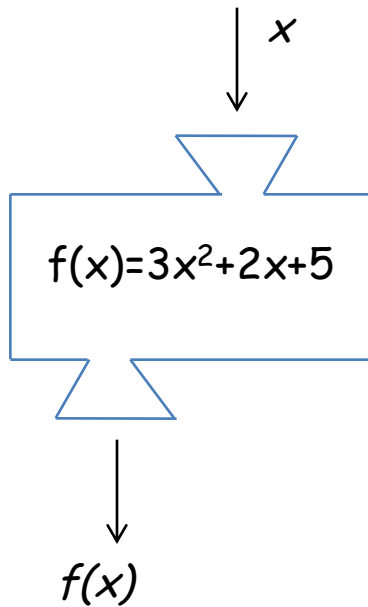
# Funciones en Python

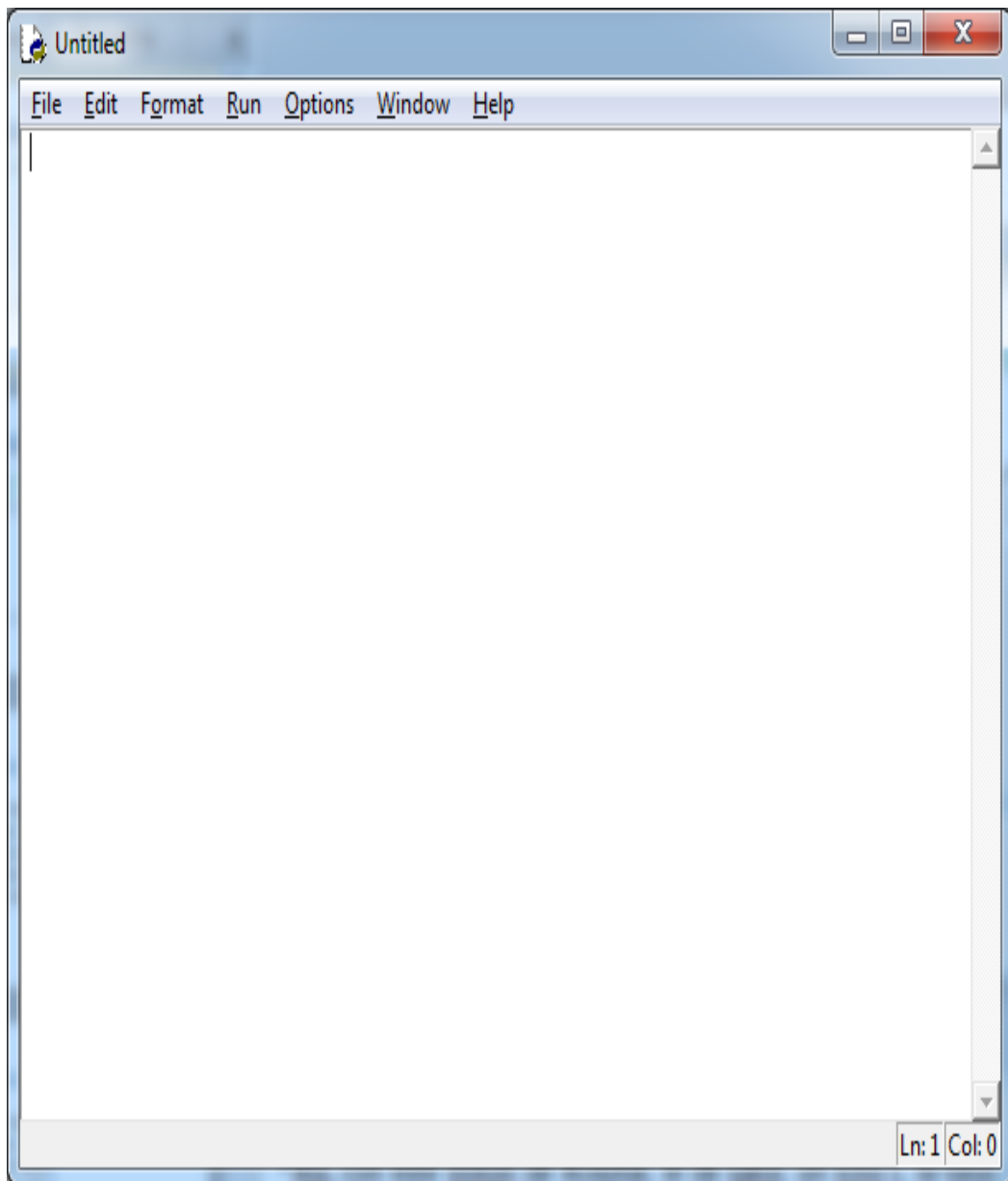
---

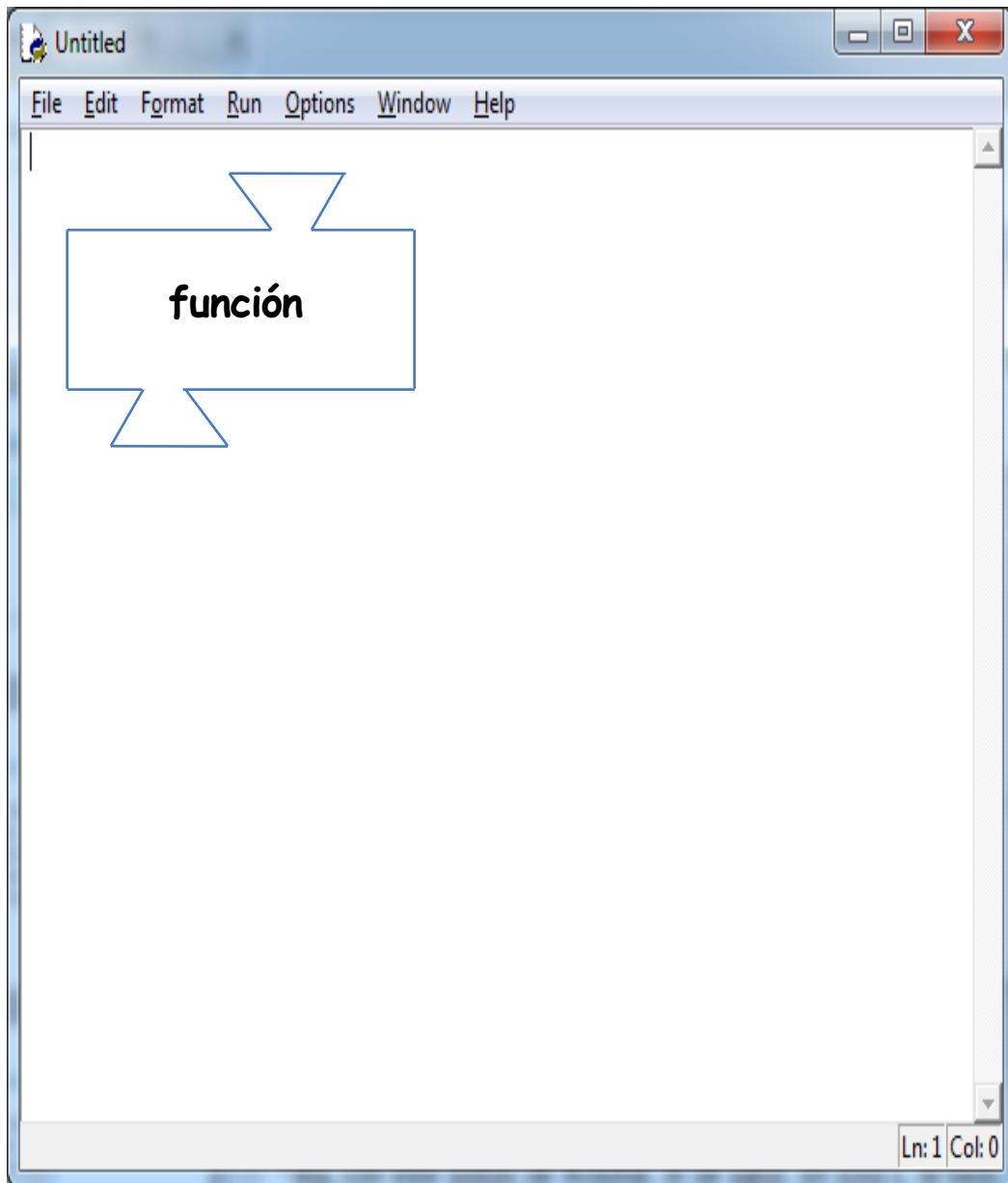


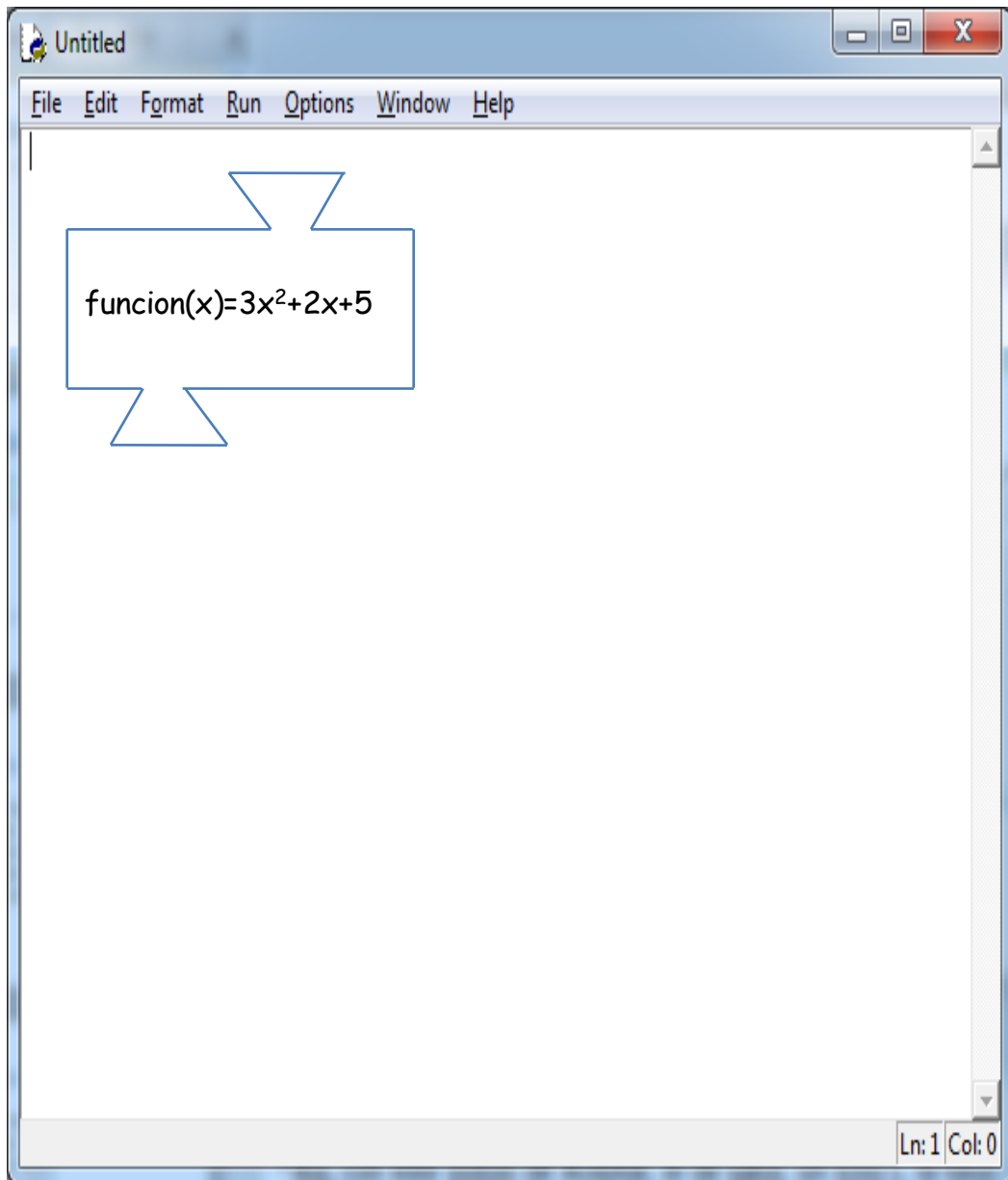
# Funciones en Python

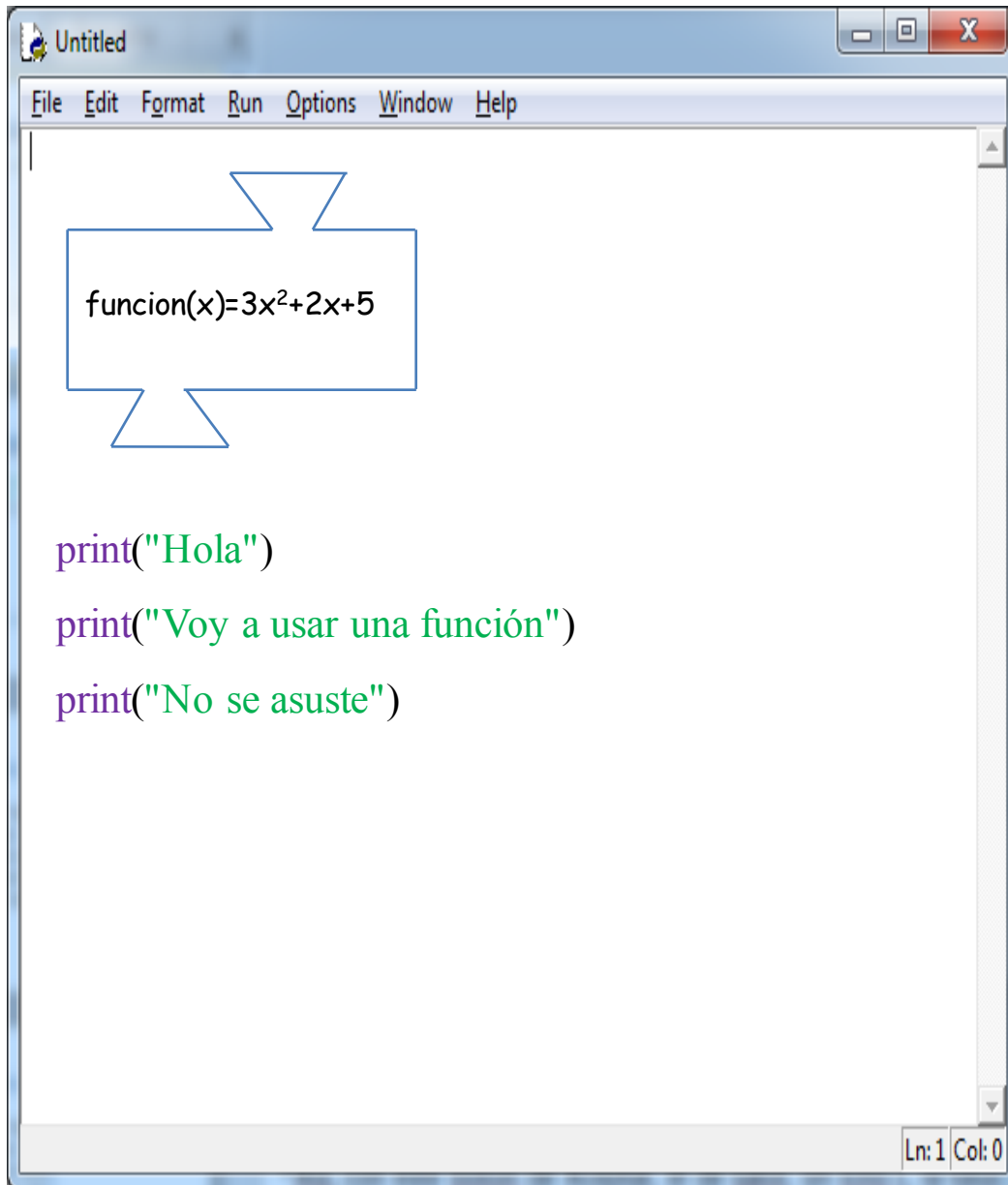
---



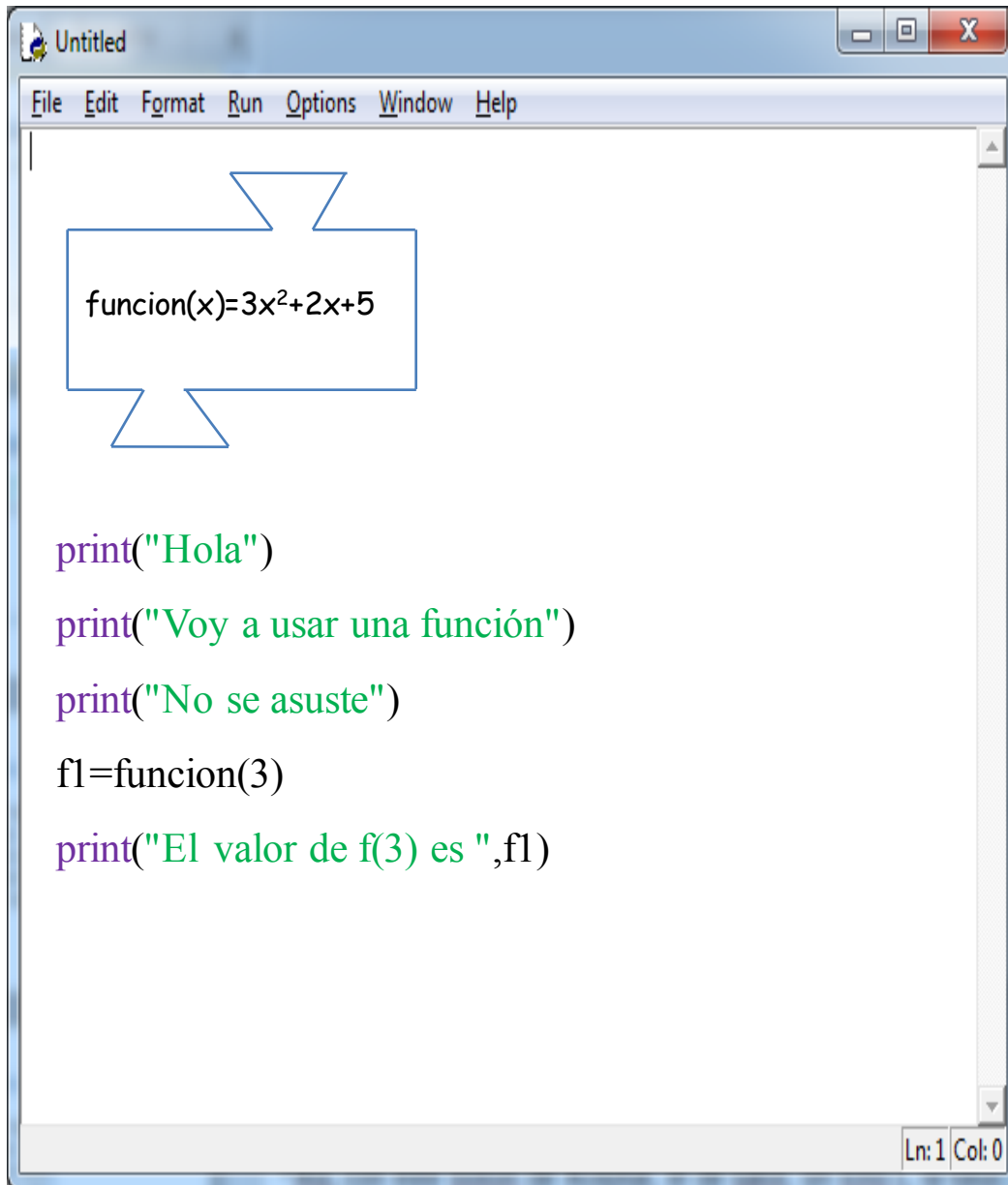


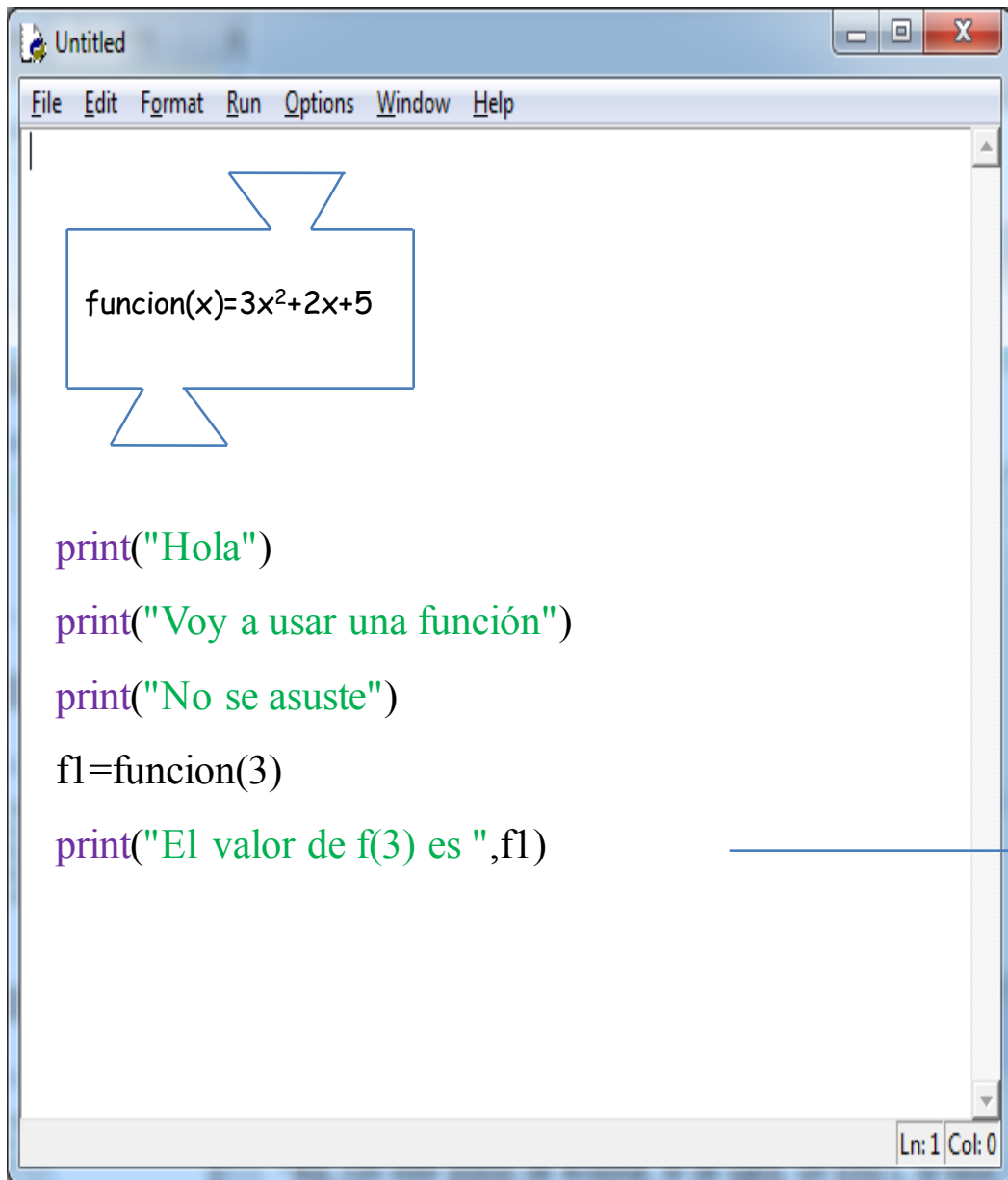




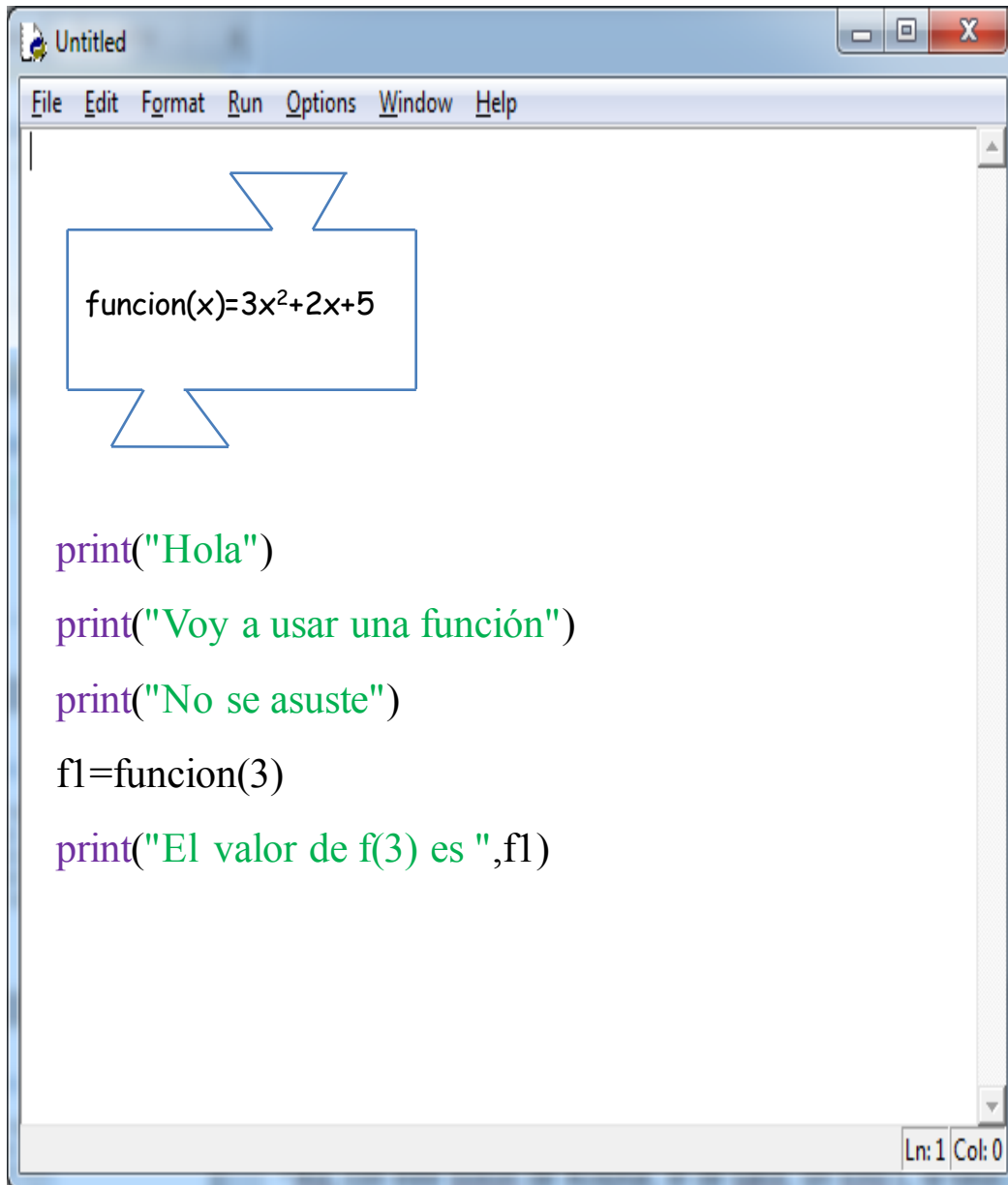


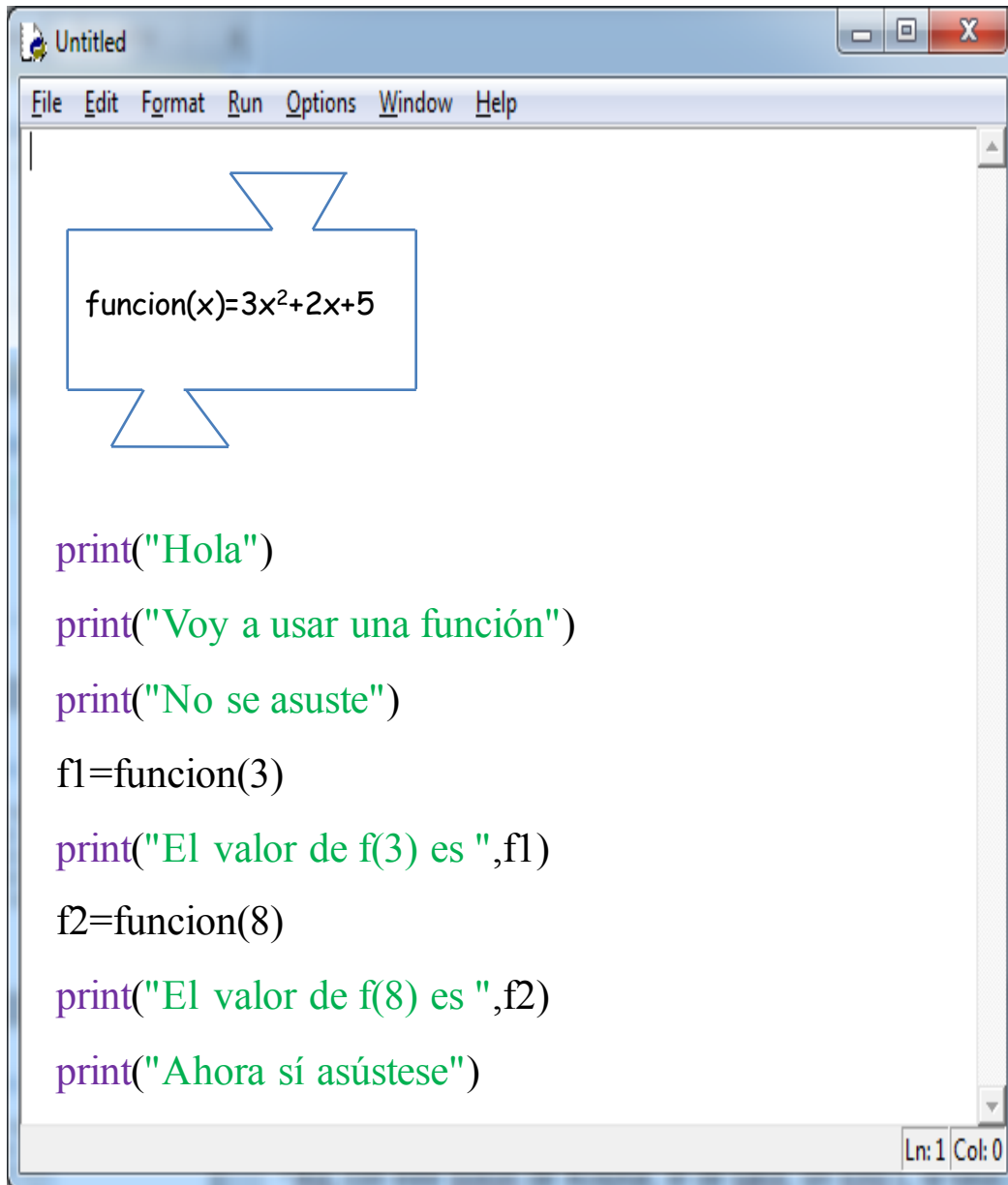


