

Прізвище: Йовбак  
Ім'я: Андріанна  
Група: КНСП-11  
Варіант: 2  
Дата захисту: 15.10.2020



Кафедра: САПР  
Дисципліна: Методи та засоби комп'ютерного навчання  
Перевірів: Андрущак Н.А.

## ЗВІТ

до лабораторної роботи №1  
на тему "Вступ в Python"

**Мета роботи:** розглянути можливості для ініціалізації змінних, виконання арифметичних операцій, ввід та вивід результатів, написання та використання функції, а також використання циклів в програмі.

### Відповіді на контрольні запитання:

**1. Як відбувається ініціалізації змінних в Python?**

Ініціалізація змінних в Python відбувається дуже просто: потрібно всього лиш ввести ім'я змінної, потім знак присвоєння (рівності) "=" та значення чи вираз, якому ми хочемо, щоб наша змінна була рівною. Якщо ж назва змінної складається з двох чи більше слів, прийнято використовувати символ нижнього підкреслення "\_" для зв'язки слів в назві. Наприклад, `student = "Andrianna"`, `current_year = 2020`. Також існують певні обмеження до назви змінної, а саме: назва змінної повинна починатись з нижнього підкреслення чи букви, після чого можуть йти цифри, букви чи символи підкреслення. Важливо зазначити те, що Python чутлива до регістру мова програмування, тому змінні `Person` і `person` - є двома різними змінними та займають окремі місця в пам'яті.

**2. Що таке оператори присвоювання?** Оператори присвоєння – це інструкції, які присвоюють значення змінним. Вони мають одну змінну та знак рівності, який слідує за нею. Після знаку рівності записуються значення, які ми хочемо присвоїти змінній. Це можуть бути якісь константні дані або вирази, які містять змінні і/або значення даних. Оператор присвоєння позначається "=". Також існує комбіноване присвоєння. В цьому випадку використовуються, наприклад, такі оператори `+=`, `-=`, `*=`, `/=` та `in`.

**3. Які способи вводу та виводу даних в Python ви знаєте?** Для вводу та виводу даних в Python використовуються функції `input()` та `print()` відповідно. Функція `input()` може приймати один параметр – стрічку, що є повідомленням, яке користувач бачить, перед вводом даних. Функція `print()` виводить дані, які були передані в якості аргументів. Також є опціональні аргументи. Наприклад, `sep="separator"` – символ-розділювач, при переданні багатьох об'єктів (за замовчуванням, символ " "). Аргумент `end="end"` визначає, як саме буде закінчуватись рядок (за замовчуванням "\n").

**4. Які оператори циклу в Python вам відомі?** В Python існують такі оператори циклу як *for* і *while*. Тіло циклу – це інструкції, які повторюються в циклі. *While* виконує тіло циклу до тих пір, поки умова циклу істинна. Цикл *for* – це не цикл з лічильником, яким є *for* в багатьох інших мовах. В мові програмування Python цикл *for* призначений для перебору елементів структур даних і деяких інших об'єктів.

**5. Як працює оператор умови if-else?** Оператор умови *if* перевіряє задану умову і якщо вона є правдивою (*true*), то виконує інструкції, записані нижче оператора *if*. Якщо ж, наприклад,

умова, задана для оператора if, є не вірною, то в блоці else ми можемо записати інструкції, які повинні будуть виконатись в такому випадку. Наприклад:

```
if person_age >= 18:
    print("Людина – повнолітня»)
else:
    print("Людина – не повнолітня»)
```

6. **Що таке масив? Способи ініціалізації масивів.** У Python масив реалізований у вигляді списку даних. Значення масиву (списку) вказуються всередині квадратних дужок, де перераховуються через кому. Елемент списку елемент можна викликати за індексом і привласнити йому нове значення. Щоб дізнатися довжину масиву можна скористатись функцією len().
7. **Використання функцій в Python.** Функції представляють блок коду, який виконує певне завдання і який можна повторно використовувати в інших частинах програми. Щоб оголосити функції в Python, потрібно написати ключове слово def, далі назву функції, набір дужок з параметрами, двокрапку, після чого слідуватиме тіло функції. Функції можуть приймати як кілька аргументів, так і жодного.
8. **Для чого потрібний оператор повернення? Приклад використання.** Функції в Python можуть повертати результат. Для цього в функції використовується оператор return, після якого вказується значення, яке ми хочемо повернути. Приклад використання оператора повернення: нехай в нас є функція, яка приймає значення градусів за Цельсієм, а нам потрібно перевести їх до градусів за Фаренгейтом. В такому випадку можна написати дану функцію, яка буде повертати результат обчислень:

```
def convert_c_to_f(degree):
    return (degree * 1.8) + 32
```

## Індивідуальне завдання:

### Варіант 2.

#### Завдання 1.

1. Використовуючи мову програмування Python, написати програму для виведення значення температури в Фаренгейтах, якщо значення температури в Цельсіях змінюється від 20°C до 34°C, з кроком 2°C.
2. Вивести значення непарних чисел в діапазоні від 1 до 21. В кінці програми вивести добуток значень цих непарних чисел.

