Муниципальное бюджетное общеобразовательное учреждение
«Школа № 13 имени Героя Советского Союза Санчирова Ф.В.»
городского округа Самара

РАССМОТРЕНО	ПРОВЕРЕНО	УТВЕРЖДАЮ
Руководитель МО классных руководителей	Заместитель директора по УВР	Директор МБОУ "Школа № 13" г.о. Самара
_____ Колесникова О.А.	_____ Бирюкова Н.Б.	_____ Окуленко Д.В.
Протокол №1 от 26.08.2026г.	26.08.2026 г.	Приказ №233-од от 26.08.2026 г.

Окуленко Д.В.
O=МБОУ Школа №13 г.о. Самара, CN=Окуленко Д.В., E=shkola13-sanchirov@yandex.ru
2026.10.31 19:55:09 +04'00'

РАБОЧАЯ ПРОГРАММА

курса внеурочной деятельности для учащихся 9 классов

Основы программирования на языке Python

Направление: Общекультурное
Срок реализации: 1 год

Составитель: учитель информатики Матназарова А.В.

Самара, 2026

Пояснительная записка

Рабочая программа курса внеурочной деятельности «Основы программирования на языке Python» на основе основной образовательной программы среднего общего образования МБОУ Школы № 13 г.о. Самара.

Рабочая программа «Основы программирования на языке Python» ориентирована на работу по учебным пособиям: «Программируем на Python» (автор: М. Доусон; СПб.: Питер, 2014г.), «Python 3. Самое необходимое» (авторы: Н. Прохоренок , В. Дронов; БХВ-Петербург, 2016г.), рассчитана на 34 часов (1 час в неделю).

При составлении данной программы автором использованы следующие нормативно-правовые документы:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» Федеральные государственные образовательные стандарты (ФГОС) начального общего, основного общего и среднего общего образования
- Методическое письмо Минпросвещения России от 09.07.2026 № ОК-1984/03 «Об организации внеурочной деятельности обучающихся».
- Письмо Министерства просвещения РФ от 7 мая 2020 г. № ВБ-976/04 “О реализации курсов внеурочной деятельности, программ воспитания и социализации, дополнительных общеразвивающих программ с использованием дистанционных образовательных технологий”
- «Методические рекомендации для общеобразовательных организаций по организации внеурочной деятельности обучающихся» (разработка 2026 года). Письмо министерства образования и науки Самарской области от 17.02.2016 № МО-16-09-01/173-ТУ «О внеурочной деятельности»
- Основная образовательная программа (ООО, СОО)
- Положение об организации внеурочной деятельности.

Новые образовательные стандарты предъявляют к участникам образовательного процесса высокие требования. Одной из важнейших педагогических задач становится повышение общекультурного уровня обучающихся, а также формирование потребности в духовной культуре.

Срок реализации программы составляет 1 года.

Курс обучения рассчитан на детей в возрасте от 14-15 лет (9 классы).

Календарно-тематическое планирование рассчитано на 1 час в неделю,– 34 часа в год.

Цель программы внеурочной деятельности «Основы Python» для 9 класса — сформировать у школьников комплекс цифровых и метапредметных компетенций, а также поддержать профессиональное

самоопределение через практико-ориентированное освоение языка программирования.

Задачи:

- Освоить базовый синтаксис: переменные, типы данных, ввод/вывод, условия (if/elif/else), циклы (for, while).
- Научиться работать со структурами данных: списки, кортежи, словари, множества — и понимать, когда что применять.
- Писать и вызывать функции: передавать аргументы, возвращать значения, делить код на логические блоки.
- Обрабатывать простые ошибки и исключения: понимать разницу между синтаксическими и логическими ошибками, делать базовую отладку.
- Работать с файлами: читать и записывать текстовые данные, обрабатывать построчно, понимать, зачем закрывать файлы или использовать with.
- Применять язык для решения типовых задач: подсчёт статистики, фильтрация списков, поиск максимума/минимума, обработка последовательностей.

Планируемые результаты (личностные, метапредметные, предметные)

Личностные результаты

- Воспринимать ошибки как естественную часть обучения: спокойно разбирать сбои в коде, не бояться сообщений об ошибках, учиться на неудачных попытках и последовательно искать решение.
- Соблюдать этику работы с кодом: не выдавать чужой код за свой, не копировать без понимания сути, уважать труд других разработчиков, избегать плагиата в проектах.
- Понимать основы информационной безопасности и цифровой гигиены: не хранить пароли и чувствительные данные в коде, осознанно относиться к запуску чужих скриптов, аккуратно работать с персональными и чужими данными.
- Проявлять самостоятельность и инициативу: выбирать темы мини-проектов, предлагать свои способы решения задач, искать дополнительные возможности для развития (курсы, олимпиады, открытые задачи).
- Развивать устойчивость к трудностям и установку на рост: видеть в сложных задачах возможность научиться новому, а не повод отказаться от работы.

