



Pensamiento Computacional

Formateo de salidas,
documentación y llamado a
librerías.

Cadenas

Algunos caracteres no pueden ser incluidos directamente en una cadena... por ejemplo las comillas o el apóstrofe.

Uso de \

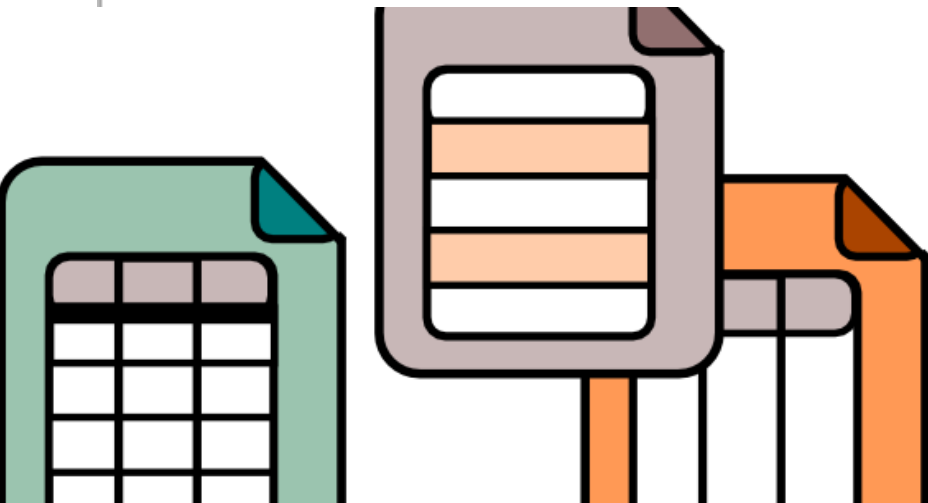
Al utilizar la diagonal invertida (slash \) antes de las comillas, puedo tener un texto que CONTENGA comillas.

```
print ("Quiero que se impriman \"comillas\" aqui")
```

Observa que el slash debe escribirse antes de cada comilla que desees que aparezca en pantalla.

```
[>>> print ("Quiero que se impriman \"comillas\" aqui")
Quiero que se impriman "comillas" aqui
>>> █
```

También puedes jugar con apóstrofes para el texto principal y comillas para donde quieres que aparezcan comillas.



Shell ×

Python 3.7.7 (bundled)

```
>>> print('Quiero que aparezcan "comillas" aquí')
Quiero que aparezcan "comillas" aquí
>>> |
```

Otros usos

- `print ("Quiero que el texto \t tenga una tabulación")`
- `print ("De igual manera \n inserto un cambio de linea")`

`\n` ó `"""`

- Si tu escribes un texto iniciándolo con `"""`, cada vez que presiones **<enter>** en la captura, *python* lo interpretará como `\n`.

```
AnaRaquel — Python — 80x15
>>>
>>> print (""" Probemos si es verdad
... que Python entiende
... donde quiero un <enter>""")
Probemos si es verdad
que Python entiende
donde quiero un <enter>
>>>
```

Al iniciar la función *print* con 3 comillas, si yo presiono <enter> la función me permite continuar en el siguiente renglón y “guarda” dónde hubo un <enter>.

Comentarios y documentación

Los comentarios son anotaciones al código utilizadas para hacerlo más fácil de entender .
No afectan la ejecución del código.

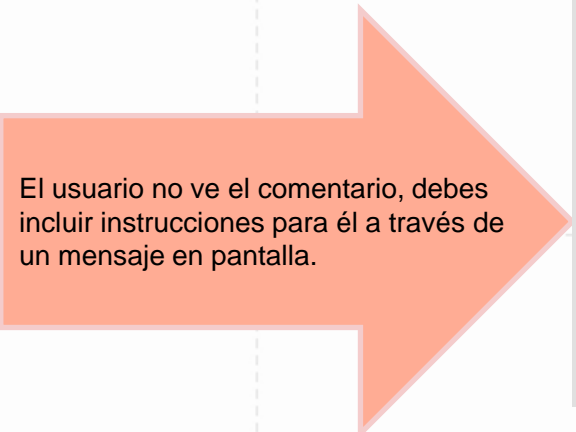
Se crean insertando un numeral #.

```
1  """
2  Mi primer programa de cálculo de edad
3  Ana Raquel
4  """
5
6  print(" Este programa calcula tu edad en el 2037")
7  nacimiento=int(input("Dime en que año naciste: "))
8  edad2037=2037-nacimiento  #es una simple resta
9  print("Tu edad en el año 2037 será ",edad2037)  #agrego la varial
```

Por el momento lo correcto es
utilizar comentarios con #

Comentarios y documentación

Como ya lo mencionamos, los comentarios no aparecen en la ejecución. Por lo tanto NO sustituyen en ninguna forma las instrucciones que el usuario del programa debe recibir mientras lo está corriendo.



El usuario no ve el comentario, debes incluir instrucciones para él a través de un mensaje en pantalla.

