

# Linux i Python w Elektronicznej Sieci – ćwiczenia #02: Wprowadzenie do programowania w Pythonie

Projekt „Matematyka dla Ciekawych Świata”,  
Robert Ryszard Paciorek  
<rrp@opcode.eu.org>

2023-07-03

**Uwaga (1):** Informacje teoretyczne przydatne do wykonania zadań oraz linki do prezentacji wideo znajdziesz w [skrypcie wykładowym](#). Gorąco zachęcamy do korzystania z tych materiałów w trakcie rozwiązywania zadań.

**Uwaga (2):** Po skończeniu zadania prosimy pokazać rozwiązanie prowadzącemu - może mieć on jakiś wartościowy komentarz dotyczący tego rozwiązania.

## Zadanie 1.0.1

Napisz funkcję, która przyjmuje dwa argumenty i zwróci ich iloczyn (wynik mnożenia). Użyj jej do obliczenia (i wypisania z użyciem print) wartości kilku iloczynów sum.

## Zadanie 2.0.1

Napisz funkcję, przyjmującą dwa argumenty  $a$  i  $b$ , która obliczy i zwróci sumę liczb całkowitych większych od  $a$  i mniejszych od  $b$ .

## Zadanie 2.0.2

Rozwiąż zadanie 2.0.1 używając pętli `while`

## Zadanie 2.0.3

Napisz funkcję, która przyjmuje argument będący napisem i wypisuje go powtarzając każdą małą literę dwukrotnie. Np. dla 'Ala ma Kota' funkcja powinna wypisać:

```
Allaa mmaa Koottaa
```

## Zadanie 2.0.4

Napisz program dekodujący napis kodowany w UTF8 zakodowany przy pomocy base64 mający postać: `b'UH10aG9uIGplc3QgZmFqbmk8J+Yjg==\n'`.

Wskazówka: dane wejściowe funkcji `decode()` muszą być typu "bytes", można to uzyskać poprzedzając napis prefiksem `b`, tak jak powyżej.

## Zadanie 3.0.1

Napisz funkcję która sprawdzi z użyciem wyrażeń regularnych czy dany napis kończy się xyz.

## Zadanie 3.0.2

Napisz program przyjmujący dwa argumenty z linii poleceń – napis i liczbę. Program ma wypisać napis (pierwszy argument) tyle razy ile wynosi podana liczba (drugi argument). Program powinien wypisać komunikat dotyczący użycia gdy wywołany jest z niedostateczną ilością argumentów.

### Zadanie 3.0.3

Napisz funkcję, która sprawdzi czy podana liczba ma ustawiony bit numer 2 (ma on wartość 1).

*Wskazówka: pamiętaj że bity w liczbie numerujemy od zera.*

### Zadanie 3.0.4

Napisz funkcję, która zwróci podana liczbę z ustawionym bitem o podanym numerze. Na przykład: `ustaw_bit(13, 1)` powinno zwrócić 15.

## Zadania dodatkowe

### Zadanie 4.0.1

Korzystając z metod typu napisowego napisz funkcję `szukaj(napis, szukany)`, która wypisze wszystkie pozycje na których w napisie `napis`, znajduje się podnapis przekazany w argumencie `szukany`. Jeżeli ciąg `szukany` występuje kilka razy pod rząd powinny być wypisane pozycje wszystkich nie nakładających się wystąpień. Przykład użycia:

```
>>> szukaj("abbccbbbbebbffbbgg", "bb")
1
5
9
13
>>> szukaj("abbccbbbbebbffbbgg", "bb")
1
5
7
10
14
```

### Zadanie 4.0.2

Używając dwóch pętli `for`, jedna wewnątrz drugiej, napisz program, który wypisze na ekranie *trójkąt z iksów*, taki jak poniżej:

```
X
XX
XXX
XXXX
XXXXX
XXXXXX
XXXXXXX
```

### Zadanie 4.0.3

Zmodyfikuj rozwiązanie zadania 4.0.2 tak aby korzystało tylko z jednej pętli.

### Zadanie 4.0.4

Zmodyfikuj rozwiązanie zadania 4.0.2 tak aby zamiast co najmniej jednej pętli zastosować rekurencję. *Wskazówka: Funkcja rekurencyjna to funkcja, która wywołuje samą siebie (typowo ze zmodyfikowanymi argumentami), dopóki zachodzi jakiś ustalony warunek (typowo zależny od argumentów).*

#### Zadanie 4.0.5

Napisz funkcję `spr`, korzystając z dopasowywania wyrażeń regularnych, która sprawdzi czy podany napis jest palindromem postaci `aa...aabb...bbaa...aa`. Czyli czy składa się z ciągu liter `a` o dowolnej długości, następnie ciągu liter `b` i ponownie ciągu liter `a` o takiej samej długości jak ten pierwszy. Przykłady użycia:

```
>>> spr("aaabba")
NIE
>>> spr("aaabbbaaa")
TAK
>>> spr("aaabaaa")
TAK
```

#### Zadanie 4.0.6

Napisz funkcję, przyjmującą dwa argumenty `a` i `b`, która będzie realizować funkcję xor (zwróci wartość `a XOR b`).

*Wskazówka: dla ułatwienia można przyjąć że argumenty są zawsze typu logicznego `True/False`.*

#### Zadanie 4.0.7

Napisz funkcję która wypisze liczby od 0 do 20 z pominięciem liczb podzielnych przez wartość określoną w jej argumencie.

#### Zadanie 4.0.8

Napisz funkcję `toStr(liczba, podstawa)`, która konwertuje podaną liczbę do reprezentacji napisowej w systemie o podanej podstawie.

*Wskazówka: do testowania poprawności działania możesz użyć funkcji `int(napis, podstawa)`, możemy przyjąć że podstawa jest mniejsza od 37 tak aby starczyło liter alfabetu łacińskiego.*