Олексій Васильєв

Мова програмування Python

Київ 2020

Лекція 2.
Керуючі інструкції

- Умовний оператор if
- Оператор циклу while
- Оператор циклу for
- Тернарний оператор
- Основи обробки помилок

Умовний оператор if

Синтаксис:

```
if умова:
    # команди
else:
    # команди
```

Блок **else** не є обов'язковим

Умова - вираз логічного типу (тип **bool**). Можливі значення **True** (правда) та **False** (неправда)

```
if умова:
    # команди
elif умова:
    # команди
elif умова:
    # команди
elif умова:
    # команди
...
elif умова:
    # команди
else:
    # команди
```

Умовний оператор

Програма (UsingIf.py):

```
# Зчитуємо число:
num=int(input("Уведіть число: "))
# Умовний оператор:
if num%2==0:
    print("Це парне число")
else:
    print("Це непарне число")
# Після умовного оператора:
print("Дякую за співпрацю!")
```

Результат виконання:

```
Уведіть число: 123
Це непарне число
Дякую за співпрацю!
```

```
Уведіть число: 124
Це парне число
Дякую за співпрацю!
```

Умовний оператор

При формуванні умови можна:

- Використовувати оператори порівняння
  == (дорівнює), != (не дорівнює), < (менше),
  <= (менше або дорівнює), > (більше),
  >= (більше або дорівнює)

- Використовувати логічні оператори
  and (логічне і), or (логічне або), not (заперечення)

- Використовувати оператори приналежності
  in (належить), not in (не належить)

Правила приведення до логічного типу:

- Ненульові значення - True, нульові значення - False
- Пусті послідовності - False, непусті - True
- Об'єкти - відповідає метод __bool__()

Умовний оператор - 2

Програма (UsingIf_2.py):

```
# Зчитуємо текст:
txt=input("Уведіть текст: ")
# Символ:
symb='ч'
# Умовний оператор:
if symb in txt:
    print("У тексті є буква",symb)
else:
    print("У тексті немає букви",symb)
```

Результат виконання:

```
Уведіть текст: Вивчаємо Python
У тексті є буква ч
```

```
Уведіть текст: Всім привіт
У тексті немає букви ч
```

Умовний оператор - 3

Програма (UsingIf_3.py):

```
# Уводяться параметри рівняння:
a,b=eval(input("Уведіть (через кому) два числа: "))
# Якщо параметри некоректні:
if (type(a)!=int and type(a)!=float) or (type(b)!=int and type(b)!=float):
    print("Уведені некоректні значення!")
    # Завершення виконання програми:
    raise SystemExit(0)
# Якщо перший параметр ненульовий:
elif a!=0:
    txt="Розв'язок x="+str(b/a)
# Якщо другий параметр ненульовий (при нульовому першому):
elif b!=0:
    txt="Розв'язків немає!"
# Якщо обидва параметри нульові:
else:
    txt="Розв'язок - довільне число!"
# Вигляд рівняння:
print("Рівняння "+str(a)+"x="+str(b))
# Результат пошуку кореня:
print(txt)
print("Пошук розв'язку завершено.")
```

Розв'язок рівняння ax=b

Умовний оператор - 3

Результат виконання:

```
Уведіть (через кому) два числа: 2.5,3.6
Рівняння 2.5x=3.6
Розв'язок x=1.44
Пошук розв'язку завершено.
```

```
Уведіть (через кому) два числа: 2,3
Рівняння 2x=3
Розв'язок x=1.5
Пошук розв'язку завершено.
```

```
Уведіть (через кому) два числа: 0,3.5
Рівняння 0x=3.5
Розв'язків немає!
Пошук розв'язку завершено.
```

```
Уведіть (через кому) два числа: 0,0
Рівняння 0x=0
Розв'язок - довільне число!
Пошук розв'язку завершено.
```

```
Уведіть (через кому) два числа: "2",3
Уведені некоректні значення!
```

