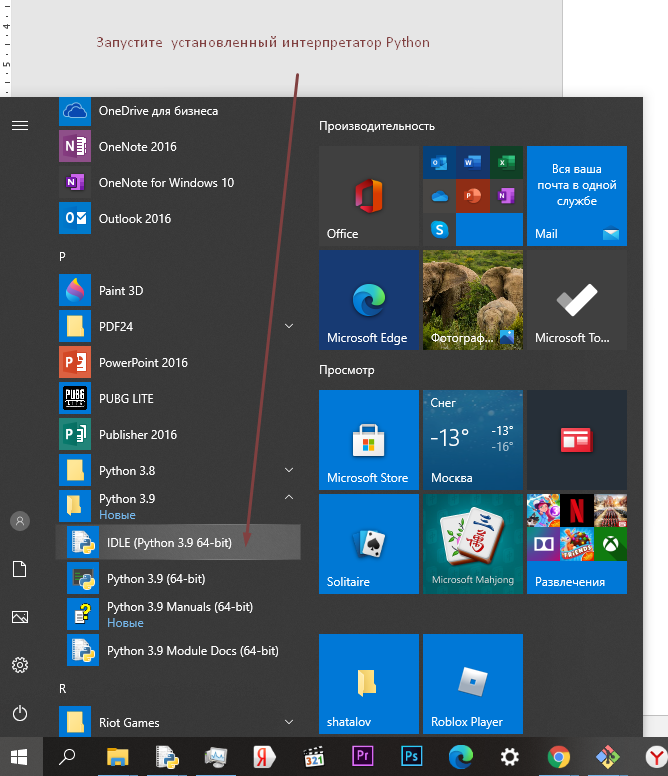
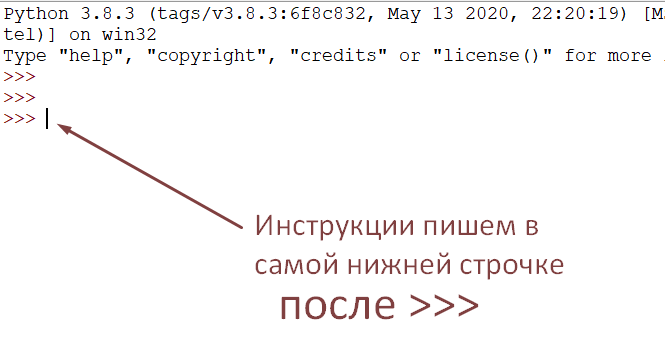
После установки Python нажмите на меню Пуск в левом нижнем углу экрана и найдите папку с установленным Питоном(это безопасно, питон не кусается). Нажмите на пункт IDLE для вызова программы.



Откроется новое окно:



1.Основные типы данных в Питоне:

int - целое число

str - строка любых символов

float - вещественное число(число с точкой)

list - список

2. тип переменной определяется данными, которые ей были присвоены

Если переменной а мы присвоили значение 12

a = 12

,то тип переменной а ( type(a) ) будет int(целое число)



Если переменной b присвоить число 1.12

b = 1.12

,то тип переменной b ( type(b) ) будет float(число с точкой)

3.Строка является последовательностью любых символов любого размера. Перед началом и после окончания строки должны быть поставлены кавычки.

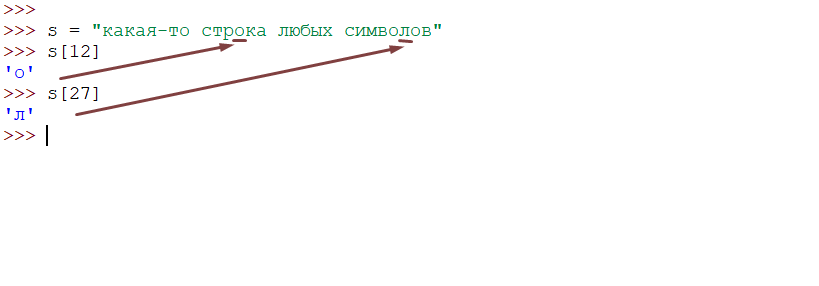
Каждый элемент строки имеет свой порядковый номер. Номера начинаются с нуля и продолжаются увеличиваясь на 1 до конца строки.

