



WPROWADZENIE

Wyprowadzanie danych: Wyprowadzanie na ekran komunikatów i wyników umożliwia instrukcja **wyjścia** – funkcja `print()`. Argumentami funkcji (podanymi w nawiasach) mogą być teksty, wyrażenia arytmetyczne lub zmienne oddzielone przecinkami.

Na przykład:

```
print("Informatyka")
print("suma liczb wynosi:", suma)
```

Deklarowanie zmiennych:

W języku Python nie deklarujemy zmiennych.

Wprowadzanie danych z klawiatury

Zmiennej stosowanej w programie możemy nadać wartość za pomocą instrukcji przypisania, wprowadzając wartość z klawiatury po uruchomieniu programu. W tym celu stosujemy **instrukcję wejścia** – funkcję `input()`.

Na przykład:

```
a = int(input("Wprowadź liczbę: "))
```


– jeśli wpisujemy z klawiatury liczbę 346, to w zmiennej `a` zostanie zapamiętana liczba 346.

Instrukcja warunkowa `if`:

```
if warunek:
    instrukcja1
else: instrukcja2

lub

if warunek:
    instrukcja1

if a>0:
    print("liczba dodatnia")
else:
    print("liczba niedodatnia")
```

W języku Python blok kodu wyróżnia się poprzez wcięcia na przynajmniej jedną spację. Na ogół przyjęte wcięcia na cztery spacje.

Instrukcja iteracyjna `for`:

```
for zmienna to lista wartości:
    lista instrukcji
```

Zastosowanie instrukcji `for` do wprowadzenia n liczb z klawiatury:

```
for i in range(n):
    print("Podaj liczbę:")
    a = int(input("Wprowadź liczbę: "))
```

Funkcje:

W języku Python wszystkie podprogramy nazywane są **funkcjami**. Dzielimy je na dwie grupy:

- funkcje niezwracające wartości (odpowiednik procedury w języku Pascal):

```
def nazwa_funkcji(lista_parametrów):
    lista_instrukcji
```

- funkcje zwracające wartość (odpowiednik funkcji w języku Pascal):

```
def nazwa_funkcji(lista_parametrów):
    lista_instrukcji
    return wartość
```

Listy:

Aby utworzyć zmienne indeksowane w języku Python, możemy zdefiniować specjalną strukturę danych – **listę**.

```
nazwa_listy = [element1, element2, ..., elementn]
```

Do elementów listy odwołujemy się, podając nazwę listy i indeks elementu umieszczony w nawiasach kwadratowych, np. `a[0]`, `a[1]`, ..., `a[n - 1]` – dla listy n -elementowej o nazwie `a`. W języku Python pierwszy indeks jest zawsze równy 0.

Na przykład:

```
moja_lista = [0] * 10
```

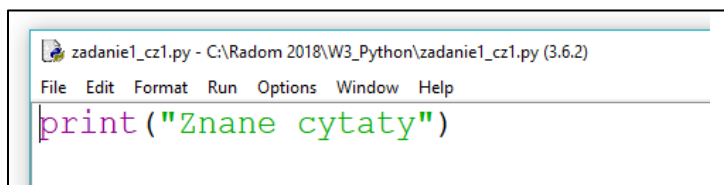
– oznacza zdefiniowanie listy o nazwie `moja_lista` składającej się z dziesięciu elementów, z których każdy jest równy zero (wyzerowanie wszystkich elementów listy).

ZADANIA

Uwaga: Utwórz na pulpicie swój folder do zapisywania rozwiązań poniższych zadań.

Zadanie 1. Wyprowadzanie napisów

1. Napisz program wyprowadzający na ekran napis: "Znane cytaty". Zapisz program w pliku pod nazwą *Zadanie1_cz1* (rys. 1.). Uruchom program.



```
zadanie1_cz1.py - C:\Radom 2018\W3_Python\zadanie1_cz1.py (3.6.2)
File Edit Format Run Options Window Help
print("Znane cytaty")
```