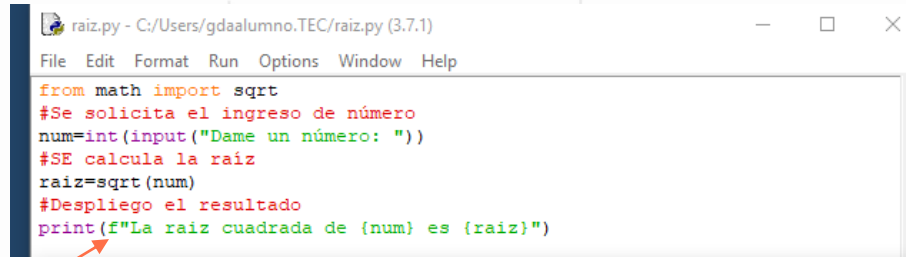
```
1  """
2  Mi primer programa de cálculo de edad
3  Ana Raquel
4  """
5
6  print(" Este programa calcula tu edad en el 2037")
7  nacimiento=int(input("Dime en que año naciste: "))
8  edad2037=2037-nacimiento  #es una simple resta
9  print("Tu edad en el año 2037 será ",edad2037)  #agrego la variabel
```

Formateo de salidas decimales

Python tiene muchas maneras de formatear una salida. Para nuestras necesidades inmediatas, sólo utilizaremos el atributo *f*

f

f me permite llamar a la variable JUSTO donde la voy a necesitar, sólo debo encerrarla en corchetes dentro del string de salida: `print()`.



```
from math import sqrt
#Se solicita el ingreso de número
num=int(input("Dame un número: "))
#SE calcula la raíz
raiz=sqrt(num)
#Despliego el resultado
print(f"La raíz cuadrada de {num} es {raiz}")
```

Se llama al contenido de cada variable

Solo se agrega una *f* antes de empezar el mensaje entrecomillado

```
raiz.py - C:/Users/gdaalumno.TEC/raiz.py (3.7.1)
File Edit Format Run Options Window Help
from math import sqrt
#Se solicita el ingreso de número
num=int(input("Dame un número: "))
#SE calcula la raíz
raiz=sqrt(num)
#Despliego el resultado
print(f"La raíz cuadrada de {num} es {raiz}")
```

```
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/gdaalumno.TEC/raiz.py =====
Dame un número: 125
La raíz cuadrada de 125 es 11.180339887498949
>>> |
```

¡Son muchos
decimales!



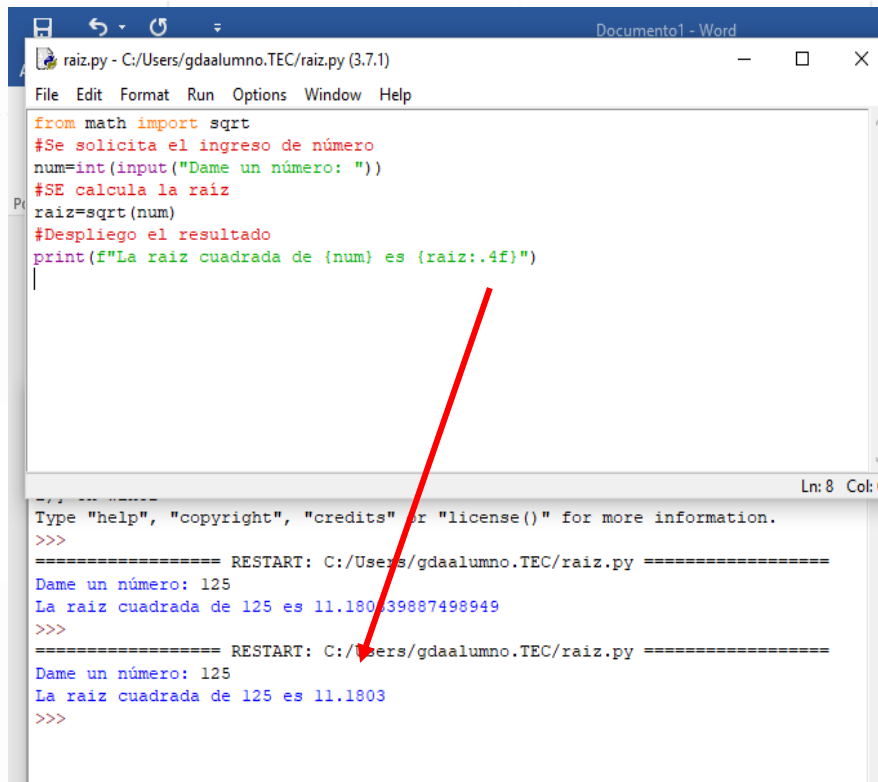
Eso es lo que vamos a formatear

El formato lo asignaremos a cada elemento del “texto” a imprimir.

