**1. Базовые операции в языке Python**

print("5 + 2 =", 7, ".", sep="", end="!\n")

- sep="" – разделитель между элементами вывода;

- end="!" – конец для строки без перевода на новую;

- \n – перевод на новую строку;

- \" – указание как обычный символ;

- \t – табуляция;

print(max/min(1, 2, 100, 55, 276, 34, -115)) – макс/мин число;  
print(abs(-5)) – число по модулю(5);  
print(pow(5, 2)) – возведение числа в степень;  
print(round(5 / 3)) – округление к ближайшему целому;

- 5 // 3 - округление к целому после деления

- 5 \*\* 2 - возведение в квадрат;

word = input("Введите свой возраст: ") – введение данных от пользователя;

ord('A') - возвращает целое число, представляющее Unicode-символ;

-----------------------------------------------------------------------------

marks = [65, 71, 68, 74, 61]  
  
total\_marks = sum(marks) – сумма элементов;  
print(total\_marks)

-----------------------------------------------------------------------------

number = 9   
square\_number = eval('number \* number')- eval() выполняет операцию умножения

a = 5.56  
print(math.ceil/floor(a)) – округление к большему/меньшему.

**2. Переменный и типы данных**

number = 5

del number – удаление переменной;

-----------------------------------------------------------------------------

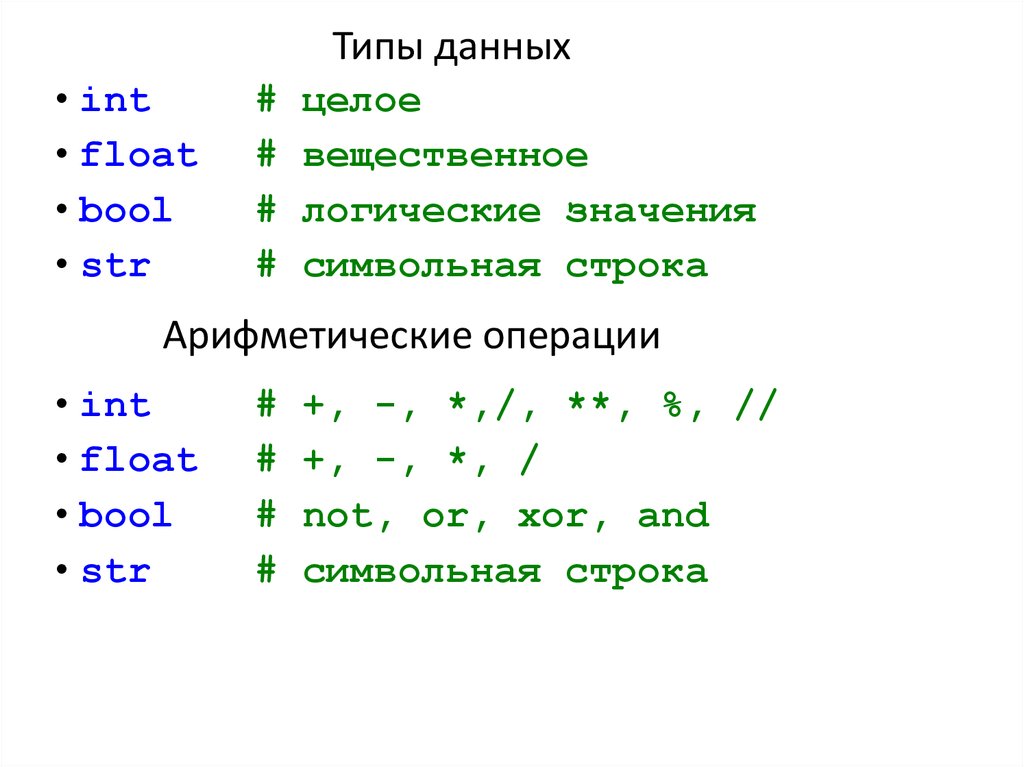
digit = 10.2  
word = "Сумма = "  
print(word + str(digit)) – приведение типа переменной int к string;

