# ALGORYTMY ITERACYJNE W JĘZYKU PYTHON.

**Iteracja** - jest to czynność powtarzania tej samej operacji w pętli z góry określoną ilość razy lub aż do spełnienia określonego warunku.

**Pętla** - jedna z konstrukcji programowania strukturalnego; pozwala na cykliczne wykonywanie instrukcji określoną ilość razy, do momentu zajścia pewnych warunków, dla każdego elementu zbioru lub w nieskończoność.

Aby w języku programowania napisać program realizujący algorytm iteracyjny, stosujemy instrukcje iteracyjne (zwane też instrukcjami pętli).

**Przykład 1.**
**Obliczenie sumy pięciu liczb.**

1. Zacznij algorytm.
2. Suma jest równa zero.
3. Wprowadź pierwszą liczbę.
4. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
5. Wprowadź drugą liczbę.
6. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
7. Wprowadź trzecią liczbę.
8. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
9. Wprowadź czwartą liczbę.
10. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
11. Wprowadź piątą liczbę.
12. Do poprzedniego wyniku sumowania dodaj wprowadzoną liczbę.
13. Wyprowadź wynik sumowania.
14. Zakończ algorytm.

W przedstawionej sytuacji możemy powtarzające się ciągi operacji (tu kroki 3. i 4.) zapisać tylko raz i zastosować pętlę, czyli po wykonaniu czwartego kroku wrócić do trzeciego, to znaczy wykonać kroki 3. i 4. pięć razy

**Przykład 2.**
**Zapisujemy algorytm sumowania n liczb w postaci listy kroków.**

**Zadanie:** Oblicz sumę **n** liczb całkowitych.
**Dane:** liczba naturalna **n** większa od zera, oznaczająca, ile liczb ma być sumowanych, **n** dowolnych liczb całkowitych.
**Wynik:** wartość sumy: **suma**.

**Lista kroków:**

1. Zacznij algorytm.
2. Zmiennej **suma** przypisz wartość 0.
3. Zmiennej **i** przypisz wartość 0.
4. Podaj, ile liczb ma być zsumowanych. Zapamiętaj wprowadzoną liczbę w zmiennej **n**.
5. Wprowadź liczbę do sumowania – zapamiętaj ją w zmiennej **a**.
6. Zmiennej suma przypisz wartość zmiennej suma powiększoną o wartość zmiennej a: **suma = suma + a**.
7. Zmiennej **i** przypisz wartość zmiennej **i** (kolejna wprowadzona liczba) powiększoną o 1: **i = i + 1**.
8. Jeśli **i < n**, wróć do kroku 5.
9. Wyprowadź wynik: **suma**.
10. Zakończ algorytm.

Przypisanie **suma = suma + a** oznacza „pod zmienną **suma** podstaw poprzednią wartość **suma** zwiększoną o wartość kolejnej liczby **a**".

suma = 4
suma = suma + 40

Wynikiem będzie 44

**Przykład 3.**
**Analiza działania algorytmu dodawania n liczb.**

Działanie algorytmu dla **n** = 5 i dla kolejnych wartości zmiennej **a**: 7, 20, 13, 15, 8.

Zmienna **i** oznacza w tym przykładzie numer wprowadzonej liczby. W analizie występują wartości zmiennych po wykonaniu każdego cyklu iteracji.

Na początku przyjmujemy: **suma** = 0, **i** = 0, **n** = 5.

| | | | | |
|---|---|---|---|---|
| cykl 1. a = 7 | suma = 0 + 7 = 7 | i = 0 + 1 = 1 | czy i < n, i < 5? | TAK |
| cykl 2. a = 20 | suma = 7 + 20 =27 | i = 1 + 1 = 2 | czy i < n, i < 5? | TAK |
| cykl 3. a = 13 | suma = 27 + 13 = 40 | i = 2 + 1 = 3 | czy i < n, i < 5? | TAK |
| cykl 4. a = 15 | suma = 40 + 15 = 55 | i = 3 + 1 = 4 | czy i < n, i < 5? | TAK |
| cykl 5. a = 8 | suma = 55 + 8 = 63 | i = 4 + 1 = 5 | czy i < n, i < 5? | NIE |

Wyprowadzenie wartości zmiennej suma: 63

**Ćwiczenie 1.**
**Analiza działania algorytmu dodawania liczb.**

Prześledź działanie algorytmu dodawania n liczb dla n = 7 i dla kolejnych wartości zmiennej a: 21, -30, 14, -7, 28, 9, 19.

# Schemat instrukcji iteracyjnej for w języku Python.

Instrukcja **for** określa długość listy wartości po słowie **in**.

```
for zmienna in lista_wartości:
    lista_instrukcji
```

Jako *lista_instru*kcji może wystąpić pojedyncza instrukcja (w tym instrukcja pętli) lub więcej instrukcji (blok instrukcji).

*Lista_instrukcji* musi być przesunięta w prawo względem for przynajmniej o jedną spację (przyjęte jest wcięcie składające się z czterech spacji lub tanulator).

Liczbę iteracji w instrukcji **for** określa długość *listy_wartości* po słowie **in**.

*Lista_instrukcji* zostanie wykonana dla wszystkich wartości z *listy_wartości*.

*Listę_wartości* możemy zapisać w różny sposób.

Przykład iteracji:

a. Wykonaj trzy razy: wrzuć łopatę węgla do piwnicy.
b. Niech *n* = 10; wykonaj *n* razy: wrzuć łopatę węgla do piwnicy.
c. Dopóki przed wsypem leży węgiel, powtarzaj: wrzuć łopatę węgla do piwnicy.

