LISTY W JĘZYKU PYTHON
Definiowanie listy, operacje na listach

Większość programów działa nie tylko ze zmiennymi. Korzystają również z list zmiennych.

Lista jest to zmienna zawierająca uporządkowany zbiór elementów. Listy są modyfikowalne, czyli można dodawać i zmieniać poszczególne elementy danej listy.

W zmiennej *imię* możemy zapisać Jan, Paweł, Ewa. Za każdym razem imię zostanie nadpisane przez wpisanie nowszego imienia. Lista pozwala nam na przechowywanie wielu imion - elementów.

Aby zapamiętać wszystkie dane, nadajemy im różne nazwy. Stosujemy wówczas tzw. **zmienne indeksowane**, np. a0, a1, …, an-1 - zmiennej *a* dodajemy kolejny numer (indeks). W takiej sytuacji dla każdej zmiennej komputer rezerwuje oddzielna komórkę pamięci.

Listę tworzymy za pomocą nawiasów kwadratowych [], pomiędzy nimi wypisane są jej elementy rozdzielone przecinkami.

```
nazwa_listy = [element_1, element_2, ...,element_n]
```

Przykład:

definiowanie trzyelementowej listy i przypisanie jej do zmiennej 'moja_lista'

```
moja_lista = [1,2,3]
```

Elementami listy mogą być dowolne typy danych. Nie określamy typu elementów listy. Lista może być również pusta.

```
a = 5
moja_lista = [1, "dwa", 4.75, a]
moja_pusta_lista = []
```

Definiowanie listy w Python

Do elementów listy odwołujemy się podając nazwę listy i indeks elementu umieszczony w nawiasach kwadratowych, np. :

moja_lista[0], moja_lista[1], … , moja_lista[n-1] - dla listy n-elementowej o nazwie moja_lista. Pierwszy indeks jest zawsze równy 0.

```
moja_lista = [1,2,3]
print(moja_lista[2]) #wyświetlona zostanie "3"
```

Definicja listy	Opis	Sposób odwołania
uczniowie = [1, 3, 7]	Oznacza zdefiniowanie listy o nazwie uczniowie, składającej się z 3 elementów o indeksach od 0 do 2	Do elementów listy odwołujemy się przez zmienne: uczniowie[0], .., uczniowie[2],
lista = [0] * 10	Oznacza zdefiniowanie listy o nazwie lista składającej się z 10 elementów równych 0.	Do elementów listy odwołujemy się przez zmienne lista[0], …, lista[9]
n = 30 lista = [0] * n	Oznacza zdefiniowanie listy o nazwie lista składającej się z 30 elementów o wartości początkowej 0	Do elementów listy odwołujemy się przez zmienne lista[0], lista[1], …, lista[n-1]

Cechy listy

Lista jest strukturą pozwalającą na przechowywanie elementów w sposób uporządkowany i posiada następujące cechy:

- ✓ przechowuje elementy,
- ✓ zachowuje kolejność (indeksy 0,1,2 …),
- ✓ elementy mogą się powtarzać,
- ✓ jedna lista może zawierać zmienne różnych typów.

Wprowadzanie elementów do listy i wyprowadzanie elementów na ekran.

Do wprowadzania i wyprowadzania elementów z listy można zastosować funkcję niezwracającą wartości bez parametrów lub z parametrami.

Przykład 1. Definiowanie i stosowanie listy w języku Python

Zadanie: Wprowadź n liczb całkowitych do listy o nazwie a, następnie wyprowadź w kolumnie elementy listy na ekran. Zdefiniuj dwie funkcje niezwracające wartości bez parametrów: wprowadz_dane() i wyprowadz_dane(). Wywołaj je na końcu w programie głównym.

Dane: stała wartość n, lista liczb lista[n]

Wynik: wyświetlone w kolumnie elementy listy lista: lista[0], .., lista[n-1]

```
n = 5
moja_lista = [0] * n

def wprowadz_dane():
    for i in range(n):
        moja_lista[i] = int(input("podaj liczbę: "))
def wyprowadz_dane():
    for i in range(n):
        print("element",i,":",moja_lista[i])
wprowadz_dane()
wyprowadz_dane()
```

lub inny zapis:

```
n = 5
moja_lista = [0] * n

def wprowadz_dane():
    for i in range(n):
        moja_lista[i] = int(input("podaj liczbę: "))
def wyprowadz_dane():
    for i in range(n):
        print(f"element {i}: {moja_lista[i]}")
wprowadz_dane()
wyprowadz_dane()
```

Ćwiczenie 1. Wprowadzamy dane do listy i wyprowadzamy je na ekran

1. Przepisz poniższy kod i sprawdź jaki jest wynik:

```
n = 5
moja_lista = [0] * n
print(moja_lista)
```

2. Przepisz kodu z Przykładu 1 (dowolna wersję).
3. Zapisz plik jako **lista_n_liczb**
4. Przetestuj program dla różnych wartości elementów listy.

Ćwiczenie 2. Wyprowadzamy elementy z listy w odwrotnej kolejności.

1. Otwórz plik lista_n_liczb
2. Zmodyfikuj program tak, aby dane wyświetlały się w odwrotnej kolejności.

   Dodatkowo zwiększ liczbę danych do dziesięciu.
3. Podpowiedź: w pętli for w funkcji range użyj trzech argumentów.
4. Zapisz plik pod nazwa **lista_n_liczb_odwrotnie**

Ćwiczenie 3. Wyświetlamy na ekranie wybrany element listy

1. Otwórz program lista_n_liczb
2. Zmodyfikuj program tak, aby na ekranie wyświetlać tylko k-ty element, gdzie k jest liczbą całkowitą wprowadzoną przez użytkownika z klawiatury.
3. Podpowiedź: należy sprawdzić czy wprowadzona wartość k jest >= 0 oraz < n.
4. Zapisz plik pod nazwą **lista_n_liczb_element**

Ćwiczenie 4. Wyświetlanie na ekranie kilku wybranych elementów listy

1. Otwórz program lista_n_liczb.
2. Zmodyfikuj program tak, aby na ekranie wyświetlić tylko pierwszy i ostatni element.
3. Zapisz plik pod nazwą **lista_n_liczb_pierwszyiostatni**