

Урок №19-20

Цикл for. Цикл з параметром



“І нащо мені вчити ще один цикл 🤖 ? Тільки з while розібрався! 😭”.



“Тут як з виделкою і ложкою. Їсти можна тільки виделкою. Або тільки ложкою. Але зручніше чергувати залежно від страви. Схоже і з циклами. Будь-яку програму, написану на while можна переписати на for. І навпаки. Але є випадки, коли краще працює цикл for у ваших Python програмах, і є випадки, де зручніше з while”

Цикл for варто використати тоді, коли ми завчасно знаємо кількість ітерацій (запусків циклів). Також його зручно запускати по списку чи об'єкту.

Задача 1: Найпростіший For

Снігопад. Запишіть собі приклади схожих циклів, які малюють сніжинки. Змініть цикли так, щоб в останньому рядку було 10 сніжинок.

```
i=0
while (i<5):
    print ("*" * i)
    i+=1
```

```
for i in range (5):
    print ("*" * i)
```

У найпростішому варіанті FOR у функцію range передається тільки кількість запусків циклу.

Синтаксис циклу for



Цикл for зручно використовувати, якщо відома кількість повторень.

Цикл з параметром використовується, коли потрібно забезпечити повторення блоку команд вказану кількість разів. Параметр при цьому має набувати значень, що входять у певну шкалу (чисел, букв чи інших елементів).

for змінна **in** послідовність
— блок операторів
↖ обов'язковий відступ від
лівого краю

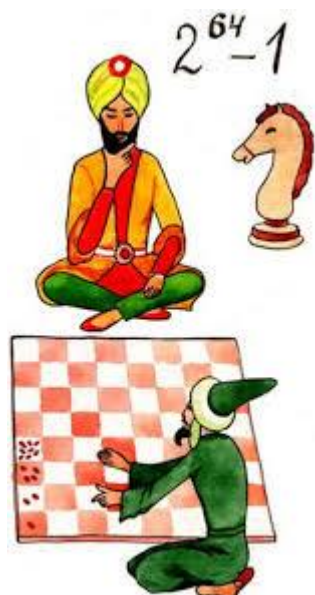
Змінна - це параметр, лічильник, котрий змінює своє значення щоразу при виконанні команд циклу.

Одне виконання команд циклу називається ітерацією.

Тіло циклу складається з команд, котрі виконуються кілька разів.

Важливо дотримуватись правил синтаксичного оформлення циклів у Python, тобто рядок з командою for завершується двокрапкою, а команди тіла циклу записуються з відступом.

Задача 2: Легенда про створення шахів



За легендою, коли творець шахів давньоіндійський математик-мудрець, на ім'я **Cicca** показав свій винахід правителю країни, тому так сподобалася гра, що він проводив години граючи у неї. За таку винахідливість правитель дозволив винахіднику вибрати нагороду.

Мудрець подумав та попросив у короля за **першу** клітку шахівниці заплатити йому **одне зерно** рису, за **другу** – **два**, за **третю** – **чотири** і т. д., подвоюючи кількість зерен на кожній наступній клітинці.

Правитель, який не знався на математиці, швидко погодився, навіть трохи образившись на таку невисоку оцінку винаходу, і наказав керівнику скарбниці підрахувати і видати винахіднику потрібну кількість зерна.

•	••	•••	••••	•••••	••••••	•••••••	128
256	512	1,024	2,048	4,096	8,192	16,384	32,868
64K	128K	256K	512K	1M	2M	4M	8M
16M	32M	64M	128M	256M	512M	1G	2G
4G	8G	16G	32G	64G	128G	256G	512G

Однак, коли через тиждень скарбник все ще не зміг підрахувати, скільки потрібно зерен, правитель запитав, у чому причина такої затримки. Скарбник показав йому розрахунки і сказав, що розплатитися неможливо, хіба що осушити моря та океани і засіяти весь простір пшеницею ☐.

Кількість зерна, що запросив хитрий винахідник, приблизно в 1800 разів перевищує сучасний світовий врожай пшениці за рік (у 2008-09 аграрному році врожай становив 686 млн тонн), і також

перевищує весь урожай пшениці,
зібраний за всю історію людства.

Порахувати суму можна так:

$$1 + 2 + 4 + 8 + \dots = 2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{63}.$$

Онлайн середовище: [Trinket](#)



Давай тепер складемо цикл **for** на python, яка буде писати кількість зерен на кожній клітинці.

Білу клітинку заміни на кількість клітинок на шахівниці.

Чорну клітинку заміни на число в яке збільшується кількість зерна від клітинки до клітинки.

```
for i in range ( ):  
    print ("На", i, "клітинці", 2**i, "зерен")
```



Задача 3: Вигідна угода.

