Самостоятельная работа 2

Контрольные вопросы:

1. Как изменить значение элемента списка?

list[index] = value

**Ответ**: для начала создаем список, затем создаем переменную ‘index’, которая будет выводить индекс указанного элемента в списке. Далее создаем переменную ‘value’, в которой указываем новое значение. Обращаемся к списку и в квадратных скобочках указываем переменную, которая выводит индекс элемента, значение которого мы хотим заменить. Затем знак равно и переменную с новым значением.

list1 = [12, 45, 78, 9]

index = list1.index(78)

value = int(input('Введите значение: ')) # Введите значение: 111

list1[index] = value

print(list1) # Результат программы: [12, 45, 111, 9]

1. Что выведет данный код

a = [1, 3]

print(int(a))

**Ответ**: программа выведет ошибку. Поскольку в тип данных ‘int’ нельзя преобразовывать списки, кортежи, множества, словари. Но можно преобразовать строку, содержащую целочисленное значение, или значение с типом данных ‘float’.

1. Что выведет данный код

a = str([1, 1])

print(a)

**Ответ**: программа выведет список [1, 1]. Ошибки не будет, т.к. аргументом функции str() может выступать число, список, кортеж, множество, словарь. Но и строки при выводе не будет, т.к. список в строку можно перевести либо с помощью цикла, либо с помощью метода join(), но перед этим нужно преобразовать int в str.

1. Что выведет данный код

a = [2, 4, 8]

print(a[::-2])

**Ответ**: [8, 2]

1. Как объединить два списка?

**Ответ**: с помощью операции сложения или с помощью метода extend(). Разница в том, что extend() не вернет новый список, как это делает операция сложения. Метод extend() производит операцию над каким-то списком (т.е. мы не создаем новый список, а расширяем уже имеющийся).

1. Как отсортировать список в порядке убывания?

**Ответ**: sort(reverse=True)

1. Как объединить два кортежа?

**Ответ**: с помощью операции сложения.

1. Можно ли умножать кортеж на число?

**Ответ**: умножать кортеж на целое число можно.

1. Что выведет данный код

a = list(‘py’)

print(len(a))

**Ответ**: 2, т.к. каждый элемент строки станет отдельным элементом списка.

1. Что выведет данный код

a = [2, 4, 8]

print(a[::4])

**Ответ**: [2]

1. Как добавить элемент в словарь?

dict[key] = value

update({key:value, key2:value})

**Ответ**: с помощью функции update() либо с помощью dict[key] = value

dict1 = {'a': 12, 'b': 23, 'c': 1}

key = input('Введите ключ: ')

key2 = input('Введите ключ: ')

value = input('Введите значение: ')

dict1.update({key: value, key2: value})

print(dict1)

dict1 = {'a': 12, 'b': 23, 'c': 1}

dict1['new'] = 18

print(dict1)

1. Как получить значение элемента в словаре?

print(dict[key])

**Ответ**: print(dict[key]), print(dict.get(key))

1. Как удалить элемент из словаря?

pop(key)

del dict[key]

**Ответ**: pop(key), del dict[key]

1. Что выведет данный код

a = [1, 2]

b = (1, 2)

print(a == b)

**Ответ**: False, т.к. нельзя сравнивать разные типы данных

1. Что выведет данный код

a = []

for i in range(3): [] +[0], [0] + [1], [0,1,] + [2]

a += [i]

print(a)

**Ответ**: [0, 1, 2]

1. Что такое исключение (exception) в Python?

**Ответ**: исключение – тип данных, необходимый для того, чтобы сообщать программисту об ошибках.

1. Как обработать несколько разных исключений?

**Ответ**: несколько раз использовать блок except или в блок except поместить несколько исключений.

try:  
 new\_dct = {'age': 11, 'height': 134}

new\_list = [12, 34, 6]

print(new\_dct + new\_list)

except KeyError:

print('Обработано исключение (KeyError)'

except TypeError:

print('Обработано исключение (TypeError)')

try:  
 new\_dct = {'age': 11, 'height': 134}

new\_list = [12, 34, 6]

print(new\_dct + new\_list)

except (KeyError, TypeError):

print('Обработано исключение (KeyError или TypeError)')

1. Как использовать блок finally при обработке исключений?

**Ответ**: после блока exception добавляем оператор ‘finally’. Этот блок выполняется в любом случае, независимо от того, было исключение или нет.

try:  
 new\_dct = {'age': 11, 'height': 134}

new\_list = [12, 34, 6]

print(new\_dct + new\_list)

except (KeyError, TypeError):

print('Обработано исключение (KeyError или TypeError)')

finally:  
 print('Программа выполнена')

1. Как открыть файл для чтения в Python?

