## 2.3. Листинги

Списки и кортежи

|  |  |
| --- | --- |
| s1 = [1, 2, 3, 4, 5]  print(s1[0]) # Выведет 1  s3 = ['a', 'b', 'c', 'w', 'z']  print(s3[3]) # Выведет w  s6 = [[1, 2, 3], ['apple', 'banana']]  print(s6[0]) # Выведет [1, 2, 3]  print(s6[0][0]) # Выведет 1 | индексы |
| s1 = [1, 2, 3, 4, 5]  s1.append(6)  print(s1) # Выведет [1, 2, 3, 4, 5, 6]  s3 = ['a', 'b', 'c']  s3.append('f')  print(s3) # Выведет ['a', 'b', 'c', 'f'] |  |
| s1 = [2, 4, 8, 16, 32]  s1.insert(1, 10)  print(s1) # Выведет [2, 10, 4, 8, 16, 32]  s2 = [2, 4, 8, 16, 32]  s2.insert(2, [10, 20])  print(s2)  # Выведет [2, 4, [10, 20], 8, 16, 32] |  |
| s1 = [1, 2, 3]  s2 = [4, 5, 6]  s1.extend(s2)  print(s1) # Вывод: [1, 2, 3, 4, 5, 6] |  |
| s1 = [1, 2, 3]  s2 = [4, 5, 6]  print(s1 + s2) | Конкатенация списков |
| k1 = (1, 2, 3)  k2 = (4, 5, 6)  k3 = s1 + s2  print(k3) | Конкатенация кортежей |
| s1 = [1, 2, 3]  print(s1 \* 2)  # Выведет [1, 2, 3, 1, 2, 3] | Умножение списков |
| k1 = (1, 2, 3)  k2 = k1 \* 3  print(k2) | Умножение кортежей |
| s1 = [1, 2, 3, 4, 5]  s1.remove(1)  print(s1) # Выведет [2, 3, 4, 5] | Удаление |
| s2 = ['apple', 'banana', 'orange']  s2.remove('banana')  print(s2) # Выведет ['apple', 'orange'] | Удаление |
| s1 = [1, 2, 3, 4, 5]  s1[2] = 8  print(s1) |  |
| s2 = ['a', 'b', 'f']  s2[0] = 'l'  print(s2) |  |
| s1 = [10, 20, 30, 40, 50]  print(s1[1:3]) # Выведет [20, 30] | Срезы со списком |
| s2 = ['a', 33, 'b', 45, 'f']  print(s2[:3]) # Выведет ['a', 33, 'b'] | Срезы со списком |
| s3 = ['w', 38, 'z', 49, 'f']  print(s3[-2:]) # Выведет [49, 'f'] | Срезы со списком |
| k1 = (10, 20, 30, 40, 50)  print(k1[1:3]) # Выведет (20, 30) | Срезы с кортежами |
| k1 = (10, 20, 30, 40, 50)  print(k1[-2:]) # Выведет (40, 50) | Срезы с кортежами |
| s1 = [2, 4, 8, 16, 32]  s1[:2] = [22, 44]  print(s1) # Выведет [22, 44, 8, 16, 32] |  |
| s2 = [12, 33, 'w', 4, 'c']  s2[-2:] = ['t2', 'q2']  print(s2)  # Выведет [12, 33, 'w', 't2', 'q2'] |  |
| s1 = [2, 4, 8, 16, 32]  s1[1:1] = [5, 15]  print(s1) # Выведет [2, 5, 15, 4, 8, 16, 32] |  |
| s2 = [8, 1, 'w', 4, 2]  s2[-2:-2] = ['t', 'q']  print(s2)  # Выведет [8, 1, 'w', 't', 'q', 4, 2] |  |
| s1 = [2, 4, 8, 16, 32, 64, 128]  print(s1[::2]) # Выведет [2, 8, 32, 128] |  |
| s2 = [8, 1, 'w', 't', 'q', 4, 2, 'c', 0]  print(s2[2::3]) # Выведет ['w', 4, 0] |  |
| s2 = [8, 1, 'w', 't', 'q', 4, 2, 'c', 0]  print(s2[-5::-1])  # Выведет ['q', 't', 'w', 1, 8] |  |
| s1 = [2, 4, 8, 16, 32, 64]  print(s1[::-1]) # Выведет [64, 32, 16, 8, 4, 2] | Инвертирование списка |
| s1 = [2, 4, 8, 16, 32, 64]  s1.reverse()  print(s1)  # Выведет [64, 32, 16, 8, 4, 2] | Инвертирование списка |
| s1 = [1, 2, 3, 4]  s2 = s1.reverse()  print(s2) # Выведет: None  print(s1) # Выведет: [4, 3, 2, 1] | Инвертирование списка |
| k1 = (10, 20, 30, 40, 50)  k2 = k1[::-1]  print(k2) | Инвертирование кортежа |
| s1 = [2, 64, 8, 4, 32, 16]  s1.sort()  print(s1) # Выведет [2, 4, 8, 16, 32, 64] | Сортировка списка |
| s1 = [3, 2, 1]  s2 = s1.sort()  print(s1) # Выведет [1, 2, 3]  print(s2) # Выведет None |  |
| s1 = ['в', 'б', 'а']  s1.sort()  print(s1) # Выведет ['а', 'б', 'в'] |  |
| s1 = ['машу', 'мама', 'мыла', 'мылом']  s1.sort()  print(s1)  # ['мама', 'машу', 'мыла', 'мылом'] |  |
| s1 = "Буря мглою небо кроет"  s2 = s1.split() #По умолчанию разделитель - пробел  print(s2)  # Выведет ['Буря', 'мглою', 'небо', 'кроет'] | Превращение строки в список слов |
| s1 = "Щи, каша, репа, горох"  s2 = s1.split(", ")  print(s2)  # Выведет ['Щи', 'каша', 'репа', 'горох'] | Превращение строки в список слов |
| s1 = "Вихри снежные крутя"  s2 = list(s1)  print(s2)  # Выведет ['В', 'и', 'х', 'р', 'и', ' ', 'с', 'н', 'е', 'ж', 'н', 'ы', 'е', ' ', 'к', 'р', 'у', 'т', 'я'] | Превращение строки в список символов |
| k1 = "Вихри снежные крутя"  k2 = tuple(k1)  print(k2) | Превращение строки в кортеж из символов |
| pokupki = ['молоко', 'сметана', 'сыр', 'хлеб']  pokupki.clear()  print(pokupki) | Очистка списка |
| k = (1, 2, 1, 3, 1, 4)  print(k.count(1))  # Выведет 3 | Подсчет количества |
| k = (1, 2, 1, 3, 1, 4)  print(k.index(1)) | Поиск элемента |
| k = (1, 2, 1, 3, 1, 4)  print(k.index(1,3))  # Первое число - которое ищем, второе число - с какого индекса начинаем поиск. |  |
| k = (1, 2, 3, 4, 5)  a = 4  if a in k:  i = k.index(a)  print(f"Индекс {a} в кортеже: {i}")  else:  print(f"Элемент {a} не найден в кортеже.") |  |
| s1 = [2, 64, 80, 4, 32, 16]  print(max(s1)) | Поиск максимального числа в списке |
| k = (2, 64, 80, 4, 32, 160)  print(max(k)) | Поиск максимального в кортеже |
| s1 = ['Буря', 'мглою', 'небо', 'кроет']  print(max(s1)) | Поиск максимального среди текстовых строк (ближайшего к концу алфавита) |
| s1 = ['Буря', 'мглою', 'небо', 'кроет']  print(min(s1)) | Поиск минимального среди текстовых строк (ближайшего к началу алфавита) |
| s = [-2, 64, 80, 4, 32, 16]  print(min(s)) | Поиск минимального числа |
| url = "https://example.com/path/to/file.txt"  f = url.split("/")[-1]  print(f"Имя файла из URL: {f}") | Извлекаем имя файла из URL. |
| path = 'C:\Python\Lesson\June\_2019\Lesson\_05.txt'  f = path.split("\\")[-1]  print("Имя файла из пути: {f}") | Извлекаем имя файла из пути. |
| s1 = [1, 2, 3, 4, 5]  for i in s1:  print(i \* 10) | Перебор списка в цикле |
| k = (10, 20, 30, 40, 50)  for i in k:  print(i / 2) | Перебор кортежа в цикле |
| s = ['Буря', 'мглою', 'небо', 'кроет']  for i in s:  print(i) | for со списком |
| s = ['Буря', 'мглою', 'небо', 'кроет']  for i in s:  if i == 'небо':  continue  print(i) | continue |
| s = ['Буря', 'мглою', 'небо', 'кроет']  i = 0  while i < len(s):  print(s[i])  i += 1 | while со списком |
| s = ['Буря', 'мглою', 'небо', 'кроет']  i = 0  while i < len(s):  if s[i] == 'мглою':  i += 1  continue  print(s[i])  i += 1 | while с оператором continue |

