

УТВЕРЖДАЮ
Генеральный директор
ООО «Центр образовательных
технологий»
Приказ № 4 от 01.06.2026 г.

**Дополнительная общеобразовательная
общеразвивающая программа**

«Python для начинающих»

Возраст обучающихся: 14-18 лет

Срок реализации: 4 месяца

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Python для начинающих» (далее - Программа) является дополнительной общеобразовательной общеразвивающей программой и разработана в соответствии с положениями:

- Федерального закона № 273-ФЗ от 29.12.2012 «Об образовании в Российской Федерации»;
- Концепции развития дополнительного образования детей до 2030 года (Распоряжение Правительства Российской Федерации от 31.03.2022 г. № 678-р, с изменениями от 01.07.2025 №1745-р);
- Государственной программы РФ «Развитие образования» (постановление Правительства Российской Федерации от 26 декабря 2017 г. № 1642);
- Приказа Министерства просвещения РФ 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Методических рекомендаций по проектированию дополнительных общеразвивающих программ (Письмо Минобрнауки России № 09-3242 от 18 ноября 2015 года);
- Письма Министерства просвещения РФ от 07.05.2020 г. № ВБ-976/04 «О реализации курсов внеурочной деятельности, программ воспитания и социализации, дополнительных общеразвивающих программ с использованием дистанционных образовательных технологий»;
- Постановления Правительства Российской Федерации от 11.10.2023 №1678 "Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ";
- Санитарных правил СП 2.4.3648-20.

Направленность программы: техническая.

Уровень освоения программы: общеразвивающая.

Актуальность программы

Актуальность программы обусловлена необходимостью формирования у обучающихся инженерного мышления и навыков текстового программирования в условиях стремительной цифровизации. Традиционные методы обучения часто не учитывают барьер изучения реальных языков программирования, что ведет к потере интереса. язык Python предлагает мягкий вход в профессиональную разработку благодаря лаконичному синтаксису, помогая детям преодолевать трудности при написании первых строк кода и формируя прочную базу для освоения современных ИТ-профессий и изучения более сложных технологий в будущем.

Педагогическая целесообразность

Программа направлена на системное развитие навыков написания текстового кода и алгоритмизации (структуры данных, управление потоком выполнения, модульность). Она помогает устранить пробелы в понимании того, как работают высокоуровневые системы, через отработку практических задач по созданию алгоритмов. Упор на интерактивное освоение синтаксиса и проектную деятельность позволяет обучающимся осознанно применять математические знания и логику для решения реальных задач.

Цель

Ликвидировать пробелы в знаниях обучающихся в области архитектуры программного обеспечения и основ текстового кодирования, сформировать устойчивые навыки разработки на языке Python (работа с переменными, циклами, функциями и внешними библиотеками) и обеспечить готовность к самостоятельному созданию программных инструментов для учебы и творчества.

Задачи

Образовательные:

- расширить словарный запас ИТ-понятий (интерпретатор, синтаксис, переменная, тип данных, функция) и представления о возможностях текстового программирования;
- сформировать устойчивые навыки работы в среде разработки (IDE), взаимодействия с консолью и написания чистого программного кода;
- создать условия для формирования базовой инженерной и алгоритмической компетентности;
- восполнить пробелы в технических навыках (структурирование программных файлов, работа с библиотеками, чтение и отладка сообщений об ошибках).

Развивающие:

- развивать умения анализировать, структурировать и обрабатывать информацию с помощью программных алгоритмов (текстовые строки, числовые данные, списки);
- способствовать развитию логического, абстрактного и вычислительного мышления при решении прикладных задач;
- способствовать эффективному и осознанному использованию языка Python в учебной, исследовательской и творческой деятельности.

Воспитательные:

- формировать ключевые цифровые компетенции разработчика;
- повышать интерес к профессиональному программированию и созданию полезных ИТ-продуктов;
- воспитывать культуру безопасного и ответственного поведения в цифровой среде.
- воспитывать культуру ответственного использования открытого кода и соблюдения принципов этичного программирования в цифровой среде

Категория обучающихся

Возраст обучающихся: 14-18 лет.

Форма проведения учебных занятий:

групповые занятия до 10 человек

Срок реализации Программы

Программа рассчитана на 4 месяца обучения. Продолжительность обучения составляет 48 учебных часов.

Форма и режим занятий

Форма проведения учебных занятий: комбинированная (теоретическая часть + практическая отработка).

Режим занятий определяется в зависимости от формы обучения:
групповые занятия: 3 раза в неделю по 1 академическому часу.

Режим занятий соответствует санитарным правилам СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

Продолжительность одного академического часа — 60 минут.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Название образовательной программы	Продолжительность учебного часа	Кол-во уч. недель	Кол-во уч. часов в неделю	Кол-во уч. дней в неделю	Кол-во уч. часов в год	Даты начала и окончания уч. периодов/этапов
Python для начинающих	60 мин.	16	3	3	48	14 сентября – 3 января
Итоговый контроль: итоговый проект		2 часа				
Общее количество учебных часов за весь период обучения:		48 часов				

Примечание: дата начала и окончания учебных занятий может быть скорректирована в зависимости от сроков формирования учебной группы.