Присвоим преременной s значение “какая-то строка любых символов”

s = “какая-то строка любых символов”

Данная строка состоит из тридцати символов, включая пробелы, которые тоже являются символами строки и имеют свой номер в последовательности. Нумерация символов в строке s начинается с 0(нуля) и заканчивается номером 29.

Мы можем получить любой символ так:



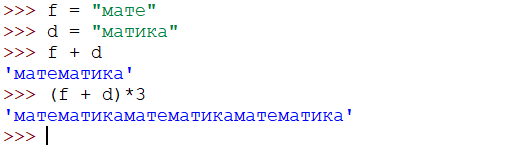
Строки похожи на поезд, вагоны которого пронумерованы от нуля и в каждом вагоне которого находится один символ

4.Между переменными и значениями определены операции.

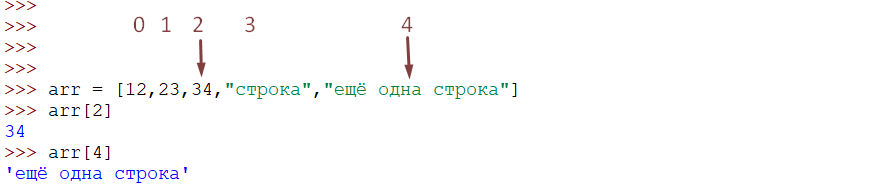
Числа можно сложить, перемножить, отнять одно от другого, возвести в степень и тд

Строки тоже можно сложить, умножить строку на число



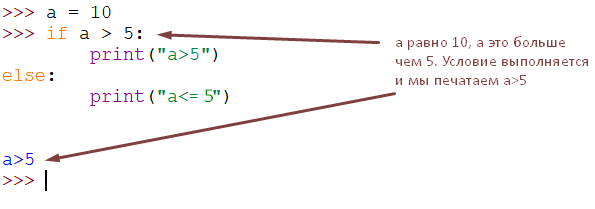


5. Списки. Списки тоже похожи на поезд, но несколько другой конструкции чем в строках. Начинаются и заканчиваются они квадратными скобками, вагоны разделены запятыми, а в каждом вагоне находится число или строка.

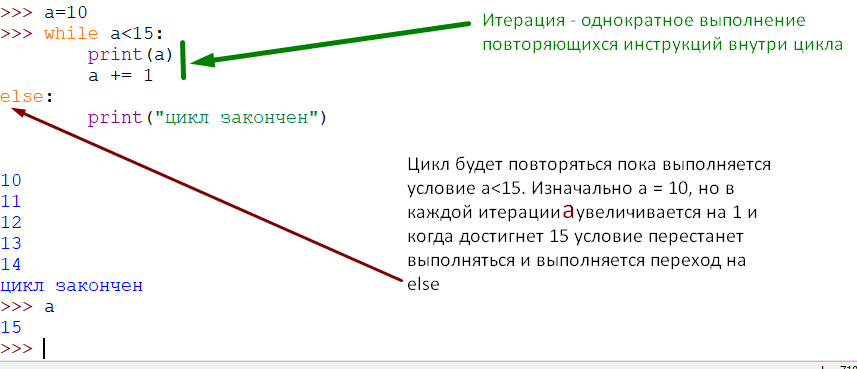


6. Алгоритмические структуры

А) Ветвление

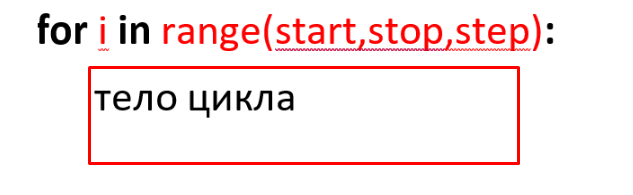


Б) Повторение



7. Цикл for

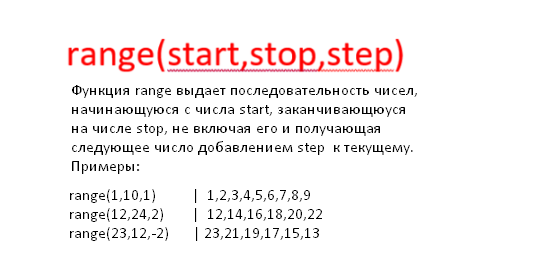
Общий вид:



