«Talento Tech»

Iniciación a la Programación con Python

CLASE 16

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Buenos Aires Ciudad

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Clase N° 16 | Proyecto final

Temario:

- Módulo Colorama.
- Retroalimentación general del curso y de los proyectos.
- Conclusiones y cierre final.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Introducción.

Estamos en la última clase del curso, una etapa de cierre que nos permitirá redondear y pulir varios conceptos importantes, desarrollados a lo largo de varias semanas. Hasta aquí, aprendimos a utilizar Python para crear una aplicación que gestione datos, trabajamos con estructuras como los diccionarios y finalmente implementamos bases de datos para que la información pueda guardarse y recuperarse cuando la necesitemos.

También conocimos los módulos en Python y estudiamos cómo nos permiten organizar y simplificar el código, ahorrando tiempo al reutilizar funciones que ya están diseñadas para tareas específicas. Empezamos con módulos simples como *math y random*, aprendiendo a realizar cálculos y generar valores aleatorios sin necesidad de escribir todo desde cero. Luego, dimos un paso más con *sqlite3*, un módulo esencial para interactuar con bases de datos y permitir que nuestros programas manejen información de manera permanente. Este avance fue fundamental en nuestro Proyecto Final Integrador, una aplicación que no sólo funciona temporalmente, sino que es capaz de guardar datos de forma persistente.

Hoy vamos a sumar un recurso más a esta caja de herramientas de Python: **el módulo Colorama**. Vamos a aprender a agregar colores y estilos a los mensajes que se muestran en la consola, lo cual nos permitirá hacer que la interfaz de nuestra aplicación sea más clara y atractiva. Este módulo nos ayudará a destacar información importante, guiar a la usuaria o usuario en el uso del programa y, sobre todo, darle un toque personal y profesional a nuestros proyectos de manera que cada mensaje sea más fácil de leer y entender.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Módulo Colorama.

Colorama es un módulo externo, por lo que generalmente necesita ser instalado, ya que no viene preinstalado con Python. Para esto, utilizaremos **el gestor de paquetes pip,** que es una herramienta muy común en Python para añadir librerías externas a un proyecto.

Instalación de Colorama.

Si nunca instalaste Colorama en tu entorno, sólo tenés que ejecutar el siguiente comando en la terminal o consola:

```
pip install colorama
```

Este comando descargará e instalará la última versión de Colorama.

Importación de Colorama.

Una vez instalado, vas a poder importarlo y utilizarlo en tu código de la siguiente manera:

```
from colorama import init, Fore

init()  # Inicializa Colorama

print(Fore.GREEN + "Este mensaje se muestra en verde.")
print(Fore.RED + "Este mensaje se muestra en rojo.")
```

Al utilizar **init()** junto con *Fore y Style*, podrás dar un toque visual a los mensajes de tu aplicación, destacando información importante y guiando mejor a la usuaria o usuario. Esta es la salida del código de la prueba anterior:

```
Este mensaje se muestra en verde.
Este mensaje se muestra en rojo.
```

¿Qué es una “secuencia ANSI”?

ANSI es una norma de codificación establecida por el Instituto Nacional Estadounidense de Estándares (American National Standards Institute) y, en el contexto de consolas de texto, se refiere a un conjunto de códigos especiales que permiten agregar colores y estilos (como texto en negrita o subrayado) a la salida de texto en la terminal. Estos códigos, conocidos como secuencias ANSI, son instrucciones que la terminal interpreta para mostrar el texto en colores específicos o con ciertos efectos visuales.

En el fondo, las secuencias ANSI son básicamente combinaciones de caracteres que la terminal reconoce como instrucciones en lugar de texto normal. Por ejemplo, para que un texto se muestre en rojo, se envía a la terminal una secuencia ANSI específica que activa el color rojo y luego se envía otra secuencia que restablece el color al valor predeterminado al final de la línea. Así, en lugar de simplemente imprimir texto plano, las secuencias ANSI le indican a la terminal que ese texto debe verse de una forma particular.

El inconveniente que tienen estas secuencias es que no siempre son sencillas de recordar. Por ejemplo, la secuencia ANSI para indicar que el texto que sigue debe verse de color rojo es **\033[31m** . Y para volver al color normal, usaríamos **\033[0m** .

Pero afortunadamente existe Colorama. En lugar de escribir estas secuencias manualmente, que pueden ser confusas y difíciles de recordar, Colorama simplifica el proceso proporcionando una interfaz más legible y práctica para aplicar colores y estilos. Esto hace que el código sea más fácil de leer y entender, especialmente si se trabaja en varios sistemas operativos, ya que Colorama se encarga de que las secuencias ANSI se interpreten correctamente en cada plataforma.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

La función init().

