

```

import ColabTurtlePlus.Turtle as turtle
import time
import random

def get_fade_color(start_hex, bg_hex, ratio):
    """Calculates a color between start_hex and bg_hex based on ratio (0 to 1)."""

    s_rgb = [int(start_hex[i:i+2], 16) for i in (1, 3, 5)]
    b_rgb = [int(bg_hex[i:i+2], 16) for i in (1, 3, 5)]
    fade_rgb = [int(s + (b - s) * ratio) for s, b in zip(s_rgb, b_rgb)]
    return '#{02x}{:02x}{:02x}'.format(*fade_rgb)

def fade_out_text(brush, text, start_x, theme, steps=4):
    """Simulates a fade-out by redrawing text closer to the background color."""
    for step in range(1, steps + 1):
        ratio = step / steps
        brush.clear()
        brush.penup()
        brush.goto(start_x, -15)

        for i, char in enumerate(text):

            current_color = theme["colors"][i % len(theme["colors"])]
            faded = get_fade_color(current_color, theme["bg"], ratio)
            brush.color(faded)
            brush.write(char, font=(theme["font"], 48, "bold"))
            brush.forward(70)

        brush.penup()
        brush.goto(start_x, -15)
        time.sleep(0.1)

def high_contrast_showcase(run_speed=8, loop_count=4):
    turtle.clearscreen()
    turtle.setup(950, 450)
    brush = turtle.Turtle()
    brush.hideturtle()

    themes = [
        {"name": "Light", "colors": ["#00FF00", "#FFFFFF", "#00FFFF"], "font": "Georgia", "bg": "#1A0033"},
        {"name": "Cold", "colors": ["#FFFFFF", "#00BFFF", "#1E90FF"], "font": "Courier", "bg": "#000000"},
        {"name": "Bold", "colors": ["#FF0000", "#FFFFFF", "#FF0000"], "font": "Arial", "bg": "#000000"},
        {"name": "Vivid", "colors": ["#FFFF00", "#FF00FF", "#00FFFF"], "font": "Verdana", "bg": "#121212"}
    ]

```

```
text = "CSTerrellART"

for cycle in range(loop_count):
    theme = themes[cycle % len(themes)]
    turtle.bgcolor(theme["bg"])
    brush.speed(run_speed)
    brush.clear()

    start_x = -400

    brush.penup()
    brush.goto(start_x, -15)
    for i, char in enumerate(text):
        brush.color(theme["colors"][i % len(theme["colors"])])
        brush.write(char, font=(theme["font"], 48, "bold"))
        brush.penup()
        brush.forward(70)
        time.sleep(0.1)

    brush.goto(0, -120)
    brush.color("white" if theme["bg"] != "#ffffff" else "black")
    brush.write(f"Active Theme: {theme['name']}", align="center", font=("Arial", 12, "bold"))

    time.sleep(2.0)

    fade_out_text(brush, text, start_x, theme)

high_contrast_showcase(run_speed=10, loop_count=4)
```