#### Завдання 2.

1. Змініть програму так, щоб вона вивела 5 календарів, запитуючи користувача значення n і d кожного разу перед виводом.
2. Написати програму виводу на екран кожного парного числа, що лежить в діапазоні від -3 до -25.

#### Завдання 3.

1. Написати програму, яка би знаходила індекс максимального елемента в масиві з 12 елементів.
2. Реалізуйте метод сортування Шелла для масиву 50 випадкових чисел.

**Код програми:**

```
import random
```

```
# TASK 1
```

```
print("<--- TASK 1 --->")
```

```
def convert_c_to_f(degree):
```

```
    return (degree * 1.8) + 32
```

```
for i in range(20, 36, 2):
```

```
    print(i, "°C ---> ", "%.f" % convert_c_to_f(i), "°F")
```

```
print("\nOdd numbers:")
```

```
product = 1
```

```
for i in range(1, 21+1):
```

```
    if i % 2 != 0:
```

```
        print(i, end=" ")
```

```
        product *= i
```

```
print("\nProduct:", product)
```

```
# TASK 2
```

```
print("\n\n<--- TASK 2 --->")
```

```
def print_calendar(days_amount, start_day):
```

```
    for j in range(start_day):
```

```
        print(" ", end=" ")
```

```
    for i in range(1, days_amount+1):
```

```
        if i < 10:
```

```

        print("", i, end=" ")
    else:
        print(i, end=" ")
    if(i+start_day) % 7 == 0:
        print("")
    print("")

```

```

for i in range(5):
    n = int(input("Input the number of days in the month (28-31): "))
    d = int(input("Input the starting day (0=Sun, 1=Mon,...):"))
    print_calendar(n, d)

```

```

print("\nEven numbers [-3;-25]:")
i = -3
while i >= -25:
    if i % 2 == 0:
        print(i, end=" ")
    i -= 1

```

```

# TASK 3
print("\n\n<--- TASK 3 --->")

```

```

def create_random_list(size):
    arr = list(range(size))
    random.shuffle(arr)
    return arr

```

```

def get_max_value_index(values_list):
    max = values_list[0]

```

```

max_index = 0
for i in range(len(values_list)):
    if max < values_list[i]:
        max = values_list[i]
        max_index = i
return max_index

```

```

def shell_sort(values_list):
    n = len(values_list)
    step = round(n / 2)
    while step > 0:
        for i in range(step, n):
            temp = values_list[i]
            j = i
            while j >= step and values_list[j - step] > temp:
                values_list[j] = values_list[j - step]
                j -= step
            values_list[j] = temp
        step = round(step / 2)
    return values_list

```

```

numbers_1 = create_random_list(12)
print("List of numbers: ", end="")
print(numbers_1)
print("Index of max number: ", get_max_value_index(numbers_1))

```

```

numbers_2 = create_random_list(50)
print("\nList of numbers: ", numbers_2)
print("Sorted list of numbers: ", end="")
print(shell_sort(numbers_2))

```

## Результати виконання програми:

```
<--- TASK 1 --->
```

```
20 °C ---> 68 °F
```

```
22 °C ---> 72 °F
```

```
24 °C ---> 75 °F
```

```
26 °C ---> 79 °F
```

```
28 °C ---> 82 °F
```

```
30 °C ---> 86 °F
```

```
32 °C ---> 90 °F
```

```
34 °C ---> 93 °F
```

```
Odd numbers:
```

```
1 3 5 7 9 11 13 15 17 19 21
```

```
Product: 13749310575
```

```
<--- TASK 2 --->
```

```
Input the number of days in the month (28-31): 28
```

```
Input the starting day (0=Sun, 1=Mon,...): 4
```

```
1 2 3
```

```
4 5 6 7 8 9 10
```

```
11 12 13 14 15 16 17
```

```
18 19 20 21 22 23 24
```

```
25 26 27 28
```

```
Input the number of days in the month (28-31): 30
```

```
Input the starting day (0=Sun, 1=Mon,...): 1
```

```
1 2 3 4 5 6
```

```
7 8 9 10 11 12 13
```

```
14 15 16 17 18 19 20
```

```
21 22 23 24 25 26 27
```

```
28 29 30
```

```
Input the number of days in the month (28-31): 29
```

```
Input the starting day (0=Sun, 1=Mon,...): 5
```

```
1 2
```

```
3 4 5 6 7 8 9
```

```
10 11 12 13 14 15 16
```

```
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29
```

```
Input the number of days in the month (28-31): 30
```

```
Input the starting day (0=Sun, 1=Mon,...):5
```

```
      1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