Предметные результаты

- Писать корректные программы на Python с использованием базовых конструкций: условия (if/elif/else), циклы (for, while), функции с аргументами и возвращаемыми значениями.
- Работать с основными типами данных и структурами: использовать переменные, списки, кортежи, словари, множества; осознанно выбирать подходящую структуру под задачу (например, словарь для хранения пар «ключ — значение», список — для упорядоченной последовательности).
- Обрабатывать ввод и вывод данных: считывать данные от пользователя, выводить результаты с форматированием, работать с текстовыми файлами (чтение и запись), использовать контекстный менеджер with для безопасной работы с файлами.
- Отлаживать и исправлять ошибки: различать синтаксические, логические и ошибки времени выполнения, применять простые приёмы отладки (вывод промежуточных значений, пошаговая проверка логики), обрабатывать базовые исключения.
- Решать типовые алгоритмические задачи: подсчёт статистики (среднее, максимум, минимум), фильтрация и преобразование списков, поиск по условию, обработка последовательностей, работа с подстроками и строковыми методами.
- Применять простые приёмы структурирования кода: разбивать решение на функции, давать осмысленные имена переменным и функциям, добавлять комментарии там, где логика неочевидна.

Метапредметные результаты

- Формализовать задачу: из словесного описания выделять входные данные, желаемый результат и ограничения, составлять пошаговый план решения.
- Строить и оценивать алгоритмы: выбирать подходящий тип алгоритма (линейный, с ветвлением, с циклом), проверять логику на простых примерах и крайних случаях, сравнивать разные подходы по понятности и эффективности.
- Искать и критически оценивать информацию: пользоваться официальной документацией Python, примерами кода и надёжными обучающими ресурсами, отличать рабочий и безопасный код от потенциально ошибочного или небезопасного.
- Объяснять и читать код: устно и письменно описывать логику своей программы, понимать и комментировать чужой код, давать конструктивную обратную связь по структуре и читаемости.
- Планировать и контролировать работу: ставить цель для мини-проекта, разбивать её на этапы, оценивать сроки, проверять соответствие результата критериям, корректировать план при необходимости.

СОДЕРЖАНИЕ КУРСА

Знакомство с языком Python	Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.
Переменные и выражения	Типы данных . Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.
Условные предложения	Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.
Циклы	Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.
Функции	Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.
Строки - последовательности символов	Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.
Сложные типы данных	Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или

	изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач. Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.
Стиль программирования и отладка программ	Стиль программирования. Отладка программ. Зачет по курсу «Программирование на языке Python»

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

Тема 1. Знакомство с языком Python – 2 часа
Тема 2. Переменные и выражения - 4 часа
Тема 3. Условные предложения – 5 часов
Тема 4. Циклы – 7 часов
Тема 5. Функции – 5 часов
Тема 6. Строки - последовательности символов - 3 часов
Тема 7. Сложные типы данных – 6 часов
Тема 8. Стиль программирования и отладка программ - 2 часа

Формы контроля и оценивания

- **Текущий контроль:** выполнение практических заданий, мини-задачи на написание функций/условий/циклов, проверка кода на соответствие критериям (работоспособность, читаемость, обработка ошибок).
- **Промежуточный контроль:** защита мини-проекта (демонстрация работы, объяснение логики, ответы на вопросы). Критерии: функциональность, структура кода, обработка крайних случаев, оформление.
- **Итоговый контроль:** комплексный мини-проект (1 занятие) либо портфолио из 2–3 работ с краткими описаниями.

Материально-техническое обеспечение

- Компьютеры/ноутбуки с доступом в интернет.
- Установленный Python (версии 3.8+).

- Среда разработки: IDLE, PyCharm Community или VS Code.
- Проектор/интерактивная доска для демонстрации кода.

Учебно-методическое обеспечение

- Официальная документация Python (docs.python.org).
- Учебные примеры и задачи по темам курса.
- Шаблоны проектов и чек-листы для самопроверки кода.
- Критерии оценивания мини-проектов.