Завдання - 1

Напишіть програму, котра визначає, чи ділиться уведене користувачем число на 7

Завдання - 2

Напишіть програму, котра визначає, чи ділиться уведене користувачем число хоча би на одне з чисел 5 або 11

Завдання - 3

Напишіть програму, котра визначає, чи міститься цифра 3 у введеному користувачем числі. Для перетворення чисел в текст використати функцію str()

Оператор циклу while

Синтаксис:

```
while умова:
    # команди
else:
    # команди
```

Блок **else** не є обов'язковим

```
while умова:
    # команди
```

Інструкція **break** завершує виконання оператору циклу (в цьому випадку код в блоці **else** не виконується)

Інструкція **continue** завершує виконання поточної ітерації

Сума чисел

Програма (UsingWhile.py):

```
# Зчитування верхньої границі суми:
n=int(input("Укажіть верхню границю суми: "))
# Початкове значення для суми:
s=0
# Початкове значення індексної змінної:
k=0
# Оператор циклу для розрахунку суми:
while k<n:
    # Збільшення значення індексної змінної на одиницю:
    k=k+1
    # Додавання доданку до суми:
    s=s+k
# Відображення результату:
print("Сума чисел від 1 до",n,"дорівнює",s)
```

Результат виконання:

```
Укажіть верхню границю суми: 100
Сума чисел від 1 до 100 дорівнює 5050
```

Сума чисел - 2

Програма (UsingWhile_2.py):

```
# Зчитування кількості доданків:
n=input("Укажіть кількість доданків: ")
# Змінна з текстовим значення:
txt="1"
# Індексна змінна:
k=1
# Оператор циклу для формування тексту з
# виразом для суми натуральних чисел:
while str(k)!=n:
    # Збільшення значення індексної змінної:
    k=k+1
    # Додавання фрагменту до тексту з виразом
    # для суми натуральних чисел:
    txt=txt+"+"+str(k)
# Відображення результату:
print(txt,"=",eval(txt))
```

Результат виконання:

```
Укажіть кількість доданків: 10
1+2+3+4+5+6+7+8+9+10 = 55
```

Склад числа

Програма (UsingWhile_3.py):

```
# Уводиться число:
number=int(input("Уведіть число: "))
# Поки число більше нуля:
while number>0:
    # Остання цифра у числі:
    digit=number%10
    # Відображення цифри:
    print("|"+str(digit),end="")
    # Відкидається остання цифра у числі:
    number=number//10
# Відображається останній роздільник:
print("|")
```

Результат виконання:

```
Уведіть число: 31049265
|5|6|2|9|4|0|1|3|
```

Прості числа

Програма (UsingWhile_4.py):

```
# Уводиться число:
number=int(input("Уведіть число: "))
# Верхня границя для дільника:
num=number//2
# Початкове значення дільника:
k=2
# Пошук дільників числа:
while k<=num:
    # Якщо число ділиться на k:
    if number%k==0:
        print("Число не є простим")
        # Завершення оператора циклу:
        break
    # Збільшується значення дільника:
    k=k+1
# Блок виконується, якщо не виконана інструкція break:
else:
    print("Це просте число")
# Повідомлення відображається завжди:
print("Перевірка завершена")
```

```
Уведіть число: 97
Це просте число
Перевірка завершена
```

```
Уведіть число: 95
Число не є простим
Перевірка завершена
```

Завдання - 4

Напишіть програму, котра розраховує суму квадратів непарних натуральних чисел

Завдання - 5

Напишіть програму, котра розраховує суму цифр в числі, уведеному користувачем

Завдання - 6

Напишіть програму, котра розраховує всі дільники уведеного користувачем числа

Оператор циклу for

Синтаксис:

```
for змінна in колекція:
    # команди
else:
    # команди
```

Блок **else** не є обов'язковим

```
for змінна in колекція:
    # команди
```

Інструкція **break** завершує виконання оператора циклу (в цьому випадку код в блоці **else** не виконується)

Інструкція **continue** завершує виконання поточної ітерації

Колекція: список (наприклад, [1,3,5]), текст (наприклад, "Python"), кортеж (наприклад, (1,3,5)), множина (наприклад, {1,3,5}) та ін.

