

Лабораторна робота № 1

Тема: Написання простих програм мовою Python.

Мета: Отримати практичні навички написання лінійних програм на мові Python. Відпрацювати функції вводу\виводу, арифметичних операцій, використання основних типів даних.

Література:

Васильєв О.М. Програмування мовою Python – Тернопіль: Навчальна книга – Богдан, 2019. – 504с. стор. 39-89

Зміст роботи:

Завдання 1.

Виведіть на екран свої групу, прізвище та ім'я.

Створіть змінну a1 в яку запишіть ціле число, a2-дійсне число.

Створіть список, кортеж, рядок, множину, словник із 7 елементів.

Використовуючи функцію type() отримайте на екрані повідомлення про тип створених змінних.

Завдання 2.

Використовуючи функцію input() введіть значення двох змінних і виконайте над ними всі можливі арифметичні операції. Результати запишіть у список та виведіть на екран. Створіть новий список в який запишіть другий і п'ятий елементи раніше створеного списку.

Завдання 3.

Ввести чотири цілих числа. Перших два – це довжина і ширина першого прямокутника, а других два – це довжина і ширина другого прямокутника. Якої довжини потрібно мати дріт, щоб виготовити такі прямокутники?

Завдання 4.

Напишіть програму для рішення задачі розрахунку оплати чеку в кафе. N друзів отримали в кафе рахунок на суму m гривень. Було прийнято рішення розділити оплату порівну. До оплати додали суму чайових яка склала 15 % від суми рахунку. Необхідно визначити суму оплати кожного з друзів. (Значення N і m введіть з клавіатури)

Завдання 5.

Напишіть програму для знаходження суми цифр введеного трьохзначного числа. Виведіть число в якому поміняйте місцями сотні і десятки; одиниці і сотні.

Завдання 6.

Створіть список використовуючи функцію **range()** в діапазоні від 0 до 56 з кроком 2. Отриманий список виведіть на екран. Елементи створеного списку збільшити в 2 рази. Зайдіть суму, мінімальний та максимальний елементи списку. Результат оформіть у вигляді словника.

Створений список перетворіть в кортеж та рядок, використовуючи команду type() отримайте підтвердження правильності виконання команд.

Методичні вказівки.

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження на офіційному сайті Python (Python.org).



Мова Python відноситься до мов з неявною строгою динамічною типізацією. Неявна типізація означає, що при оголошенні змінної її тип не вказується (при явній типізації тип змінної вказується обов'язково).

Для мов з динамічною типізацією тип змінної визначається безпосередньо при виконанні програми.

Для мов програмування з динамічною типізацією характерно:

- будь-яка змінна є посиланням;
- типом змінної є те, на що вона посилається;
- тип змінної може довільно змінюватися по ходу виконання коду, коли змінна починає посилатися на інший об'єкт.

В Python існують декілька числових типів:

int – ціле число;

float – число з плаваючою точкою (дійсне число);

complex – комплексне число.

Враховуючи неявну типізацію мови Python при оголошенні змінної, їй повинно бути надане значення (вона має бути ініціалізована). Щоб оголосити та ініціалізувати змінну необхідно: написати її ім'я, потім поставити оператор присвоєння (знак дорівнює) і вказати значення, з яким дана змінна буде створена.

Приклад:

```
a = 10
```

```
b=7.25
```

Для визначення типу призначена функція **type()**.

Приклад:

```
a = 10
```

```
print(type(a))
```

Для вводу даних визначеного використовується функція **input()**

Приклад:

```
a= int(input())
```

```
d= int(input("Задати число"))
```

Арифметичні операції

Знак	Операція
+	Додавання
-	Віднімання
*	Множення
/	Ділення
//	Цілочисельне ділення
%	Остача від ділення
**	Піднесення до степеня

Для звернення до елементів списку треба використовувати індекси, які представляють номер елемента в списку. Індеси починаються з нуля. Тобто другий елемент буде мати індекс 1. Для звернення до елементів з кінця можна використовувати негативні індекси, починаючи з -1. Тобто у останнього елемента буде індекс -1, у передостаннього - -2 і так далі.

```
numbers = [1, 2, 3, 4, 5]
print (numbers [0]) # 1
print (numbers [2]) # 3
print (numbers [-3]) # 3
numbers [0] = 125 # змінюємо перший елемент списку
print (numbers [0]) # 125
```

Якщо необхідно створити список, в якому повторюється одне і те ж значення кілька разів, то можна використовувати символ зірочки *. Наприклад, визначимо список з шести п'ятірок:

```
numbers = [5] * 6 # [5, 5, 5, 5, 5, 5]
print (numbers)
```

Щоб додати елемент застосовуються методи

append () – додає елемент в кінець списку.

http://pishachok.blogspot.com/p/blog-page_37.html методи та функції роботи зі списками.

Контрольні запитання.

1. Які типи даних ви знаєте? Опишіть їх.
2. Чи можна перетворити дробове число на ціле? ціле на дробове? У яких випадках можна рядок перетворити на число?
3. Наведіть приклади операцій. Для чого призначена операція присвоєння?
4. Які існують правила і рекомендації для іменування змінних?
5. Як список перетворити в рядок, кортеж, множину і навпаки?
6. Назвіть основні методи та функції роботи зі списками.