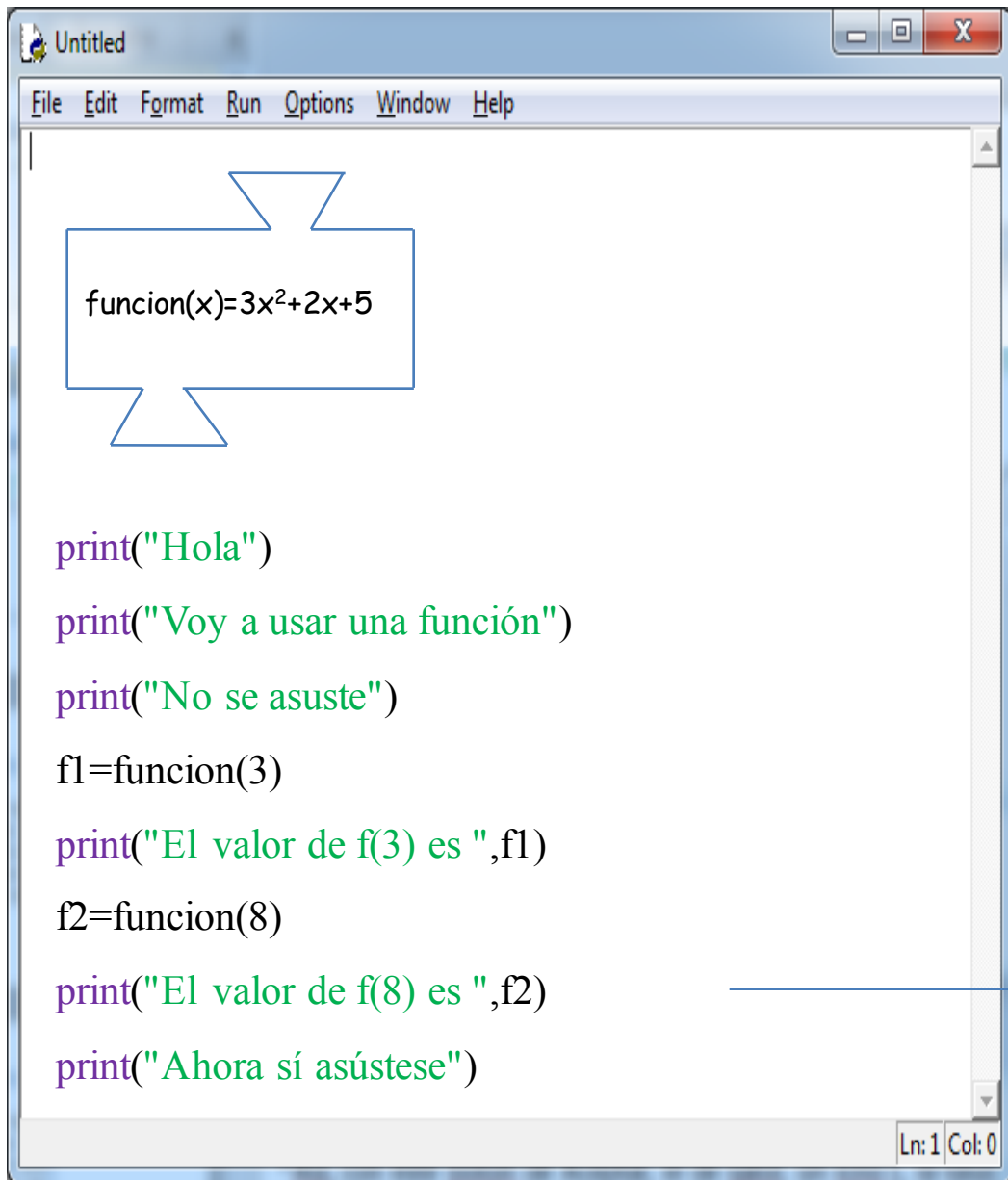




El valor de  $f(3)$  es 38



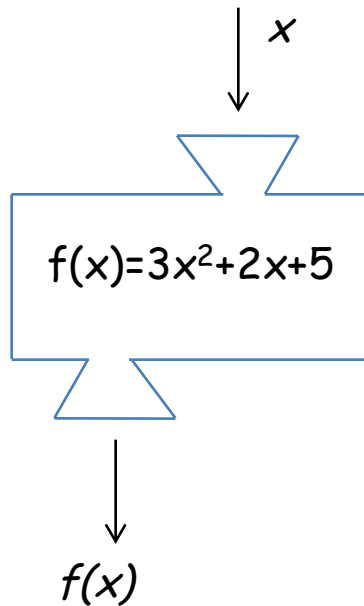




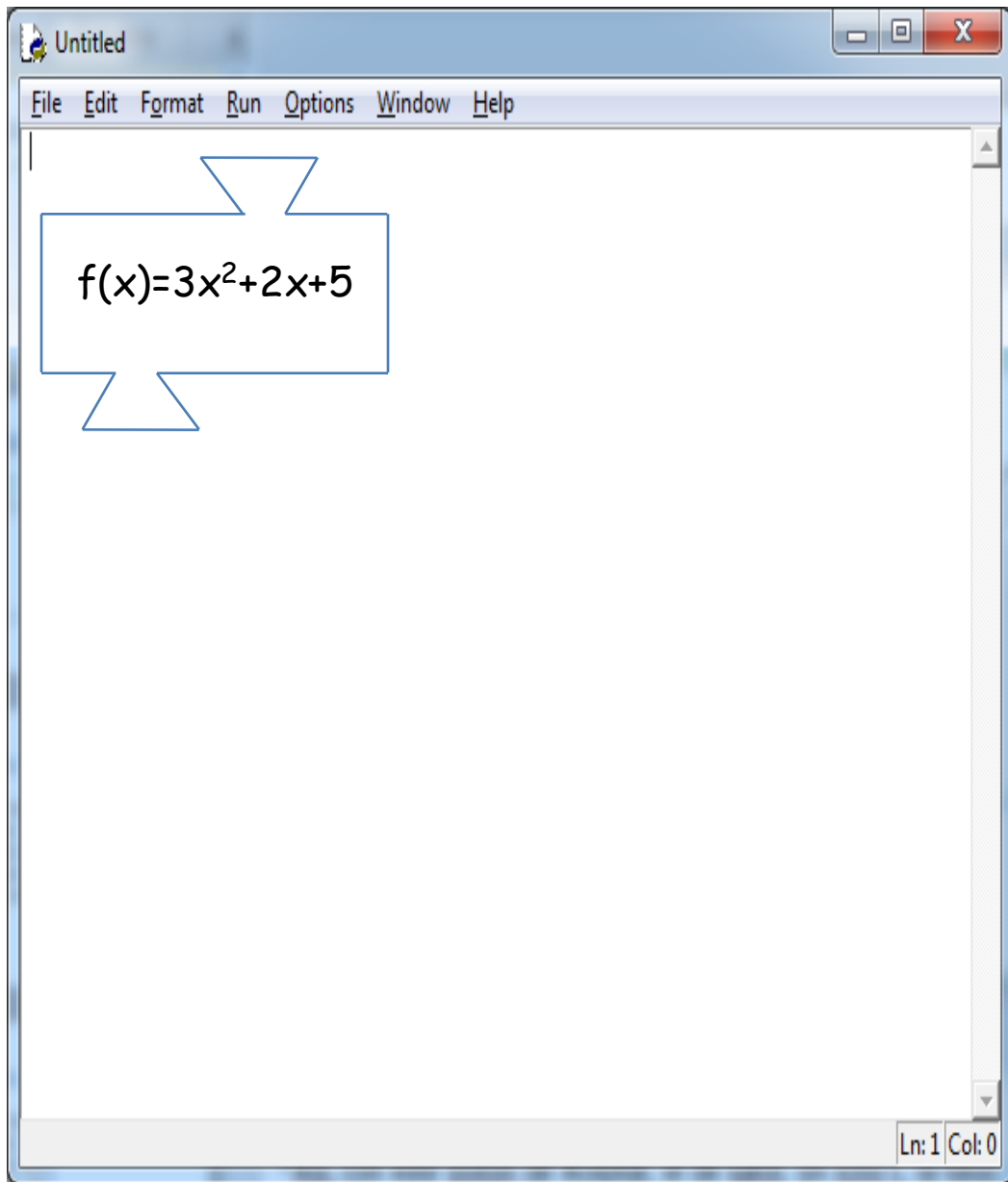
El valor de  $f(8)$  es 213

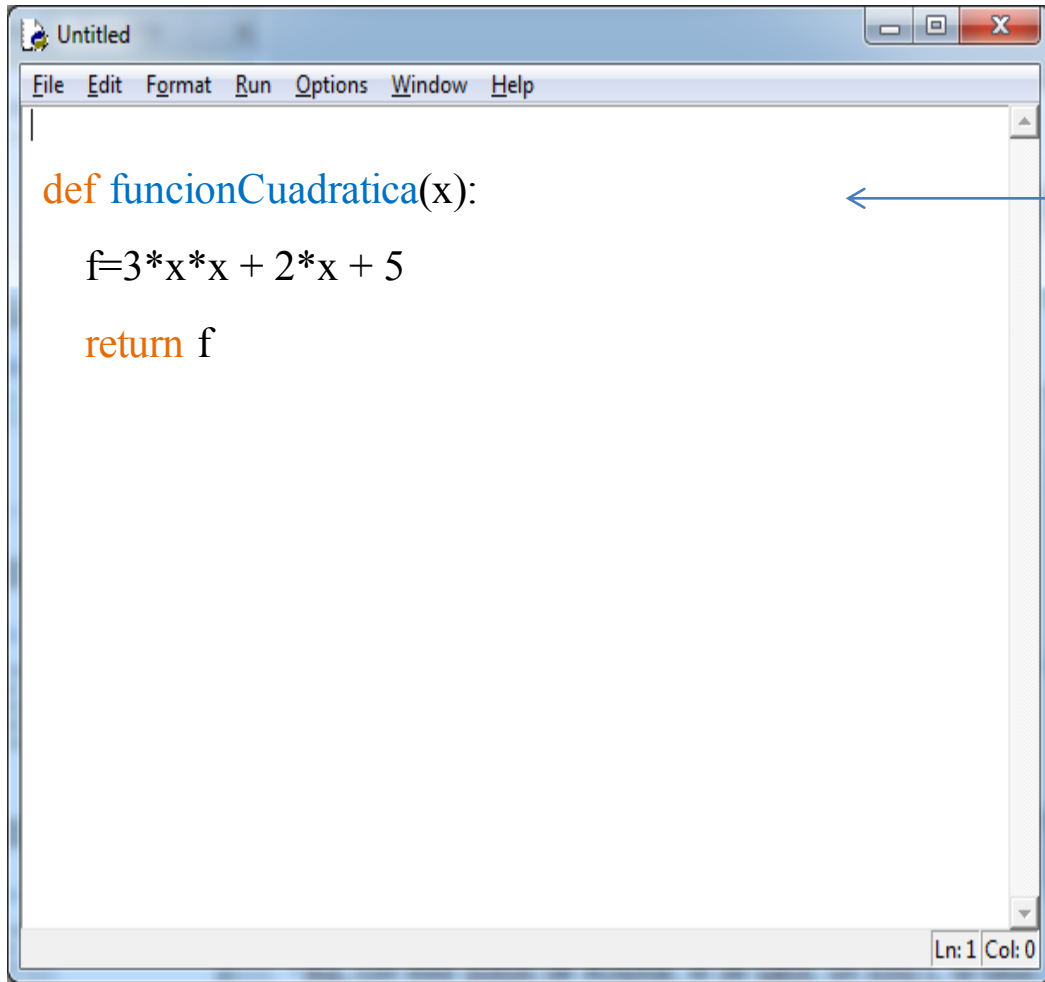
# Funciones en Python

---



Una función en Python se **especializa** en realizar un cálculo determinado



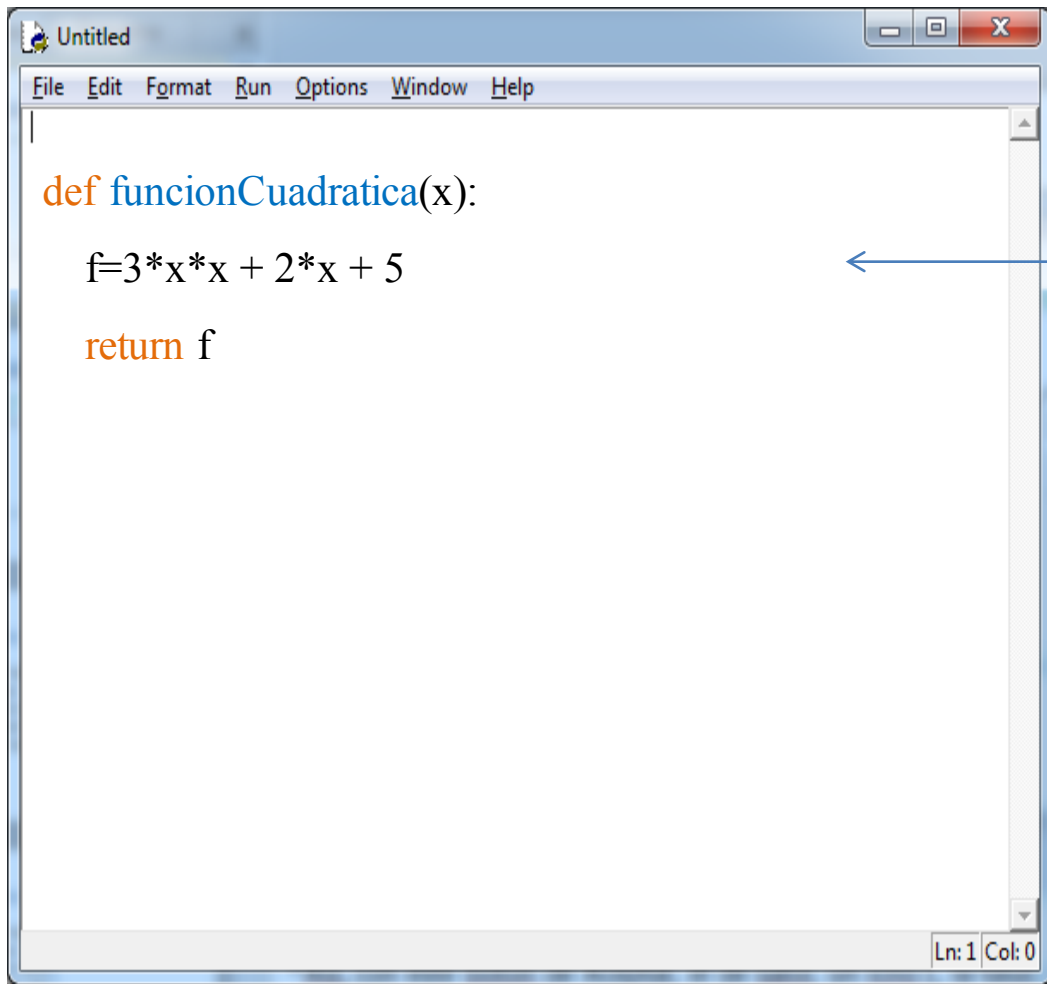


