

Listy

Python ma wbudowany typ: **lista**.

1. Lista służy do przechowywania wielu elementów w jednej zmiennej.
2. Lista to bardzo uniwersalny typ danych, w którym elementy ułożone są w liniowym porządku, określonym poprzez związane z nimi indeksy.
3. Listy są definiowane w nawiasach kwadratowych [].
4. Listy to jeden z bardziej użytecznych typów danych w Pythonie. Podobnie jak łańcuch znaków, lista stanowi sekwencję elementów. Różnica polega na tym, że jedna lista może zawierać elementy różnych typów.
5. Najważniejsze właściwości tego typu: służy do grupowania obiektów może przechowywać obiekty dowolnego typu elementy są indeksowane.
6. Istnieje kilka sposobów na stworzenie listy - najprostszy z nich to użycie nawiasów kwadratowych.
7. Przydatność list w Pythonie jest zasadna dzięki metodom jakie oferuje.

Funkcje działające na sekwencjach:

len(lista) – zwraca ilość elementów;

enumerate() – zwraca obiekt zawierający indeksy i elementy sekwencji;

Wykorzystanie **enumerate()** w pętli for:

enumerate() inaczej wyliczanie, zwróci wyliczony obiekt iterowany, który musi być sekwencją (iteratorem lub innym obiektem obsługującym iterację). Funkcja **enumerate()** zwróci krotkę zawierającą liczbę oraz wartość uzyskaną z iteracji. Liczba to element iteracyjny, który rozpoczynamy od 0 (numer indeksu).

Inne funkcje działające na listach (zmieniające listy):

reversed() – zwraca obiekt zawierający odwróconą sekwencję;

sorted(lista) – zwraca kopię listy posortowanej rosnąco;

sorted(lista, reverse=True) – zwraca kopię listy w odwrotnym porządku;

Lista ma wiele użytecznych metod:

lista.append(x) – dodaje x do listy;

lista.remove(x) – usuwa pierwszy x z listy;

lista.insert(i, x) – wstawia x przed indeksem i;

lista.count(x) – zwraca ilość wystąpień x;

lista.index(x) – zwraca indeks pierwszego wystąpienia x;

lista.pop() – usuwa i zwraca ostatni element listy;

lista.sort() – sortuje listę rosnąco;

lista.clear() – usuwa zawartość całej listy (kasuje zawartość);

lista.reverse() – sortuje listę w odwróconym porządku.

Indeksowanie list w Pythonie.

Elementy w liście są indeksowane od 0 do $n - 1$ (n ilość elementów w liście). Czyli pierwszy element listy to element o indeksie 0, zaś ostatni to $n - 1$. Istnieje jeszcze indeksowanie wtórne, oznacza to że można podać indeks ujemny. Podając indeks równy -1 wskazujemy na ostatni element listy.

```
listy.py x
1 lista = [21, 62, 23, 37]
2 print(lista[-1])
3 # ostatni element - 37
```

Przykłady:

```
listy.py x
1 colors = ['red', 'blue', 'green']
2 print(colors[4])    ## out_of_range
3 print(colors[-1])  ## ostatni elem. listy
4 print(colors[0:2]) ## red_blue
```

```
listy.py x
1 colors = ['red', 'blue', 'green']
2 # append
3 colors.append('yellow')
4 # Dodaje pojedynczy element na koniec listy.
5 # Często błąd: nie zwraca nowej listy,
6 # tylko modyfikuje oryginał.
```

```
listy.py x
1 colors = ['red', 'blue', 'green']
2 # insert
3 colors.insert(1, 'black')
4 # Wstawia element pod podanym indeksem,
5 # przesuwając elementy w prawo stawia element
6 # pod podanym indeksem, przesuwając elementy w prawo.
7
```

```
listy.py x
1 colors = ['red', 'blue', 'green']
2 # pop
3 colors.pop()
4 colors.pop(1)
5 tmp = colors.pop()
6 # Usuwa i zwraca element o podanym indeksie.
7 # Zwraca skrajny prawy element,
8 # jeśli pominięto indeks (w przybliżeniu przeciwnie do append()).
```