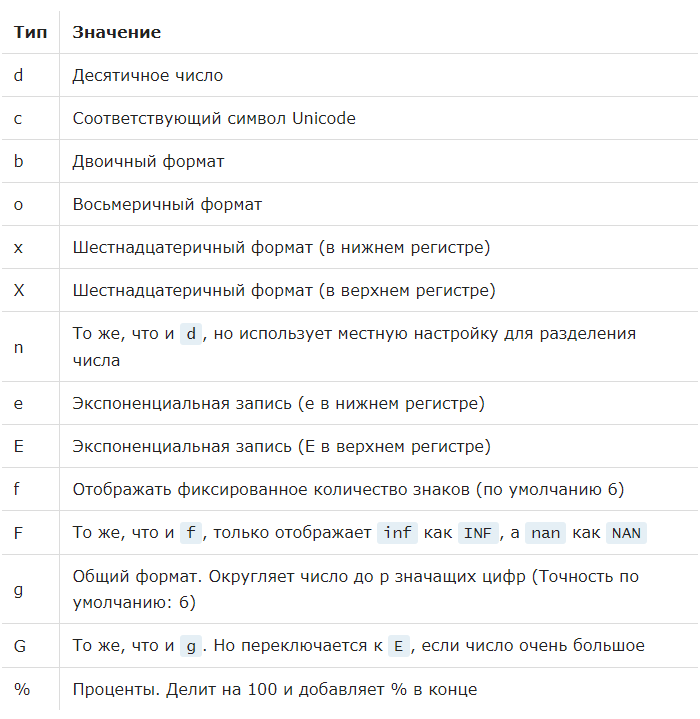
-----------------------------------------------------------------------------

word = "hi"  
print(word \* 2) – повторение слова 2 раза.

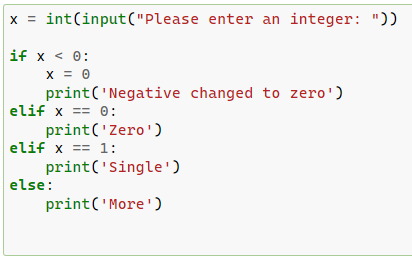
-----------------------------------------------------------------------------

value = 45  
  
binary\_value = format(value, 'b') – форматируем целое в двоичное.  
print(binary\_value)

****

****

**3. Условные операторы**

****

booleanTrue = False  
if not booleanTrue: - если не истина;  
 print("Boolean is false")

if booleanTrue: - если истина;  
 print("Boolean is false")

- and/or – и то и то/либо то, либо то.

**4. Циклы и операторы в них (for, while)**

for i in range(6): - диапазон до 6(с 0 до 5);

for i in range(1, 6): - диапазон с 1 до 6 не включительно;

for i in range(1, 6, 2): - диапазон 1-6(5) с увеличением на 2;

for i in word: цикл с перебором переменной word;

-----------------------------------------------------------------------------

i = 1  
while i < 15: - пока условие будет истинным;

for i in range(1, 11):  
 if i == 5:  
 break – выход из цикла после условия;

if i % 2 == 0:  
 continue – пропуск результата условия, и возврат в начало цикла;  
 print(i)

-----------------------------------------------------------------------------

def find\_needle(haystack):   
 if 'needle' in haystack: - если в списке есть элемент.

****

**5. Списки (list). Функции и их методы**

- его элементы можно переопределять;

- в него можно добавлять новые элементы;

- из него можно удалять элементы;

- в качестве ключей идут индексы (числа - 0, 1, 2...).

list = ["Андрей", "Иван", "Василий", "Петро", "Максим", "Дима"]

nums = [1, 2, 3, 4, 5, "Hello", [1.1, 1.2, 1.3]] – список в списке;  
nums[0] = 10 – переопределение 1 элемента списка;  
print(nums, nums[5], nums[-1], nums[-1][2]))

– [-1] – последний элемент из списка;

- [-1][1] – второй элемент из последнего элемента списка.

-----------------------------------------------------------------------------

numbers = [1, 5, 10, 15]  
numbers.append(100) – добавление числа 100 в конец списка;

numbers.insert(1, 50) – в элемент с индексом 1 вставить значение 50 со смещением заменяемого элемента;

numbers.extend([15, 20, 25]) – добавление группы элементов в конец списка;

numbers.sort() – сортировка от мин – макс. False = 0/True = 1;

numbers.reverse() – переворачивает список;

numbers.pop()/(2) – удаление последнего элемента/элемента с индексом 2;

numbers.remove(10) – удаление элемента с определённым значением;

numbers.clear() – очистка всего списка;

print(numbers.count(15)) – количество одинаковых значений элементов;

return math.prod(arr) – умножение всех элементов списка;

print(len(numbers)) – размер списка;

new\_list = list(range(1, 10, 2/-1)) – создание списка в диапазоне с шагом 2/ и -1 в порядке убывания;