Онлайн середовище: [Trinket](#)

Якщо вам запропонують угоду, чи погодитеся ви:

Вам будуть давати по 10 000 доларів на день. Натомість ви повинні повертати такі гроші протягом місяця: в 1-й день 1 цент, в другий – 2 центи, в 3-й 4 центи і т.д.

Складіть програму, яка рахуватиме, скільки людина отримує грошей в разі “Вигідної угоди” вище і скільки втратить.

Функція range()

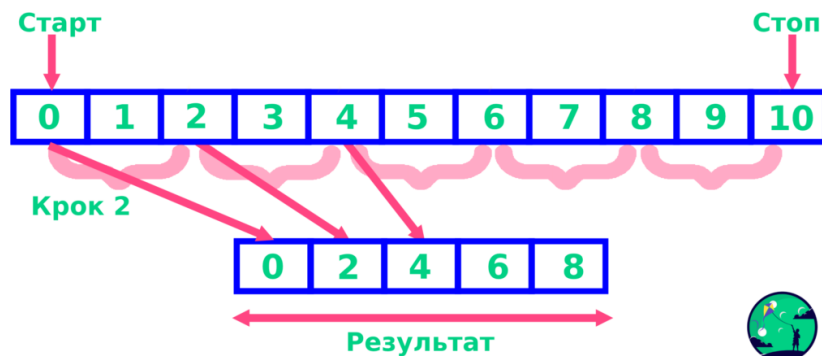
Функція **range()** вказує скільки разів потрібно повторити щось.

Функцію **range()** часто використовує цикл **For** у Python. Вона має такі аргументи:

- **start** вказує ціле число, з якого починається послідовність (включно). Якщо цей аргумент не вказано, використовується значення 0.
- **stop** потрібно використовувати завжди. Це останнє ціле число послідовності (не включно).

- **step** визначає крок: наскільки потрібно збільшити (у разі негативних чисел зменшити) наступну ітерацію. Якщо аргумент **step** пропущено, за промовчанням використовується 1.

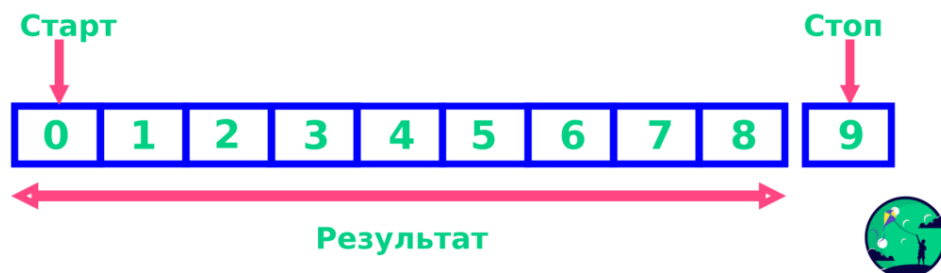
Python range (0,10,2)



Не обов'язково вказувати всі три аргументи для range.

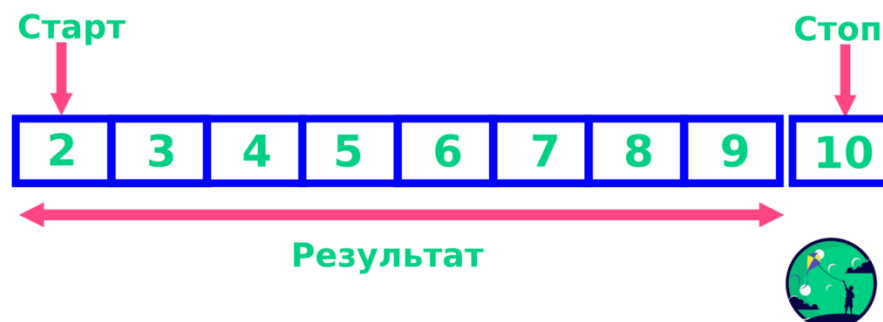
Можна вказати лише один. Тоді цикл буде від 0 (включно) до вказаного числа (не включно) з кроком 1

Python range (9)



Можна вказати два числа. Тоді цикл буде від першого числа (включно) до другого числа (не включно) з кроком 1.

Python range (2,10)



Задача 4: Знайдіть суму всіх чисел від 0 до 100 (включно) за допомогою циклу **for** та **range(stop)**:

Онлайн середовище: [Trinket](#)

```
result=0
for i in range(101):
    result+=i
print (result)
```

Задача 5: Знайдіть суму всіх чисел від 20 до 30 (включно) за допомогою циклу **for** та **range(start,stop)**

Задача 6: Знайдіть суму всіх чисел, які діляться на 5 від 100 до 200 (включно) за допомогою циклу **for** та **range(start, stop, step)**

Також в параметрі “крок” (step) можна використовувати від’ємні числа. Тоді цикл перебиратиме значення у зворотному напрямку. У такому разі не забудьте відкоригувати аргументи start і stop (поставте їх за спаданням).