Календарно-тематическое планирование

№	Тема занятия	Тип занятия	Основные виды учебной деятельности
1.	Общие сведениея о языке Python	Комбинированное занятие	*Аналитическая деятельность:* - знакомиться с языком Python; - изучать структуру программы на Python, · режимы работы с Python. *Практическая деятельность:* · выполнить установку программы; · выполнить простейшую программу в интерактивной среде; · написать комментарии в программе.
2.	Режимы работы	Комбинированное занятие	
3.	Переменные	Комбинированное занятие	*Аналитическая деятельность:* - изучать операторы ввода-вывода. *Практическая деятельность:* - работать со справочной системой; - решать задачи на элементарные действия с числами.; - пользоваться интерфейсом среды программирования Python; - использовать команды редактора; - организовывать ввод и вывод данных; - записывать арифметические выражения.
4.	Выражения	Комбинированное занятие	
5.	Ввод и вывод	Комбинированное занятие	
6.	Задачи на элементарные действия с числами	Комбинированное занятие	
7.	Логические выражения и операторы	Комбинированное занятие	*Аналитическая деятельность:* - изучать назначение условного оператора; - изучать способы записи условного оператора; - изучать логический тип данных; - изучать логические операторы or, and, not; *Практическая деятельность:* - использовать условный оператор; - создавать сложные условия с помощью логических операторов.; - решать задачи по теме "Условные операторы"; - составлять программы с ветвлением
8.	Условный оператор	Комбинированное занятие	
9.	Множественное ветвление	Комбинированное занятие	
10.	Реализация ветвления в языке Python	Комбинированное занятие	
11.	Составление программ с ветвлением	Комбинированное занятие	
12.	Оператор цикла с условием	Комбинированное занятие	*Аналитическая деятельность:* - изучать циклы с условием и их виды; - изучать правила записи циклов условием; - изучать назначение и особенности использования цикла с параметром; - изучать формат записи цикла с параметром;
13.	Оператор цикла for	Комбинированное занятие	
14.	Вложенные циклы	Комбинированное занятие	
15.	Случайные числа	Комбинированное занятие	
16.	Примеры решения задач с	Занятие контроля знаний	

	циклом	и умений	- изучать примеры использования циклов различных типов. *Практическая деятельность:* - решать задачи с циклом for; - реализацовывать циклические алгоритмы - составлять программы с циклом; - определять вид цикла, наиболее удобный для решения поставленной задачи; - использовать цикл с условием; - определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.
17.	Творческая работа "Циклы"	Комбинированное занятие	
18.	Создание функций	Комбинированное занятие	*Аналитическая деятельность:* - изучать способы описания функции; - изучать принципы структурного программирования; - изучать понятие локальных переменных подпрограмм; - изучать понятие формальных и фактических параметров подпрограмм; - изучать способы передачи параметров. *Практическая деятельность:* - решать задачи с использованием функций; - решать задачи с использованием рекурсивных функций; - создавать и использовать функции; - использовать механизм параметров для передачи значений.
19.	Локальные переменные	Комбинированное занятие	
20.	Примеры решения задач с использованием функций	Комбинированное занятие	
21.	Рекурсивные функции	Комбинированное занятие	
22.	Строки	Занятие контроля знаний и умений	*Аналитическая деятельность:* -изучать назначение строкового типа данных; - изучать операторы для работы со строками; - изучать процедуры и функции для работы со строками; - изучать операции со строками. *Практическая деятельность:* - решать задачи со строками; - описывать строки; - соединять строки; - находить длину строки; - вырезать часть строки; - находить подстроку в строке; - находить количество слов в строке.
23.	Срезы строк	Комбинированное занятие	
24.	Примеры решения задач со строками	Комбинированное занятие	
25.	Списки	Комбинированное занятие	*Аналитическая деятельность:* - изучать сложные типы данных;
26.	Срезы списков	Комбинированное занятие	

27.	Списки: примеры решения задач	Комбинированное занятие	- изучать способы описания списка; - изучать способы доступа к элементам списка; - изучать способы описания кортежа; - изучать способы описания словаря; - изучать операции, выполняемые со списками, кортежами и словарями; - изучать понятие множества; - изучать способы описания множества; - изучать операторы работы с множествами. *Практическая деятельность:* - описывать списки; - вводить элементы списка; - выводить элементы списка; - выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; - использовать вложенные списки; - приводить примеры использования вложенных списков (матриц); - описывать множества; - определять принадлежность элемента множеству; - вводить элементы множества; - выводить элементы множества; - решать задачи со списками.
28.	Матрицы	Комбинированное занятие	
29.	Кортежи	Комбинированное занятие	
30.	Введение в словари	Занятие контроля знаний и умений	
31.	Множества в языке Python	Комбинированное занятие	
32.	Стиль программирования	Комбинированное занятие	*Аналитическая деятельность:* - изучать, что такое стиль программирования; -изучать правила именования объектов; -изучать основные рекомендации при написании программ. *Практическая деятельность:* - определять вид ошибок и находить ошибки в программе. - выполнять тестирование и отладку программ.
33.	Отладка программ	Комбинированное занятие	
34.	Зачет по курсу «Программирование на языке Python»	Занятие контроля знаний и умений	