Черным цветом выделена неизменяемая часть. Верхняя строчка, начинающаяся словом «for» может быть прочитана как:

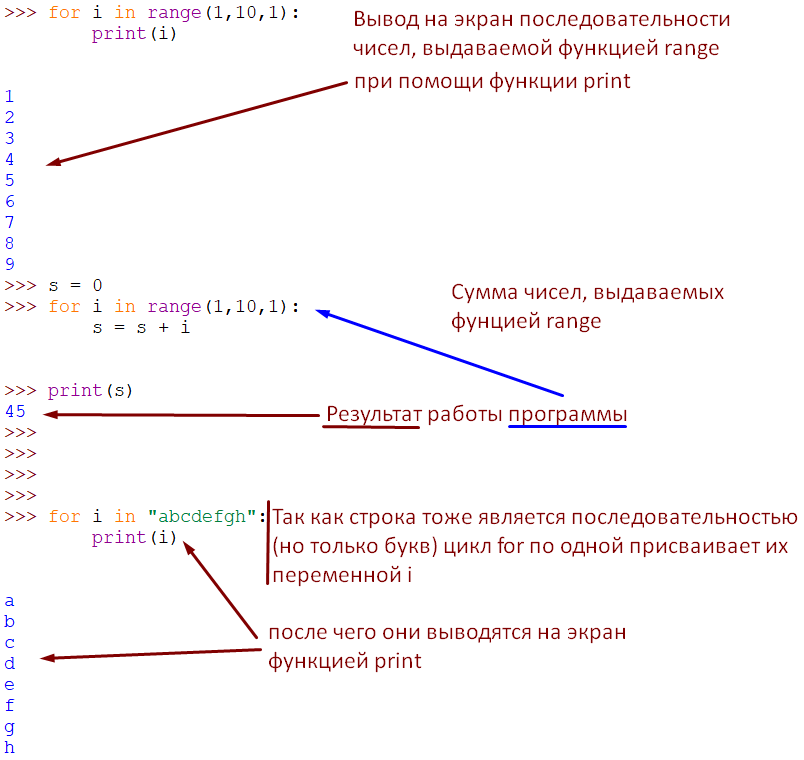
***Для всех i принимающих все значения из последовательности, определённой при помощи функции range, выполнить все команды из тела цикла.***

Тело цикла – последовательность команд(присвоить, напечатать и др), выполняемая столько раз, сколько чисел выдаст функция range.



Итерация цикла – одно выполнение команд из тела цикла.

Примеры использования цикла for



8. Знакомство с роботом

Robot – программа, которая позволит нам на практике, посредством управления ходящим и закрашивающим роботом, увидеть как устроены и работают алгоритмические конструкции в python.

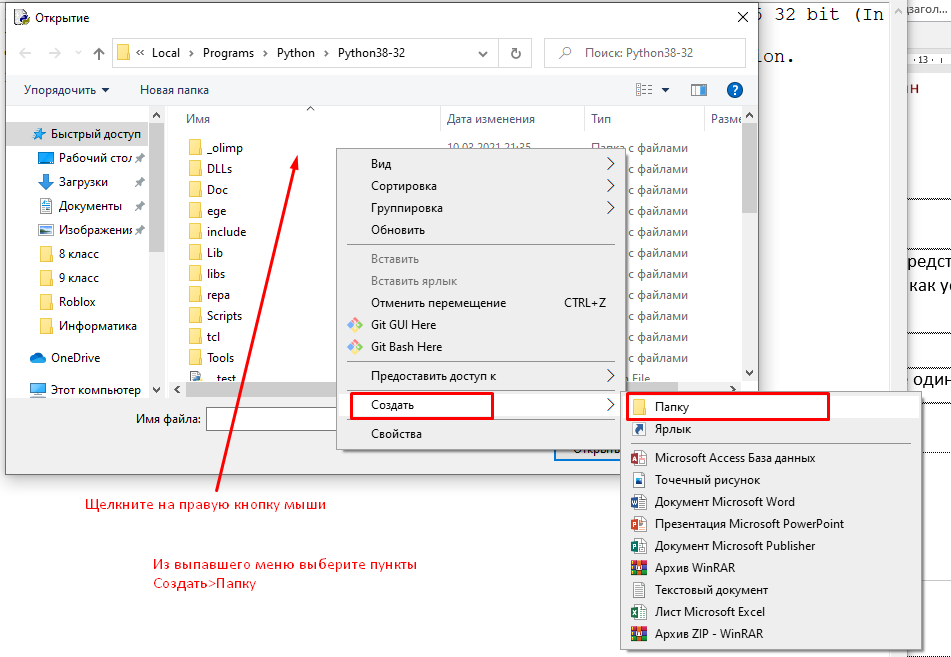
Как установить робота у себя на компьютере