`{raiz:.4f}`

implicará que la variable raiz se imprimirá como un *float* con 4 decimales

Se escribe **:** y luego un punto **.** seguido del *número de decimales* y la letra *f* para indicar float



```
raiz.py - C:/Users/gdaalumno.TEC/raiz.py (3.7.1)
File Edit Format Run Options Window Help

from math import sqrt
#Se solicita el ingreso de número
num=int(input("Dame un número: "))
#SE calcula la raiz
raiz=sqrt(num)
#Despliego el resultado
print(f"La raiz cuadrada de {num} es {raiz:.4f}")

Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/gdaalumno.TEC/raiz.py =====
Dame un número: 125
La raiz cuadrada de 125 es 11.180339887498949
>>>
===== RESTART: C:/Users/gdaalumno.TEC/raiz.py =====
Dame un número: 125
La raiz cuadrada de 125 es 11.1803
>>>
```


Librerías en python



Estructura típica de un programa

1. Importar librerías
2. Ingreso de datos
3. Instrucciones del proceso
4. Salidas

```
1 """
2 Programa que calcula la raíz cuadrada de un número dado por
3 """
4 from math import sqrt
5 print("Este programa calcula la raíz cuadrada de un número")
6 # Se solicita el ingreso de un número y se guarda en num
7 num=int(input("Dame un número para obtener su raíz "))
8 # Se calcula la raíz cuadrada con la función sqrt de la lib
9 raiz=sqrt(num)
10 print(f"La raíz cuadrada de {num} es {raiz:.3f}")
11 # Insertamos el valor de num y de raíz y formateamos raíz co
```

Shell <

```
>>> %Run raiz.py
```

```
Este programa calcula la raíz cuadrada de un número
Dame un número para obtener su raíz 125
La raíz cuadrada de 125 es 11.180
```

```
>>>
```

Recuerda que también es conveniente describir al usuario de qué se trata el programa.

Librería math

Se puede importar la librería completa

- `import math`
- `print(math.sqrt(25))`

O se puede importar una función de la librería

- `from math import pow`
- `print(pow(2,3))`

```
import math
math.pi #Pi, 3.14...
math.e  #Número de Euler, 2.71...
math.degrees(2)  #2 radianes = 114.59 grados
math.radians(60) #60 grados = 1.04 radianes
math.sin(2)      #Seno de 2 radianes
math.cos(0.5)    #Coseno de 0.5 radianes
math.tan(0.23)   #Tangente de 0.23 radianes
math.factorial(5) #1 * 2 * 3 * 4 * 5 = 120
math.sqrt(49)    #Raíz cuadrada de 49 = 7
```

<https://docs.python.org/3/library/math.html>

A programar

THINK



CODE



Recuerda...

01

Analizar

Comprender que nos está pidiendo el problema

02

Diseño

Hacemos un plan de acción, un algoritmo.

03

Codear (codificar)

Es hora de trasladar el algoritmo a código.

04

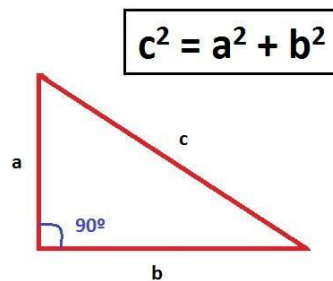
Pruebas

Comprobamos su efectividad y eficacia.

Ejercicio: Resuelve los siguientes problemas, cuida la interacción con el usuario.

1. Realizar un programa que reciba los dos catetos de un triángulo rectángulo y encuentre su hipotenusa
2. Utiliza la librería math para llamar a PI y obtener el área de un círculo dado el radio.
3. Dado el cateto opuesto y el ángulo de un triángulo rectángulo, realiza un programa para calcular la hipotenusa.

La fórmula para calcular la hipotenusa es igual al cateto opuesto / seno del ángulo.



Cuida mucho la presentación de resultados.
Documenta tu código y presume tu habilidad para formatear las salidas.

Ejercicio:

1. Desarrolla un programa que calcule el volumen de una esfera.

$$V = \frac{4}{3} \pi r^3$$



2. Crea un programa que resuelva la siguiente operación.

$$z = \frac{b}{2a} (x - y)$$

Ejercicio:

Billetes

Eres el cajero de un banco y tienes que entregar la cantidad solicitada por el usuario en billetes de \$200, \$50, \$20 y monedas de \$1

Ejemplo:	Cantidad en monedas	\$1343
	Equivale a:	6 billetes de \$200
		2 billetes de \$50
		2 billetes de \$20
		3 monedas de \$1

Recuerda hacer primero tu pseudocódigo:

Identifica qué pide el programa, revisa con qué entradas cuentas y plantea la solución.

Qué aprendimos...



Formateo de salidas

Utiliza el punto y un número para “reservar” el espacio en pantalla para desplegar un dato



Librerías y uso de Div y Mod

1. Llamar a toda la librería, llamar una función de una librería.
2. Obtener el entero de una división, obtener el residuo.



Documentación

para escribir un renglón

"""

Varias líneas de código las encierro en tres comillas

"""

```
print("¡Ya sé  
formatear mis  
salidas!")
```



Recursos recomendados

- Tu libro de texto.
 - <https://automatetheboringstuff.com/>
- Video de Yolanda Martínez para uso de funciones predefinidas:
<https://youtu.be/z8KtyZF5WLo>
- Ayuda para dar formato a números, strings y tablas.
 - <https://mkaz.blog/code/python-string-format-cookbook>