



```
main.py
1 print("Krój normalny")
2 print()
3 print("\033[1mKrój pogrubiony")
4 print("\033[22m")
5 print("\033[2mTekst trochę ciemniejszy")
6 print("\033[22m")
7 print("\033[3mKursywa (tekst pochyły)")
8 print("\033[23m")
9 print("\033[4mTekst podkreślony")
10 print("\033[24m")
11 print("\033[5mTekst mrugający (?)")
12 print("\033[25m")
13 print("\033[7mZamiana tekstu z tłem kolorami")
```

input

```
Krój normalny
Krój pogrubiony
Tekst trochę ciemniejszy
Kursywa (tekst pochyły)
Tekst podkreślony
Tekst mrugający (?)
Zamiana tekstu z tłem kolorami
```

Obsługa konsoli tekstowej w Pythonie

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Konsola tekstowa w Pythonie

- Konsola Pythona to interaktywne narzędzie, które umożliwia pisanie oraz uruchamianie kodu.
- Pozwala na skorzystanie z trybu tekstowego.
- Zaletą są małe wymagania sprzętowe, niezależność od konkretnych środowisk graficznych i możliwość pracy w różnych systemach operacyjnych.

Kody ANSI Escape

- Kody **ANSI escape** (*sekwencje ucieczki*) to zestawy znaków rozpoczynające się od znaku ESC (ASCII **27**, hex **\x1B**, ósemkowo **\033**, Unicode: **\u001b**)
- Używane do sterowania formatowaniem, kolorami i pozycją kursora w terminalach tekstowych (Linux, macOS).
- Najczęściej wykorzystują składnię ESC[(*CSI - Control Sequence Introducer*), np. `\033[1;31m` dla czerwonego tekstu.
- Kody umieszcza się bezpośrednio w drukowanym ciągu znaków, a na końcu stosuje kod resetujący (`\033[0m`), by przywrócić domyślny kolor.

Przykład użycia kodów ANSI Escape

- Czerwony tekst

```
print ("\033[1;31mHello World")
```

```
print ("\x1B[1;31mHello World")
```

- Utrzymanie koloru

```
print ("Hello World")
```

- Przywrócenie koloru domyślnego

```
print ("\033[0m")
```

```
print ("Hello World")
```

```
print ("\033[1;31mHello World")
```

```
print ("\x1B[1;31mHello World")
```

```
print ("Hello World")
```

```
print ("\033[0m")
```

```
print ("Hello World")
```

```
Hello World
```

```
Hello World
```

```
Hello World
```

```
Hello World
```

Podstawowe kody ANSI escape:

- **Kolory tekstu (Foreground):**
 - Czerwony: `\033[31m`
 - Zielony: `\033[32m`
 - Resetuj kolor: `\033[0m`
- **Kolory tekstu (Background):**
 - Żółty: `\033[43m`
 - Niebieski: `\033[44m`
- **Style tekstu:**
 - Pogrubienie: `\033[1m`
 - Podkreślenie: `\033[4m`
- **Sterowanie kursorem i ekranem:**
 - Wyczyść ekran: `\033[2J`
 - Pozycja domowa (lewy górny róg): `\033[H`
 - Przesuń w górę o n: `\033[nA`
 - Przesuń w dół o n: `\033[nB`
- **Przykład w Pythonie/Bashu:** `print("\033[1;31mTo jest czerwony tekst\033[0m")`

Kolory w konsoli tekstowej

- Można zdefiniować zarówno kolor liter jak i tła w konsoli.
- Większość terminali pozwala na użycie 8 lub 16 kolorów. Współczesne terminale obsługują również kolory 256-bitowe (`\033[38;5;Vm`) oraz pełne RGB (`\033[38;2;R;G;Bm`).

| Nazwa koloru | Kod koloru tekstu | Kod koloru tła |
|--------------|-------------------|----------------|
| Czarny | 30 | 40 |
| Czerwony | 31 | 41 |
| Zielony | 32 | 42 |
| Żółty | 33 | 43 |
| Niebieski | 34 | 44 |
| Magenta | 35 | 45 |
| Turkusowy | 36 | 46 |
| Biały | 37 | 47 |
| Domyślny | 39 | 49 |

Przykład użycia kolorów w konsoli tekstowej

```
print ("\033[1;30mKolor tekstu czarny")
print ("\033[1;31mKolor tekstu czerwony")
print ("\033[1;32mKolor tekstu zielony")
print ("\033[1;33mKolor tekstu żółty")
print ("\033[1;34mKolor tekstu Niebieski")
print ("\033[1;35mKolor tekstu różowy")
print ("\033[1;36mKolor tekstu błękitny")
print ("\033[1;37mKolor tekstu biały")
print ("\033[1;39mKolor tekstu domyślny")
```

```
print ("\033[1;40mKolor tła czarny")
print ("\033[1;41mKolor tła czerwony")
print ("\033[1;42mKolor tła zielony")
print ("\033[1;43mKolor tła żółty")
print ("\033[1;44mKolor tła Niebieski")
print ("\033[1;45mKolor tła różowy")
print ("\033[1;46mKolor tła błękitny")
print ("\033[1;47mKolor tła biały")
print ("\033[1;49mKolor tła domyślny")
```

