Katedra Informatyki i Automatyki

Programowanie w języku Python

Autorzy:
dr inż. Tomasz Krzeszowski, prof. PRz

Politechnika Rzeszowska 2023 | www.kia.prz.edu.pl

1. Środowisko programistyczne i pierwszy program

1. Uruchomić system operacyjny wskazany przez prowadzącego.
2. Uruchomić zintegrowane środowisko programistyczne Spyder (ikonka z czerwoną pajęczyną i literką S w prawym dolnym rogu).
3. Utworzyć nowy plik (skrypt) i zapisać go po nazwą „program1.py”
4. Wpisać w skrypcie wywołanie funkcji print z parametrem „Hello world” (print(”Hello world”) i uruchomić skrypt (zielony trójkąt w IDE).
5. Uruchomić terminal (znajduje się pod ikonką IDE Spyder), po uruchomieniu wpisać polecenie „ls” i sprawdzić czy wśród wyświetlonych plików znajduje się plik „program1.py”.
6. Uruchomić skrypt poleceniem „python3 program1.py”

2. Komentarze

```
#Komentarz zaczyna się od znaku #

"""
Komentarz
na kilka linii
"""

'''
Komentarz
na kilka linii
'''
```

3. Zmienne

```
x = 4     # x jest typu int (całkowitego)
x = "Sally" # x jest teraz łańcuchem znaków
print(x)

x = str(3)    # x będzie zawierać '3'
y = int(3)    # y będzie zawierać 3
z = float(3)  # z będzie zawierać 3.0
```

4. Funkcje we/wy

print(arg, arg, …, arg) – wypisuje argumenty na konsolę

input() – pozwala na wpisanie wartości z wejścia (Uwaga! Funkcja input() zwraca łańcuch znaków, więc jeśli chcemy wczytać liczbę musimy dokonać konwersji do odpowiedniego formatu, np. a = int(input()))

input(arg) – wypisuje argument i pozwala na wpisanie wartości z wejścia

Zadania do wykonania:

1. Napisz program, który pobiera dwie wartości całkowite z klawiatury i zapisuje je w zmiennych ‘a’ i ‘b’. Następnie należy zapisać wynik sumowania wartości zmiennych ‘a’ i ‘b’ w zmiennej ‘c’ i wypisać następujący komunikat „Suma a i b = c” (w miejscach znaków ‘a’, ‘b’ i ‘c’ powinny się znaleźć odpowiednie wartości liczbowe).

5. Instrukcja warunkowa if

if ***warunek1***:
    wyrażene1
elif ***warunek2***:
    wyrażenie2
else:
    wyrażenie3

Przykład:

```
wiek = int(input('Podaj swój wiek '))
if wiek >= 18:
    print("Jesteś pełnoletni. ")
else:
    print("Brakuje Ci jeszcze ", 18-wiek, " do pełnoletności.")
```

Zadania do wykonania:

1. Napisz program wczytujący z klawiatury liczbę całkowitą i wypisujący informację o tym, czy wczytana liczba jest parzysta, czy nieparzysta. Wskazówka: należy wykorzystać operator modulo (%), który zwraca resztę z dzielenia.
2. Napisz program wczytujący z klawiatury długość trzech odcinków (wartości typu float), sprawdzający i wypisujący, czy z odcinków można zbudować trójkąt.

6. Pętle

- Pętla while

while ***warunek***:
    wyrażenia

Przykład:

```
i  = 0
while i < 10:
    print(f"Number is {i}!")
    i = i + 1
```

- Pętla for

for ***nazwaZmiennej*** in ***kolekcjaWartości***:
    wyrażenia

Przykład:

```
for i in range(10):
    print(f"Number is {i}!")
```

Zadania do wykonania:

1. Napisać program wczytujący n liczb (n należy wpisać z klawiatury) i wypisujący ich średnią arytmetyczną. Przygotować wersję z pętlą while i for.
2. Napisać program, który wczytuje liczby całkowite do czasu wpisania liczby 0. Wśród wczytanych liczb program powinien znaleźć liczbę najmniejszą i największą.
3. Napisz program, który wyświetli wszystkie liczby pierwsze z podanego przedziału.

