**Typy sekwencyjne w Pythonie**

**Sekwencyjne typy danych umożliwiają zapisywanie wielu wartości w jednej zmiennej. W języku Python wyróżniamy sekwencje zmienne, którym można zmieniać wartości, oraz sekwencje niezmienne, których wartości nie można zmieniać. Do sekwencji zmiennych należą listy, a przedstawicielami sekwencji niezmiennych są krotki i napisy.**

**Listy – sekwencje zmienne**

**Lista to typ sekwencyjny zmienny, więc możliwe jest przypisywanie wartości pojedynczym elementom tego typu.**

**Do zapisywania listy wykorzystujemy nawiasy kwadratowe, a poszczególne elementy rozdzielamy przecinkami. Listy mogą zawierać wartości różnego typu**

|  |  |
| --- | --- |
| **Konstrukcja języka** | **Opis** |
| **lista1 = [0 for row in range(n)]**  **lista2 = [[0 for col in range(n)]**  **for row in range(m)]** | **Deklaracja listy jednowymiarowej lista1 i listy dwuwymiarowej lista2, przypisanie elementom list wartości początkowych 0** |
| **lista = [1.0, 2, ’’trzy”, 4.0, 5]** | **Przypisanie wartości zmiennej lista** |
| **lista[0]**  **lista[-1]**  **lista[-2]** | **Odwołanie do pierwszego elementu listy(1.0),**  **Odwołanie do ostatniego elementu listy(5),**  **Odwołanie do przedostatniego elementu listy(4.0)** |
| **lista[2:]** | **Od elementu lista[2], wynik:[‘trzy’, 4.0, 5]** |
| **lista[1::2]** | **Co drugi element od elementu lista[1], wynik: [2, 4.0]** |
| **lista[0:4:2]** | **Od elementu lista[0] do elementu lista[3] co drugi,**  **Wynik: [1.0, ‘trzy’]** |
| **Lista \*= 2** | **Wynik: [1.0, 2, ‘trzy’, 4.0, 5, 2.0, 2, ‘trzy’, 4.0, 5]** |
| **lista = lista[:2]** | **Skracanie listy, wynik: [1.0, 2]** |
| **len(lista)** | **Długość listy (liczba elementów), wynik: 5** |
| **lista[1] = 6** | **Przypisywanie elementów lista[1] wartości 6,**  **Instrukcja przypisania (elementom listy można przypisać wartości), wynik: [1.0, 6, ‘trzy’ , 4.0 , 5]** |
| **lista[0] += 7** | **Element lista[0] zwiększony o 7, wynik:[8.0 , 2 , ‘trzy’ , 4.0 , 5]** |
| **del lista[2]** | **Usunięcie elementu lista[2] z listy wynik: [1.0 , 2 , 4.0 , 5]** |
| **== != <>** | **Relacje na listach (listy można porównywać)** |
| 1. **in lista** 2. **not in lista** | **Sprawdzenie, czy na liście są określone wartości** |
| **lista[i], lista[k] = lista[k], lista[i]** | **Zamiana elementów listy** |
| **for indeks, wartosc in enumerate(lista):**  **print(indeks, “=”, wartosc** | **Wypisywanie kolejnych elementów listy: indeksów i wartości.** |

**Indeksowanie list**

|  |  |
| --- | --- |
| **Konstrukcja języka** | **Wynik** |
| **lista = [9, 2, 3, 4, 5, 6, 7, 8, 1, 10]** | |
| **lista** | **[9, 2, 3, 4, 5, 6, 7, 8, 1, 10]** |
| **lista[2:]** | **[ 3, 4, 5, 6, 7, 8, 1, 10]** |
| **lista[3::2]** | **[4, 6, 8, 10]** |
| **lista[::-1]** | **[10, 1, 8, 7, 6, 5, 4, 3, 2, 9]** |
| **lista[::-2]** | **[10, 8, 6, 4, 2, ]** |
| **lista[::3]** | **[9, 4, 7, 10]** |
| **lista[4::-2]** | **[10, 8, 6, ]** |
| **lista[4::2]** | **[9, 3]** |
| **lista[0:5:2]** | **[9, 3, 5]** |
| **lista[3:9:2]** | **[4, 6, 8]** |

