

Listy - dostęp do elementów

Elementy listy są indeksowane i można uzyskać do nich dostęp, odwołując się do numeru indeksu zaczynając od 0, np. odczytaj 2 element listy, czyli indeks 1

```
cars = ["Audi", "Skoda", "Opel", "Toyota"]  
print(cars[1])
```

Indeksowanie negatywne

Indeksowanie ujemne oznacza rozpoczęcie od końca -1 odnosi się do ostatniej pozycji, -2 odnosi się do przedostatniej pozycji itd.

```
cars = ["Audi", "Skoda", "Opel", "Toyota"]  
print(cars[-1])
```

Zakres indeksów

Można określić zakres indeksów, określając, gdzie zaczynać i gdzie kończyć zakres. Podczas określania zakresu **zwracaną wartością** będzie **nowa lista** z określonymi pozycjami.

Zwróć trzeci, czwarty i piąty przedmiot:

```
fruits = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
new_fruits = fruits[2:5]  
print(new_fruits)
```

Element z indeksem 5 nie będzie już zwracany.

Pominięcie wartości początkowej spowoduje, że zakres zacznie się od pierwszego elementu, a pominięcie wartości końcowej spowoduje przesunięcie zakresu na koniec listy::

```
fruits = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
new_fruits = fruits[:5] # zapisze elementy od indeksu 0 do 4 (bez 5)  
print(new_fruits)  
another_fruits = fruits[2:] # zapisze elementy indeksu od 2 do ostatniego
```

Zakres indeksów ujemnych

Ten przykład zwraca elementy z „pomarańczy” -4 do „mango” -1, ale NIE włączając tego

```
fruits = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(fruits[-4:-1])
```

Sprawdź, czy element istnieje

Aby określić, czy określony element znajduje się na liście, użyj słowa kluczowego `in`:

Sprawdź, czy na liście znajduje się pomarańcza:

```
fruits = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"] if "orange" in fruits:  
print("Pomarańcza jest na liście")
```

Listy - zmiany elementów

Zmień wartość elementu listy

Zmieńmy 7 element listy (indeks 6)

```
cars = ["Skoda", "Opel", "Volvo", "BMW", "Ferrari", "Koenigsegg", "Baic",  
"Ford"]  
cars[6] = "Fiat"  
print(cars)
```

Zmiana zakresu wartości listy

Aby zmienić wartość pozycji w określonym **zakresie**, zdefiniuj **listę z nowymi wartościami** i odwołaj się do zakresu numerów indeksu, w którym chcesz wstawić nowe wartości:

```
cars = ["Skoda", "Opel", "Volvo", "BMW", "Ferrari", "Koenigsegg", "Baic",  
"Ford"]  
cars[1:3] = ["Toyota", "Suzuki"]  
print(cars)
```

Jeśli wstawisz *więcej* elementów niż zastąpisz, nowe elementy zostaną wstawione w określonym miejscu, a pozostałe elementy zostaną odpowiednio przesunięte:

Zmień drugą wartość, zastępując ją *dwoma* nowymi wartościami:

```
cars = ["Skoda", "Opel", "Volvo", "BMW", "Ferrari", "Koenigsegg", "Baic",  
"Ford"]  
cars[1:2] = ["Toyota", "Suzuki"]  
print(cars)
```

Długość listy ulegnie zmianie, jeśli liczba wstawionych pozycji nie będzie odpowiadać liczbie zastąpionych pozycji.

Jeśli wstawisz *mniej* elementów niż zastępujesz, nowe elementy zostaną wstawione w określonym miejscu, a pozostałe elementy zostaną odpowiednio przesunięte:

Zmień drugą i trzecią wartość, zastępując je *jedną* wartością:

```
names = ["Adam", "Ewa", "Jerzy"]  
names[1:3] = ["Zofia"]  
print(names)
```

Wstawianie nowych elementów

Aby wstawić nowy element listy, bez zastępowania żadnej z istniejących wartości, możemy skorzystać z metody `insert()`.

```
names = ["Adam", "Ewa", "Jerzy", "Małgorzata"]
names.insert(2, "Zofia")
print(names)
```

Uwaga: w wyniku powyższego przykładu lista będzie teraz zawierać 4 pozycje.

Listy - dodawanie elementów

Dołączanie elementów

Aby dodać pozycję na koniec listy, użyj metody `append()`

```
fruits = ["apple", "banana", "cherry"]
fruits.**append**("orange")
print(fruits)
```

Wstawianie elementów

Aby wstawić element listy o określonym indeksie, użyj `insert()` metody.

Metoda `insert()` wstawia element o określonym indeksie:

```
fruits = ["apple", "banana", "cherry"]
fruits.insert(1, "mango")
print(fruits)
```

Listy teraz będą zawierać 4 elementy.

Rozszerzanie listy

Aby dołączyć elementy z *innej listy* do aktualnej listy należy skorzystać z metody `extend()`.

```
cars = ["Audi", "Skoda", "Opel"]
others_cars = ["Ferrari", "Koenigsegg", "Baic", "Ford"]
cars**.extend(**others_cars**)
print(cars)
```

Elementy zostaną dodane na *końcu* listy.

Dodawanie dowolnej iteracji

Metoda `extend()` nie wymaga dołączania *list*, można dodać dowolny obiekt iterowalny (**krotki**, zbiory, słowniki itp.).

Dodaj element **krotki** do listy:

```
racers = ["Ferrari", "Koenigsegg"] cars_tuple = ("Lamborghini", "Cayman") racers.extend(cars_tuple)  
print(cars)
```