**Ответ**:

with open('task1.txt', 'r') as file:

print(file.read())

file = open('new\_task1.txt, 'r') # можно без режима ‘r’

print(\*file)  
file.close()

1. Как прочитать содержимое файла?

with open(‘file\_name.txt’) as file:

file.read(5) - hello

file.read(6) world ….

file.readline()

file.readline() …

file.readlines()

**Ответ**: чтобы вывести всю информацию из файла, нужно воспользоваться методом file.readlines().

1. file.read(5) – hello – результатом программы будут символы соответствующие индексам от 0 до 4.

2. file.read(6) world …. – результатом программы будут символы соответствующие индексам от 0 до 5.

3. file.readline() – программа выведет информацию построчно, в данном случае выведет первую строку.

4. file.readline() … – в данном случае программа выведет вторую строку.

5. file.readlines() – программа выведет содержимое файла как элементы списка.

1. Как закрыть файл после работы с ним?

**Ответ**: с помощью метода close()

file = open('new\_task1.txt') # можно без режима ‘r’

print(\*file)  
file.close()

Если мы открываем файл с помощью оператора with, то файл закроется автоматически.

with open('task1.txt', 'r') as file:

print(file.read())

1. Что такое CSV файл?

**Ответ**: CSV – один из популярных форматов данных для работы с таблицами.

1. Как записать данные в CSV файл?

**Ответ**:

import csv

name = 'Petya'

name2 = 'Misha'

with open('examples1.csv', 'w') as file:

writer = csv.writer(file, delimiter=';')

writer.writerow([name, name2])

import csv

name = 'Petya'

with open('examples1.csv', 'w') as file:

writer = csv.writer(file, delimiter=';')

writer.writerow(name)

1. Что выведет данный код

a = [1, 2]

print(a[len(a)])

**Ответ**: программа выведет ошибку.

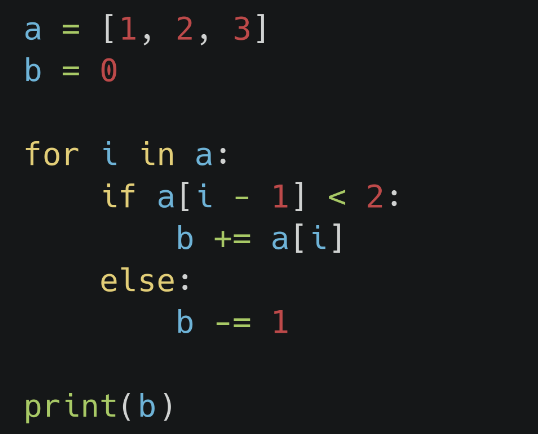
1. Что выведет данный код

a = (1, 2) + 2

print(a)

**Ответ**: программа выведет ошибку, поскольку мы не можем к кортежу прибавлять число. Кортеж – это неизменяемый тип данных.

1. Что выведет данный код



**Ответ**: итак, сначала мы берем первый индекс [i] = 0, a[0 - 1] = a [-1] = 3 (‘a’ c индексом -1 равно значению 3), 3 больше 2, поэтому в данном случае сработает блок ‘else’, b будет равно -1. Далее берем второй индекс [i] = 1, a[1 - 1] = a [0] = 1 (‘a’ c индексом 0 равно значению 1), 1 меньше 2, поэтому в данном случае сработает блок ‘if’ b += a[i], a[1] = 2, тогда b = 2.

Затем берем третий индекс [i] = 2, a[2 - 1] = a [1] = 2 (‘a’ c индексом 1 равно значению 2), 2 не меньше 2, поэтому в данном случае снова сработает блок ‘else’, b будет равно -1.

b = -1 + 2 + (-1) = 0

Результат программы будет равен 0.

1. Что выведет данный код

a = (1, 3)

b = (0, 100)

print(a > b)

**Ответ**: True, т.к. сравнение кортежей происходит последовательно элемент за элементом, а если элементы равны, то мы переходим к следующим элементам. В нашем случае 1 > 0.

1. Что выведет данный код

var = ‘{0}{0}’.format(1, 4)

print(int(var))

**Ответ**: программа выведет число 11. Здесь используется метод форматирования с повторяющимся индексом. На место {0} подставится число 1, затем снова идет {0}, и снова подставляется 1. Получилось var = 11 (тип данных - строка), а далее str преобразуется в int.

1. Что выведет данный код

lst = [1]

lst.extend(‘hi’)

lst.append(‘hey’)  
 print(lst)

**Ответ**: [1, 'h', 'i', 'hey']. Т.к. метод extend() посимвольно разобьет строку, append() добавит элемент в конец списка.

1. Что выведет данный код

string = ‘abcdefghijklmnopqrst’

print(max(string), min(string))

**Ответ**: max(string) будет равно ‘t’ – последний символ строки,

min(string) – первый символ строки – ‘a’.