Esta función prepara el entorno de nuestra consola para que podamos usar los colores y estilos sin problemas, especialmente en sistemas Windows, que suelen necesitar una configuración adicional para que los códigos de color se muestren correctamente.

La función init() es, en cierto sentido, como el botón de “encendido” de Colorama: una vez que la llamamos, el módulo está listo para que usemos sus opciones de color en nuestros textos. Dentro de init(), podemos ajustar algunos parámetros para controlar el comportamiento del módulo según nuestras necesidades:

Autoreset: si configuramos init(autoreset=True), cada vez que usemos un color o un estilo en la consola, ese formato se aplicará sólo a esa línea o bloque de texto específico y, una vez mostrado el color se “resetea” al valor predeterminado. Esto es útil para evitar que, al aplicar un color a una línea, todo lo que venga después se muestre también en ese color.

```
from colorama import init, Fore
init(autoreset=True)

print(Fore.RED + "Este texto es rojo.")
print("Este texto vuelve al color original automáticamente.")
```

Convert: otro parámetro es convert, que ajusta automáticamente la compatibilidad en sistemas Windows. Esto no suele ser necesario modificarlo manualmente, ya que Colorama detecta el sistema operativo y lo configura solo.

```
from colorama import init, Fore
init(convert=True)

print(Fore.GREEN + "Este texto debería verse en verde en todos los
sistemas, incluyendo Windows.")
```

Strip: cuando usamos strip=True, los códigos de formato que no son compatibles se eliminan de manera automática. Este parámetro es útil si vamos a ejecutar el programa en diferentes sistemas, como Windows y Linux, porque ayuda a evitar errores de formato.

```
from colorama import init, Fore
init(strip=True)

print(Fore.BLUE + "Este texto es azul en consolas compatibles.")
```

Usando init() iniciamos Colorama de manera que se adapte al sistema operativo que estemos usando. Esto hace que el módulo sea fácil de utilizar sin tener que preocuparnos por problemas de compatibilidad y nos deja concentrarnos en cómo darle un toque de color a nuestra aplicación.

Fore, Back y Style.

Para hacer que los textos se destaquen en la consola, Colorama nos ofrece tres elementos fundamentales: **Fore, Back y Style.** Con ellos, podemos aplicar colores y estilos a nuestras impresiones, lo que nos ayuda a organizar visualmente la información y a resaltar detalles importantes.

Fore.

Fore es la opción que nos permite cambiar el color del texto, es decir, el color de las letras que mostramos en pantalla. Colorama ofrece una gama de colores básicos que podemos aplicar con Fore, como Fore.RED, Fore.GREEN, Fore.BLUE, entre otros. Usar Fore es útil para resaltar mensajes específicos en la consola, como alertas, confirmaciones o errores.

Por ejemplo, si queremos destacar un mensaje de advertencia en color rojo, podemos escribir lo siguiente:

```
from colorama import init, Fore
init(autoreset=True)
```

```
print(Fore.RED + "Cuidado: estás a punto de eliminar un producto.")
```

En el contexto de nuestro TFI, podríamos utilizar Fore para mostrar mensajes en diferentes colores. Supongamos que queremos que, cuando se registre un producto correctamente, el sistema muestre el mensaje en verde para indicar que fue exitoso:

```
print(Fore.GREEN + "Producto registrado con éxito.")
```

El resultado de todo el código anterior es el siguiente:

```
Cuidado: estás a punto de eliminar un producto.
Producto registrado con éxito.
```

Como vimos, con autoreset=True, el color vuelve al original después de cada línea, lo que nos permite aplicar un color a una línea específica sin afectar al resto.

Back:

Back permite cambiar el color del fondo, lo cual puede ser útil para hacer que un mensaje resalte aún más en la pantalla. Cambiar el color de fondo puede ayudar a diferenciar secciones o crear bloques de información que sean fáciles de identificar visualmente. La paleta de colores de Back es similar a la de Fore, por lo que encontramos opciones como Back.YELLOW, Back.RED y otras.

Supongamos que queremos mostrar un aviso de alerta con fondo amarillo. Esto ayuda a captar la atención de la persona de inmediato:

```
from colorama import Back
print(Back.YELLOW + "Aviso: el stock de este producto es muy bajo.")
```

```
Aviso: el stock de este producto es muy bajo.
```

En el TFI, podríamos utilizar **Back** para resaltar errores específicos en el ingreso de datos. Por ejemplo, si se intenta ingresar un precio negativo para un producto, el mensaje de error puede mostrarse con un fondo rojo para indicar que hay un problema:

```
print(Back.RED + "Error: el precio no puede ser un valor negativo.")
```

Style

Style nos permite cambiar el estilo del texto. Las opciones principales de Style incluyen Style.DIM, Style.NORMAL y Style.BRIGHT. Estas opciones afectan la intensidad del color del texto, lo cual puede ser útil para aplicar distintos niveles de énfasis en la información que mostramos.

