

Python - Szyfr Cezara ver. 1

© Copyright by 3bird Projects 2024, <http://edukacja.3bird.pl>

Uwagi ogólne

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\Users\krzysiu> python C:\Users\krzysiu\Desktop\twójSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstęp. Kod należy przepisać ze zrozumieniem.

Nasz skrypt

Zadaniem programu jest zaszyfrowanie dowolnego tekstu w oparciu o przesunięcie liter o jedno miejsce w alfabecie. W przypadku litery „Z” następuje jej przesunięcie do litery „A”. Należy pamiętać, że Rzymianie nie używali małych liter, nie stosowali spacji i nie operowali liczbami jako osobną pulą cyfr. Zależy nam, aby zaszyfrowany tekst mógł być przedstawiony wyłącznie za pomocą liter.

Kod skryptu - wersja podstawowa

```
tekstDoZaszyfrowania = input("Wpisz tekst do zaszyfrowania: ")
```

```
zaszyfrowanyTekst = ''
```

```
for kazdaLitera in tekstDoZaszyfrowania:
```

```
    kazdaLitera = kazdaLitera.upper()
```

```
    kodASCII = ord(kazdaLitera) + 1
```

```
    if kodASCII > ord('Z'):
```

```
        kodASCII = ord('A')
```

```
    zaszyfrowanyTekst += chr(kodASCII)
```

```
print(zaszyfrowanyTekst)
```

```
input('\nNaciśnij ENTER, aby zakończyć...')
```

Kod skryptu - wersja rozszerzona

```
#!/usr/bin/env python
```

```
# Powyższa linia, jedynie dla systemu Linux.
```

```
from os import system # Wymagane do kolorowania składni w systemie Windows 10/11:
```

```
system("")
```

```
# Uwaga: Składnia nie będzie kolorowana, gdy uruchomimy kod w IDLE Shell (to nie jest
```

```
# prawdziwy terminal) oraz w Windows 7/8.
```

```
print('\n\033[1;34;40m===== SZYFR CEZARA =====\033[0m\n')
```

```
print('\033[1;30;40mTa wersja algorytmu szyfruje tekst\nprzesuwając każdą z liter do przodu\nno jedno miejsce w alfabecie.\033[0m\n')
```

```
# Inicjujemy zmienną:
```

```
zaszyfrowanyTekst = ''
```

```

tekstDoZaszyfrowania = input("Wpisz tekst do zaszyfrowania: ")
# Zamiana na duże litery (aby było łatwiej):
tekstDoZaszyfrowania = tekstDoZaszyfrowania.upper()

for kazdaLitera in tekstDoZaszyfrowania:
    # Zamieniamy każdą kolejną literę na kod ASCII:
    kodASCII = ord(kazdaLitera)
    # W przypadku, gdy znak jest z zakresu alfabetu ASCII:
    if ((kazdaLitera.isalpha()) and (kazdaLitera.isascii()):
        # Jeśli użytkownik wpisał literę Z...
        if (kodASCII == ord('Z')):
            # ... to zamieniamy ją na literę A:
            kodASCII = ord('A')
        # W przypadku pozostałych znaków alfabetycznych ASCII,
        # przesuwamy literę o jedno miejsce do przodu
        else:
            kodASCII += 1 # Równoważne: kodASCII = kodASCII + 1
        # Z powrotem zamieniamy kod ASCII na literę i każdą z nich
        # po kolei dołączamy do zmiennej "zaszyfrowanyTekst":
        zaszyfrowanyTekst += chr(kodASCII)

print('\nZaszyfrowany tekst: \033[1;33;40m' + zaszyfrowanyTekst + '\033[0m.\n')

# Teraz odszyfrowywanie:
odszyfrowanyTekst = ''
tekstDoOdszyfrowania = input("Wpisz tekst do odszyfrowania lub wciśnij ENTER,\naby
odszyfrować powyżej zaszyfrowany tekst: ")
if (tekstDoOdszyfrowania == ""):
    tekstDoOdszyfrowania = zaszyfrowanyTekst

for kazdaLitera in tekstDoOdszyfrowania:
    kodASCII = ord(kazdaLitera)
    if ((kazdaLitera.isalpha()) and (kazdaLitera.isascii()):
        # Jeśli mamy do czynienia z literą A...
        if (kodASCII == ord('A')):
            # ... to zamieniamy ją na literę Z:
            kodASCII = ord('Z')
        # Wszystkie inne litery z zakresu ASCII cofamy o jedną pozycję do tyłu:
        else:
            kodASCII -= 1
        # Z powrotem zamieniamy kod ASCII na literę i każdą z nich
        # dołączamy do zmiennej "odszyfrowanyTekst":
        odszyfrowanyTekst += chr(kodASCII)

print('\nOdszyfrowany tekst: \033[1;32;40m' + odszyfrowanyTekst + '\033[0m.\n')
input('\n\033[1;30;40mNaciśnij ENTER, aby zakończyć...\033[0m\n')

```