

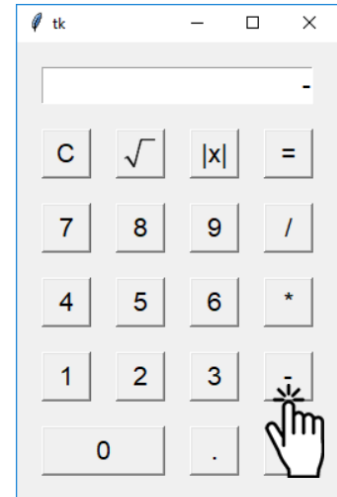
Урок 30. Проект “Калькулятор”. Етап 4. Користувацькі функції

Вивчення нового матеріалу

Слайд № 1

Щоб перевірити, як обчислюється квадратний корінь та модуль числа, нам знадобилося вводити від’ємні числа. Як ви бачили, знак “-” можна було вводити тільки з клавіатури.

Запрограмуємо можливість введення від’ємних чисел з використанням кнопки “-” нашого калькулятора.



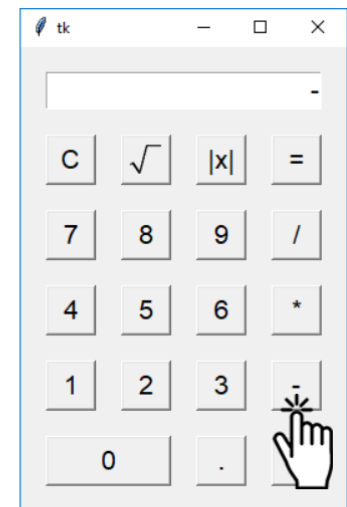
Будемо вважати, що під час введення кожного числа кнопку “-” користувач натискає лише один раз.

Вправа

Вправа № 1



Вправа 1. Змініть код обробника події натискання кнопки “-” так, щоб з її допомогою можна було вводити від’ємні числа.



Змініть код обробника події натискання кнопки “-”.

Вправа № 2



Вправа 2. Змініть код виведення результату у функції `Equal_click`.

```
ent.insert(END, a)
```

```
if a==int(a):  
    ent.insert(END,int(a))  
else:  
    ent.insert(END,a)
```



Змініть код виведення результату для кожної арифметичної операції.

Вивчення нового матеріалу

Слайд № 2

Ми вставляли той самий код у програму кілька разів, і тому програма стала доволі громіздкою. Щоб спростити програму, краще створити **користувацьку функцію**, що буде містити код виведення числа. Тоді в коді самої програми ми будемо лише **викликати** цю функцію.

Загальний вигляд опису функції є таким:

```
def ім'я функції(параметри функції):  
    тіло функції
```

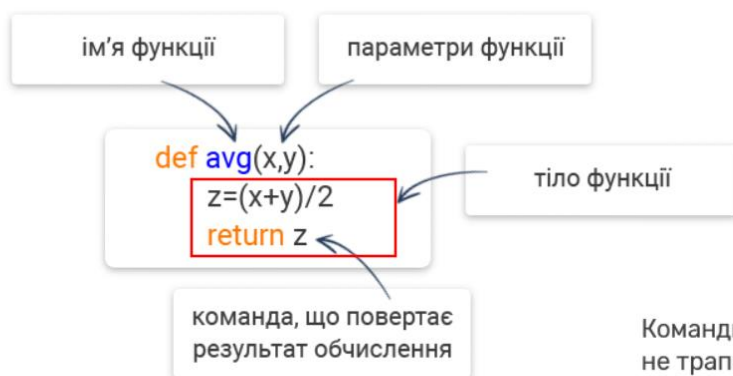
Параметри функції — список змінних, від яких залежить описаний у тілі функції процес.



Запам'ятайте загальний синтаксис опису функції.

Слайд № 3

Приклад функції знаходження середнього арифметичного:



Команди функції виконуються, доки не трапиться команда **return**. Після слова **return** вказують значення, яке функція **повертає**.



Зауважте, що всі команди тіла функції мають бути зсунуті вправо відносно її заголовку.

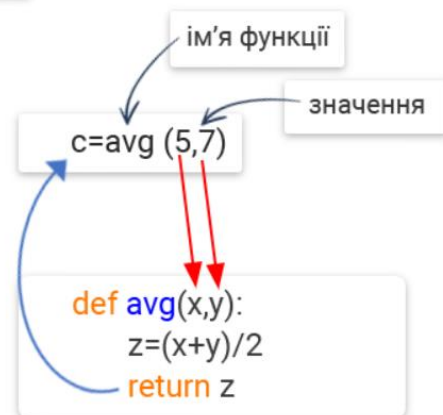
Слайд № 4

Однак якщо просто створити функцію, вона не буде виконуватися. Її потрібно ще **викликати**, тобто вказати в програмі місце, де виконуються команди функції. Для виклику функції використовують такий код:

ім'я функції (список значень)

Під час виконання команди виклику **значення** передаються відповідним **параметрам** функції, після чого виконуються її команди.

У результаті виконання цього коду змінній **c** буде присвоєно значення **6**.



Отже, замість усіх команд тіла функції ми записуємо лише її ім'я і в дужках — значення параметрів, тобто **викликаємо** функцію.

Слайд № 5

На рисунку вказано послідовність виконання команд у програмі з використанням функції.

```
4 def avg(x, y):  
5     return (x+y) / 2  
1 a=int(input("a="))  
2 b=int(input("b="))  
3 c=avg(a, b)  
6 print("c=", c)
```



Зауважте: функція описується вище від основної програми, однак першою починає виконуватися саме програма.

Вправа

Вправа № 3



Вправа 3.

1. Уведіть код функції **result**, що виводить результат обчислень.
2. Замініть у функції **Equal_click** фрагменти коду, що виводять результати виконання **vcix** операцій, викликами функції **result(a)**.

```
def result(a):  
    if a==int(a):  
        ent.insert(END,int(a))  
    else:  
        ent.insert(END,a)
```

```
def Equal_click():  
    global a,b  
    c=float(ent.get())  
    ent.delete(0,END)  
    if b=="+":  
        a=a+c
```

```
        if a==int(a):  
            ent.insert(END,int(a))  
        else:  
            ent.insert(END,a)
```

result(a)



Зауважте, що код функції **result** необхідно розташувати **вище** за обробники подій натискання кнопок.

Вправа № 4



Вправа 4. Замініть команду на виклик функції **result(c)** в обробниках подій натискання кнопок $\sqrt{\quad}$ і $|x|$.

```
#Radical
def Radical_click():
    global c
    a=float(ent.get())
    ent.delete(0,END)
    if a>=0:
        c=sqrt(a)
        ent.insert(END,c)
    else:
        ent.insert(0,"помилка")
```

функція **result(c)**
замінює цю команду



У функціях обчислення кореня та модуля виклик функції **result** виглядатиме однаково.

Вивчення нового матеріалу

Слайд № 6

Зауважте, що якщо змінна-параметр та змінна в основній програмі мають однакові імена, це все одно **різні змінні**.

```
def f(a):
    a=a+3
    return a

a=5
c=f(a)
print("a=", a, "c=", c)
```

одна змінна **a**

друга змінна **a**



Ця програма виведе **a=5 c=8**, оскільки на змінну **a**, якій присвоїли 5, змінна **a**, що є параметром функції, ніяк не вплине.