Сума чисел

Програма (UsingFor.py):

```
print("Сума натуральних чисел")
n=100 # Кількість доданків
# Формуємо текст для відображення результату:
text="1+2+...+"+str(n)+" ="
# Змінна для запису суми:
s=0
# Оператор циклу для розрахунку суми:
for i in range(1,n+1):
    # Додаємо доданок до суми:
    s=s+i
# Відображаємо результат:
print(text,s)
```

Результат виконання:

```
Сума натуральних чисел
1+2+...+100 = 5050
```

Числа Фібоначчі

Програма (UsingFor_2.py):

```
# Кількість чисел в послідовності:
n=15
# Перші два числа:
a,b=1,1
# Відображення перших двох чисел:
print(a,b,end=" ")
# За кожен цикл розраховується одне нове число:
for k in range(n-2):
    # Розрахунок нового числа в послідовності:
    a,b=b,a+b
    # Відображення нового числа:
    print(b,end=" ")
```

Результат виконання:

```
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610
```

Пошук букв в тексті

Програма (UsingFor_3.py):

```
# Текст для пошуку букв:
mytext=input("Уведіть текст для перевірки: ")
# Букви для пошуку:
symbs=['а','у','я']
print("Шукаємо такі букви:",symbs)
# Пошук букв:
for s in symbs:
    # Якщо букву знайдено:
    if s in mytext:
        print("В тексті є буква '"+s+"'")
        # Завершення оператору циклу:
        break
    # Якщо букви немає:
    else:
        print("В тексті немає букви '"+s+"'")
# Блок else оператору циклу:
else:
    print("Таких букв в тексті немає")
# Останнє повідомлення програми:
print("Пошук завершено")
```

```
Уведіть текст для перевірки: текст
Шукаємо такі букви: ['а', 'у', 'я']
В тексті немає букви 'а'
В тексті немає букви 'у'
В тексті немає букви 'я'
Таких букв в тексті немає
Пошук завершено
```

```
Уведіть текст для перевірки: рядок
Шукаємо такі букви: ['а', 'у', 'я']
В тексті немає букви 'а'
В тексті немає букви 'у'
В тексті є буква 'я'
Пошук завершено
```

Завдання - 7

Напишіть програму, котра для уведеного числа визначає, скільки в цьому числі цифр 0, 1, 2 і так далі, до 9

Завдання - 8

Напишіть програму, в котрій користувач уводить ціле число, а програма кожну цифру в цьому числі заміняє на "доповнення до 9": цифра 0 заміняється на 9, цифра 1 заміняється на 8, цифра 2 заміняється на 7, і так далі — цифра 8 заміняється на 1, а цифра 9 заміняється на 0

Тернарний оператор

Синтаксис:

```
значення if умова else значення
```

Программа (OddEven.py):

```
# Уводиться число:
num=int(input("Уведіть ціле число: "))
# Використання тернарного оператора:
res="парне" if num%2==0 else "непарне"
# Повідомлення:
print("Це "+res+" число")
```

Результат виконання:

```
Уведіть ціле число: 123
Це непарне число
```

```
Уведіть ціле число: 124
Це парне число
```

Обробка помилок

Синтаксис:

```
try:
    # команди
except:
    # команди
```

- Якщо в try-блоці помилок немає, except-блок ігнорується
- Якщо в try-блоці виникає помилка, починає виконуватися except-блок