Задача 7: Давайте напишемо програму-таймер, яка відрховує задану кількість секунд

```
import time
for i in range(30,0,-1):
    time.sleep(1)
    print(i)
```

time.sleep(1) – ця команда допомагає програмі заснути на одну секунду

Задача 8: Змініть таймер, щоб він рахував 20 секунд.

Онлайн середовище: [Trinket](#)

Задача 9: Подана програма виводить всі дні січня. Перепишіть її, використовуючи цикл **for** замість **while**

```
date = 1
while date <= 31:
    print ( str(date) + " of January")
    date = date + 1
```

Задача 10: Дана програма виводить табличку множення на 8 для чисел від 0 до 10. Ваша задача змінити програму так, щоб вона виводила табличку множення на 7 для чисел від 11 до 20.

```
for i in range (11):
    res=i*8
    print (i, "*8=", res)
```

Задача 11: Виведіть всі числа, які діляться на 3 із діапазону, який введе користувач.
Приклад роботи:

Введіть перше число: 6
Введіть друге число: 15
6, 9, 12

Додаткові бали, якщо в програмі будуть перевірки: Додайте необхідні перевірки того, що користувач ввів дані правильно. Якщо дані неправильні, виведіть повідомлення про помилку.

Задача 12: У юного тітокера зараз 5000 підписників. Він дізнався, що якщо буде працювати над власним контентом і викладати вчасно нові відео, то до нього на канал будуть додаватися нові підписники, при чому у їх кількість буде зростати щомісяця. У перший місяць може підписатися 600 людей, у 2-й – 700, у 3-й – 800 і т.д. Створіть програму, як буде показувати, яка кількість користувачів буде у блогера в кожний місяць року.

Підказка по формулі: $500+i*100$ (якщо місяці рахувати з 1)

Онлайн середовище: [Trinket](#)

Додаткові завдання

- 1) Програма має вивести на екран n (запитується у користувача) разів слово «Привіт».
- 2) З клавіатури вводиться число. Знайти суму чисел від 1 до цього числа.
- 3) З клавіатури вводиться два числа. Знайти суму чисел, які розміщені між ними.
- 4) Спортсмен розпочав тренування, пробігаючи 10 км в день. Щодня від збільшував норму на 10% від пробігу попереднього дня. Який шлях пробіжить спортсмен за 10 днів?
- 5) У гусей та кроликів разом 64 лапи. Скільки може бути кроликів та гусей (всі можливі комбінації)?
- 6) Надрукувати таблицю множення на 2.
- 7) Морозиво коштує 1,75. Надрукувати табличку у допомогу продавцеві:
1 порцій коштують 1.75 грн
2 порцій коштують 3.5 грн
3 порцій коштують 5.25 грн
4 порцій коштують 7.0 грн
5 порцій коштують 8.75 грн
6 порцій коштують 10.5 грн

7 порцій коштують 12.25 грн

8 порцій коштують 14.0 грн

9 порцій коштують 15.75 грн

8) Надрукувати всі двозначні числа з різними цифрами.

9) Надрукувати всі дільники числа.

Пробуємо ділити число на всі числа від 1 до цього числа. Якщо знаходимо число, при діленні на яке одержуємо остачу 0, то це - дільник, друкуємо його.

10) Обчислити значення функції $y=4x^2+3x+2$ на проміжку від -10 до 10.

11) У бригади є 10 косилок. Перша працювала m годин, кожна наступна - на 10 хвилин довше. Скільки годин і хвилин працювали косилки?

12) Знайти всі дільники натурального числа k . Число 1 і саме число k не включати до переліку дільників. Алгоритм розв'язування задачі: перебрати всі значення числа i від 2 до $k // 2$; якщо i є дільником числа k ($k \% i == 0$), вивести значення i .

13) Досконалими давні греки називали числа, які дорівнюють сумі всіх своїх дільників, окрім самого числа ($6=1+2+3$). Знайти досконалі числа в діапазоні від m до n . Алгоритм розв'язування задачі: для кожного числа k від m до n знайти суму дільників від 1 до $k // 2$; якщо сума дільників дорівнює числу k , вивести значення k .

14) За допомогою методу перебору знайти розв'язок системи рівнянь

$$\begin{cases} 2x + 3y = 20; \\ 3x - y = 19 \end{cases}$$

Значення x не може бути більшим за 7, т.к., якщо $y=0$, то $23//3=7$. Значення y не може бути більшим за 6, т.к., якщо $x=0$, то $20//3=6$. У зовнішньому циклі переберемо значення x з діапазону $[0; 8)$. Для кожного x у внутрішньому циклі переберемо значення y з діапазону $[0; 8)$. Для кожного сполучення значень потрібно перевіряти виконання умови $(2*x-3*y==8)$ and $(3*x+y==23)$.