

РУТНОН

input() ВВОД
print() ВЫВОД
= присваивание

[Списки]

list() приведение к списку
listing[index] получение элемента по индексу
.append(что) добавить в конец
.insert(куда, что) вставить
len() длина
+ объединение
[от:до:шаг] нарезка
for i in listing: цикл
map(функция, список) цикл map()
filter(функция, список) цикл с фильтрацией
что in где проверка содержания
range(от, до, шаг) диапазон

Булев тип

True и False булевы литералы
bool() приведение к булеву
== сравнение
!= не равно
>, <, >=, <= больше, меньше...
and и
or или
if если
else иначе
elif ...а иначе
a = 'a=b' if a==b else 'a!=b' тернарный оператор

Циклы

while пока...
for i in l: для каждого i в...
рекурсия самовывоз функции
break остановить цикл совсем
continue прекратить текущую итерацию

Числа

int() приведение к целому
float() приведение к вещественному
+= инкремент
-= декремент

Организация кода

def название(аргументы): декларация функции
lambda аргументы: результат лямбда-функция
pass пропустить
a, b = ['Николай', 'Семенович'] реструктурирующее присваивание
a, b = b, a перевернуть
try: экскерт: обработка исключений
class Имя_класса: декларация класса
class Имя_класса(Родитель): ...с наследованием
def __init__(self): декларация конструктора
def название(self): декларация метода
объект = Класс() создание объекта
объект = Класс(аргументы) ...с передачей аргументов в конструктор
self ссылка на объект, сделанный из этого класса

'Строки'

str() приведение к строке
f'{a + b}' это f-строка f-строка
'%s и %.2f' % ('S-строка', 3.141592) s-строка
.index() позиция подстроки
.count() количество подстрок
.startswith() начинается на
.endswith() заканчивается на
len() длина
+ конкатенация
string[index] получение одной буквы
[от:до:шаг] нарезка
.replace('шило', 'мыло') замена

{Словари}

dict() приведение к словарю
d[index] получение по индексу
.items() итератор для цикла
del удалить
push(что) вытолкать

Множества

set() приведение к множеству
.intersection() пересечение
.symmetric_difference() симметричная разница
.difference() разница
.union() объединение