-----------------------------------------------------------------------------

word = 'nikita, roma, sasha'

hobby = word.split(', ') – возвращает список, которые разделены запятой в строке, и разделяет символом в скобках;  
  
for i in range(len(hobby)):  
 hobby[i] = hobby[i].capitalize()  
  
result = ", ".join(hobby) – объединяет элементы списка в строку с разделителем в скобках;  
print(result)

-----------------------------------------------------------------------------

lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] – генератор списка;  
a = [i for i in lst if i > 0]  
print(a)

-----------------------------------------------------------------------------

nums = [1, 3, 5, 7, "9", False] – вывод всех элементов списка;  
for el in nums:  
 print(el)

-----------------------------------------------------------------------------

n = int(input("Enter length: ")) – создание пользовательского списка.  
user\_list = []  
i = 0  
while i < n:  
 string = "Enter element №" + str(i + 1) + ": "  
 user\_list.append(input(string))  
 i += 1  
print(user\_list)

**6. Функции строк. Индексы и срезы**

- Исключительно для типа str.

word = 'nikiTa, rOma, sasHa'

print(len(word)) – длинна строки;

new\_x = '' – создание пустой строки, new\_x += '0' – добавление символа в строку;

print(word.count('a'))/count(True) – количество одинаковых элементов строки;

print(word.upper()) – переводит всю строку в верхний регистр;  
print(word.lower()) – переводит всю строку в нижний регистр;  
print(word.islower()) - если строка в нижнем регистре – True, если в верхнем – False;  
print(word.isupper()) – если строка в нижнем регистре – False, если в верхнем – True;

print(word.capitalize()) – переводит первый элемент строки в верхний, остальные – в нижний;  
print(word.find('k')) – поиск символа в строкe, возвращает индекс символа;  
print(word.split(', ')) – возвращает список, которые разделены запятой в строке;

return x.replace(" ", "", 1) – заменяет символы/удаляет пробелы, заменяет только 1 обнаруженный символ;

return "{}".format(5) – перевод числа в строку;

indx = haystack.index('needle') – возвращает индекс элемента в списке;

-----------------------------------------------------------------------------

n = "Hello"  
print(list(reversed(n))) – переворачивает строку и выводит списком;

-----------------------------------------------------------------------------

a = 'abc12345rttyy'  
print(a.translate({ord('t'): None})) – удаляет/заменяет t;

-----------------------------------------------------------------------------

word = 'Football'

print(word[0:]) – вывод элемента с 0 индекса до конца строки;  
print(word[0: 4]) – вывод элементов с 0 до 4 индекса;  
print(word[1: -1: 2]) – вывод элементов с 1 до предпоследнего индекса с шагом 2(прыгает через 1 символ);  
print(word [::]) – вывод всей строки;

print(lis[::-2]) – список перевернётся и вывод элементов через 1 символ;

-----------------------------------------------------------------------------

words = "Начинается предложение"  
if words.startswith('Начин'): - проверяет начало строки на символы в скобках.

**7. Кортежи(tuple)**

**-** Кортежи нельзя изменять после их создания, а также они занимают чуть меньше памяти, нежели списки;  
- Если обычная передача информации пользователю без необходимости её изменения – использовать кортежи.

data = (1, 3, 5, 7, 9)/ data = 1, 5, 7, 9/ data = (5,)/ data = 5, – в круглых скобках, либо без;  
print(data[1:3]) – вывод элементов кортежа в диапазоне индексов;  
print(data.count(7)) – количество элементов в кортеже;  
print(len(data)) – подсчёт длины;

-----------------------------------------------------------------------------

nums = [1, 3, 5, 7]  
new\_nums = tuple(nums) – преобразование списка в кортеж;  
word = tuple('Hello World!')  
print(new\_nums)  
print(word)

**8. Словари(dict)**

new\_dict = zip(key\_list, value\_list)- создание словаря из двух списков;

**11. Работа с файлами**

file = open('data/text.txt', 'w');

- создание файла автоматически в папке “data”;  
- ‘w’ – режим открытия файла;

file.close() – закрытие файла;

file.write('Hello') – запись в файл;

file.read()- чтение из файла(в скобках указать кол-во выводимых символов);

-----------------------------------------------------------------------------

for line in file:  
 print(line)

- построчный вывод текста из файла.

