

Лабораторна робота № 6

Тема: Функції в Python.

Мета: Ознайомитися з призначенням, правилами створення функцій та їх видами. Отримати практичні навички роботи з функціями в Python.

Васильєв О.М. Програмування мовою Python – Тернопіль: Навчальна книга – Богдан, 2019. – 504с. стор. 139-198

Порядок виконання завдань.

Завдання 1.

Розробіть функції для здійснення наступних операцій зі списками:

- 1 Швидке сортування;
- 2 Пошук елемента за значенням;
- 3 Пошук перших п'яти мінімальних елементів;
- 4 Пошук середнього арифметичного;
- 5 Повернення списку, що сформований з початкового списку, але не містить повторів.

Завдання 2.

Розробити функцію яка приймає послідовність чисел $a_1, a_2, \dots, a_n$. Визначити в якій половині послідовності додатних елементів більше. Якщо n не парне то кількість елементів у першій половині заокруглити до більшого числа, другої половини до меншого.

Завдання 3.

Розробіть функцію яка на основі введеного значення n обчислює тригонометричний вираз за наступним правилом:

- $\sin(x) \cdot \cos(x)$, якщо $n \equiv 1$;
- $\sin(x^2) + \cos(x)$, якщо $n \equiv 2$;
- $1 - \sin(x)$, в противному випадку.

Методичні рекомендації.

Інструкція `def` створює об'єкт функції та зв'язує його з ім'ям. Інструкція `def` складається з рядка заголовка і слідуючого за ним блоку інструкцій, зазвичай з відступами (або простої інструкції слідом за двокрапкою).

Блок інструкцій утворює тіло функції, тобто програмний код, який виконується інтерпретатором щоразу, коли здійснюється виклик функції. У рядку заголовка інструкції `def` визначаються ім'я функції, з яким буде пов'язаний

об'єкт функції, і список з нуля або більше аргументів (іноді їх називають параметрами) в круглих дужках.

Імена аргументів в рядку заголовка будуть пов'язані з об'єктами, переданими в функцію, в точці виклику. Тіло функції часто містить інструкцію return:

```
def (arg1, arg2,... argN):  
    ... return
```

Інструкцію return можна розташувати в будь-якому місці в тілі функції – вона завершує роботу функції і передає результат програмі, яка її викликає. Інструкція return містить об'єктний вираз, який надає результат функції. Інструкція return є необов'язковою – якщо вона відсутня, робота функції завершується, коли потік управління досягає кінця тіла функції. Функція без інструкції return повертає об'єкт None, проте це значення зазвичай ігнорується.

Приклад 1.

```
def function(c, d):  
    # a, b – глобальні змінні; c, d -- локальні global a, b  
    # зміна значення глобальної змінної  
    a = 5  
    b = 7  
    # зміна значення локальної змінної  
    c = 10  
    d = 12  
a, b, c, d = 1, 2, 3, 4    # множинне присвоєння  
print(a, b, c, d)    # 1 2 3 4  
function(c, d)  
print(a, b, c, d)    # 5 7 3 4  
Результат  
1 2 3 4  
5 7 3 4
```

Приклад 2.

Функція, яка реверсує рядок

```
def Reverse(s):  
    s2 = ""  
    i = len(s)-1  
    while i>=0:  
        s2 = s2 + s[i]  
        i = i-1  
    return s2
```

Виклик функції Reverse()

```
s1 = "abcdef"
s2 = Reverse(s1) # s2 = 'fedcba'
print("s2 = ", s2)
```

Результат роботи програми

```
s2 = fedcba
```

Контрольні запитання.

1. Як створити функцію у мові Python?
2. Що таке анонімні функції та інструкція `lambda`?
3. Який синтаксис оголошення функції?
4. Що таке позиційні та непозиційні аргументи функції?
5. Як створити функцію зі значеннями за замовчуванням?
6. Який порядок передачі аргументів у функцію, якщо вона містить позиційні та непозиційні аргументи та аргументи із значеннями за замовчуванням?
7. Для чого використовується інструкція `return`? Чи обов'язково вона присутня у функції?
8. Що повертає функція, якщо в її тілі відсутня інструкція `return`?