7. Łańcuchy znaków

```
‘Łańcuch znaków’, "Łańcuch znaków", "Tytuł książki to 'Pan
Tadeusz'.", 'Temat wykładu to "Programowanie w języku Python".'

’’’Długi
łańcuch
znaków’’’

"hello"+"world"            # łączenie, wynik - "helloworld"
"hello"*3                  # powtarzanie, wynik -  "hellohellohello"
"hello"[0]                 # indeksowanie, wynik - "h"
"hello"[-1]                     # indeksowanie od końca, wynik - "o"
len("hello")                    # rozmiar, wynik - 5
"hello"[1:4]                    # wycinanie, wynik - "ell"
"hello"[:4]      # pobranie wszystkich elementów do danego indeksu
                 # wynik - "hell"
"hello"[2:]        # pobranie wszystkich elementów od danego
                   # indeksu, wynik - "llo"
"hello" < "jello"               # porównywanie, wynik - true
"e" in "hello"                  # wyszukiwanie, wynik -  true
len("hello")                    # długość łańcucha, wynik – 5
```

Zadania do wykonania:

1. Mając łańcuch znaków zawierający kombinację małych i wielkich liter napisz program, który ułoży znaki w łańcuchu tak, aby wszystkie małe litery były na początku.
   Przykład: str1 = "PythonJestSuper"
   Na wyjściu: str2 = "ythonestuperPJS"
   Wskazówki:
   - wykorzystać pętlę for aby pobierać kolejne elementy łańcucha
   - wykorzystać funkcję islower() (lub isupper()) aby sprawdzić czy dany znak jest małą literą czy dużą
   - wykorzystać funkcję join() aby połączyć listę znaków w łańcuch znaków.
2. Napisać program, który policzy wszystkie litery, cyfry i symbole specjalne w podanym łańcuchu znaków. Wskazówka: wykorzystać funkcję isalpha() żeby sprawdzić czy znak jest literą oraz isdigit() aby sprawdzić czy jest cyfrą.
3. Mając łańcuch s1, napisz program, który zwróci sumę i średnią cyfr występujących w łańcuchu, ignorując wszystkie inne znaki. Wskazówka: wykorzystaj funkcję isdigit().

8. Typy złożone

- Listy

```
l1 = [1, 3, 6]
l2 = ["abc", "de"]
l3 = [100, "hello", ["abc", "de",3]]
l1[1] = 5 #zmiana wartości drugiego elementu w liście (listy
          #są indeksowane od indeksu 0 [1, 5, 6]
print(l1[2]) #wypisanie 3 elementu
```

- Krotki

```
t1 = (0,1,2,3)
t2 = ('aaa','bbb')
t3 = (100, "hello", ("abc", "de",3))
t4 = ()
       # odwołania do poszczególnych elementów wykonuje się
       # analogicznie jak w przypadku list
```

- Słowniki

```
kontakty = {"Jacek": 606555333, "Henek": 505333444, "Adam":
605777444, "Paweł": 804444333}
```

- Zbiory

```
owoce = {'pomarańcza','wiśnia','truskawka'}
s = set('12345abc')   #{1,2,3,4,5,a,b,c}
```

Zadania do wykonania:

1. Napisz program zamieniający każdy element listy liczb na kwadrat tej liczby.
2. Sprawdzić dla następujących zbiorów set1 = {10, 20, 30, 40, 50}, set2 = {30, 40, 50, 60, 70} działanie następujących funkcji: intersection(), union(), difference(), symmetric_difference(). Co robią poszczególne funkcje? Sprawdzić działanie operatorów &, |, -, ^.

9. Funkcje

Składnia:

```
def func(args):
    return values
```

Przykład 1

```
def add(a, b):
    return a + b

res = calc(10, 30)
print(res)
```

Przykład 2

```
def calc(a, b):
    add = a + b
    sub = a - b

    return add, sub

# pobierz wynik w formie krotki
res = calc(10, 30)
print(res)
```

Zadania do wykonania:

1. Napisać funkcję, która przyjmuje 1 parametr będący liczbą całkowitą i zwraca wartość bezwzględną tej liczby.
2. Napisać funkcję, która przyjmuje 3 parametry opisujące funkcję kwadratową (a, b i c) oraz oblicza i wypisuje miejsca zerowe tej funkcji.
3. Napisać funkcje wykonujące podstawowe operacje arytmetyczne dla dwóch parametrów (dodawanie, odejmowanie, mnożenie, dzielenie), a następnie stworzyć prosty kalkulator. Program powinien pytać użytkownika jaką operację chce wykonać i zależnie od tego co wpisze użytkownik wywoływać odpowiednią funkcję.
4. Napisz funkcję, która przyjmuje dwa parametry ‘n’ i ‘z’, a następnie rysuje na konsoli choinkę zbudowaną ze znaków ‘z’ o wielkości ‘n’.

10. Literatura

1. https://www.python.org/ (strona umożliwiająca pobranie Pythona, dokumentacja)
2. https://pynative.com/python/ (nauka Pythona)
3. https://www.w3schools.com/python/ (nauka Pythona)
4. https://winpython.github.io/ (dystrybucja Pythona – Python + dodatkowe programy)