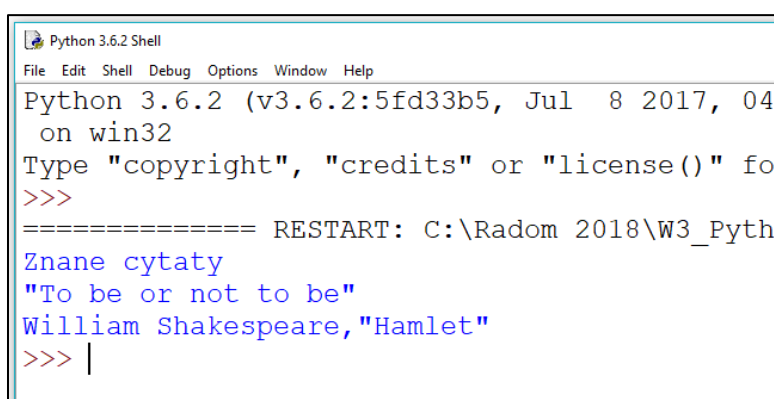
Rys. 1. Kod źródłowy programu *Zadanie1_cz1*

2. Uzupełnij program: w drugim wierszu wyświetlaj cytat "To be, or not to be ", a w trzecim imię i nazwisko autora cytatu "William Shakespeare" i po przecinku tytuł sztuki "Hamlet". Zapisz program w pliku pod nazwą *Zadanie1_cz2*. Uruchom program. Dodatkowo zadbaj, aby cytat i tytuł utworu były wyświetlone w cudzysłowach (rys. 2.).

Wskazówki: Po wykonaniu funkcji `print()` kursor przechodzi do nowego wiersza.

Aby tekst został wyświetlony w cudzysłowach należy dodać wyświetlanie cudzysłowu: `"\""`

```
print("William Shakespeare," "\"\" \"Hamlet\" \"\"")
```



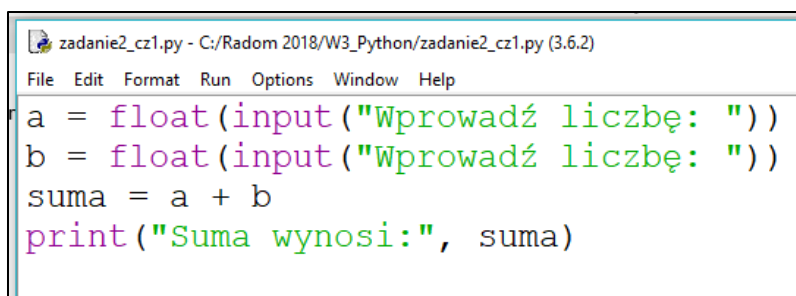
```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04
on win32
Type "copyright", "credits" or "license()" fo
>>>
===== RESTART: C:\Radom 2018\W3_Pytho
Znane cytaty
'To be or not to be'
William Shakespeare, 'Hamlet'
>>> |
```

Rys. 2. Wynik działania programu *Zadanie1_cz2*

Zadanie 2. (zad. 1., str. 125) Wprowadzanie i wyprowadzanie danych, wykonywanie obliczeń – obliczanie sumy i średniej arytmetycznej

1. Napisz program umożliwiający wprowadzenie dwóch liczb rzeczywistych *a* i *b*, obliczenie ich sumy (*suma*) i wyprowadzenie wyniku (*suma*) na ekran monitora. Uruchom program. Zapisz program w pliku pod nazwą *Zadanie2_cz1* (rys. 3.).
2. Uzupełnij program o obliczenie średniej arytmetycznej liczb *a* i *b*. Wyświetlaj też napis "Średnia wynosi: ". Zapisz program pod nazwą w pliku pod nazwą *Zadanie2_cz2*.

Wskazówki: W instrukcji przypisania używamy znaku „=”.



```
zadanie2_cz1.py - C:/Radom 2018/W3_Python/zadanie2_cz1.py (3.6.2)
File Edit Format Run Options Window Help
a = float(input("Wprowadź liczbę: "))
b = float(input("Wprowadź liczbę: "))
suma = a + b
print("Suma wynosi:", suma)
```

Rys. 3. Kod źródłowy programu Zadanie2_cz1

Zadanie 3. (zad. 4., str. 125) Stosowanie instrukcji warunkowej – algorytm sprawdzania, która liczba jest większa

