

Listy

Podstawy programowania w języku Python

A word cloud featuring various Python list methods and related concepts. The most prominent words are 'list.remove()', 'list()', and 'list.pop()'. Other visible words include 'list.append()', 'list.reverse()', 'list.sort()', 'sorted()', 'deque', 'IndexError', 'list.insert()', 'list.copy()', 'element in list', 'list.reverse()', 'reversed()', and '[]'. The words are arranged in a dense, overlapping manner, with colors ranging from dark blue to light blue.

Przechowywanie wielu wartości

- zastosowanie zmiennej pozwala nam przechować pojedynczą wartość

```
1 fruit = "banan"  
2 number = 7
```

- do przechowania większej ilości wartości możemy zastosować listę

```
1 fruits = ["banan", "jabłko", "czereśnia"]  
2 numbers = [1, 2, 3, 7, 9]
```

Co to jest lista

- uporządkowana kolekcja elementów
- reprezentuje zbiór wartości
- tworzymy ją za pomocą nawiasów kwadratowych, oddzielając elementy przecinkami np. [a, b, c]

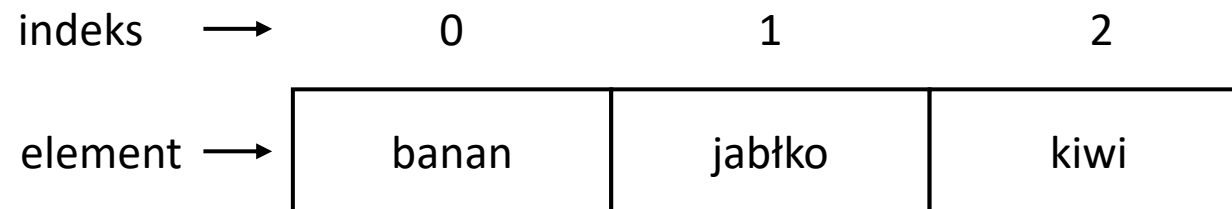
```
1 fruits = ["banan", "jabłko", "czereśnia"]  
2 print(fruits)
```

Właściwości listy

- jest uporządkowana
- pozwala na dodawanie i usuwanie elementów
- pozwala przechowywać duplikaty
- pozwala przechowywać elementy o różnych typach
- każdy element listy posiada indeks
- elementy listy są zawsze numerowane od zera

Odwoływanie się do elementów listy

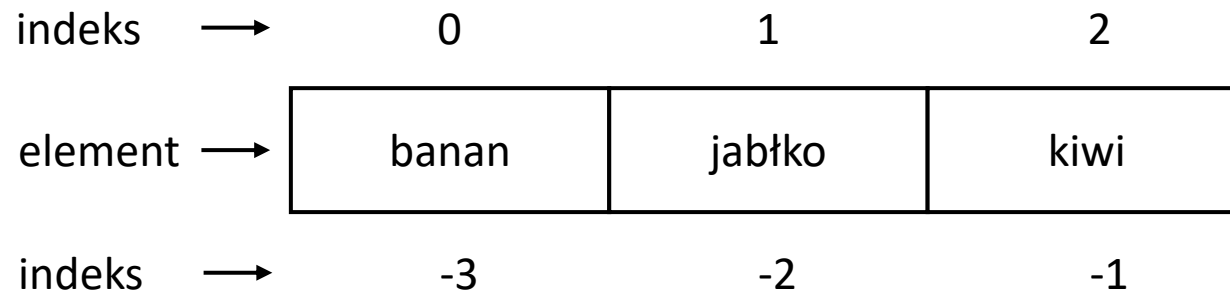
- do elementów listy odwołujemy się za pomocą indeksów
- pierwszy element ma indeks zero



```
1 fruits = ["banan", "jabłko", "kiwi"]
2 print(fruits[0]) #banan
3 print(fruits[2]) #kiwi
```

Odwoływanie się do elementów listy

- można także stosować indeksy ujemne
- ostatni element listy ma zawsze indeks -1



```
1 fruits = ["banan", "jabłko", "kiwi"]
2 print(fruits[-1]) #kiwi
3 print(fruits[-2]) #jabłko
```

Wyznaczanie długości listy

- funkcja **len()** podaje aktualną długość listy
- przyjmuje nazwę listy jako argument
- zwraca liczbę elementów aktualnie przechowywanych w liście

```
1 fruits = ["ananas", "banan"]  
2 print(len(fruits)) #wynik: 2
```

Usuwanie elementów z listy

- do usuwania elementów z listy służy instrukcja **del**
- należy wskazać element do usunięcia
- usunięty element zniknie z listy a długość listy zmniejszy się o jeden

```
1 numbers = [1, 5, 7]
2 del numbers[1]
3 print(numbers) #[1, 7]
```

del jest instrukcją a nie funkcją

Funkcje a metody

- metoda jest specjalnym rodzajem funkcji
- funkcja nie należy do żadnych danych - pobiera dane, może tworzyć nowe dane i generalnie daje wynik
- metoda robi wszystkie te rzeczy, ale jest również w stanie zmienić stan wybranej jednostki
- metoda jest własnością danych, na których operuje, podczas gdy funkcja jest własnością całego kodu

wynik = funkcja(argument)

wynik = dane.metoda(argument)

Dodawanie elementów do listy

- metoda **append()** dodaje element na końcu listy

```
1 numbers = [1, 5, 7]
2 numbers.append(4)
3 print(numbers) #[1, 5, 7, 4]
```

- metoda **insert()** potrafi dodać element w dowolnym miejscu listy

```
1 numbers = [1, 5, 7]
2 numbers.insert(1, 4) #wstawiamy na pozycji 1 element 4
3 print(numbers) #[1, 4, 5, 7]
```

Iterowanie po liście

- z użyciem indeksów

```
1 fruits = ["ananas", "gruszka", "banan"]  
2 for i in range(len(fruits)):  
3     print(fruits[i])
```

- bez użycia indeksów

```
1 fruits = ["ananas", "gruszka", "banan"]  
2 for fruit in fruits:  
3     print(fruit)
```

Pytanie

Co wyświetli się na ekranie po wykonaniu poniższego skryptu?

- a) 3
- b) [1, 2, 3]
- c) wystąpi błąd `IndexError`
- d) 2
- e) wystąpi błąd `NameError`

```
1 numbers = [1, 2, 3]
2 print(numbers[3])
```

Odpowiedź: c)

Na liście znajdują się tylko elementy o indeksach 0, 1 i 2. Zawsze po odwołaniu do nieistniejącego elementu na liście otrzymamy błąd **`IndexError: list index out of range`**.

Pytanie

Wskaż poprawny zapis usunięcia elementu z listy **numbers**:

- a) `del(numbers)`
- b) `del(numbers[1])`
- c) `del numbers[1]`
- d) `del numbers(1)`
- e) `del numbers`

Odpowiedź: c)

Do usunięcia elementu listy stosujemy instrukcję **del**. W przeciwieństwie do funkcji, nie stosujemy tutaj nawiasów okrągłych.

Pytanie

Co wyświetli się na ekranie po wykonaniu poniższego skryptu?

- a) wystąpi błąd
- b) True
- c) konewka
- d) 7
- e) False

```
1 elements = [True, "konewka", 7]
2 elements.append(False)
3 print(elements[-1])
```

Odpowiedź: e)