Usar Style.BRIGHT hace que el texto sea más intenso, lo cual es ideal para títulos o mensajes importantes. Por ejemplo, para una sección de bienvenida en el TFI, podríamos escribir:

```
from colorama import Style, Fore
print(Style.BRIGHT + Fore.BLUE + "Bienvenido/a a la gestión de
inventario.")
```

Este es el resultado:

```
Bienvenido/a a la gestión de inventario.
```

Por otro lado, Style.DIM reduce la intensidad del color, algo útil para mostrar mensajes secundarios o instrucciones que no requieren tanta atención. Por ejemplo, si queremos mostrar una nota en segundo plano, podríamos hacer lo siguiente:

```
print(Style.DIM + "Nota: asegúrate de ingresar un ID válido.")
```

Uniendo Fore, Back y Style, podemos crear mensajes más complejos y visualmente atractivos. Supongamos que queremos mostrar un aviso completo para cuando el stock de un producto esté por debajo de un límite crítico en el TFI. Utilizando los tres elementos juntos, podríamos hacer algo como esto:

```
from colorama import Style, Back, Fore
print(Back.YELLOW + Fore.RED + Style.BRIGHT + "Alerta: el stock de este
producto es crítico. Se recomienda reponer.")
```

Se ve así:

```
Alerta: el stock de este producto es crítico. Se recomienda reponer.
```

De esta manera, Fore, Back y Style nos brindan opciones flexibles para destacar información en pantalla. Al aplicarlos de manera coherente, logramos que los mensajes importantes se resalten y que se pueda comprender mejor el flujo de nuestra aplicación.

Buenos Aires aprende
Agencia de Habilidades para el Futuro

Aplicación práctica de Colorama.

Este podría ser el código que escribiste para gestionar el menú de opciones en tu TFI:

```
# Función del menú principal que permite acceder a cada función
def mostrar_menu():
    while True:
        print("\nMenú de Gestión de Personas:")
        print("1. Registrar persona")
        print("2. Mostrar personas")
        print("3. Actualizar edad de una persona")
        print("4. Eliminar persona")
        print("5. Buscar persona")
        print("6. Reporte de personas menores de edad")
        print("7. Salir")

        opcion = input("Seleccione una opción: ")


        if opcion == "7":
            print("Saliendo del programa...")
            break
        else:
            print("Opción inválida. Por favor, intente nuevamente.")

# Iniciar el programa llamando al menú principal
mostrar_menu()
```

Y cuando lo ejecutás ves algo como esto:

```
Menú de Gestión de Personas:
1. Registrar persona
2. Mostrar personas
3. Actualizar edad de una persona
4. Eliminar persona
5. Buscar persona
6. Reporte de personas menores de edad
7. Salir
Seleccione una opción:
```

Usando Colorama, podés reescribirlo y hacer que la persona que utiliza la aplicación tenga una experiencia de uso más atractiva. Con estos cambios:

```
from colorama import init, Fore, Back, Style

init(autoreset=True)

# Función del menú principal que permite acceder a cada función
def mostrar_menu():
    while True:
        print(Style.BRIGHT + Fore.BLUE + Back.WHITE + "\nMenú de Gestión
de Personas".center(40, " "))
        print(Style.BRIGHT + Fore.CYAN + "1. Registrar persona")
        print(Style.BRIGHT + Fore.CYAN + "2. Mostrar personas")
        print(Style.BRIGHT + Fore.CYAN + "3. Actualizar edad de una
persona")
        print(Style.BRIGHT + Fore.CYAN + "4. Eliminar persona")
        print(Style.BRIGHT + Fore.CYAN + "5. Buscar persona")
```

```
        print(Style.BRIGHT + Fore.CYAN + "6. Reporte de personas menores
de edad")
        print(Style.BRIGHT + Fore.RED + "7. Salir")

        opcion = input(Fore.YELLOW + "\nSeleccione una opción: ")

        if opcion == "7":
            print(Fore.RED + "Saliendo del programa...")
            break
        else:
            print(Back.RED + Fore.WHITE + "Opción inválida. Por favor,
intente nuevamente.")

# Iniciar el programa llamando al menú principal
mostrar_menu()
```

Tu menú se vería así:

```
Menú de Gestión de Personas                    .
1. Registrar persona
2. Mostrar personas
3. Actualizar edad de una persona
4. Eliminar persona
5. Buscar persona
6. Reporte de personas menores de edad
7. Salir

Seleccione una opción: 7
Saliendo del programa...
```

Mucho mejor, ¿verdad? 🤔

Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad