

Python - Silnia rekurencyjnie

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

Uwagi ogólne

Silnią liczby 5 jest pięć kolejno pomnożonych przez siebie liczb **naturalnych** zaczynając od 1, tj. $1*2*3*4*5 = 120$. Kilka pierwszych wyników silni:

```
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40 320
9! = 362 880
10! = 3 628 800
```

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:\> python naszSkrypt.py
```

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

Kod skryptu - wersja podstawowa

```
def silnia_rekurencyjna(podanaLiczba):
```

```
    if podanaLiczba > 1:
        return podanaLiczba * silnia_rekurencyjna(podanaLiczba-1)
    else:
        return 1
```

```
definicja = """\n\nDEFINICJA SILNI\n\n
```

```
    Silnią liczby 5 jest pięć kolejno pomnożonych przez siebie liczb naturalnych zaczynając od 1,  
    tj. 1*2*3*4*5 = 120."""
```

```
print(definicja)
```

```
podanaLiczba = int(input('\n\nPodaj liczbę naturalną: '))
```

```
print('\n\nWynik: ', podanaLiczba, '! = ', silnia_rekurencyjna(podanaLiczba), '\n\n', sep='')
```

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

Kod skryptu - wersja rozbudowana

```
#!/usr/bin/env python
# Powyższa linia, tylko dla systemu Linux.

# SILNIA REKURENCYJNIE (wolniejsza)
from os import system # Wymagane do kolorowania składni w systemie Windows 10/11:
system(" ")
# Uwaga: Tekst nie będzie kolorowany, gdy uruchomimy go w „IDLE Shell” (bo to nie jest
prawdziwy terminal) oraz w systemach Windows 7/8.

def silnia_rekurencyjna(podanaLiczba):
    if podanaLiczba > 1: # Warunek przerwania rekurencji
        # Po każdym przebiegu rekurencji zmniejszamy wartość mnożonych liczb o 1
        # aż instrukcja warunkowa przestanie być prawdą (podanaLiczba > 1),
        # czyli 5x4x3x2.
        return podanaLiczba * silnia_rekurencyjna(podanaLiczba-1) # 5 x 4 x 3 x 2
    else:
        # Albo od razu zwraca 1 (gdyż 1!=1 oraz 0!=1),
        # albo zwraca 1 jako ostatni argument mnożenia w rekurencji: 5 x 4 x 3 x 2 x 1
        return 1
```

definicja = ""\n\n\033[1;30;40mDEFINICJA SILNI\n\n

*Silnią liczby 5 jest pięć kolejno pomnożonych przez siebie liczb naturalnych zaczynając od 1, tj. $1*2*3*4*5 = 120$.*

Filozoficznie: Silnia oznacza ilość permutacji w zbiorze n-elementowym (na ile sposobów można przedstawić te elementy).

W przypadku 1! (zbiór z jednym elementem), ilość możliwych przedstawień (permutacji) byłaby równa jeden.

W przypadku 0! (zbiór pusty), ilość możliwych permutacji byłaby... zerowa (ewentualnie nie miałyby sensu) bądź (jak chce matematyka formalna na zasadzie umowy) byłaby równa także jeden (gdyż zbiór pusty można przedstawić na jeden możliwy sposób).

Można by to także wyobrazić sobie jako hasło do systemu składające się z zerowej ilości znaków (tzw. puste hasło). W tym przypadku możemy dostać się do systemu na jeden możliwy sposób: nie podając hasła (jedna możliwa permutacja).

Taka argumentacja wydaje się jednak naciągana. Formalnie można by w ten sposób udowodnić, że $0=1$, gdyż $0!=1$ i $1!=1$; skoro prawe strony równań są takie same (można je do siebie przyrównać: $1=1$), to lewe strony równania także można przyrównać: $0! = 1!$, co po uproszczeniu daje $0 = 1$.

Kontekst historyczny także wskazuje, że zero i liczby ujemne nie biorą udziału w silni. Pierwsza wzmianka o silni występuje w Talmudzie (II-V w. CE), następnie w Indiach (XII w. CE). W roku 1808 Christian Kramp wprowadził oznaczenie $n!$. To wszystko przed powstaniem

matematyki formalnej. \033[0m\n""

print(definicja)

```
podanaLiczba = -1    # Podajemy tutaj jakąkolwiek wartość wstępną, która uruchomi poniższą
                    # pętlę WHILE.
while not podanaLiczba >= 0:
    try:
        podanaLiczba = int(input("\n\nPodaj liczbę naturalną: "))
        if podanaLiczba >= 1:
            # Poniżej, to jest jeden długi wiersz:
            print('\n\nWynik: \033[1;32;40m', podanaLiczba, '! = ',
                  silnia_rekurencyjna(podanaLiczba), '\033[0m\n\n', sep='')

            elif podanaLiczba == 0:
                print('\n\n\033[1;36;40mMatematyka formalna przyjmuje na zasadzie umowy,
że 0! = 1.\nZ filozoficznego punktu widzenia, nie ma to jednak sensu.\033[0m\n\n')

            else:
                print('\n\n\033[1;37;41m BŁĄD: \033[0m')
                print('\033[1;31;40mMusisz podać liczbę naturalną (czyli liczbę całkowitą
dodatnią, większą od zera)!\033[0m')

        except ValueError:
            print('\n\n\033[1;37;41m BŁĄD: \033[0m')
            print('\033[1;31;40m1. Wprowadzone wartości albo nie są liczbami naturalnymi...\n
2. Albo w ogóle nie wprowadzono żadnych wartości.\n\nSpróbuj jeszcze raz.\033[0m\n')

input('\n\n\033[1;30;40mNaciśnij ENTER, aby zakończyć...\033[0m\n')
```

Ostatnia aktualizacja: 6 grudnia 2022.