Napisz program sprawdzający, która z dwóch różnych liczb całkowitych x i y wprowadzanych z klawiatury jest większa. Wyświetl na ekranie wartość większej liczby (rys. 4.). Zapisz program w pliku pod nazwą *Zadanie 3*.

```
if x > y:
    print("Większa jest liczba pierwsza", x)
else:
    print("Większa jest liczba druga", y)
```

Rys. 4. Stosowanie instrukcji warunkowej – fragment kodu źródłowego programu Zadanie3

Zadanie 4. Stosowanie instrukcji iteracyjnej for – przykład algorytmu iteracyjnego

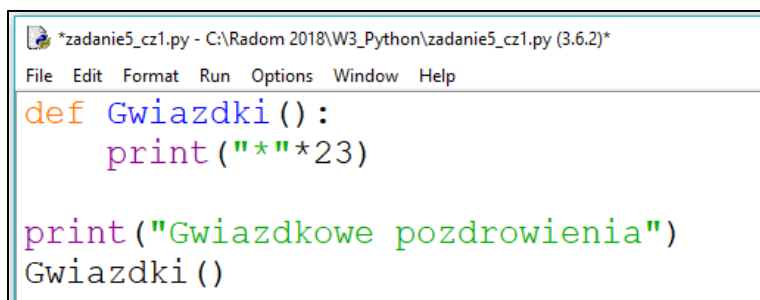
Zmodyfikuj program z zadania 3. tak, aby sprawdzał n par liczb wprowadzanych z klawiatury (rys. 5.). Wartość n wprowadzaj z klawiatury. Dodaj przed wprowadzeniem n napis "Podaj liczbę sprawdzanych par liczb". Zapisz program w pliku pod nazwą *Zadanie4*.

```
for i in range(n):
    x = int(input("Wprowadź pierwszą liczbę: "))
    y = int(input("Wprowadź drugą liczbę: "))
    if x > y:
        print("Większa jest liczba pierwsza", x)
    else:
        print("Większa jest liczba druga", y)
```

Rys. 5. Stosowanie instrukcji iteracyjnej `for` – fragment kodu źródłowego programu Zadanie4

Zadanie 5.* (zad. 2., str. 141) Stosowanie funkcji niezwracającej wartości, bez parametrów

1. Napisz program umożliwiający wyprowadzenie na ekran monitora napisu „Gwiazdkowe pozdrowienia”, a pod nim dwudziestu trzech gwiazdek (znaków *). Zdefiniuj funkcję *Gwiazdki* jako niezwracającą wartości i bez parametrów. Zadaniem funkcji jest wyprowadzenie w jednym wierszu dwudziestu trzech gwiazdek. Funkcję wywołaj w programie głównym (rys. 6.). Zapisz program w pliku pod nazwą *Zadanie5_cz1*.



```
*zadanie5_cz1.py - C:\Radom 2018\W3_Python\zadanie5_cz1.py (3.6.2)*
File Edit Format Run Options Window Help
def Gwiazdki():
    print("*"*23)

print("Gwiazdkowe pozdrowienia")
Gwiazdki()
```

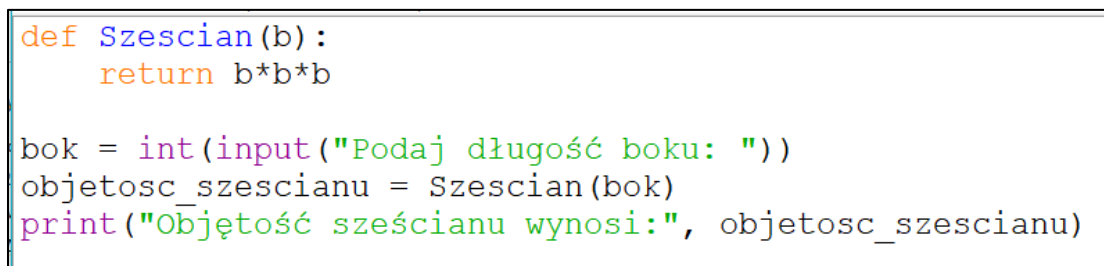
Rys. 6. Definicja funkcji *Gwiazdki* i wywołanie jej w programie głównym – fragment kodu programu *Zadanie5_cz1*

2. Uzupełnij program, wyświetlając w kolejnym wierszu napis „z konferencji w Radomiu”, pod którym również wyświetlaj 23 gwiazdki. Zapisz program w pliku pod nazwą *Zadanie5_cz2*.