```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f
```

Ln: 1 Col: 0

Una función se define indicando un **nombre** y las **variables** que necesita para hacer el cálculo correspondiente

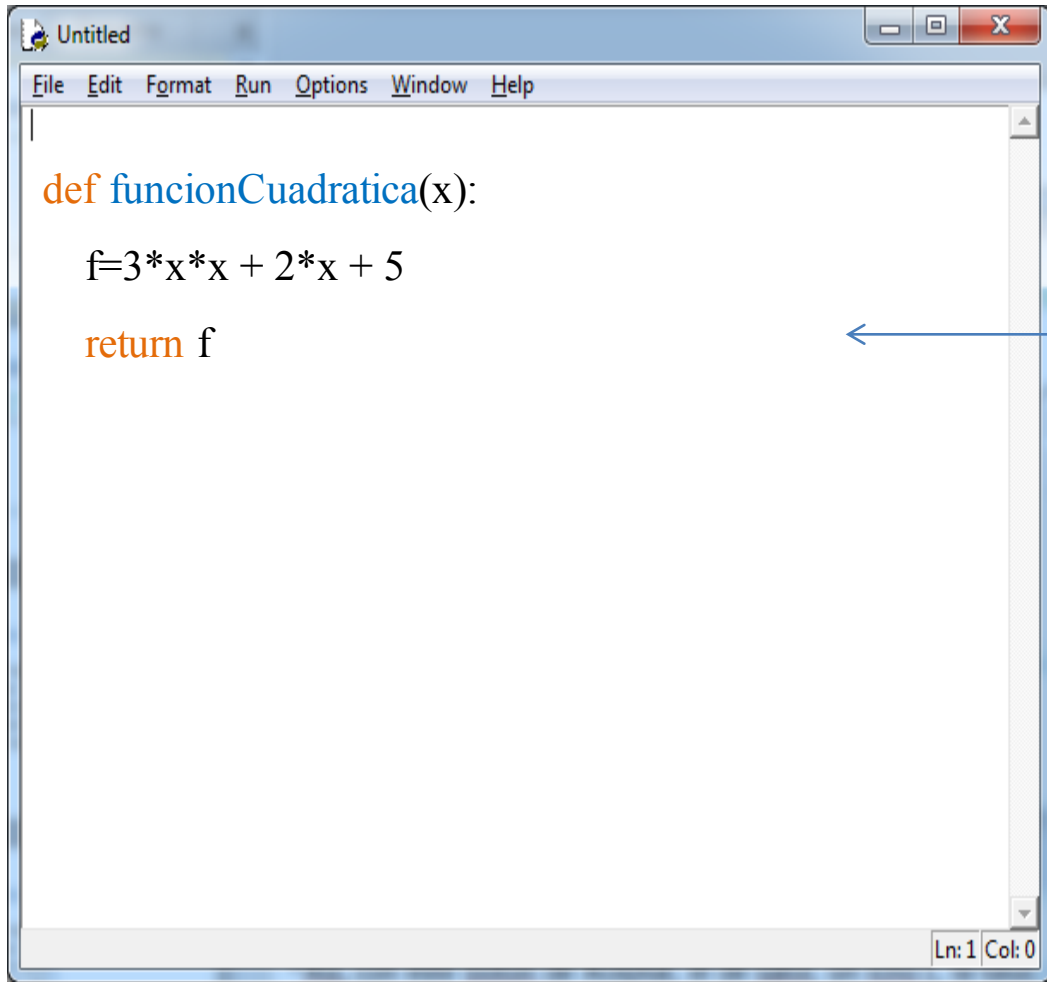




```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f
```

Ln: 1 Col: 0

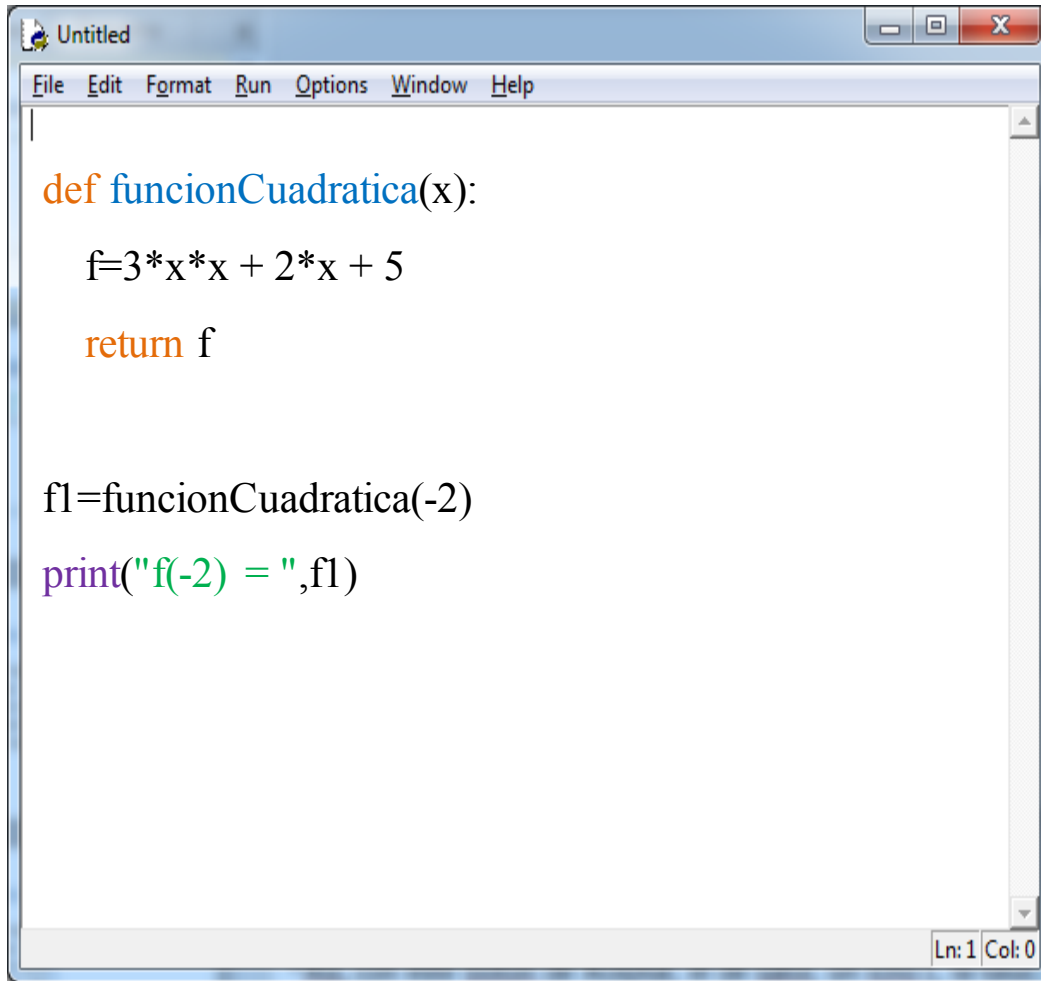
lo que calcula la función



```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f
```

