

TRABAJANDO CON VENTANA DE CONSOLA

POSICIONAMIENTO DEL CURSOR.

Forma 1:

```
01 from colorama import init
02 init()
03
04 y = 10000
05 pos = lambda y,x: '\x1b[%d;%dH'%(y,x)    # Estas líneas se deben especificar para que
06                                           # pos funciones correctamente.
07 print(pos(13,17) + "PRUEBA 1")
08 print(pos(10,50) + "PRUEBA 2")
09 print(pos(1,1)   + "PRUEBA 3")
10 input()
```

Forma 2:

```
01 from colorama import init
02 init()
03
04 def pos(x,y):
05     return "\033[" + str(x) + ";" + str(y) + "H"
06
07 print(pos(13,17) + "PRUEBA 1")
08 print(pos(10,50) + "PRUEBA 2")
09 print(pos(1,1)   + "PRUEBA 3")
10 input()
```

Forma 3 (Recomendada):

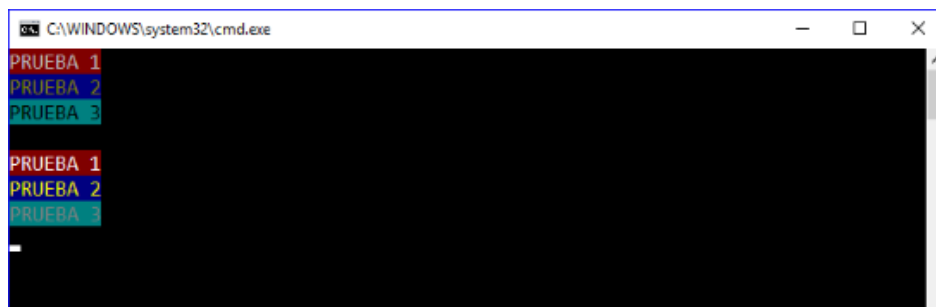
```
01 import os
02 from colorama import init
03 init()
04
05 def pos(x,y):
06     print("\033[" + str(x) + ";" + str(y) + "H", sep="", end="")
07     return None
08
09 pos(13,17); print("PRUEBA 1")
10 pos(10,50); print("PRUEBA 2")
11 pos(1,1);   print("PRUEBA 3")
12 input()
```



DESPLEGAR EN COLORES.

Ejemplo: El siguiente código despliega en pantalla las palabras **PRUEBA 1**, **PRUEBA 2** y **PRUEBA 3** con distintos Colores Opacos y Brillos. Observe que entre las líneas 04 y 09 se despliegan los colores opacos. En cambio, entre las líneas 11 a la 16 se despliegan con colores brillantes.

```
01 from colorama import init, Fore, Back, Style
02 init()
03
04 print(Back.RED + Fore.WHITE, end="")
05 print("PRUEBA 1")
06 print(Back.BLUE + Fore.YELLOW, end="")
07 print("PRUEBA 2")
08 print(Back.CYAN + Fore.BLACK, end="")
09 print("PRUEBA 3")
10 print()
11 print(Back.RED + Fore.WHITE + Style.BRIGHT, end="")
12 print("PRUEBA 1")
13 print(Back.BLUE + Fore.YELLOW + Style.BRIGHT, end="")
14 print("PRUEBA 2")
15 print(Back.CYAN + Fore.BLACK + Style.BRIGHT, end="")
16 print("PRUEBA 3")
17 input()
```



El mismo resultado se obtiene con el siguiente código:

```
01 from colorama import init, Fore, Back, Style
02 init()
03
04 print(Back.RED + Fore.WHITE + "PRUEBA 1")
05 print(Back.BLUE + Fore.YELLOW + "PRUEBA 2")
06 print(Back.CYAN + Fore.BLACK + "PRUEBA 3")
07 print("PRUEBA 3")
08 print()
09 print(Back.RED + Fore.WHITE + Style.BRIGHT + "PRUEBA 1")
10 print(Back.BLUE + Fore.YELLOW + Style.BRIGHT + "PRUEBA 2")
11 print(Back.CYAN + Fore.BLACK + Style.BRIGHT + "PRUEBA 3")
12 input()
```

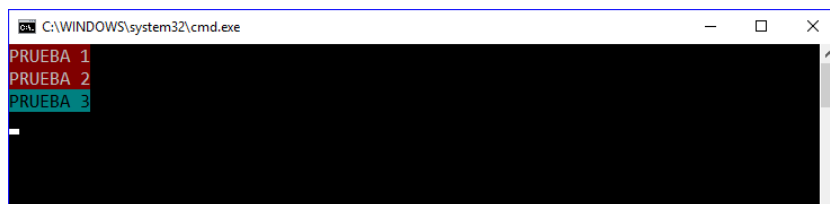
Existen 2 formas de aplicar colores: Persistente y Momentáneamente.

