Lista de comandos de Hermoso.py

Podés encontrar la lista actualizada en <https://github.com/despediteerik/hermoso/wiki/Comandos>.

Las librerías que se pueden encontrar en *Hermoso.py* son seis: tres regulares (para poder crear un párrafo de un color/fondo/formato específico), y tres avanzadas (para poder cambiar el texto en medio de un párrafo). Las regulares empiezan con una letra p (pcolor, pfondo, pformato), y las avanzadas no tienen un prefijo (color, fondo, formato).

**LIBRERÍAS REGULARES**

**pcolor**

La librería pcolor tiene los comandos para poder colorear el texto en específico de un párrafo. El sintáxis es pcolor.COLOR("TEXTO").

Los colores disponibles en pcolor/color son negro, rojo, verde, amarillo, azul, violeta, celeste, gris, rojoclaro, verdeclaro, amarilloclaro, azulclaro, violetaclaro, celesteclaro, grisclaro y blanco.

**pfondo**

La librería pfondo tiene los comandos para poder colorear el fondo de el texto en específico de un párrafo. El sintáxis es pfondo.COLOR("TEXTO").

Los colores disponibles en pfondo/fondo son negro, rojo, verde, amarillo, azul, violeta, celeste, gris, rojoclaro, verdeclaro, amarilloclaro, azulclaro, violetaclaro, celesteclaro, grisclaro, blanco, blancopuro (texto blanco arriba de fondo blanco) y negropuro (texto negro arriba de fondo negro).

**pformato**

La librería pformato tiene los comandos para poder formatear un párrafo de distintas formas. El sintáxis es pformato.FORMATO("TEXTO").

Los formatos disponibles son negrita, subrayado, reverso, y cursiva. El soporte entre líneas de comandos de estos es limitado.

**LIBRERÍAS AVANZADAS**

**color**

La librería color tiene los comandos para poder colorear el texto en específico de una parte de un print(""). El sintáxis es color.COLOR("TEXTO"), y se usa dentro de un print de la forma: print("color.COLOR("TEXTO")).

Los colores disponibles en pcolor/color son negro, rojo, verde, amarillo, azul, violeta, celeste, gris, rojoclaro, verdeclaro, amarilloclaro, azulclaro, violetaclaro, celesteclaro, grisclaro y blanco.

Prácticamente, esto se usa de esta forma: print("Este texto está en " + color.rojo("rojo") + ".")

**fondo**

La librería fondo tiene los comandos para poder colorear el fondo de el texto en específico de una parte de un print(""). El sintáxis es fondo.COLOR("TEXTO"), y se usa dentro de un print de la forma: print("fondo.COLOR("TEXTO")).

Los colores disponibles en pfondo/color son negro, rojo, verde, amarillo, azul, violeta, celeste, gris, rojoclaro, verdeclaro, amarilloclaro, azulclaro, violetaclaro, celesteclaro, grisclaro, blanco, blancopuro (texto blanco arriba de fondo blanco) y negropuro (texto negro arriba de fondo negro).

Prácticamente, esto se usa de esta forma: print("El fondo de este texto está en " + fondo.rojo("rojo") + ".")

**formato**

La librería formato tiene los comandos para poder formatear un párrafo de distintas formas. El sintáxis es formato.FORMATO("TEXTO"), y se usa dentro de un print de la forma: print("formato.FORMATO("TEXTO")).

Los formatos disponibles son negrita, subrayado, reverso, y cursiva. El soporte entre líneas de comandos de estos es limitado.

Los colores disponibles en pfondo/fondo son negro, rojo, verde, amarillo, azul, violeta, celeste, gris, rojoclaro, verdeclaro, amarilloclaro, azulclaro, violetaclaro, celesteclaro, grisclaro, blanco, blancopuro (texto blanco arriba de fondo blanco) y negropuro (texto negro arriba de fondo negro).

Prácticamente, esto se usa de esta forma: print("Este texto está en " + formato.negrita("negrita") + ".")

**CÓDIGOS ESPECIALES**

**formato.limpiar()**

Limpia la pantalla. Este comando depende de la librería os, y solo está confirmado que funcione en Windows.

**color.custom()**

Podés modificar el color de el texto y el fondo, individualmente! Esto toma tres entradas: el TEXTO, el COLOR DEL TEXTO, y el COLOR DEL FONDO. Su uso es de color.custom("Esto es un texto personalizado!",39,42).

El número del texto tiene que ser de 30-39, y el número del color de fondo tiene que ser de 40-49. También podés usarlo en un parágrafo con pcolor.custom("Esto es un párrafo personalizado!",30,45).

Instalando Hermoso.py

Podés encontrar las instrucciones en <https://github.com/despediteerik/hermoso/wiki/Instalaci%C3%B3n>.

Hay dos formas de instalar la aplicación, ya que de ambas formas requiere dependencias externas.

Si estás usando una distribución de linux o tenés Windows 10 con las últimas actualizaciones, no ténés que hacer nada – ¡ya funciona por defecto! Los caractéres de escape ANSI ya son compatibles con tu versión de la terminal.

La primera (y más fácil) forma de instalarlo si no tenés una pestaña de comandos compatible con ANSI es descargar e instalar conemu, un emulador de la pestaña de comandos que agrega soporte para colores, pestañas (para tener varias aplicaciones abiertas), etcétera. podés descargarla en [conemu.github.com](https://conemu.github.com).

La forma más definitiva, si todavía querés usar CMD tradicional, involucra descargar ANSICON. ANSICON es un programa que da soporte para poder agregar a la terminal carácteres de secuencia ANSI, lo cual va a permitir el uso de colores y formato dentro de la terminal.

Descargá la última versión en [adoxa.altervista.com/ansicon](http://adoxa.altervista.com/ansicon). dejalo en una carpeta del sistema donde no se elimine y [instalálo en tu PATH](https://medium.com/@01luisrene/como-agregar-variables-de-entorno-s-o-windows-10-e7f38851f11f), agregando una nueva variable de entorno con el nombre ANSICON y el directorio donde está tu carpeta de ANSICON. Cuando termines, abrí CMD y dirigite a la carpeta de ANSICON/x64 si tenés una PC de 64 bits, o x86 si es de 32 bits. Finalmente, ejecutá el comando ansicon.exe –h. ¡listo! ANSICON ya está instalado en la computadora.

Para terminar la instalación, dejá los archivos de la carpeta "hermoso" en el Lib de tu directorio de Python, comumnente encontrado en C:/Python/Lib. ¡Ya podés usar hermoso.py!