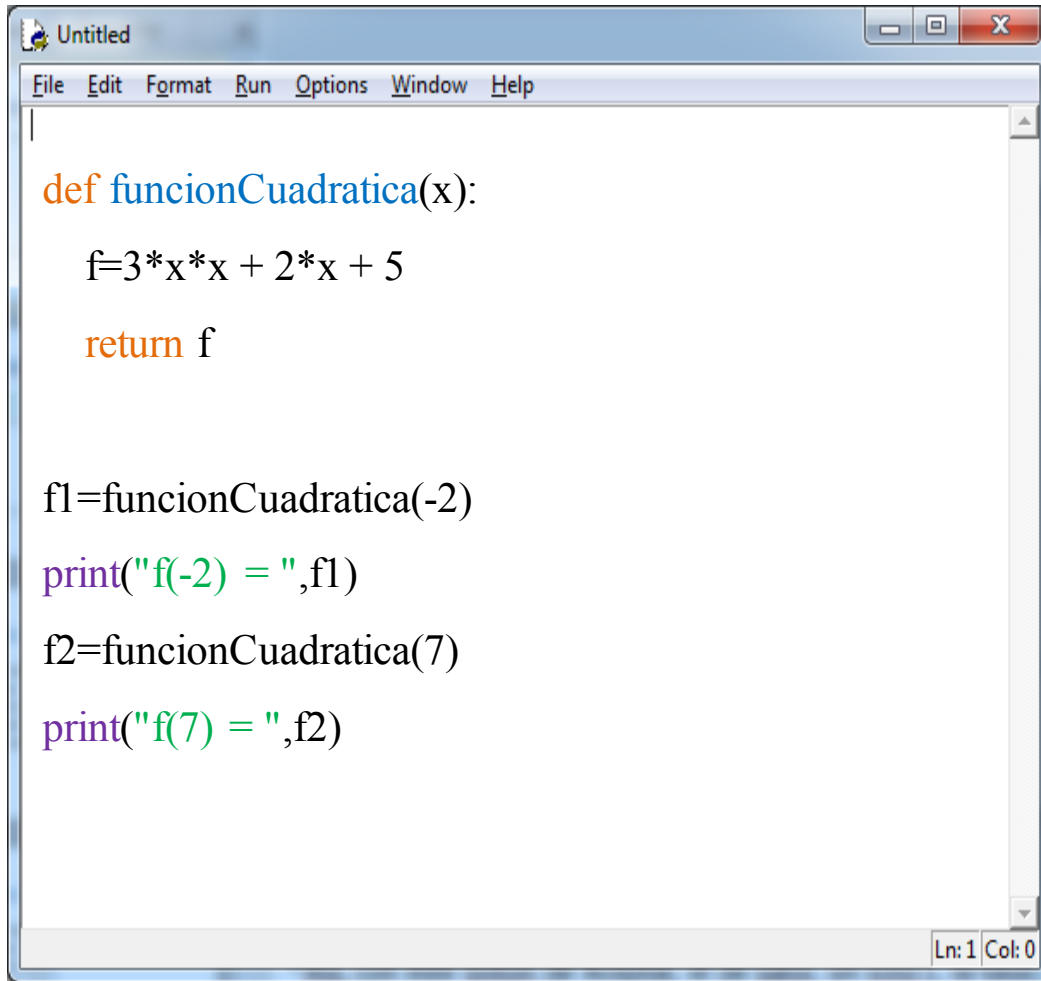
Ln: 1 Col: 0

valor que devuelve la función



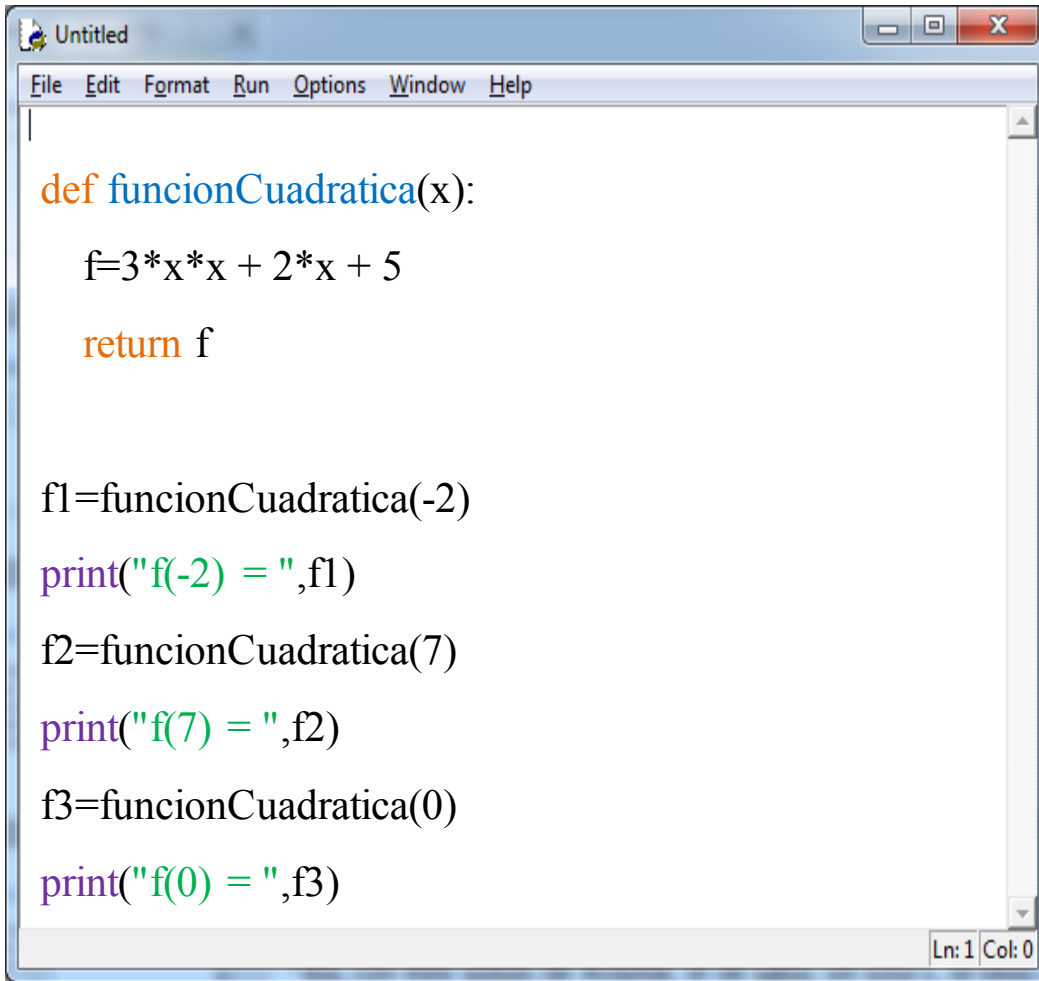
```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f  
  
f1=funcionCuadratica(-2)  
print("f(-2) = ",f1)
```

Ln: 1 Col: 0



```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f  
  
f1=funcionCuadratica(-2)  
print("f(-2) = ",f1)  
f2=funcionCuadratica(7)  
print("f(7) = ",f2)
```

The image shows a screenshot of a Python IDE window titled "Untitled". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains a function definition for "funcionCuadratica" and its usage. The function takes a parameter "x" and returns the value of the quadratic expression  $f = 3x^2 + 2x + 5$ . The function is called twice: first with  $x = -2$  and the result is stored in "f1", and then with  $x = 7$  and the result is stored in "f2". The code is color-coded: "def" is blue, "funcionCuadratica" is blue, "x" is blue, "f" is orange, "return" is orange, "f1" is black, "f2" is black, "print" is purple, and the string literals are green. The status bar at the bottom right shows "Ln: 1 Col: 0".



```
def funcionCuadratica(x):  
    f=3*x*x + 2*x + 5  
    return f  
  
f1=funcionCuadratica(-2)  
print("f(-2) = ",f1)  
f2=funcionCuadratica(7)  
print("f(7) = ",f2)  
f3=funcionCuadratica(0)  
print("f(0) = ",f3)
```

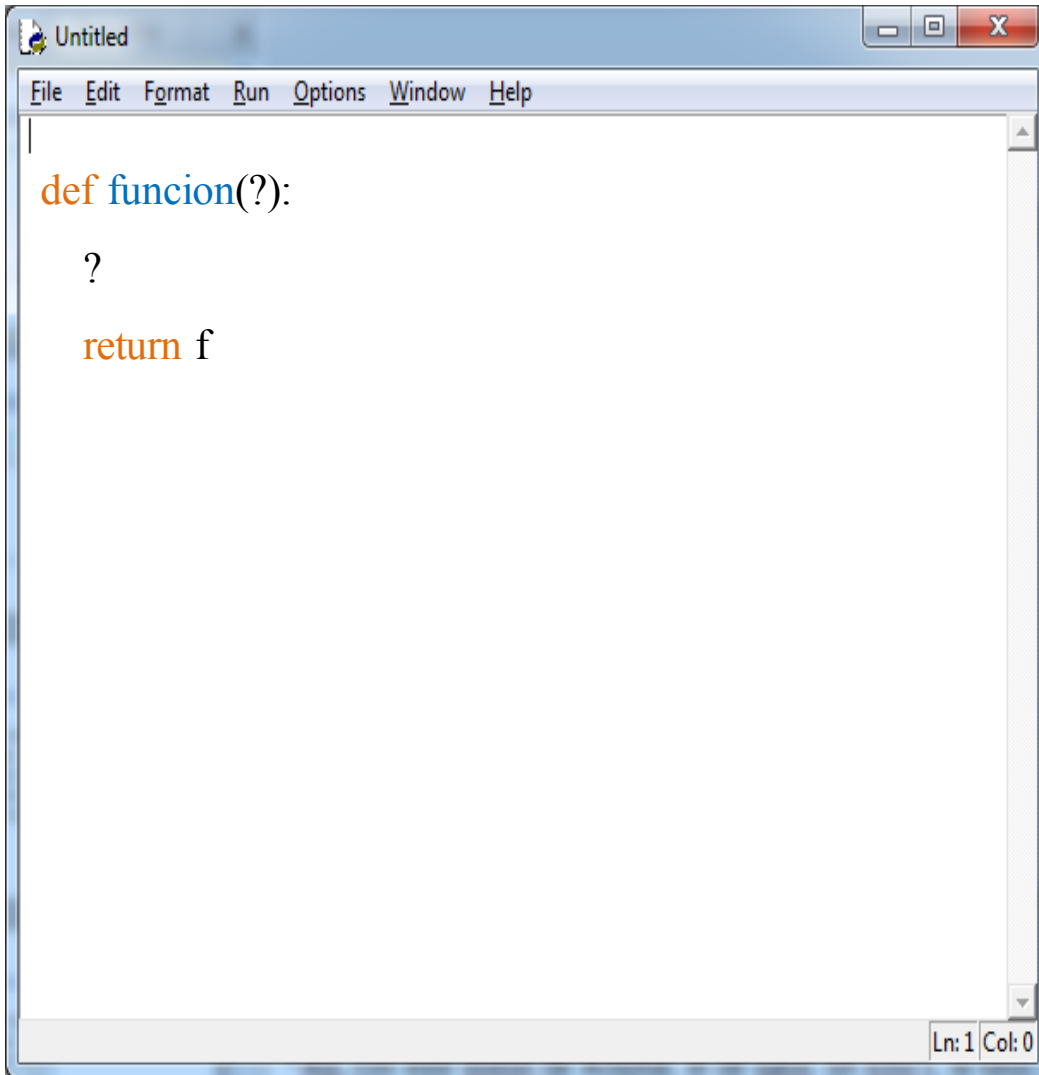
The image shows a screenshot of a Python IDE window titled "Untitled". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code is written in a light blue font. The function definition is: `def funcionCuadratica(x):`, followed by an indented line `f=3*x*x + 2*x + 5`, and then `return f`. Below the function definition, there are three calls to the function: `f1=funcionCuadratica(-2)`, `print("f(-2) = ",f1)`, `f2=funcionCuadratica(7)`, `print("f(7) = ",f2)`, `f3=funcionCuadratica(0)`, and `print("f(0) = ",f3)`. The status bar at the bottom right shows "Ln: 1 Col: 0".

# Funciones en Python

---

**Problema:** Desarrollar una función en Python que permita calcular el valor de  $f(a,b)$  definida de la siguiente manera:

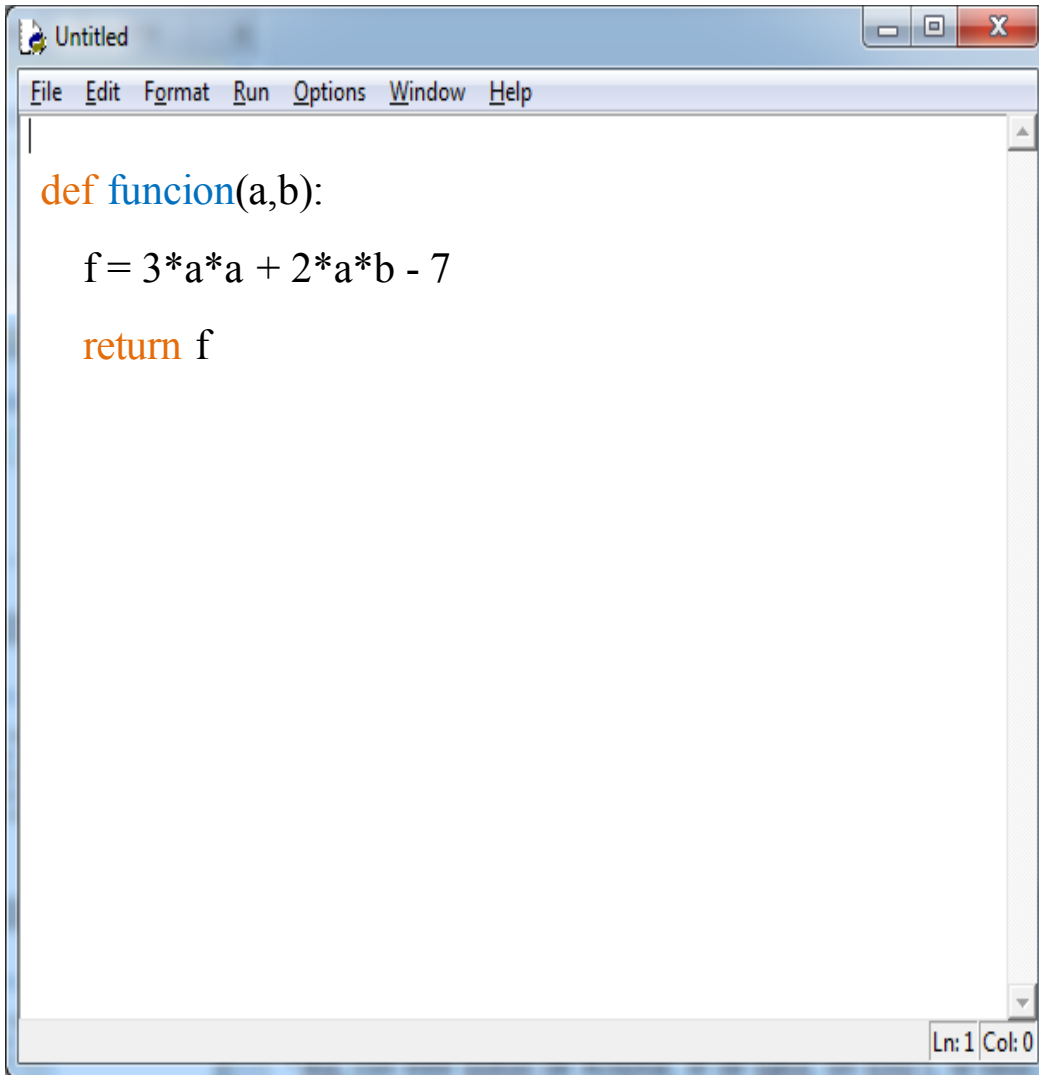
$$f(a,b) = 3a^2 + 2ab - 7$$



```
def funcion(?):  
    ?  
    return f
```

Ln: 1 Col: 0

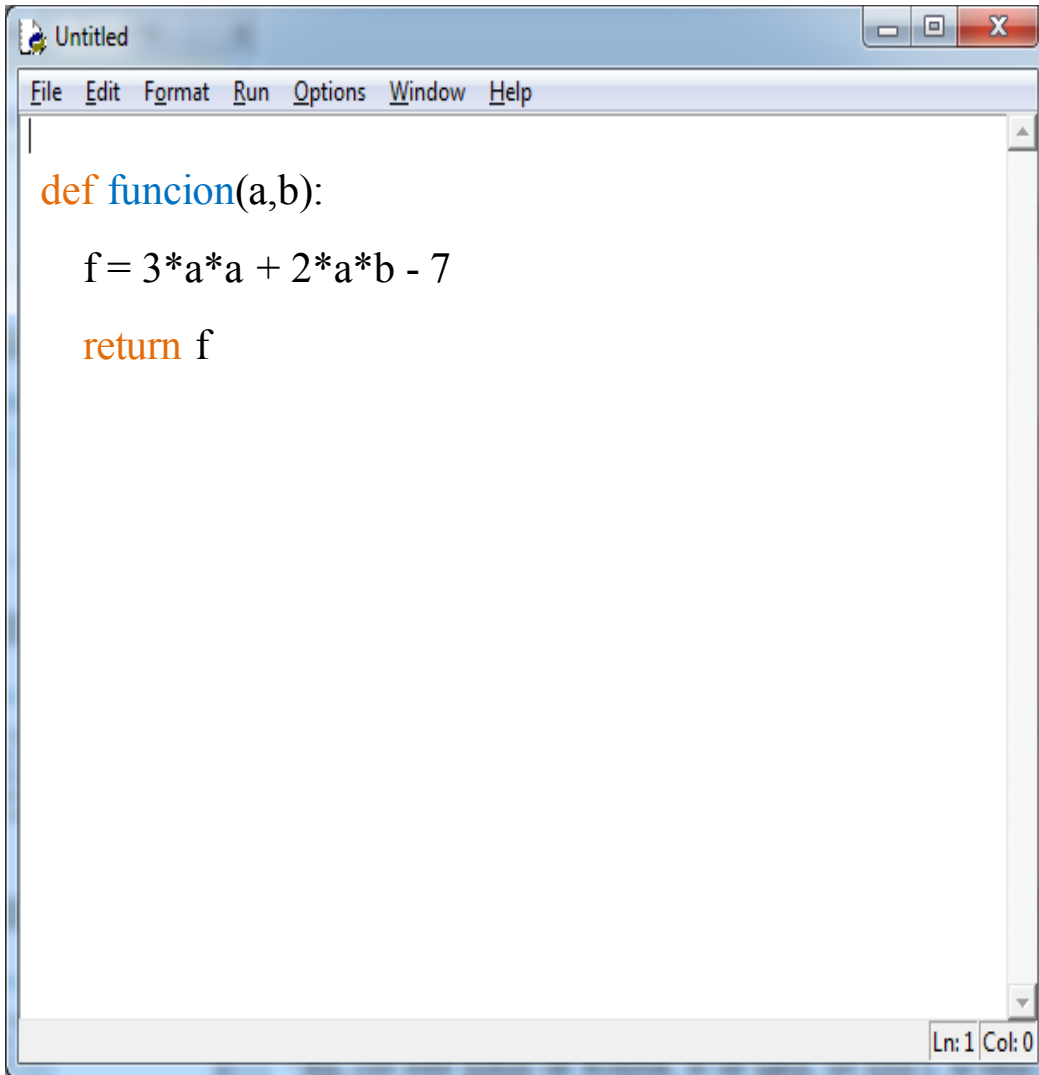
$$f(a,b) = 3a^2 + 2ab - 7$$



```
def funcion(a,b):  
    f = 3*a*a + 2*a*b - 7  
    return f
```