|  |  |
| --- | --- |
| print('00000000011111111112222222222')  print('12345678901234567890123456789')  print('\*\*\*\t\*\*\*\t\*\*\*\t\*\*\*')  print('100\t100500\t1000\t10') | Табуляция |
| print('00000000011111111112222222222')  print('12345678901234567890123456789')  print('Winter\tSpring\tSummer\tAutumn')  print('Зима\tВесна\tЛето\tОсень')  Все числа у нас однозначные, поэтому табличка получилась ровная. | Табуляция |
| import random  print("-" \* 13)  for i in range(10):  a = str(random.randint(0,9))  b = str(random.randint(0,9))  c = str(random.randint(0,9))  print(f"| {a} | {b} | {c} |")  print("-" \* 13) | Табличка со случайными числами от 0 до 9 |
| import random  print("-" \* 18)  for i in range(10):  a = str(random.randint(0,99))  b = str(random.randint(0,999))  c = str(random.randint(0,9999))  print(f"| {a} | {b} | {c} |")  print("-" \* 18) | Табличка со случайными числами от 0 до 10000 с пробелами и без \t |
| import random  print("-" \* 25)  for i in range(10):  a = str(random.randint(0,99))  b = str(random.randint(0,999))  c = str(random.randint(0,9999))  print(f"| {a} \t| {b} \t| {c} \t|")  print("-" \* 25) | Табличка со случайными числами от 0 до 10000 с табуляцией \t. |
| import random  print("-" \* 25)  for i in range(10):  a = str(random.randint(0,99))  b = str(random.randint(0,999))  c = str(random.randint(0,9999))  print(f"| {a:>2} \t| {b:>3} \t| {c:>4} \t|")  print("-" \* 25) | Выравнивание чисел по правому краю |
| import random  print("-" \* 25)  for i in range(10):  a = str(random.randint(0,99))  b = str(random.randint(0,999))  c = str(random.randint(0,9999))  print(f"| {a:>4} \t| {b:>4} \t| {c:>4} \t|")  print("-" \* 25) | Выравнивание чисел по правому краю с отступом в 2 пробела от вертикальной черты |