Программа (TryExcept.py):

```
print("Обробка помилок")
# Контрольований код:
try:
    num=int(input("Уведіть ціле число: "))
    print("Ви ввели число "+str(num))
# Обробка помилки:
except:
    print("Треба було ввести ціле число!")
print("Дякую за співпрацю!")
```

```
Обробка помилок
Уведіть ціле число: 123
Ви ввели число 123
Дякую за співпрацю!
```

```
Обробка помилок
Уведіть ціле число: 123.0
Треба було ввести ціле число!
Дякую за співпрацю!
```

Обробка помилок - 2

Синтаксис:

```
try:
    # команди
except Тип_помилки:
    # команди
except Тип_помилки:
    # команди
...
except Тип_помилки:
    # команди
```

Для помилок різних типів (класів) можна використовувати різні except-блоки

ValueError - помилка при перетворенні значень (наприклад, тексту в число)

ZeroDivisionError - помилка ділення на нуль

TypeError - помилка типу

IndexError - помилка індексу

SyntaxError - синтаксична помилка

NameError - помилка доступу до змінної (немає змінної з таким іменем)

Обробка помилок - 2

Програма (TryExcept_2.py):

```
print("Операції зі списком чисел...")
# Контрольований код:
try:
    nums=eval(input("Уведіть числовий список: "))
    print("Отримано значення: "+str(nums))
    a=int(nums[0])
    b=int(nums[3])
    print(str(a)+"/"+str(b)+"="+str(a/b))
# Обробка помилок:
except ValueError:
    print("ValueError: помилка при перетворенні!")
except ZeroDivisionError:
    print("ZeroDivisionError: спроба ділення на нуль!")
except TypeError:
    print("TypeError: неприпустима операція!")
except IndexError:
    print("IndexError: неправильний індекс елементу!")
except SyntaxError:
    print("SyntaxError: неможливо розрахувати вираз!")
except NameError:
    print("NameError: неправильний ідентифікатор!")
print("Завершення програми.")
```

```
Операції зі списком чисел...
Уведіть числовий список: [2,4,6,5,1]
Отримано значення: [2, 4, 6, 5, 1]
2/5=0.4
Завершення програми.
```

Обробка помилок - 2

Результат виконання:

```
Операції зі списком чисел...
Уведіть числовий список: [2,4,6,5,1]
Отримано значення: [2, 4, 6, 5, 1]
2/5=0.4
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: "Python"
Отримано значення: Python
ValueError: помилка при перетворенні!
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: 12345
Отримано значення: 12345
TypeError: неприпустима операція!
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: [1,2,3]
Отримано значення: [1, 2, 3]
IndexError: неправильний індекс елементу!
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: Python
NameError: неправильний ідентифікатор!
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: Hello World!
SyntaxError: неможливо розрахувати вираз!
Завершення програми.
```

```
Операції зі списком чисел...
Уведіть числовий список: [1,2,3,0,5]
Отримано значення: [1, 2, 3, 0, 5]
ZeroDivisionError: спроба ділення на нуль!
Завершення програми.
```

Завдання - 9

Напишіть програму, котра визначає, яке число (додатне чи від'ємне) увів користувач. Використати тернарний оператор

Завдання - 10

Напишіть програму, в котрій користувач уводить два дійсних числа, а програма перевіряє, яке з чисел більше. Використайте тернарний оператор і обробку помилкових ситуацій

Домашнє завдання

[1] Напишіть програму, в котрій користувач уводить список цілочислових значень, а також верхню границю для розрахунку суми. Програма розраховує суму натуральних чисел, але за виключенням тих, які входять в список. Так, якщо користувач увів список [2,5,6] і 10 як верхню границю суми, то розраховується сума чисел від 1 до 10, але без урахування чисел 2, 5 та 6

[2] Напишіть програму, в якій користувач уводить три числа, а програма визначає, чи може існувати трикутник зі сторонами, довжина яких визначається уведеними значеннями. Умова існування трикутника така: сума двох довільних (з трьох уведених) чисел має бути більше третього числа