$$f(a,b) = 3a^2 + 2ab - 7$$



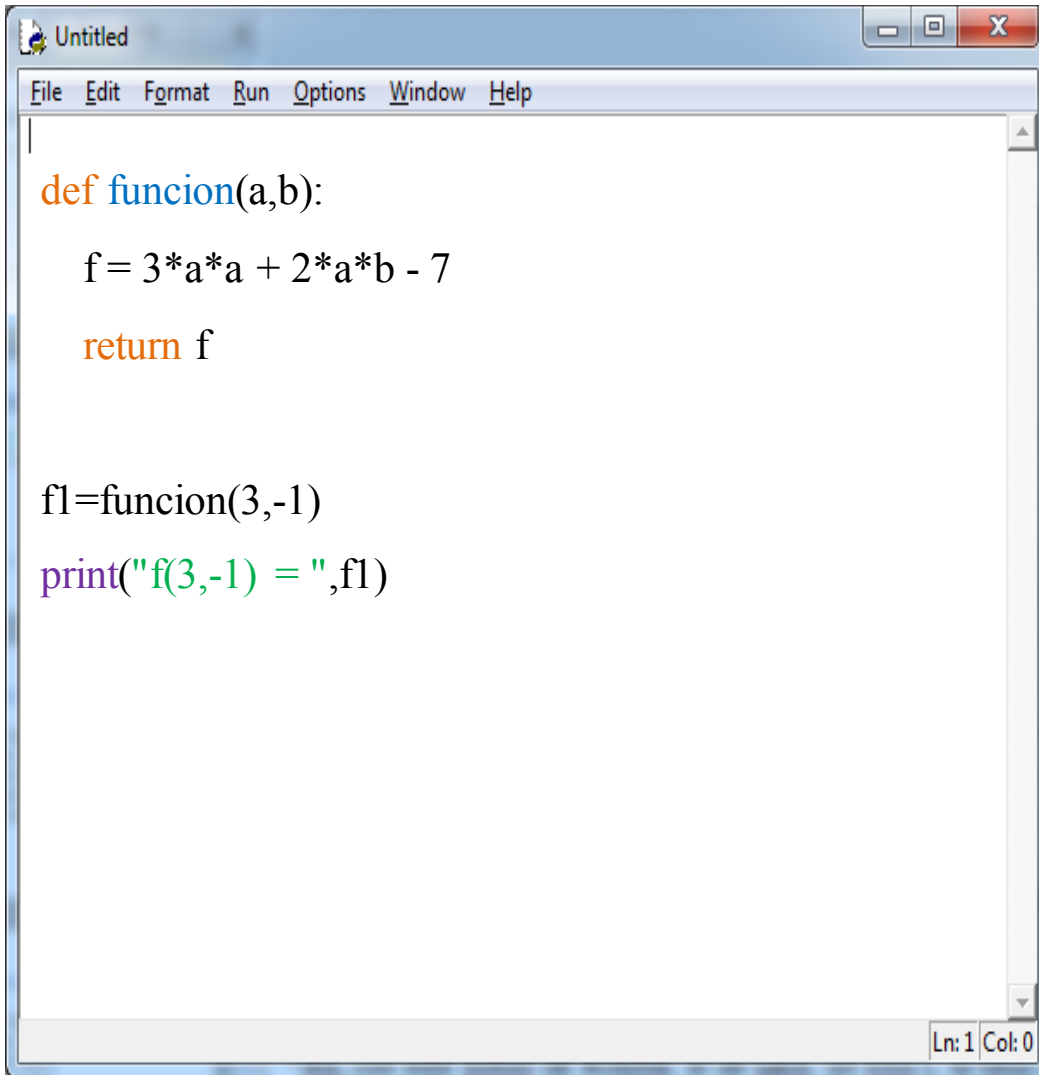


```
def funcion(a,b):  
    f = 3*a*a + 2*a*b - 7  
    return f
```

The screenshot shows a Python IDE window with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a status bar at the bottom right indicating 'Ln: 1 Col: 0'.

Utilice la función por partes  
para los valores en la  
siguiente tabla:

a	b
3	-1
-7	0
6	2

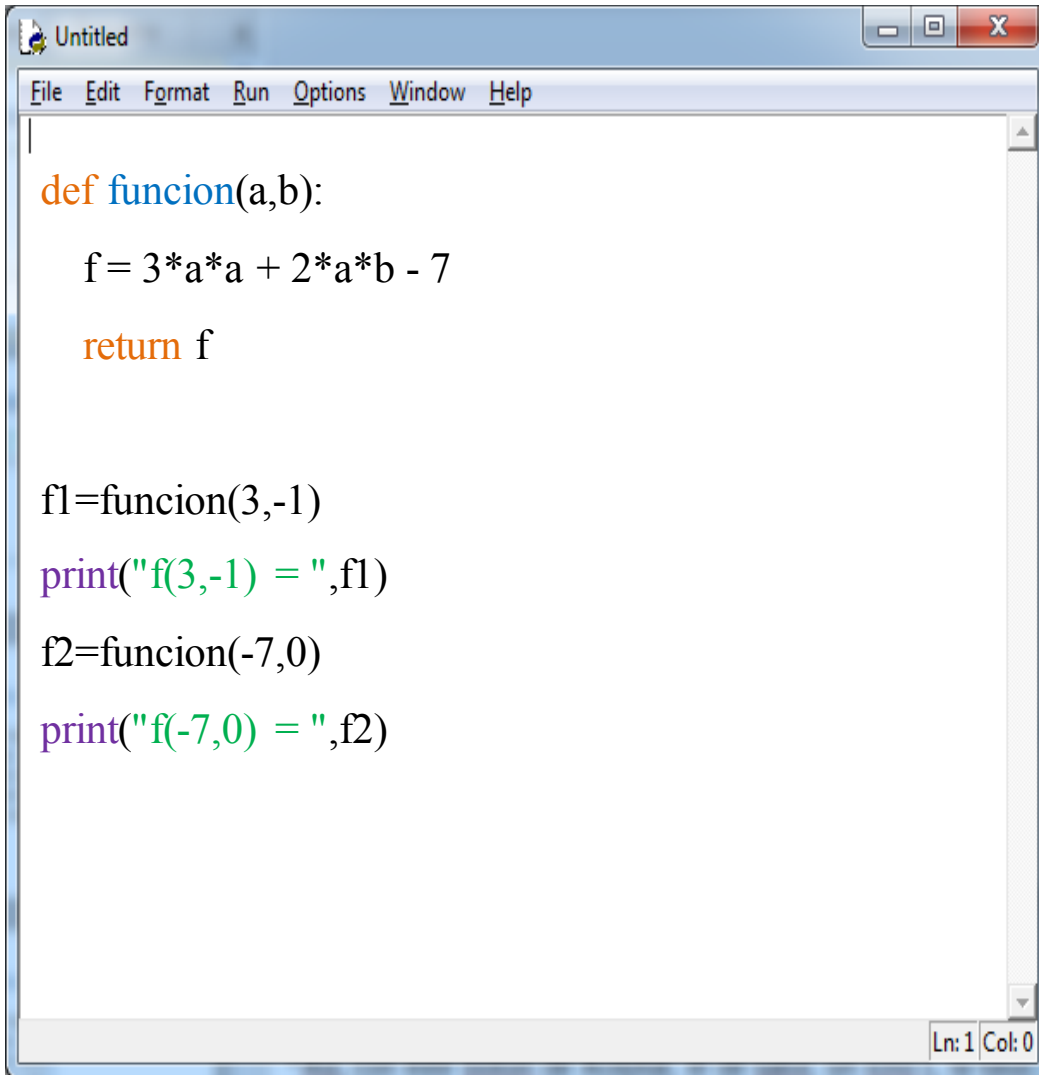


```
def funcion(a,b):  
    f = 3*a*a + 2*a*b - 7  
    return f  
  
f1=funcion(3,-1)  
print("f(3,-1) = ",f1)
```

The screenshot shows a Python IDE window with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a status bar at the bottom indicating 'Ln: 1 Col: 0'. The code defines a function 'funcion' that takes two arguments 'a' and 'b', calculates the value of the expression  $f = 3a^2 + 2ab - 7$ , and returns it. Below the function definition, the function is called with arguments 3 and -1, and the result is printed.

Utilice la función por partes  
para los valores en la  
siguiente tabla:

a	b
3	-1
-7	0
6	2

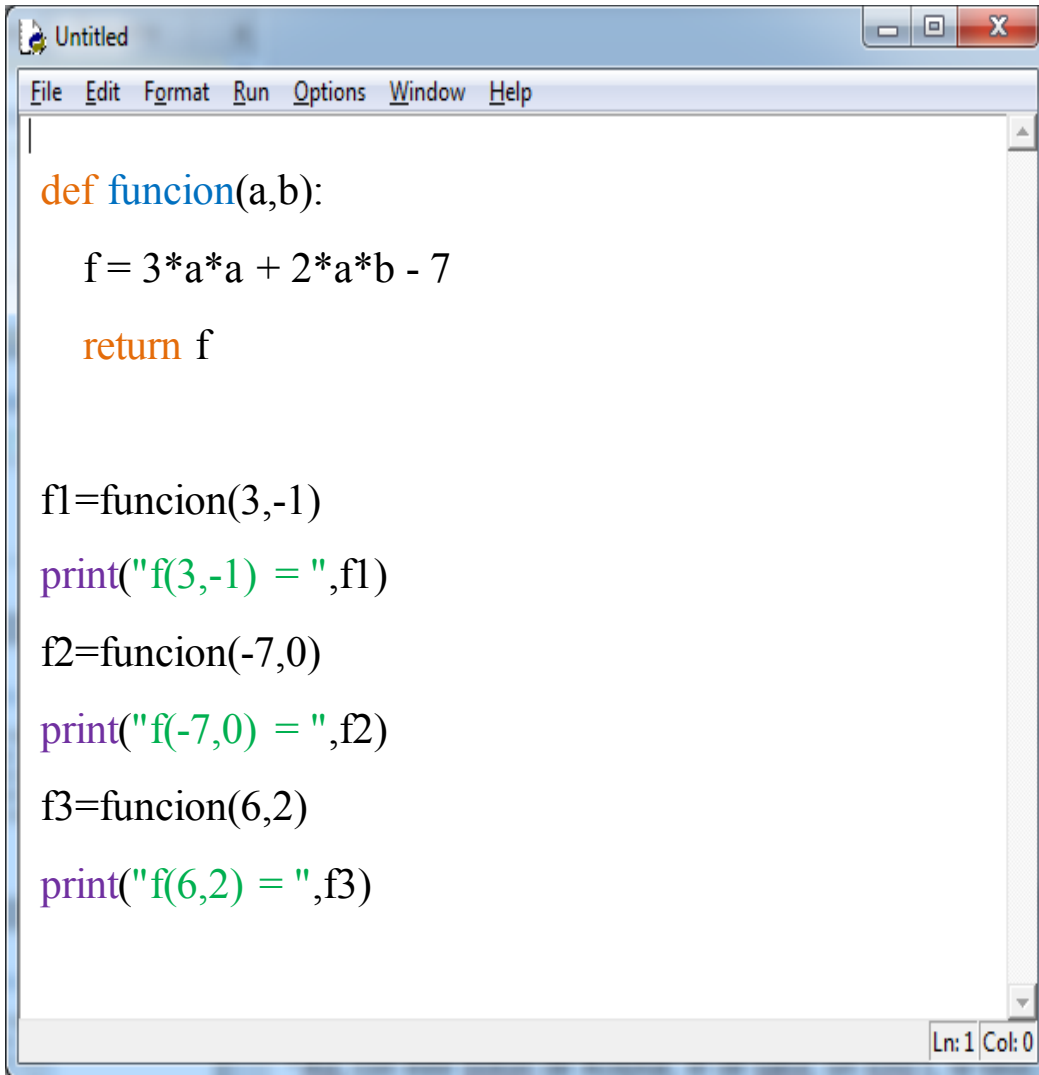


```
def funcion(a,b):  
    f = 3*a*a + 2*a*b - 7  
    return f  
  
f1=funcion(3,-1)  
print("f(3,-1) =",f1)  
f2=funcion(-7,0)  
print("f(-7,0) =",f2)
```

Ln: 1 Col: 0

Utilice la función por partes  
para los valores en la  
siguiente tabla:

a	b
3	-1
-7	0
6	2



```
def funcion(a,b):  
    f = 3*a*a + 2*a*b - 7  
    return f  
  
f1=funcion(3,-1)  
print("f(3,-1) = ",f1)  
f2=funcion(-7,0)  
print("f(-7,0) = ",f2)  
f3=funcion(6,2)  
print("f(6,2) = ",f3)
```

Ln: 1 Col: 0

Utilice la función por partes  
para los valores en la  
siguiente tabla:

a	b
3	-1
-7	0
6	2

# Funciones en Python

---

**Problema:** Desarrollar un programa que permita calcular el área de un triángulo dados su base y su altura

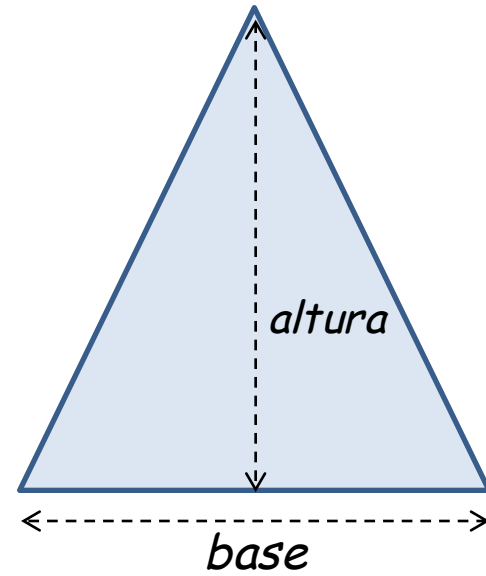
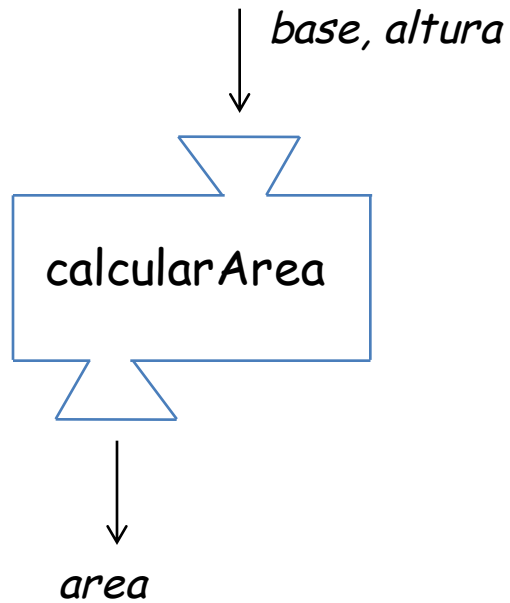
```
Digite la base: 3  
Digite la altura: 5
```