К письму, в котором вы получили этот файл, приложен ещё один файл:

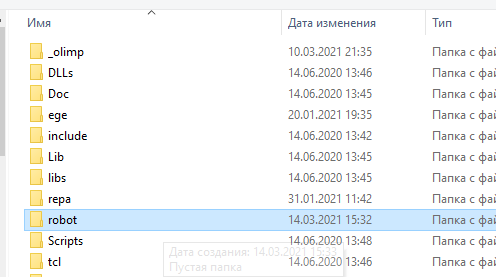
robot.py

Сохраните его на свой компьютер. Зайдите в IDLE (если забыли где он находится, смотрите самую первую картинку в этом файле) и нажмите на пункт меню File->Open

После чего откроется окно:



После чего будет создана папка «Новая папка» которую надо переименовать в **robot**



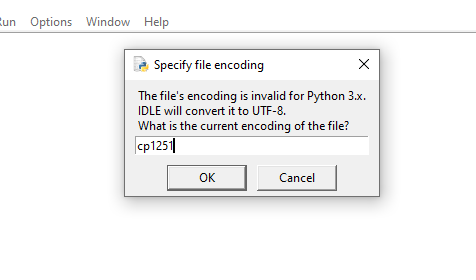
Скопируйте в эту папку ранее скачанный файл **robot.py**(он находится в папке «Загрузки») Если вы не можете найти файл **robot.py** скачайте его ещё раз и внимательно посмотрите куда вы его сохраняете.

Откройте IDLE файл **robot.py** из папки **robot** и запустите его командой Run(клавиша F5). После этого в папке robot появится файл task. Дальше работать мы будем только с ним.

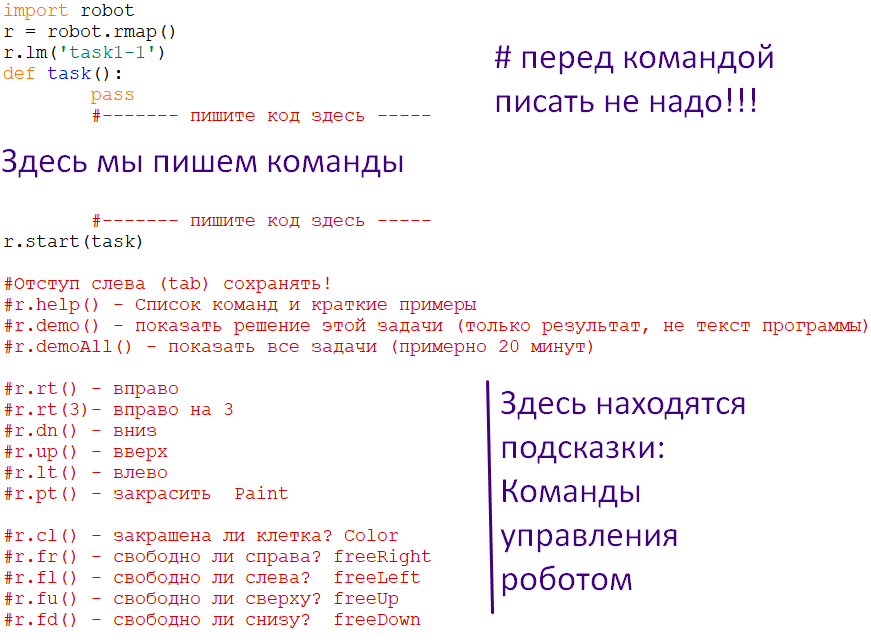
Закройте файл **robot.py** и больше его не открывайте.