**Metody przetwarzania list**

|  |  |
| --- | --- |
| **Konstrukcja języka** | **Opis** |
| **lista = [9, 2, 3, 4, 5, 6, 7, 8, 1, 10]** | |
| **lista.append(11)** | **Dołączanie elementu do listy**  **Wynik: [9, 2, 3, 4, 5, 6, 7, 8, 1, 10, 11]** |
| **lista.extend([12, 13])** | **Dołączanie listy[12, 13] do listy [9, 2, 3, 4, 5, 6, 7, 8, 1, 10]**  **Wynik: [9, 2, 3, 4, 5, 6, 7, 8, 1, 10, 12, 13]** |
| **lista.count(9)** | **Obliczanie ile razy podana wartość 9 występuje na liście**  **Wynik: 1** |
| **lista.index(8)** | **Wyznaczanie pozycji (licząc od 0) pierwszego wystąpienia podanej wartości 8**  **Wynik:7** |
| **lista.insert(3, 33)** | **Wstawianie liczby 33 do listy na podaną pozycję 3**  **Wynik: [9, 2, 3, 33, 4, 5, 6, 7, 8, 1, 10]** |
| **lista.pop(4)** | **Zwracanie wartości z podanej pozycji i usunięcie tego elementu z listy**  **Wynik: [9, 2, 3, 4, 6, 7, 8, 1, 10]** |
| **lista.remove(2)** | **Usunięcie z listy pierwszej znalezionej wartości 2**  **Wynik: [9, 3, 4, 5, 6, 7, 8, 1, 10]** |
| **lista.reverse()** | **Odwracanie kolejności elementów listy**  **Wynik [10, 1, 8, 7, 6, 5, 4, 3, 2, 9, ]** |
| **lista.sort()** | **Niemalejące sortowanie listy**  **Wynik [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]** |

**Krotki – sekwencje niezmienne**

**Krotka to typ sekwencyjny przypominający listę. Jednak w odróżnieniu od listy krotki to sekwencje niezmienne.**

**Nie możemy więc przypisać wartości pojedynczym elementom krotki.. Do zapisu tej sekwencji wykorzystujemy nawiasy okrągłe, a poszczególne elementy rozdzielamy przecinkami**

**Typ sekwencyjny - krotka**

|  |  |  |
| --- | --- | --- |
| **Konstrukcja języka** | **Opis** | |
| **krotka = (1, 2, 3)**  **krotka = 1, 2, 3** | | **Przypisanie wartości zmiennej krotka** |
| **krotka1 = (1.0, 2, „trzy”, 4.0, 5)** | **Przypisanie wartości zmiennej krotka1** | |
| **krotka \*=2** | **Wynik : (1, 2, 3, 1, 2, 3)** | |
| **~~krotka[1] = 6~~** | **Brak możliwości przypisania wartości elementom krotki** | |
| **len(krotka)** | **Długość krotki (liczba elementów)**  **Wynik: 3** | |

**Indeksowanie krotek**

|  |  |
| --- | --- |
| **Konstrukcja języka** | **Wynik** |
| **krotka = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)** | |
| **krotka** | **(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)** |
| **krotka[3:]** | **(3, 4, 5, 6, 7, 8, 9, 10)** |
| **krotka[2::3]** | **(2,5,8)** |
| **krotka[::-1]** | **(10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)** |
| **krotka[::-2]** | **(10, 8, 6, 4, 2, 0)** |
| **krotka[::2]** | **(0, 2, 4, 6, 8,10) )** |
| **krotka[4::-2]** | **(10, 8, 6)** |
| **krotka[3::2]** | **(0, 2)** |
| **krotka[0:3:2]** | **(0, 2)** |
| **krotka[3:9:3]** | **(3, 6)** |