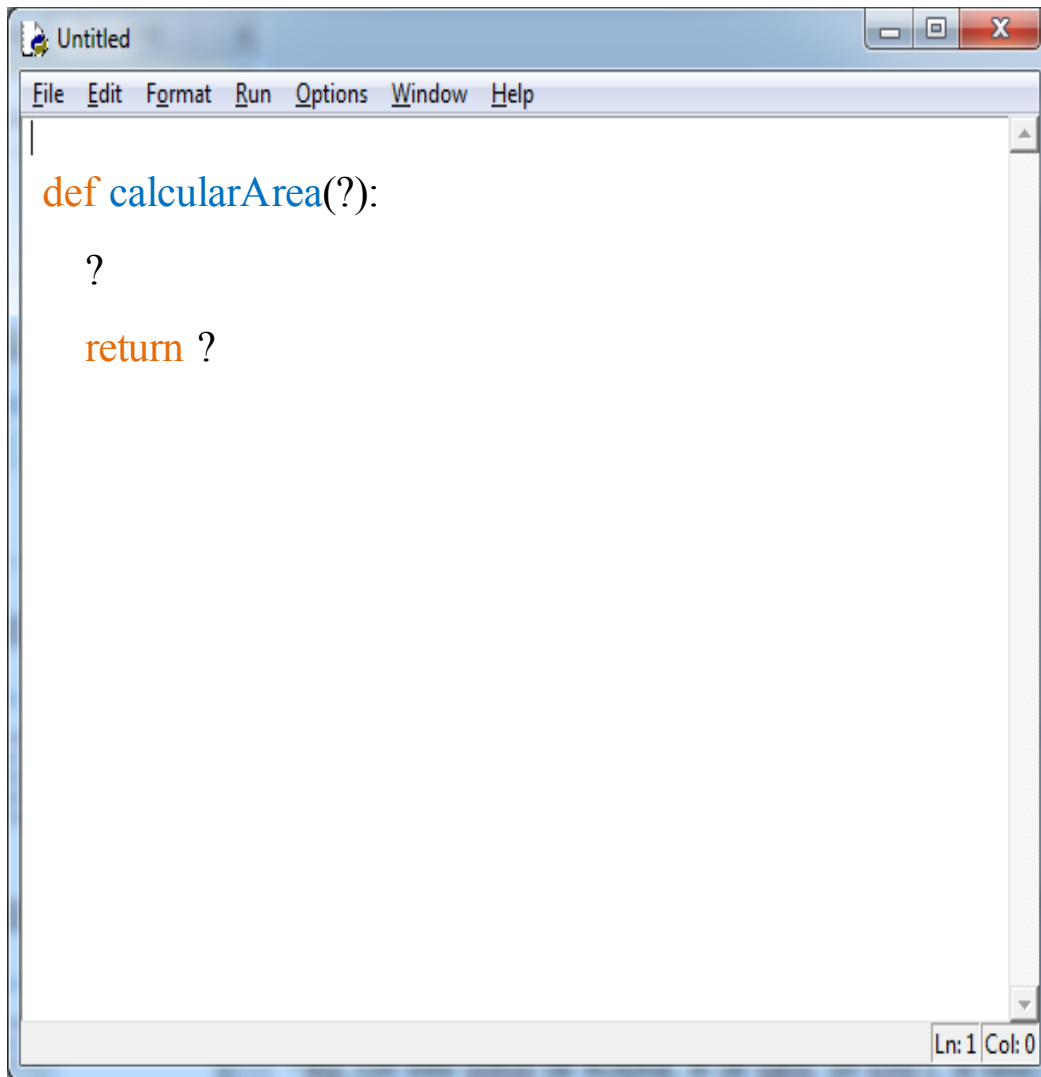


```
Area del triangulo: 7.5
```

# Funciones en Python

---

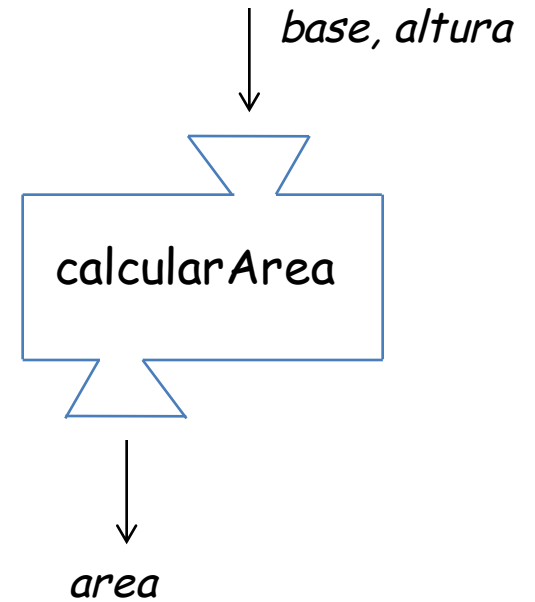


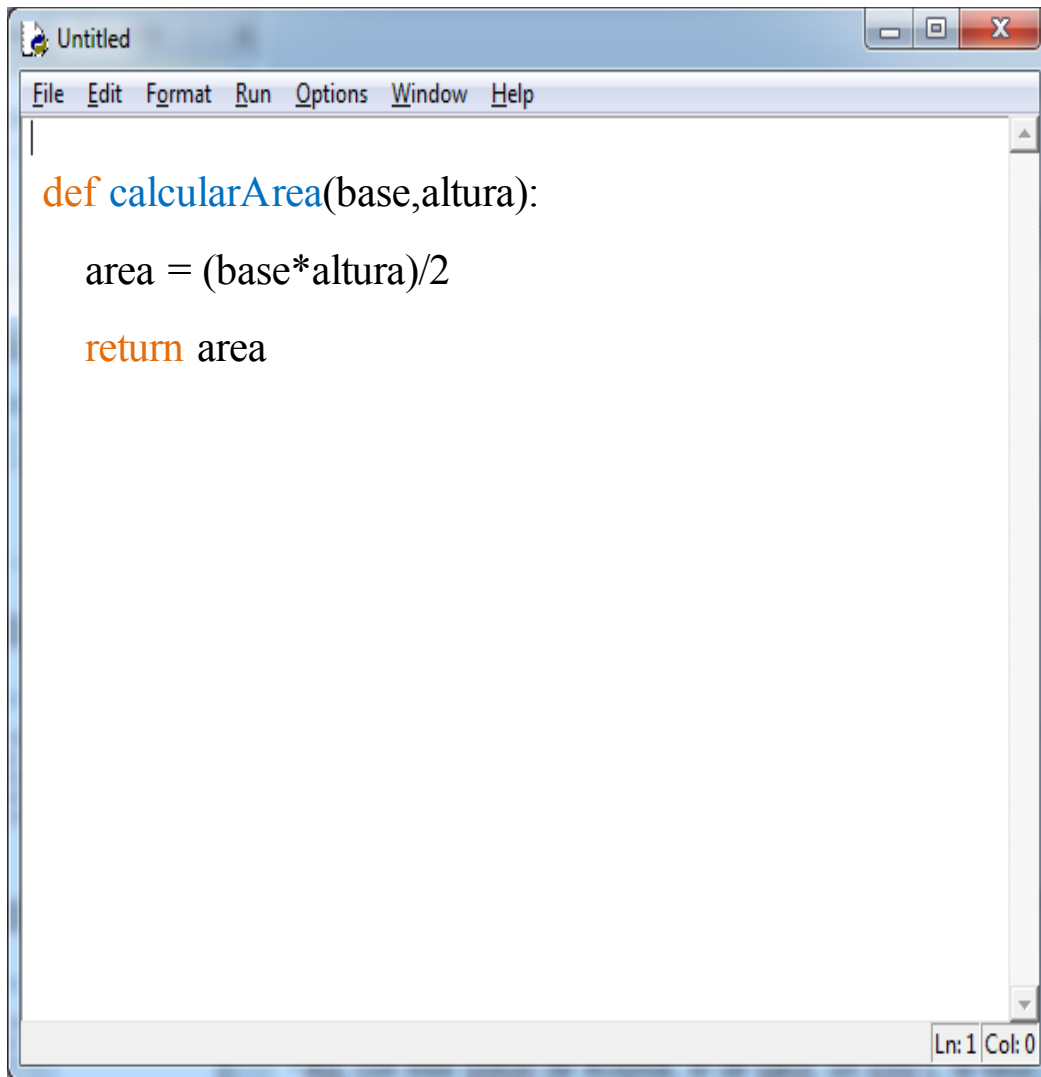


A screenshot of a code editor window titled "Untitled". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
def calcularArea(?):  
    ?  
    return ?
```

The status bar at the bottom right indicates "Ln: 1 Col: 0".

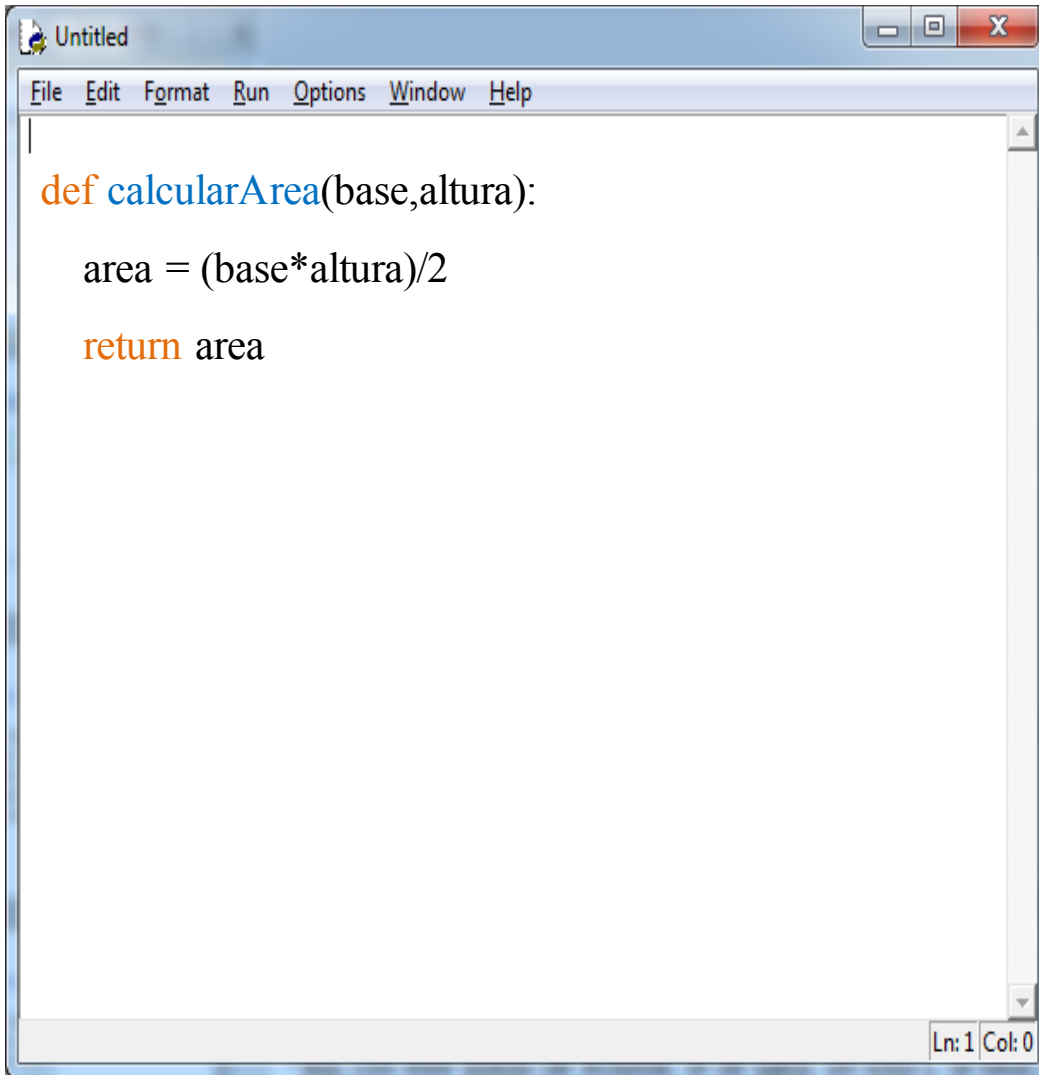




```
def calcularArea(base, altura):  
    area = (base*altura)/2  
    return area
```

Ln: 1 Col: 0



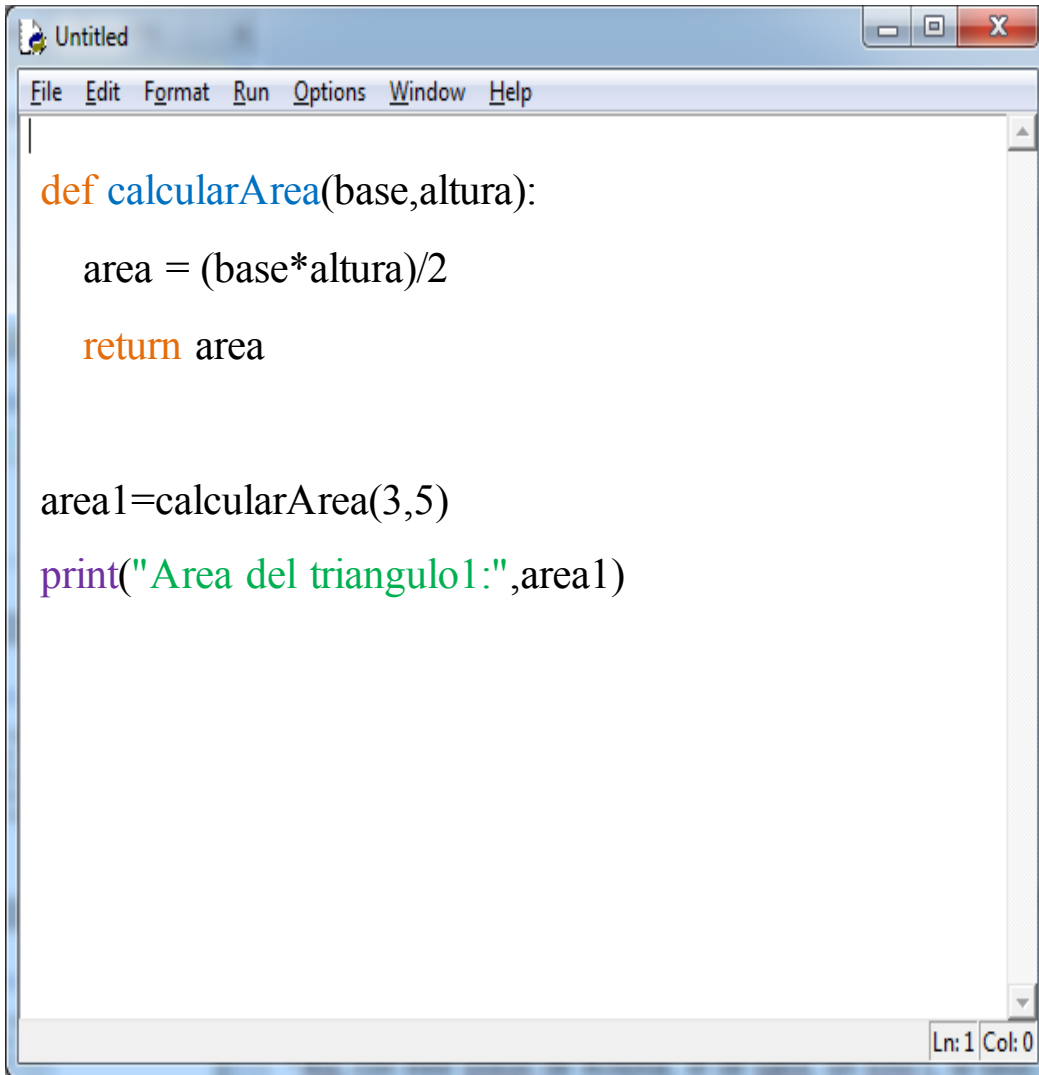


```
def calcularArea(base, altura):  
    area = (base*altura)/2  
    return area
```

The screenshot shows a Python IDE window with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a status bar at the bottom indicating 'Ln: 1 Col: 0'.

Utilice la función **calcularArea** para los valores en la siguiente tabla:

base	altura
3	5
10	12

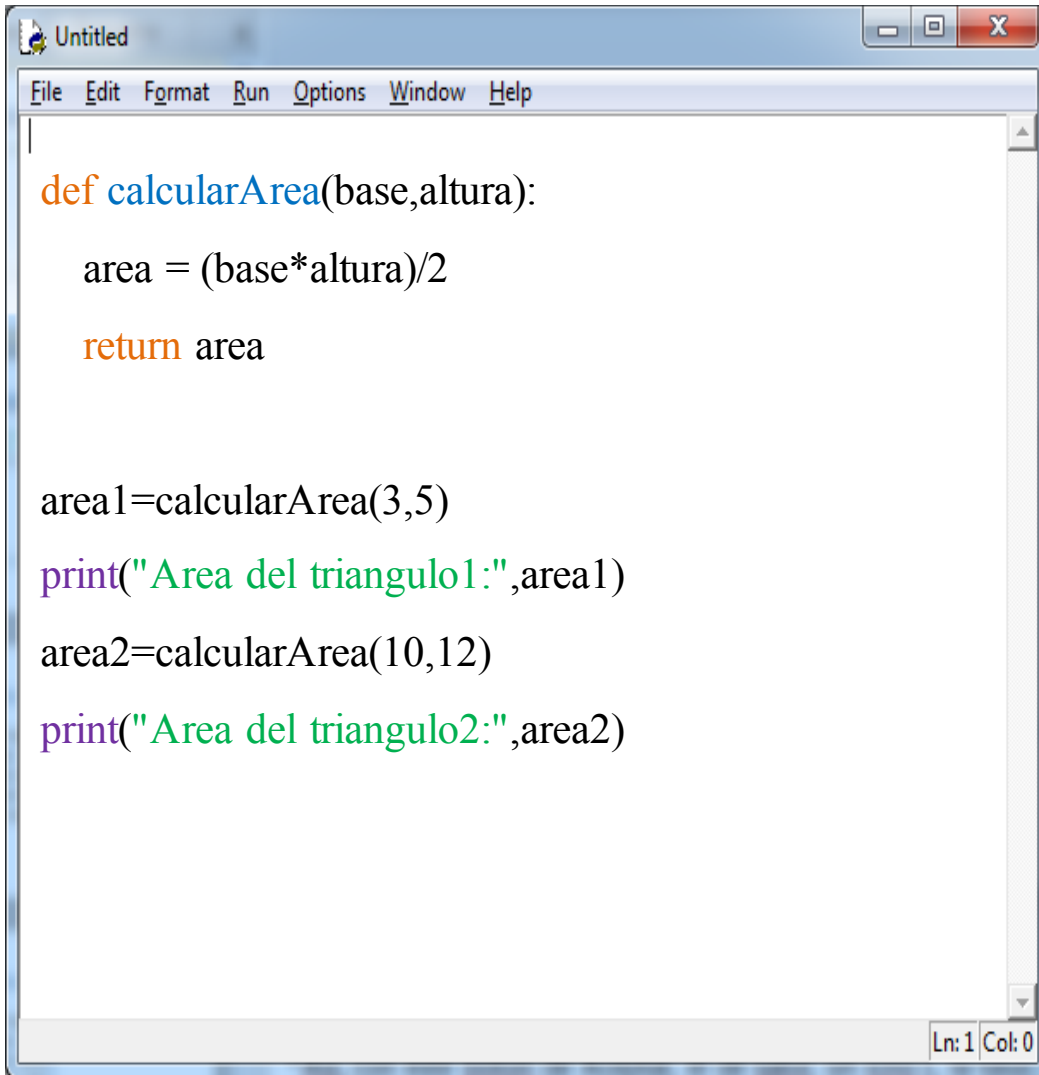


```
def calcularArea(base,altura):  
    area = (base*altura)/2  
    return area  
  
area1=calcularArea(3,5)  
print("Area del triangulo1:",area1)
```

The screenshot shows a Python IDE window with a menu bar (File, Edit, Format, Run, Options, Window, Help) and a status bar at the bottom indicating 'Ln: 1 Col: 0'. The code defines a function 'calcularArea' that takes 'base' and 'altura' as arguments, calculates the area of a triangle, and returns it. It then calls this function with base 3 and height 5, and prints the result.

Utilice la función **calcularArea** para los valores en la siguiente tabla:

base	altura
3	5
10	12