Откройте файл task.py

При открывании может появиться такое окошко

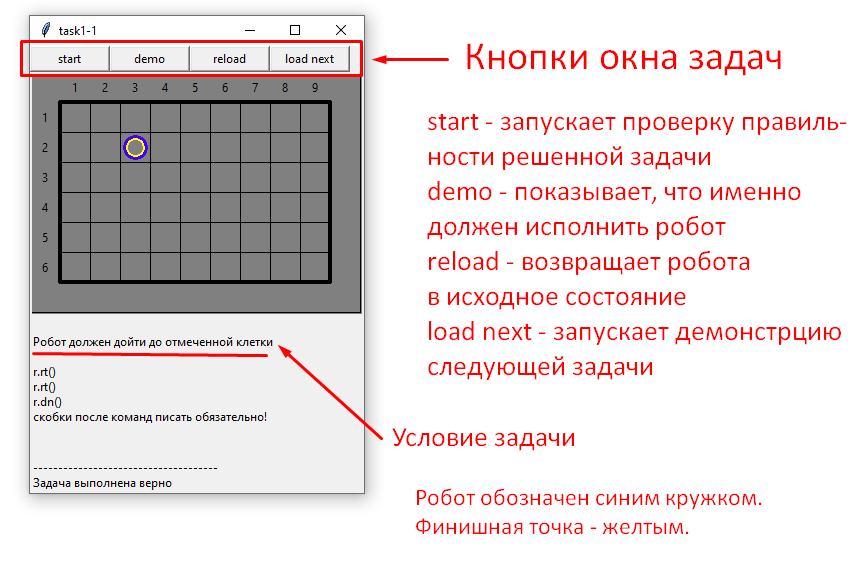


Введите в поле cp1251(в английской раскладке) и нажмите ОК



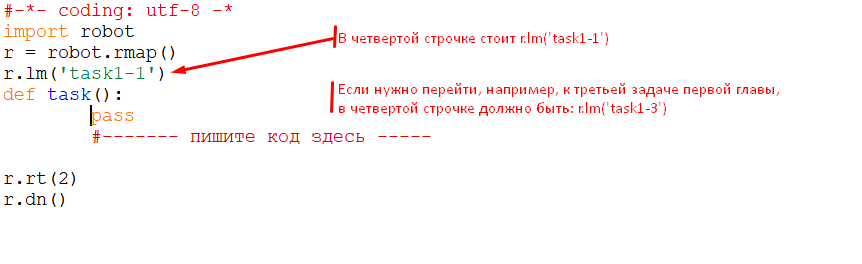
Знак # в начале строки ставится, чтобы ввести в текст программы пояснения. Команды, стоящие после # не выполняются.

Запустите файл task при помощи команды Run(клавиша F5)



Задачи разделены на главы. Первая задача первой главы вызывается, если в четвертой строке стоит:

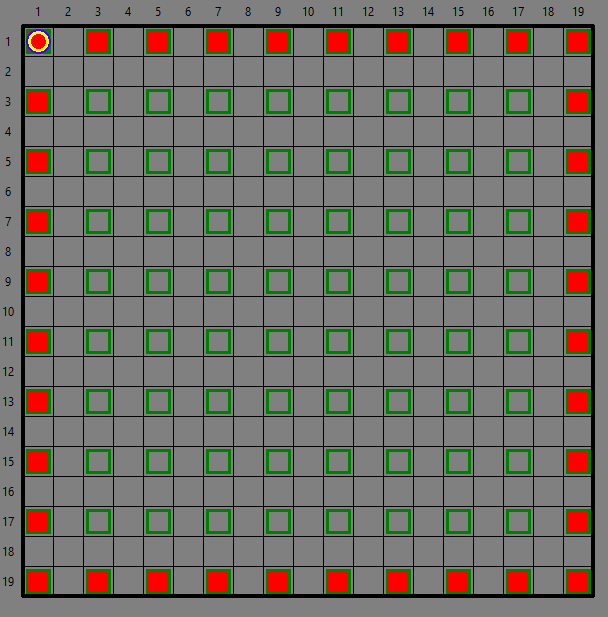
r.lm(‘task1-1’)