Przykład użycia kolorów w konsoli tekstowej

```
class Kolory:
    CZERWONY = '\033[31m'
    ZIELONY = '\033[32m'
    ZOLTY = '\033[33m'
    RESET = '\033[0m'

print(Kolory.CZERWONY + "Błąd!" +
Kolory.RESET)print(f"{Kolory.ZIELONY}Sukces!{Kolory.RESET}")
```

Styl liter

| Nazwa stylu | ESC Code Sequence | Reset Sequence |
|---------------------------|-------------------|----------------|
| Reset wszystkich ustawień | ESC[0m | |
| Pogrubiony | ESC[1m | ESC[22m |
| Tryb ciemniejszy | ESC[2m | ESC[22m |
| Kursywa | ESC[3m | ESC[23m |
| Podkreślenie | ESC[4m | ESC[24m |
| Mrugające litery | ESC[5m | ESC[25m |
| Odwrócenie z tłem | ESC[7m | ESC[27m |
| Tryb ukryty | ESC[8m | ESC[28m |
| Przekreślenie | ESC[9m | ESC[29m |

Przykład użycia stylów liter w konsoli tekstowej

```
print ("Krój normalny")
print ()
print ("\033[1mKrój pogrubiony")
print ("\033[22m")
print ("\033[2mTekst trochę ciemniejszy")
print ("\033[22m")
print ("\033[3mKursywa (tekst pochyły)")
print ("\033[23m")
print ("\033[4mTekst podkreślony")
print ("\033[24m")
print ("\033[5mTekst mrugający (?)")
print ("\033[25m")
print ("\033[7mZamiana tekstu z tłem kolorami")
print ("\033[27m")
print ("\033[8mTekst ukryty")
print ("\033[28m")
print ("\033[9mTekst przekreślony (?)")
print ("\033[29m")
print ("\033[0m")
```

Podstawowe kody ANSI escape:

- **Sterowanie kursorem i ekranem:**
 - Wyczyść ekran: `\033[2J`
 - Pozycja domowa (lewy górny róg): `\033[H`
 - Przesuń w górę o n: `\033[nA`
 - Przesuń w dół o n: `\033[nB`

Sterowanie kursorem

| Nazwa stylu | ESC Code Sequence |
|---|--|
| Kursor na pozycje początkową (0.0) | ESC[H |
| Kursor na pozycję: linia#, kolumna# | ESC[<code>{line};{column}</code> H ESC[<code>{line};{column}</code> f |
| Kursor do góry o # linii | ESC[#A |
| Kursor w dół o # linii | ESC[#B |
| Kursor w prawo o # kolumn | ESC[#C |
| Kursor w lewo o # kolumn | ESC[#D |
| Kursor w dół o # linii i na początek następnej linii | ESC[#E |
| Kursor do góry o # linii i na początek poprzedniej linii | ESC[#F |
| Kursor do # kolumny | ESC[#G |
| Podanie pozycji kursora -ESC(#C,#R) | ESC[6n |
| Kursor jedną linię wyżej, skroluje ekran w razie potrzeby | ESC M |
| Zapisuje pozycje kursora | ESC 7 |
| Przywraca ostatnio zapisane pozycję kursora | ESC 8 |
| Zapisuje pozycje kursora | ESC[s |
| Przywraca ostatnio zapisane pozycję kursora | ESC[u |

Przykład użycia sterowania kursorem w konsoli tekstowej

```
import time, os
print ("Start u góry")
time.sleep(1)
os.system("clear||cls")

print ("\033[8B8 linijek niżej")
time.sleep(2)

os.system("clear||cls")
print ("\033[2B2 linijki wyżej")
time.sleep(1)
os.system("clear||cls")

print ("pozycja kursora")
print ("\033[6n")
time.sleep(1)
os.system("clear||cls")

print ("\033[MHej")
time.sleep(2)
os.system("clear||cls")
```

Czyszczenie ekranu

| Nazwa operacji | ESC Code Sequence |
|---|--------------------------|
| Czyszczenie reszty ekranu | ESC[J |
| Czyszczenie od pozycji kursora do końca ekranu | ESC[0J |
| Czyszczenie od początku ekranu do pozycji kursora | ESC[1J |
| Czyszczenie całego ekranu | ESC[2J |
| Czyszczenie zapisanych linii | ESC[3J |
| Czyszczenie reszty linii | ESC[K |
| Czyszczenie linii od położenia kursora do końca | ESC[0K |
| Czyszczenie od początku linii do pozycji kursora | ESC[1K |
| Czyszczenie całej linii | ESC[2K |

Przykład użycia czyszczenia ekranu w konsoli tekstowej

```
print ("\033[1;30mKolor tekstu czarny")
print ("\033[1;31mKolor tekstu czerwony")
print ("\033[1;32mKolor tekstu zielony")
print ("\033[1;33mKolor tekstu żółty")
print ("\033[1;34mKolor tekstu Niebieski")
print ("\033[1J")
print ("\033[1;35mKolor tekstu różowy")
print ("\033[1;36mKolor tekstu błękitny")
print ("\033[1;37mKolor tekstu biały")
print ("\033[1;39mKolor tekstu domyślny")
```

Style programowania

- **Sequences**
- ESC - sequence starting with ESC (`\x1B`)
- CSI - Control Sequence Introducer: sequence starting with ESC [or CSI (`\x9B`)
- DCS - Device Control String: sequence starting with ESC P or DCS (`\x90`)
- OSC - Operating System Command: sequence starting with ESC] or OSC (`\x9D`)
- Any whitespaces between sequences and arguments should be ignored. They are present for improved readability.