```
def calcularArea(base,altura):  
    area = (base*altura)/2  
    return area  
  
area1=calcularArea(3,5)  
print("Area del triangulo1:",area1)  
area2=calcularArea(10,12)  
print("Area del triangulo2:",area2)
```

Ln: 1 Col: 0

Utilice la función **calcularArea** para los valores en la siguiente tabla:

base	altura
3	5
10	12

# Funciones en Python

---

**Problema:** Desarrollar un programa que permita calcular el área y el perímetro de un rectángulo de dimensiones alto y ancho

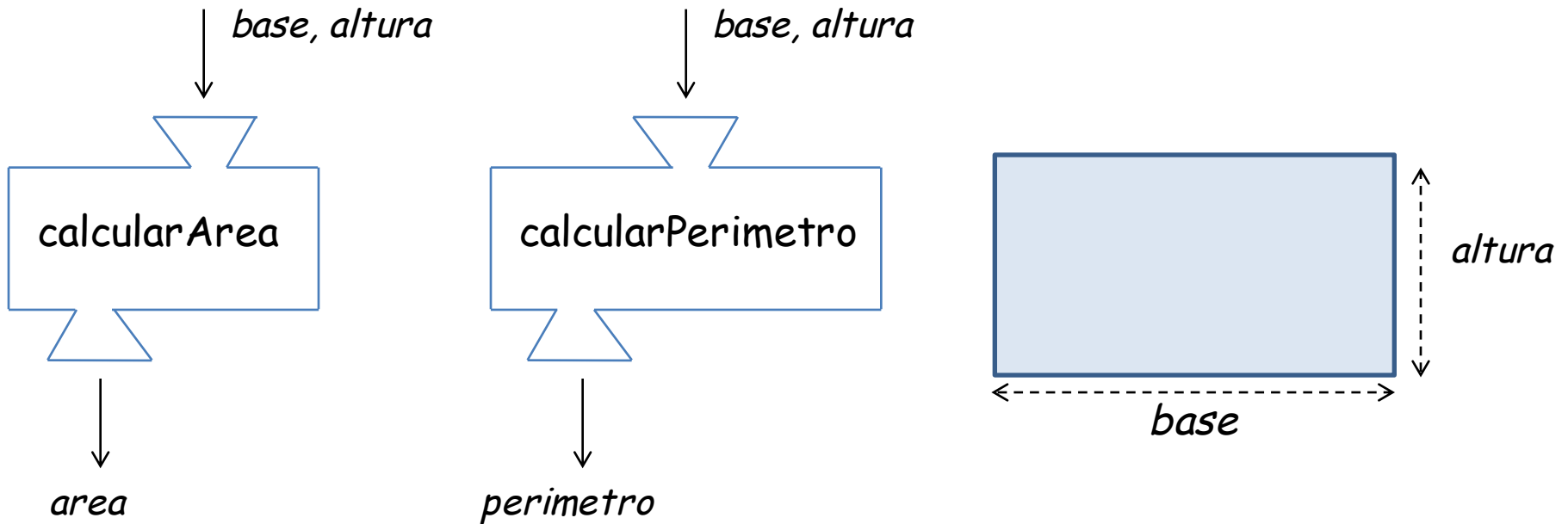
```
Digite el alto: 6.5  
Digite el ancho: 10.98
```

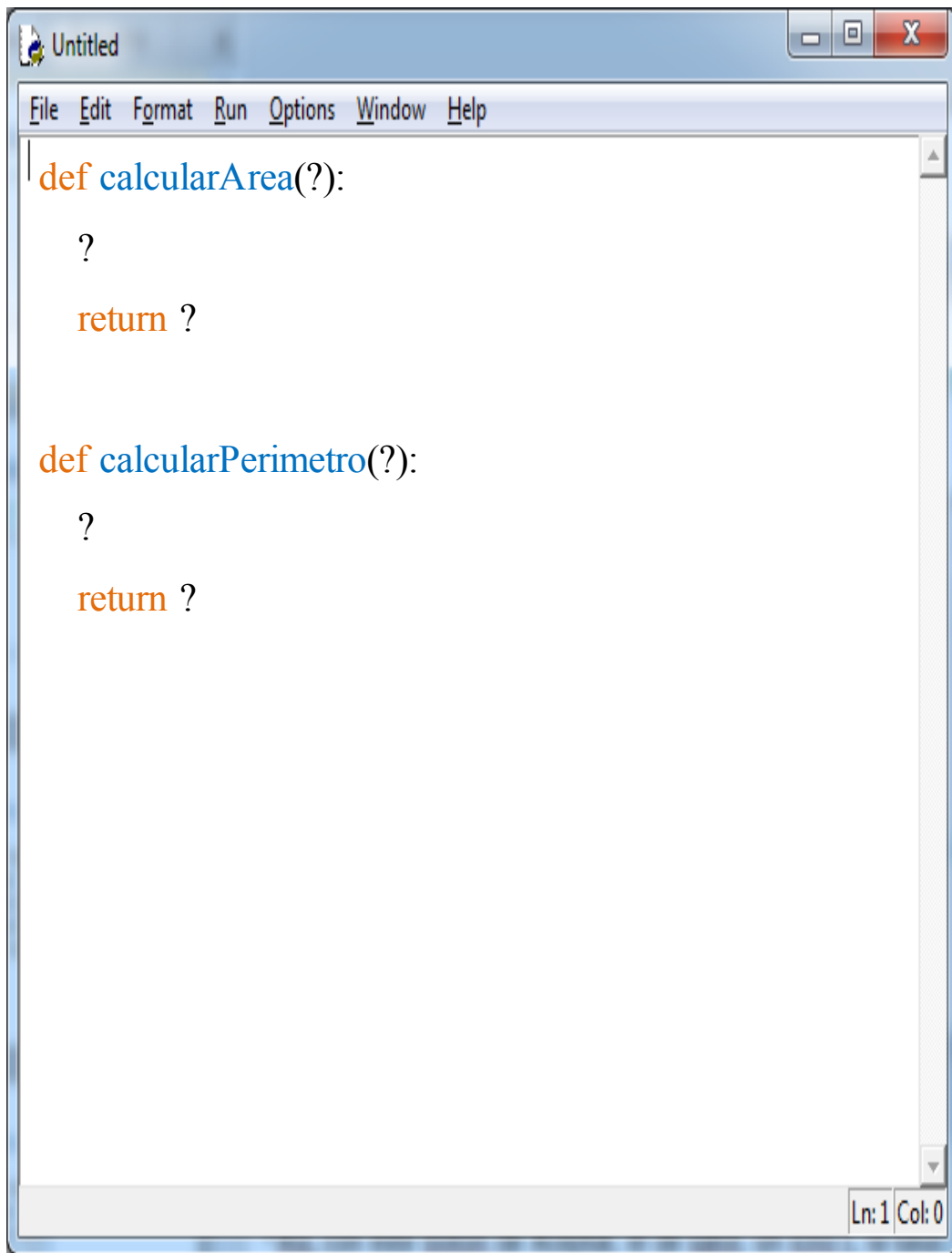


```
Area del rectangulo: 71.37  
Perimetro del rectangulo: 34.96
```

# Funciones en Python

---



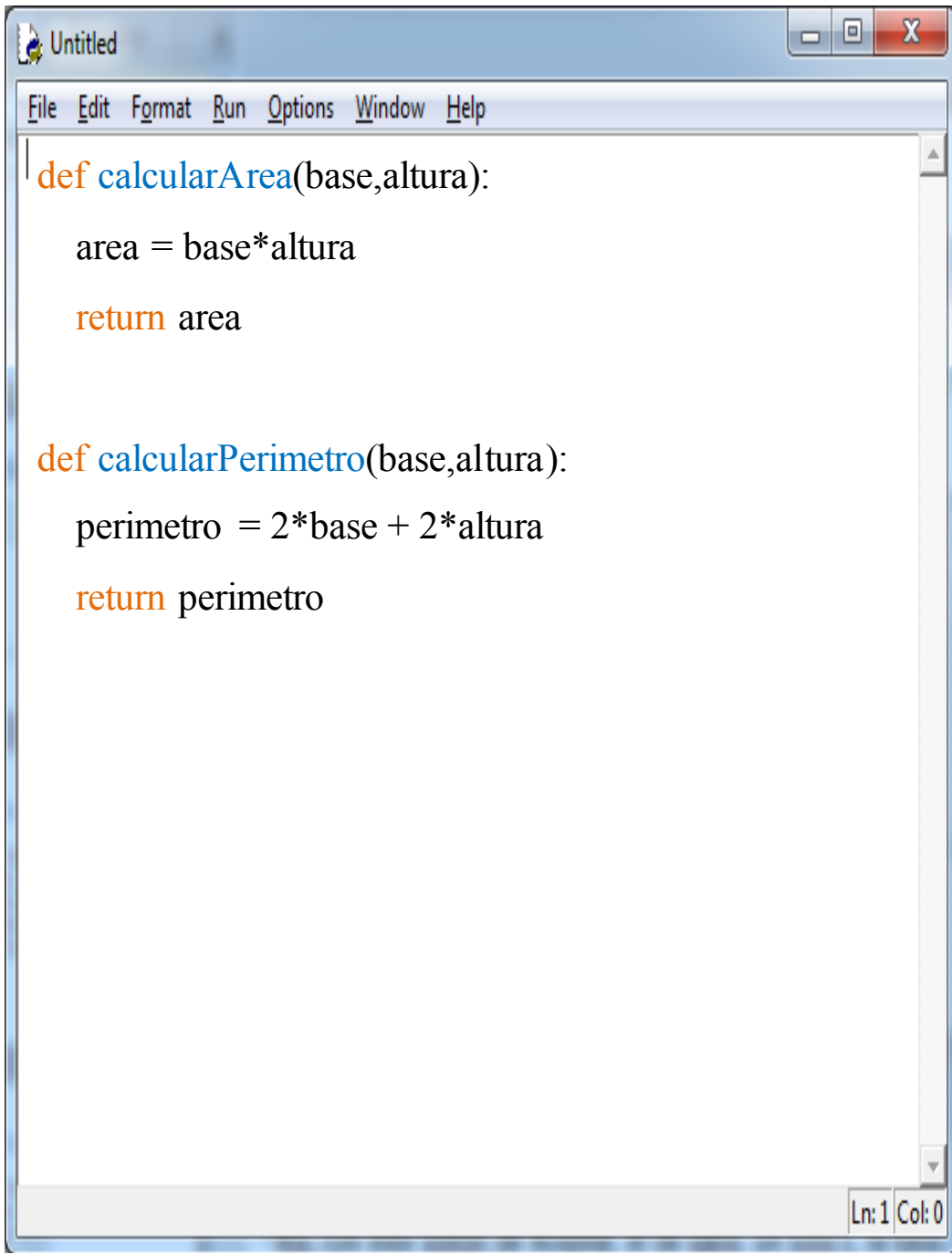


Untitled

File Edit Format Run Options Window Help

```
def calcularArea(?):  
    ?  
    return ?  
  
def calcularPerimetro(?):  
    ?  
    return ?
```

Ln: 1 Col: 0



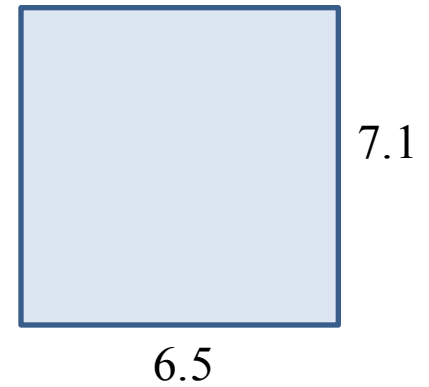
The image shows a screenshot of a Python IDE window titled "Untitled". The window has a menu bar with the following options: File, Edit, Format, Run, Options, Window, and Help. The main text area contains two Python functions. The first function, `def calcularArea(base, altura):`, calculates the area of a rectangle by multiplying the base and height, and returns the result. The second function, `def calcularPerimetro(base, altura):`, calculates the perimeter of a rectangle by multiplying the base and height by 2 and adding the results, and returns the result. The status bar at the bottom right indicates the current line and column as "Ln: 1 Col: 0".

```
def calcularArea(base, altura):  
    area = base*altura  
    return area  
  
def calcularPerimetro(base, altura):  
    perimetro = 2*base + 2*altura  
    return perimetro
```

Ln: 1 Col: 0

```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro
Ln: 1 Col: 0
```



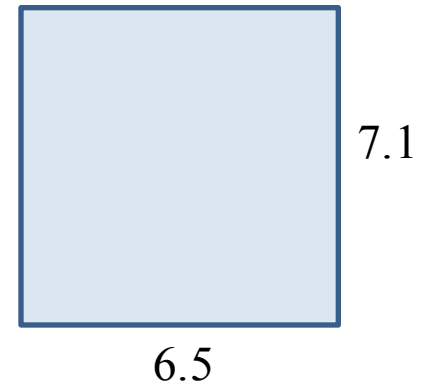


```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro

area1=calcularArea(6.5,7.1)

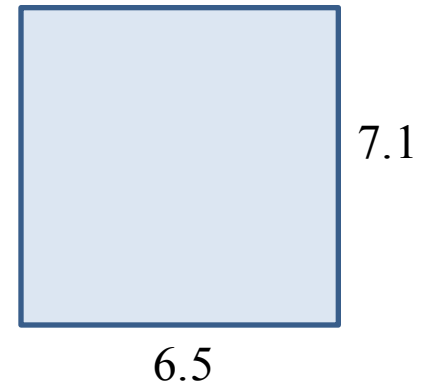
Ln: 1 Col: 0
```



```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro

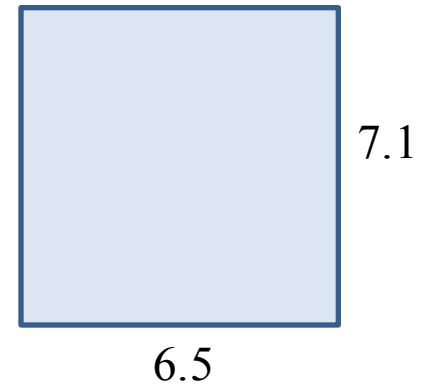
area1=calcularArea(6.5,7.1)
perimetro1=calcularPerimetro(6.5,7.1)
Ln: 1 Col: 0
```



```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro

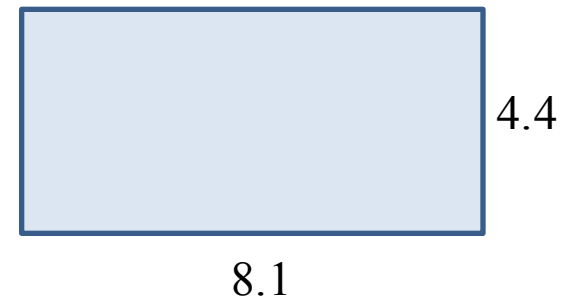
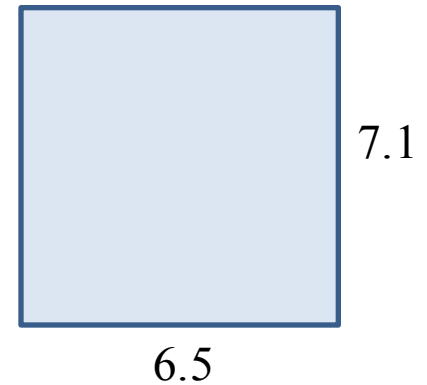
area1=calcularArea(6.5,7.1)
perimetro1=calcularPerimetro(6.5,7.1)
print("Area:",area1,"Perimetro:",perimetro1)
Ln: 1 Col: 0
```



```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro

area1=calcularArea(6.5,7.1)
perimetro1=calcularPerimetro(6.5,7.1)
print("Area:",area1,"Perimetro:",perimetro1)
Ln: 1 Col: 0
```

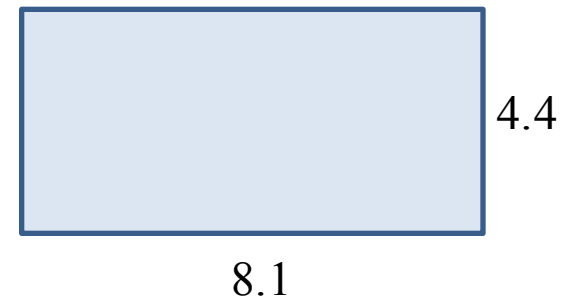
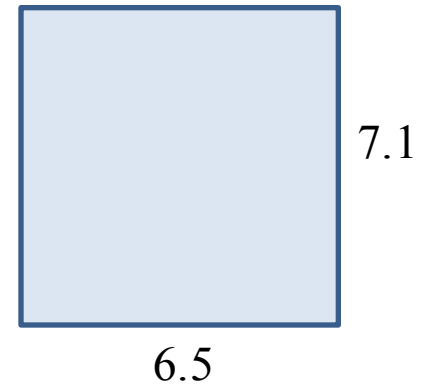


```
Untitled
File Edit Format Run Options Window Help
def calcularArea(base,altura):
    area = base*altura
    return area

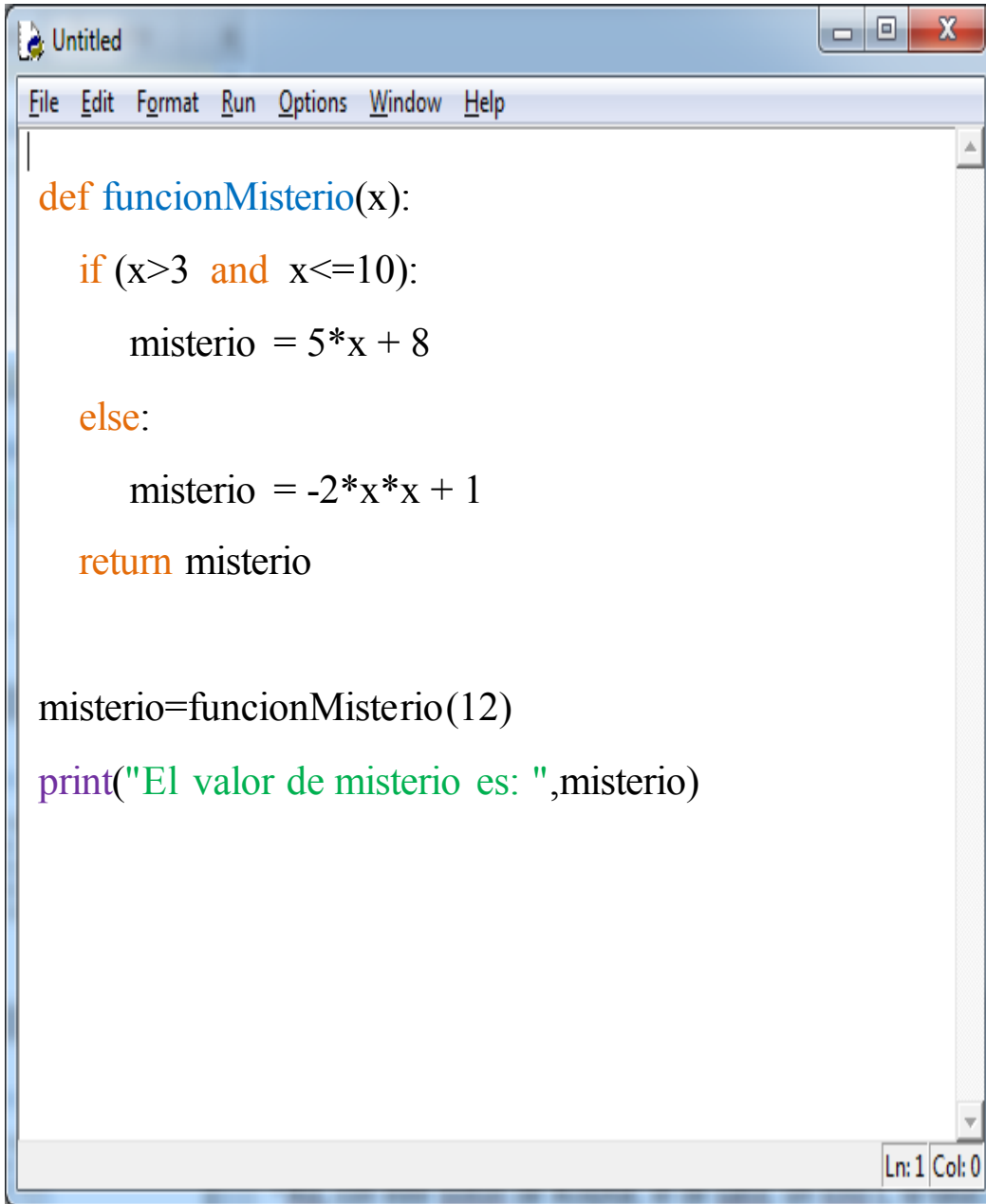
def calcularPerimetro(base,altura):
    perimetro = 2*base + 2*altura
    return perimetro