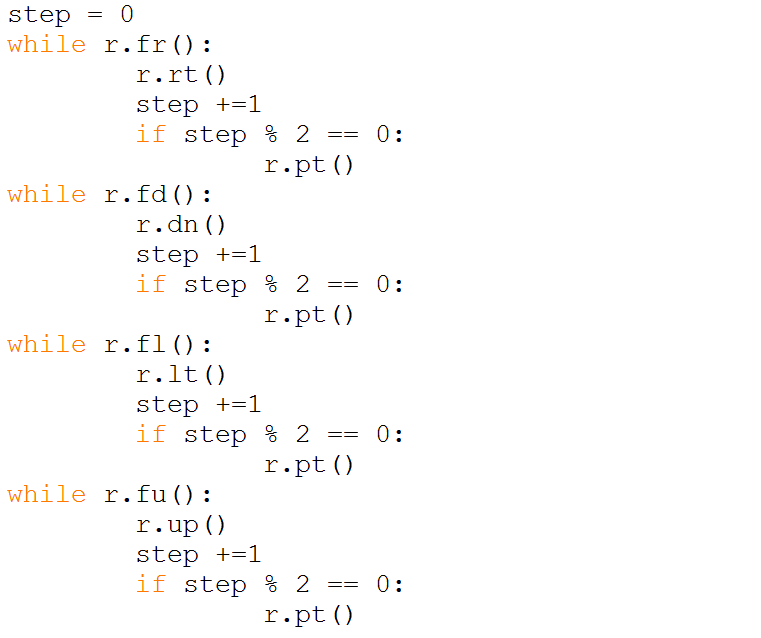
9. Робот. Решение простых задач.

pass

10. Робот. Решение задач посложнее.

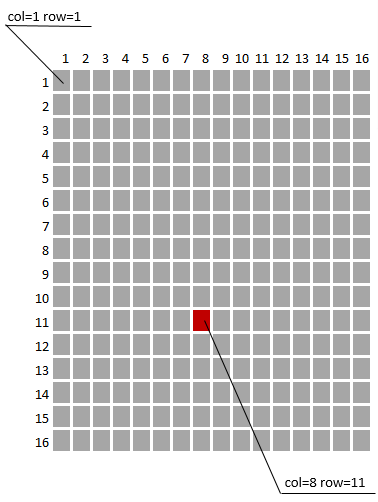
Задача: Робот должен обойти квадрат по краю и на своём пути закрашивать клетки через одну, чтобы получился пунктирный след. Для того чтобы обойти весь контур нам потребуется 4 цикла while, первый из которых будет спрашивать, свободно ли справа и если свободно - делаем шаг направо.

Для подсчёта количества шагов введём дополнительную переменную step в которой будем хранить сколько шагов сделано на данный момент. Если количество чётное – закрашиваем. То есть, на каждом шаге мы проверяем его чётность.



**Поиск закрашенной ячейки.**

Задача: Робот стоит в верхнем углу поля. Одна из ячеек закрашена. Найдите закрашенную ячейку и выведите ее координаты.



Введем для вычисления координат закрашенной клетки две переменных col и row, смысл которых пояснен на рисунке. Робот начинает двигаться направо до стенки. С каждым шагом мы прибавляем единицу к переменной col. Если дошли до стенки, а закрашенной клетки не нашли – прейдём к исследованию следующего ряда. Для этого возвращаем робота к левой стенке, и если внизу свободно – делаем шаг вниз. После чего снова движемся направо как было описано выше.

При нахождении нужной клетки программа выдает ее координаты и заканчивает работу по команде exit()-безусловное прекращение выполнения программы.