```
Input the number of days in the month (28-31): 31
```

```
Input the starting day (0=Sun, 1=Mon,...):3
```

```
      1  2  3  4
  5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

```
Even numbers [-3;-25]:
```

```
-4 -6 -8 -10 -12 -14 -16 -18 -20 -22 -24
```

```
<--- TASK 3 --->
```

```
List of numbers: [11, 3, 10, 0, 5, 7, 8, 1, 2, 4, 6, 9]
```

```
Index of max number: 0
```

```
List of numbers: [4, 19, 38, 28, 14, 44, 13, 16, 22, 18, 17, 1, 2, 40, 5, 30, 24,
```

```
Sorted list of numbers: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
```

```
3, 42, 6, 45, 11, 34, 27, 26, 18, 32, 49, 8, 37, 29, 41, 46, 36, 23, 15, 0, 7, 31, 48, 21, 43, 12, 39, 25, 33, 47, 28, 35, 9]
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]
```

### Аналіз результатів:

В першому завданні потрібно було реалізувати функцію для перетворення градусів з одної температурної шкали в іншу з визначеного проміжку. Для цього я написала функцію `convert_c_to_f(degree)`, яка приймала параметром градуси Цельсія та повертала перетворені градуси за Фаренгейтом. Потім викликала її в циклі, який проходив по заданому проміжку, при тому виводячи результат. Також треба було визначити непарні числа в заданому проміжку та знайти їх добуток, що я успішно зробила використовуючи умовний оператор `if-else` в середині циклу `for`.

Друге завдання полягало в тому, що потрібно було модифікувати програму виведення календаря та вивести 5 різних календарів, при тому кожного разу давати змогу користувачеві ввести нові дані. Для цього я написала функцію виводу календаря `print_calendar(days_amount, start_day)`, та в циклі давала змогу користувачеві вводити `n` та `d`, відповідно кількість днів у місяці та початковий день тижня. Ці дані передавала в якості аргументів функції, яку також

викликала в циклі. Другим підзавданням було знайти непарні числа з проміжку від'ємних чисел. Його я зробила аналогічно як і в першому завданні, використовуючи умовний оператор if-else, тільки в даному випадку використала цикл while.

Третє завдання полягало в знаходженні індексу максимального числа масиву та упорядкуванні масиву за допомогою сортування Шелла. Для створення масиву випадкових чисел я написала функцію `create_random_list(size)`, яка приймала аргументом розмір масиву. Всередині створювала список з впорядкованими числами, а потім перемішувала їх за допомогою функції `shuffle` бібліотеки `random`. Використовуючи цикл `for`, знайшла індекс максимального елементу. Для цього теж написала окрему функцію `get_max_value_index(values_list)`, яка аргументом приймала список, результатом, який вона повертала був індекс максимального елементу. Для сортування Шелла написала функцію `shell_sort(values_list)`, яка приймала список, який потрібно упорядкувати, та повертала його вже правильно організованим.

**Висновок:** Під час виконання лабораторної роботи я ознайомилась з мовою програмування Python, розглянула та освоїла базові речі для написання програми, такі як ініціалізація змінних, виконання арифметичних операцій, ввід та вивід результатів, написання та використання функцій, а також використання циклів в програмі. Виконала 3 завдання, згідно зі своїм варіантом. Під час виконання поставленої задачі, я змогла опанувати базовий синтаксис мови Python та провести його порівняльний аналіз відносно інших мов програмування, які вчила раніше, таких як C++, C# чи JavaScript. Йому не притаманні “;”, “{” чи “}”. Тут логічні блоки виділяються відступами. Також оголошення функції відбувається за допомогою ключового слова `def`, а не `function`.

#### **Перелік використаних посилань:**

1. Андрущак Н.А. Методи та засоби комп'ютерного навчання: лабораторний практикум для студентів другого (магістрського) рівня вищої освіти Інституту комп'ютерних наук та інформаційних технологій. – Львів: Видавництво Національного університету «Львівська політехніка», 2018. – 125 с.
2. Руководство по программированию на Python: <https://metanit.com/python/tutorial/>