area1=calcularArea(6.5,7.1)
perimetro1=calcularPerimetro(6.5,7.1)
print("Area:",area1,"Perimetro:",perimetro1)

area2=calcularArea(8.1,4.4)
perimetro2=calcularPerimetro(8.1,4.4)
print("Area:",area2,"Perimetro:",perimetro2)
Ln: 1 Col: 0
```

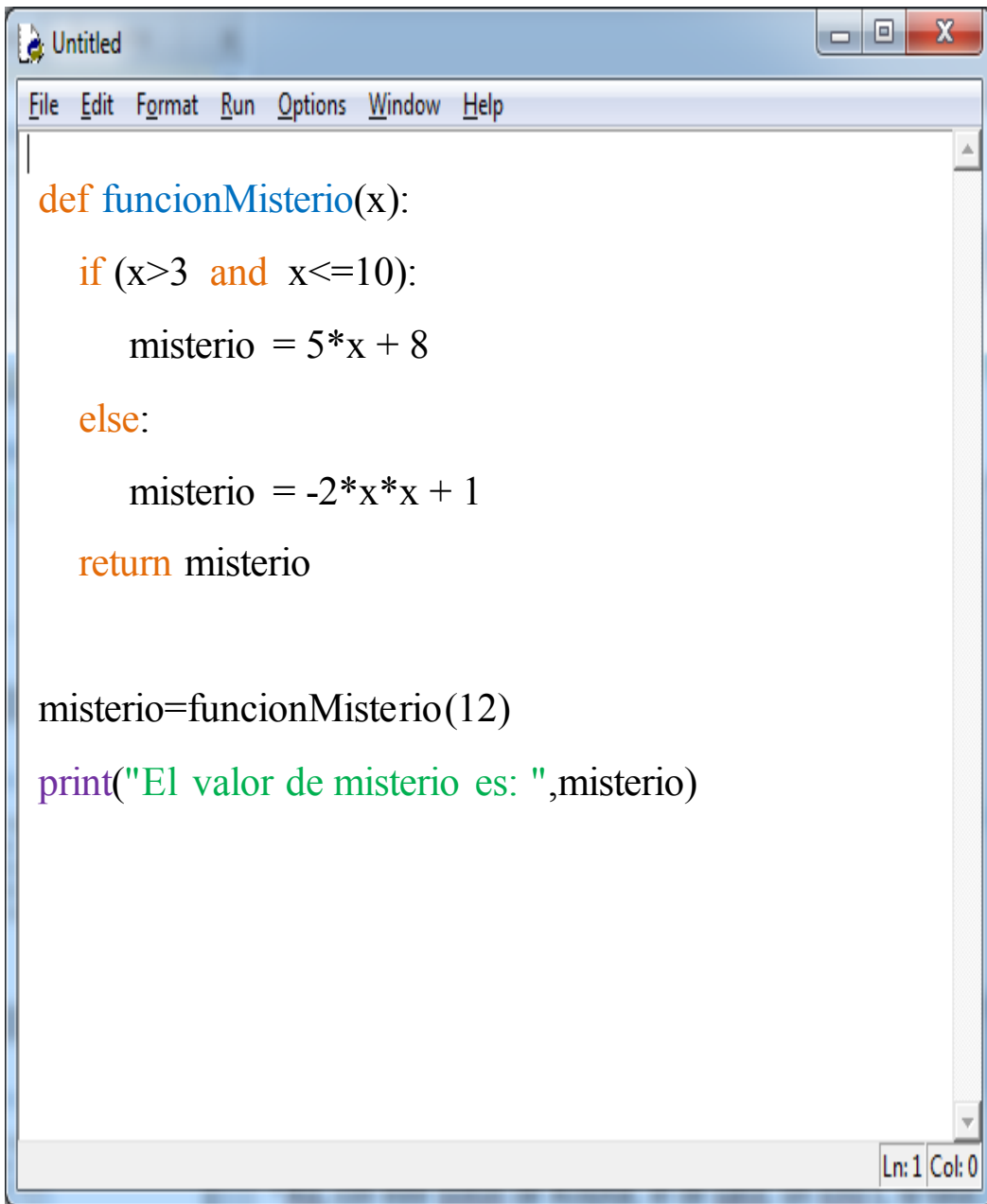






```
def funcionMisterio(x):  
    if (x>3 and x<=10):  
        misterio = 5*x + 8  
    else:  
        misterio = -2*x*x + 1  
    return misterio  
  
misterio=funcionMisterio(12)  
print("El valor de misterio es: ",misterio)
```

The image shows a screenshot of a Python IDE window titled "Untitled". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code is written in a text area with a vertical scrollbar on the right. The code defines a function "funcionMisterio" that takes a parameter "x". Inside the function, there is an "if" statement: "if (x>3 and x<=10):". If this condition is true, it calculates "misterio = 5\*x + 8". If the condition is false, it goes to the "else" block and calculates "misterio = -2\*x\*x + 1". After the "if-else" block, it returns "misterio" with "return misterio". Below the function definition, the function is called with "misterio=funcionMisterio(12)". Finally, it prints the result with "print('El valor de misterio es: ',misterio)". The status bar at the bottom right shows "Ln: 1 Col: 0".

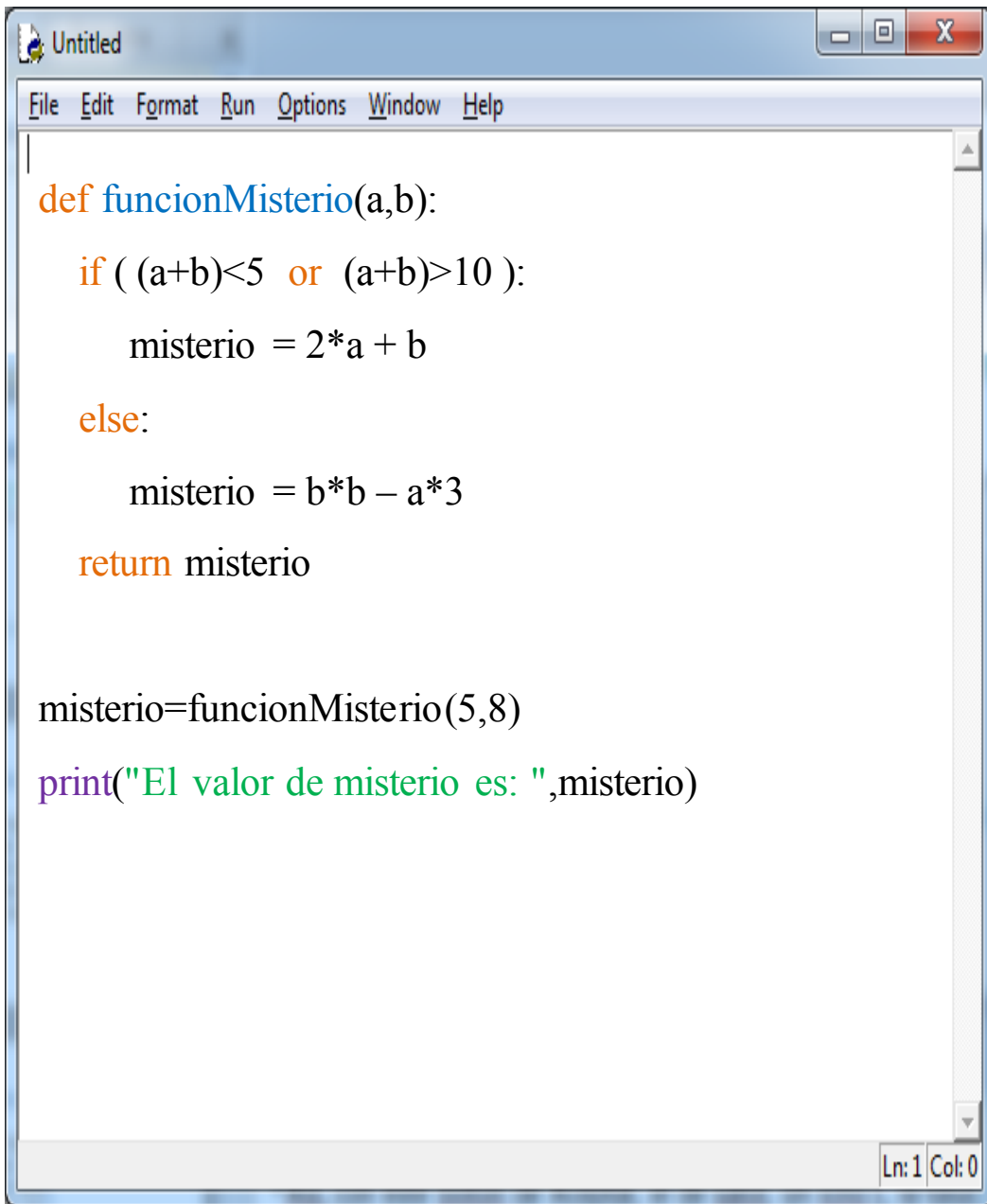


```
def funcionMisterio(x):  
    if (x>3 and x<=10):  
        misterio = 5*x + 8  
    else:  
        misterio = -2*x*x + 1  
    return misterio  
  
misterio=funcionMisterio(12)  
print("El valor de misterio es: ",misterio)
```

Ln: 1 Col: 0

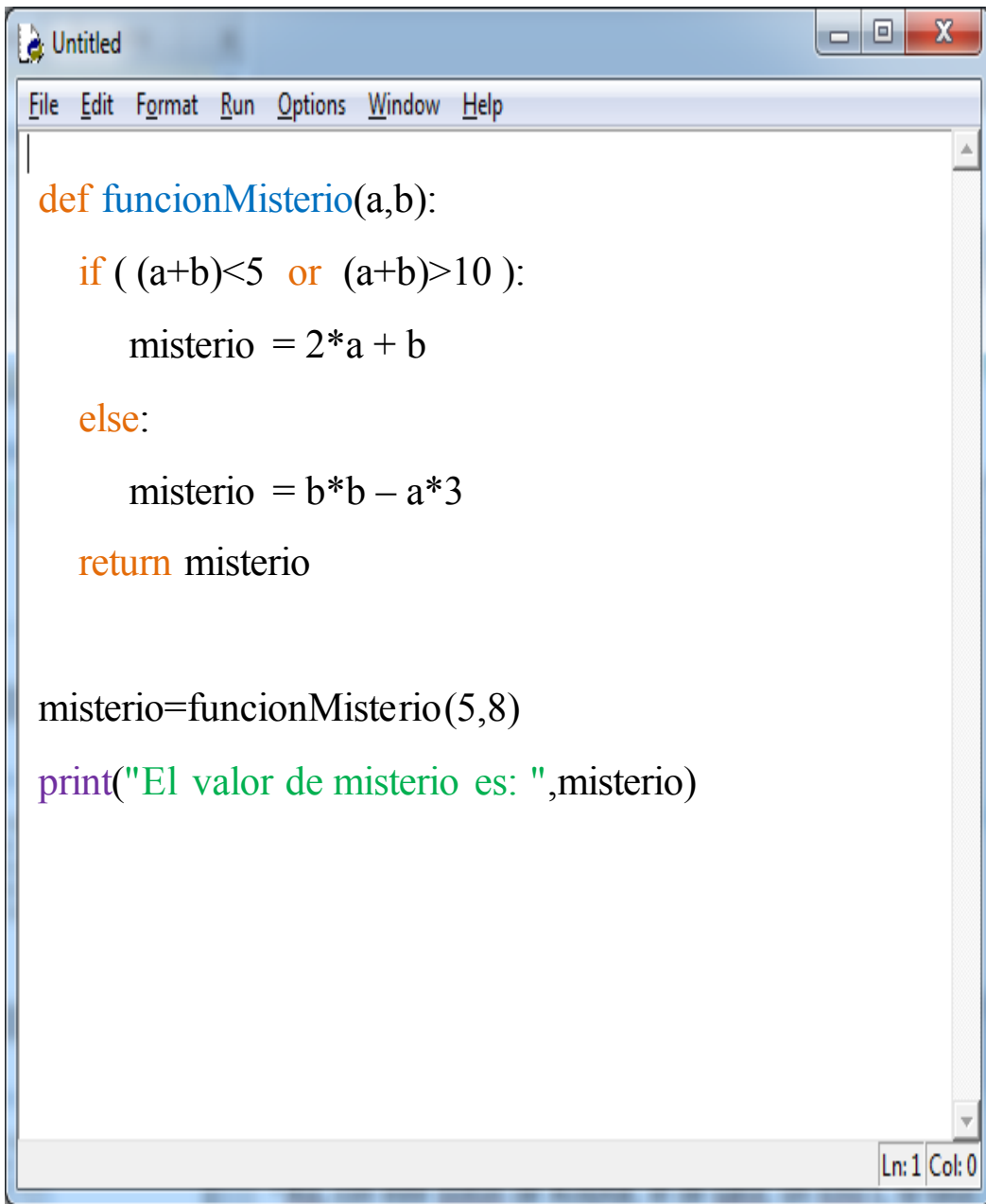
El valor de misterio es -287





```
def funcionMisterio(a,b):  
    if ( (a+b)<5 or (a+b)>10 ):  
        misterio = 2*a + b  
    else:  
        misterio = b*b - a*3  
    return misterio  
  
misterio=funcionMisterio(5,8)  
print("El valor de misterio es: ",misterio)
```

The image shows a screenshot of a Python IDE window titled "Untitled". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains a Python function definition and its usage. The function is named "funcionMisterio" and takes two arguments, "a" and "b". It uses an "if" statement to check if "(a+b) < 5 or (a+b) > 10". If true, it calculates "misterio = 2\*a + b". Otherwise, it calculates "misterio = b\*b - a\*3". The function returns "misterio". Below the function definition, the function is called with "misterio=funcionMisterio(5,8)" and the result is printed using "print('El valor de misterio es: ',misterio)". The status bar at the bottom right shows "Ln: 1 Col: 0".



```
def funcionMisterio(a,b):  
    if ( (a+b)<5 or (a+b)>10 ):  
        misterio = 2*a + b  
    else:  
        misterio = b*b - a*3  
    return misterio  
  
misterio=funcionMisterio(5,8)  
print("El valor de misterio es: ",misterio)
```

Ln: 1 Col: 0

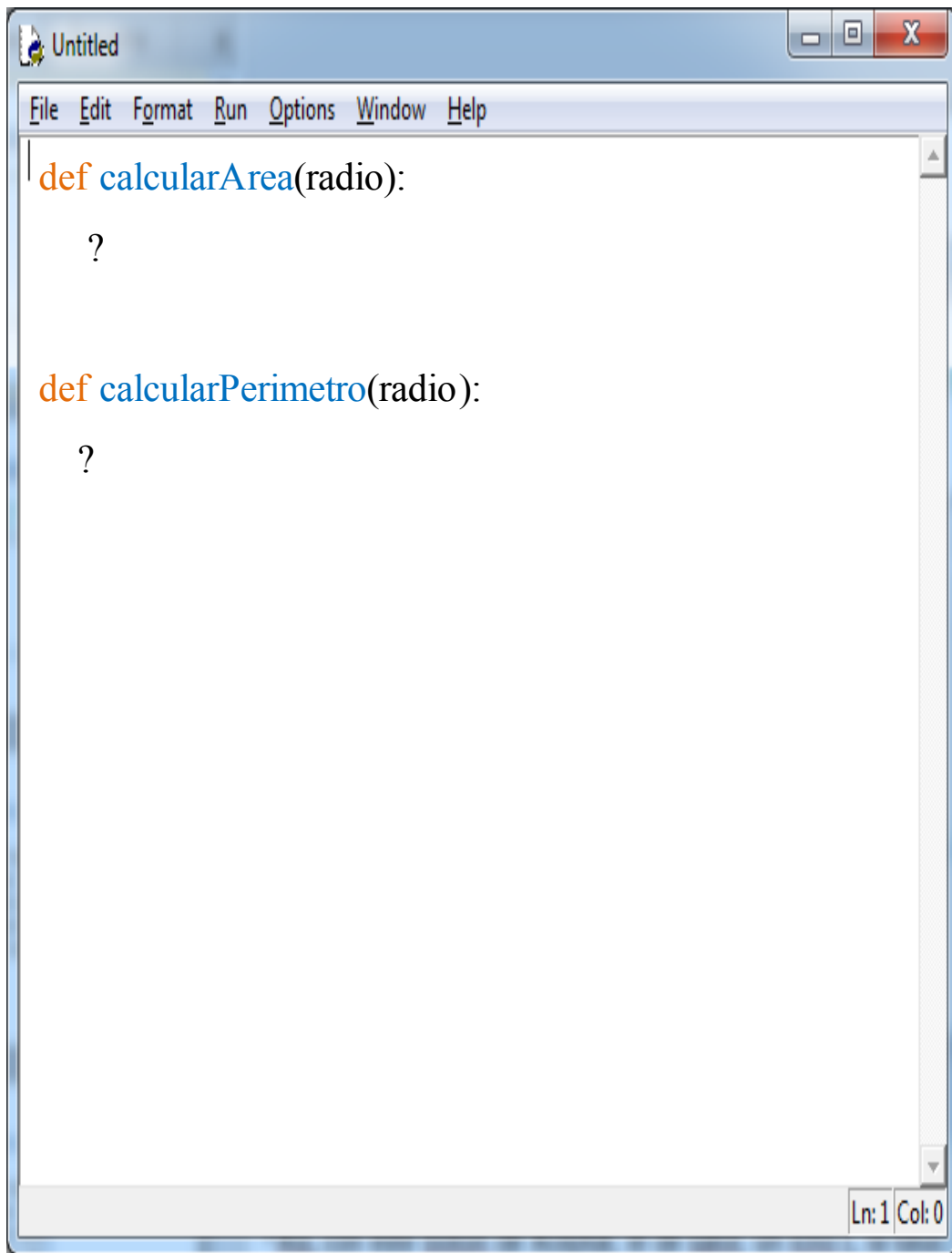
El valor de misterio es 18

# Funciones en Python

---

**Problema\*:** Desarrollar dos funciones, una que calcula el área de un círculo y otra para el perímetro de un círculo. Usar las funciones tres veces para calcular el área y el perímetro de los círculos con los siguientes tres radios:

radio
1.5
5.4
7.8



```
def calcularArea(radius):  
    ?  
  
def calcularPerimetro(radius):  
    ?
```

Ln: 1 Col: 0