- **Persistentemente:** Para que el color persista, de tal forma que todo lo escrito en pantalla sea con el color especificado, hasta que se vuelva a cambiar por otro, a la función `init()` no se le especifican el parámetro de inicialización (se omite) o se anota `autoreset = False`, por ejemplo: `init(autoreset = False)`

- **Momentáneamente:** Para que el color se aplique solamente a lo desplegado con la función `print`, la función `init()` debe llevar el parámetro `autoreset = True`, por ejemplo: `init(autoreset = True)`. En tal caso la configuración de colores y el mensaje a desplegar debe estar incorporados en el mismo `print`.

Ejemplo de Colores Persistentes: En el siguiente ejemplo, observe que el color definido en la línea 4 persiste en las líneas 5 y 6.

```
01 from colorama import init, Fore, Back
02 init() # También se puede emplear init(autoreset = False)
03
04 print(Back.RED + Fore.WHITE, end="")
05 print("PRUEBA 1")
06 print("PRUEBA 2")
07 print(Back.CYAN + Fore.BLACK, end="")
08 print("PRUEBA 3")
09 input()
```

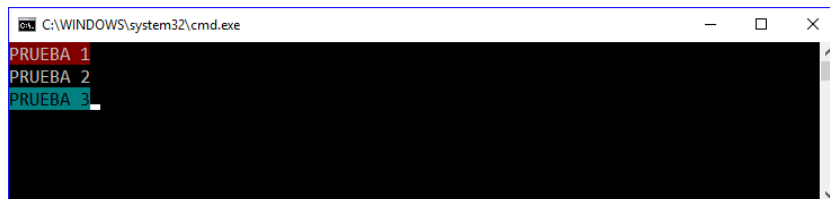


El siguiente código genera el mismo resultado: Los colores definidos en la línea 4 persisten en la línea 5.

```
01 from colorama import init, Fore, Back
02 init() # También se puede emplear init(autoreset = False)
03
04 print(Back.RED + Fore.WHITE + "PRUEBA 1", end="")
05 print("PRUEBA 2")
06 print(Back.CYAN + Fore.BLACK, end="")
07 print("PRUEBA 3")
08 input()
```

Ejemplo de Colores Momentáneos: En el siguiente ejemplo, observe que el color definido en la línea 4 no afecta al de la línea 5.

```
01 from colorama import init, Fore, Back
02 init(autoreset = True)
03
04 print(Back.RED + Fore.WHITE + "PRUEBA 1", end="")
05 print("PRUEBA 2")
06 print(Back.CYAN + Fore.BLACK + "PRUEBA 3", end="")
07 input()
```



DESPLEGAR VENTANA DE MENSAJE (MessageBox)


Permite desplegar una ventana de diálogo, la que puede configurarse en diversos tipos. **MessageBox** es una función que entrega como resultado un valor que representa al botón seleccionado para cerrar la ventana.

SINTAXIS: win32ui.MessageBox(MENSAJE, TÍTULO, TIPO)

BIBLIOTECA: win32ui

TIPO	EJEMPLO
0	
1	
2	
3	
4	
5	
6	

Al valor de TIPO se le puede sumar los valores, 16, 32 y 64 para incorporar uno de los siguientes íconos:

16		Ejemplo: import win32ui win32ui.MessageBox("ESTO ES UN MENSAJE", "ESTO ES UN TÍTULO", 1 + 64)
32		
64		

Valor de retorno	Botón
1	Aceptar
2	Cancelar
3	Anular
4	Reintentar
5	Omitir
6	Sí
7	No
11	Continuar

Para que sea más cómo su uso, lo emplearemos de la siguiente forma:

```
01 import win32ui
02
03 def MsgBox(Mensaje="", Titulo="", Tipo=0):
04     return win32ui.MessageBox(Mensaje,Titulo,Tipo)
05
06 MsgBox("Este es un mensaje", "Titulo", 1 + 64)
```

LIMPIAR PANTALLA.

```
01 import os
02 os.system("cls")
```

GENERAR UNA PAUSA.

```
01 import time
02
03 time.sleep(5)      # Espera por 5 segundos
04 time.sleep(.300)   # Espera por 300 milisegundos
```