**Списки:List**

|  |  |
| --- | --- |
| **massiv\_list= ["Telefon","Televizor"]**  **massiv\_list1= ["Komputer","Holodilnik"]** | Обявление массива |
| **massiv\_list[1]**  **massiv\_list1[1]** | Вывод нужного элемента списка |
| **massiv\_list=massiv\_list+massiv\_list1** | Слияние списков |
| **massiv\_list.append("Stiralka")** | Добавление элементов в массив |
| **massiv\_list.extend([3,4,5])** | *Добавление нескольких элементов в конец списка* |
| **massiv\_list[3]="Utug"** | Замена элемента списка |
| **massiv\_list.pop(2)** | Удаляет последний, либо по индексу элемент списка. Возвращает удаленный элемент |
| **zz.remove("Silnuy")** | Удаляет по значению |
| **massiv\_list.sort(key=len)**  result = [(1, 6.06), (2, 6.23), (3, 7.03), (4, 6.88), (5, 6.43), (6, 6.57)]  result.sort(key=lambda x:x[1],reverse=True) | Сортировка списка. reverse=True, key=len, key=str.lower(переведет все в нижний регистр и отсортирует) |
| **L = [‘abc’, ‘ABD’, ‘aBe’]**  **sorted(L, key=str.lower, reverse=True)**  **[‘aBe’, ‘ABD’, ‘abc’]** | Встроенная функция сортировки |
| **massiv\_list.reverse()** | Обратный порядок всех элементов |
| **massiv\_list.insert(2,"Avtomobil")** | Вставляет новый элемент в середину списка |
| **massiv\_list.index("Telefon",0,5)** | Возвращает индекс искомого элемента в заданном промежутке от 0 до 5 |
| **massiv\_list.count("Telefon")** | Возвращает количество искомых элементов в списке |
| **copy=massiv\_list.copy()** | Копирует список. Есть подводные камни. Смотреть справку |
| **massiv\_list.clear()** | Удаляет все элементы списка. Возвращает None |
| **list(a)** | Определяет в переменной тип данных |
| **del massiv\_list[3]**  **del L[1:]** | Удаляет по индексу элемент списка.  Либо удаляет целый срез элементов. |
| **zz = ["Krasniy", "Bolshoy","Silnuy"]**  **cvet, razmer, sila= zz**  **print(cvet)**  **print(razmer)**  **print(sila)** |  |
|  |  |
| **«, “.join([“slovo1”, “slovo2”])**  **«: “.join([“slovo1”, “slovo2”])**  **«:: “.join([“slovo1”, “slovo2”])** | Вставляет нужный разделитель между элементами списка и возвращает строку |
|  |  |
| List2= List1[:] | Создает КОПИЮ ОБЕКТА списка. |
|  |  |
|  |  |
|  |  |

**Генератор списков**

***[ выражение* for *цель!* in *итерируемый— объект!* if *условие!***

**for *цель2* in *итерируемый—объект 2* if *условие2 . . .***

**for *цель!!* in *итерируемый\_объектЫ* if *условие!! ]***

1. **List compehension**

* **Генерирует саму последовательность в списке**

a= [0 for i in range(10)]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

a= [i for i in range(10)]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

**- Генерирует комбинированные последовательности из двух циклов**

m = [(i+str(y)) for i in "abc" for y in [1,2,3]]

['a1', 'a2', 'a3', 'b1', 'b2', 'b3', 'c1', 'c2', 'c3']

m = [(i, y) for i in "abc" for y in [1,2,3]]

[('a', 1), ('a', 2), ('a', 3), ('b', 1), ('b', 2), ('b', 3), ('c', 1), ('c', 2), ('c', 3)]

**- Генерирует список по индексу каждого вложенного элемента в списке списков.**

list1 =[[1,2,3],

        [4,5,6],

        [7,8,9]]

lis\_= [element[0] for element in list1]

print(lis\_)

lis\_= [element[1] for element in list1]

print(lis\_)

lis\_= [element[2] for element in list1]

print(lis\_)

[1, 4, 7]

[2, 5, 8]

[3, 6, 9]

**- Генерирует список, обединяя в один список списков**

lis\_= [y for i in list1 for y in i]

[1, 2, 3, 4, 5, 6, 7, 8, 9]

**- Генерирует список с условием, выбирает только четные элементы**

lis\_= [y for i in list1 for y in i if y%2==0]

[2, 4, 6, 8]

**-Увеличивает на 10 все элементы сгенерированного списка (можно подставить любой математическое выражение для каждого элемента)**

lis\_= [y+10 for i in list1 for y in i]

[11, 12, 13, 14, 15, 16, 17, 18, 19]

**- Генератор списка элементов по диагоналям матрицы списка списков**

list1 =[[1,2,3],

        [4,5,6],

        [7,8,9]]

lis\_= [list1[i][i] for i in [0,1,2]]

[1, 5, 9]

lis\_= [list1[i][len(list1)-1-i] for i in [0,1,2]]

[3, 5, 7]

**- Разбивает строку на элементы и дублирует значение каждого**

doubles = [c \* 2 for c in 'spam']

['ss', 'pp', 'aa', 'mm']

**- Сумирует все элементы в списка в списке**

sum\_el = [sum(i) for i in list1]

[6, 15, 24]

**- Применяет функцию для обработки каждого элемента**

res = (ord(x) for х in 'spam']

*Таблица 8.1. Литералы списков и операции*

|  |  |
| --- | --- |
| **Операция** | **Интерпретация** |
| L = [] | Пустой список |
| L = [0, 1, 2, 3] | Четыре элемента с индексами 0..3 |
| L = [‘abc’, [‘def’, ghi’]] | Вложенные списки |
| L = list(‘spam’)  L = list(range(-4, 4)) | Создание списка из итерируемого объекта.  Создание списка из непрерывной последова­тельности целых чисел |
| L[i]  L[i][j]  L[i:j]  len(L) | Индекс, индекс индекса, срез, длина |
| L1 + L2  L \* 3 | Конкатенация, дублирование |
| for x in L: print(x)  3 in L | Обход в цикле, проверка вхождения |
| L.append(4)  L.extend([5,6,7])  L.insert(I, X) | Методы: добавление элементов в список |
| L.index(1)  L.count() | Методы: поиск |
| L.sort()  L.reverse() | Методы: сортировка, изменение порядка сле­дования элементов на обратный |
| del L[k]  del L[i:j]  L.pop()  L.remove(2)  L[i:j] = [] | Уменьшение списка |
| L[i] = 1  L[i:j] = [4,5,6] | Присваивание по индексу, присваивание срезу |
| L = [x\*\*2 for x in range(5)]  list(map(ord, ‘spam’)) | Генераторы списков и отображение (главы 14 и 20) |