Powtarzanie odbywa się ustaloną liczbę razy (a, b), albo tak długo, jak długo spełniony jest warunek, którego sprawdzenie następuje przed każdorazowym powtórzeniem czynności (c).

**Przykład 4.**
**Stosowanie instrukcji for.**

Podajemy przykłady określania listy_wartości w instrukcji for:

```
for i in [0, 1, 2, 3, 4, 5]:
    print(i)
```

Instrukcja **print(i)** zostanie wykonana sześć razy dla listy wartości [0, 1, 2, 3, 4, 5];
zmienna **i** będzie przyjmować kolejne wartości z tej listy, czyli: 0, 1, 2, 3, 4, 5.

```
for i in [2, 4, 6, 8, 10]:
    print(i)
```

Instrukcja **print(i)** zostanie wykonana pięć razy dla listy [2, 4, 6, 8, 10];
zmienna **i** będzie przyjmować kolejne wartości z tej listy, czyli: 2, 4, 6, 8, 10.

Nie zawsze chcemy wpisywać wszystkie wartości listy, zwłaszcza dla dużej liczby iteracji.

DO UTWORZENIA LISTY_WARTOŚCI MOŻNA UŻYĆ FUNKCJI **RANGE()**, KTÓRA TWORZY SEKWENCJĘ WARTOŚCI CAŁKOWITYCH.

**Przykład 5.**
**Stosowanie funkcji range() do określania listy_wartości w instrukcji for.**

Argumentami funkcji **range()** mogą być konkretne wartości lub zmienne, np. range(10), range(n). Wartość zmiennej **n** możemy wprowadzać z klawiatury.

- ✓ Dla jednego argumentu: **range***(koniec)*

```
for i in range(7):
    a = int(input("Podaj liczbę: "))
```

Instrukcja **a = int(input("Podaj liczbę: "))** zostanie wykonana siedem razy.
Funkcja range() wygeneruje kolejne liczby całkowite z przedziału ⟨0, koniec).
Zmienna **i** będzie przyjmować kolejno wartości: 0, 1, 2, 3, 4, 5, 6.

```
for i in range(10):
    print("Witaj")
```

Instrukcja **print("Witaj")** zostanie wykonana dziesięć razy.
Funkcja range() wygeneruje kolejne liczby całkowite z przedziału ⟨0, koniec).
Zmienna **i** będzie przyjmować kolejno wartości: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

- ✓ Dla dwóch argumentów: **range***(początek, koniec)*

```
for i in range(18, 95):
    print(i)
```

Instrukcja **print(i)** zostanie wykonana siedemdziesiąt siedem (95 – 18 = 77) razy.
Funkcja range() wygeneruje kolejne liczby całkowite z przedziału ⟨początek, koniec).
Zmienna **i** będzie przyjmować kolejno wartości: 18, 19, 20, …, 94.

- ✓ Dla trzech argumentów: **range***(początek, koniec, krok)*

```
for i in range(2, 102, 2):
    print(i)
```

Instrukcja **print(i)** zostanie wykonana pięćdziesiąt razy.
Funkcja range() wygeneruje kolejne liczby całkowite z przedziału ⟨początek, koniec), zmieniające się o krok.
Zmienna **i** będzie przyjmować kolejno wartości: 2, 4, 6, …, 100. Trzeci argument (krok) określa tym samym, o jaką wartość zmienia się zmienna **i**.

```
for i in range(10, 0, -1):
    print(i)
```

Instrukcja **print(i)** zostanie wykonana dziesięć razy. Krok wynosi -1.
Zmienna **i** będzie przyjmować kolejno wartości: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

**Przykład 6.**
**Chcemy zapytać 3 użytkowników o imię, a następnie przywitać każdego po imieniu.**

Możemy zrobić to tak:

```
name = input("Jak masz na imię? ")
print("Cześć", name)
name = input("Jak masz na imię? ")
print("Cześć", name)
name = input("Jak masz na imię? ")
print("Cześć", name)
```

Szybciej i wygodniej możemy to uzyskać używając pętli for:

```
for user in range(3):
    name = input("Jak masz na imię? ")
    print("Cześć",name)
```

**Ćwiczenie 2.**
**Stosujemy instrukcję for.**

1. Korzystając z **Przykładów 4.** i **5.**, napisz poniższe programy:
   a) Program wyświetlający imiona trzech osób znajdujących się nad Tobą w dzienniku. Każde imię powinno być w kolejnej linijce. Zapisz program jako **osoby_z_dziennika**
   b) Program wyświetlający w kolumnie liczby całkowite nieparzyste od 1 do 33. Zapisz program jako **nieparzyste_1_33**
   c) Program wyświetlający w kolumnie liczby całkowite od 10 do -10. Zapisz program jako **całkowite_10_do_-10**
   d) Program wyświetlający na ekranie w kolejnych wierszach n napisów „Lubię szkołę” (wartość zmiennej n wprowadzaj z klawiatury). Zapisz program jako **lubie_szkole**
   e) Program wyświetlający w kolumnie liczby całkowite od 1 do k (wartość zmiennej k wprowadzaj z klawiatury). Zapisz program jako **liczby_calkowite_k**
   f) Napisz specyfikację zadania i program, który obliczy sumę dwunastu comiesięcznych wpłat na zakup telewizora i wyświetli wynik na ekranie. Zapisz program jako **suma_tv**
   g) Zmodyfikuj powyższy program, aby mógł obliczyć sumę wpisywanych z klawiatury wybranej ilości comiesięcznych wpłat (użytkownik decyduje ile będzie rat) na zakup telewizora i wyświetlić wynik na ekranie. Zapisz program jako **dowolna_suma_tv**

2. Każdy program zapisz w pliku i uruchom.