Zadanie 6.* (zad. 4., str. 141) Stosowanie funkcji zwracającej wartość, z jednym parametrem

Napisz program umożliwiający obliczenie objętości sześcianu. Zdefiniuj funkcję *Szescian* z jednym parametrem typu całkowitego o nazwie *b*, obliczającą sześcian liczby *b* i zwracającą do programu głównego wynik obliczenia. Wywołaj funkcję w programie głównym z parametrem aktualnym *bok*. Wartość parametru wprowadzaj z klawiatury (rys. 7.). Zapisz program w pliku pod nazwą *Zadanie6*.

Uwaga: Rozwiązanie pokazane na rysunku 7. można zmodyfikować, rezygnując ze zmiennej *objetosc_szescianu*. Spróbuj poprawić program.



```
def Szescian(b):
    return b*b*b

bok = int(input("Podaj długość boku: "))
objetosc_szescianu = Szescian(bok)
print("Objętość sześcianu wynosi:", objetosc_szescianu)
```

Rys. 7. Definicja funkcji *Szescian* i wywołanie jej w programie głównym – fragment kodu programu *Zadanie6*

Zadanie 7.* (ćw. 5. i 6., str. 133-134) Stosowanie listy do wprowadzania i wyprowadzania danych

1. Napisz funkcję o nazwie *WprowadzDane* wprowadzającą dane do tablicy *a*, składającej się z dziesięciu liczb całkowitych. Zapisz program w pliku pod nazwą *Zadanie7*. Skompiluj i uruchom program.
2. Zdefiniuj funkcję o nazwie *WyprowadzDane* wyprowadzającą w kolejnych wierszach dane z tablicy na ekran. W funkcji głównej `main()` po poleceniu wywołania funkcji *WprowadzDane* dodaj wywołanie funkcji *WyprowadzDane*. Zapisz program w pliku pod tą samą nazwą. Skompiluj i uruchom program.

```
N = 10
a = [0] * N

def WprowadzDane():
    for i in range(N):
        a[i] = int(input("Podaj liczbę: "))

def WyprowadzDane():
    for i in range(N):
        print("a[" + str(i) + "] = " + str(a[i]))

WprowadzDane()
WyprowadzDane()
```

Rys. 8. Stosowanie tablicy do wprowadzania i wyprowadzania danych – fragment kodu programu Zadanie7

3. Zmodyfikuj program *Zadanie7* tak, aby dane wyświetlały się w odwrotnym porządku. Zapisz program pod nazwą *Zadanie7_odwrotnie*.

Wskazówka: Należy zastosować funkcję `range()` z trzema parametrami:

```
range(początek, koniec, krok)
```

Na przykład:

```
for i in range(1, 20, 2):
    print(i)
```

– instrukcja `print(i)` zostanie wykonana dziesięć razy. Funkcja `range()` wygeneruje kolejne liczby całkowite z przedziału \langle *początek*, *koniec* \rangle , zmieniające się o *krok*, czyli zmienna *i* będzie przyjmować kolejno wartości: 1, 3, 5, ..., 19. Trzeci argument (*krok*) określa tym samym, o jaką wartość zmienia się zmienna *i*.

Środowisko programowania dostępne bezpłatnie w Internecie:

Python 3 (np. wersja 3.6) <https://www.python.org/downloads/>

Kompilator Dev-C++ <https://sourceforge.net/projects/orwelldevcpp/>

Podręczniki zawierające tematy z programowania w języku C++ i Python

- G. Koba, *Teraz bajty. Informatyka dla szkoły podstawowej. Klasa VIII*
- G. Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych – zakres podstawowy,*
- G. Koba, *Teraz bajty. Informatyka dla szkół ponadpodstawowych – zakres rozszerzony (w przygotowaniu)*

Podręczniki zawierające tematy z programowania w języku C++

- G. Koba, *Z nowym bitem. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy,*

Poradniki metodyczne:

- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy,*
- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres rozszerzony.*

Materiały metodyczne do pozostałych podręczników dostępne poprzez wyszukiwarkę:

<https://www.migra.pl/wyszukiwarka>