Для обучения по данной программе промежуточные каникулы не предусмотрены.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Личностные результаты:

- осознают ценность текстового программирования как универсального языка для решения технических и творческих задач в современном мире;
- формируют внутреннюю мотивацию к углубленному изучению ИТ-сферы и развитию навыков разработки программного обеспечения;
- имеют достаточный уровень алгоритмических знаний и практических умений для самостоятельного написания простых скриптов и автоматизации задач;
- развивают способность к рефлексии и аналитическому мышлению на основе поиска и исправления логических ошибок в собственном коде.

Предметные результаты:

Знать:

- базовые понятия языка Python (переменные, типы данных, арифметические и логические операторы, циклы, условия, функции);
- синтаксические правила оформления кода, принципы работы интерпретатора и структуры программного файла;
- нормы корректного использования библиотек и безопасного скачивания программных инструментов из сети Интернет.

Уметь:

- уверенно работать в среде разработки (IDE), использовать горячие клавиши, сохранять и запускать программные файлы через консоль;
- создавать работоспособные цифровые продукты (калькуляторы, консольные игры, программы для обработки текста) на языке Python;
- находить и исправлять синтаксические и логические ошибки в коде, опираясь на диагностические сообщения системы или подсказки педагога;
- осознанно применять навыки программирования для решения математических и творческих учебных задач.

Метапредметные результаты:

- оценивать эффективность и чистоту своего кода, опираясь на критерии работоспособности и понятности алгоритма;
- анализировать причинно-следственные связи при возникновении багов и затруднений в процессе кодирования;
- осознанно выстраивать четкую последовательность действий (алгоритм) при реализации программного замысла от идеи до финального запуска;
- признавать возможность существования разных программных решений (алгоритмов) для одной и той же задачи и уважать архитектурный выбор других;
- внимательно слушать объяснения логики построения функций и циклов, кратко формулировать суть алгоритма при обсуждении технических тем.

СОДЕРЖАНИЕ ПРОГРАММЫ

УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН

№ п/п	Наименование разделов и тем	Всего, час.	теор. занятия	практич. занятия	самост. работа
1	Раздел 1. Python для начинающих	96	24	24	48
1	Введение в программирование: что такое Python и зачем он нужен	2	0,5	0,5	1
2	Установка языка и запуск первой программы	2	0,5	0,5	1
3	Вводное занятие: работа с компьютером и безопасность	2	0,5	0,5	1
4	Вывод информации на экран: функция print()	2	0,5	0,5	1
5	Числа в Python: целые и дробные	2	0,5	0,5	1
6	Простые математические операции (сложение, вычитание, умножение, деление)	2	0,5	0,5	1
7	Работа с текстом: знакомство со строками	2	0,5	0,5	1
8	Ввод данных от пользователя: функция input()	2	0,5	0,5	1
9	Преобразование данных: как сделать из текста число и наоборот	2	0,5	0,5	1
10	Калькулятор с помощью Python	2	0,5	0,5	1
11	Ошибки в коде: как их читать и исправлять	2	0,5	0,5	1
12	Логика в коде: Истина (True) и Ложь (False)	2	0,5	0,5	1
13	Операторы сравнения: больше, меньше, равно	2	0,5	0,5	1
14	Условный оператор if	2	0,5	0,5	1
15	Альтернативные пути: блоки else и elif	2	0,5	0,5	1
16	Логические and, or, not	2	0,5	0,5	1
17	Создание текстового квеста с помощью ветвлений	2	0,5	0,5	1
18	Введение в циклы	2	0,5	0,5	1
19	Цикл while: повторяем, пока условие истинно	2	0,5	0,5	1
20	Остановка цикла: как использовать break	2	0,5	0,5	1

21	Цикл for: перебор элементов по очереди	2	0,5	0,5	1
22	Функция random() : генерация случайных чисел, букв, случайного выбора элементов	2	0,5	0,5	1
23	Длина строки и индексы символов	2	0,5	0,5	1
24	Удаление ненужного кода	2	0,5	0,5	1
25	Полезные инструменты для текста: перевод букв в заглавные и строчные	2	0,5	0,5	1
26	Списки (list): как хранить много данных в одной переменной	2	0,5	0,5	1
27	Изменение списков: добавление и удаление элементов	2	0,5	0,5	1
28	Индексы в списках: как достать нужный элемент	2	0,5	0,5	1
29	Перебор элементов списка с помощью цикла for	2	0,5	0,5	1
30	Умный список покупок с помощью Python	2	0,5	0,5	1
31	Введение в функции	2	0,5	0,5	1
32	Создание собственных функций (команда def)	2	0,5	0,5	1
33	Как передавать информацию внутрь функции (аргументы)	2	0,5	0,5	1
34	Как получать результат работы функции (команда return)	2	0,5	0,5	1
35	Знакомство с модулями: встроенные "помощники" Python	2	0,5	0,5	1
36	Модуль random: генерация случайных чисел и случайный выбор	2	0,5	0,5	1
37	Модуль time: как ставить программу на паузу	2	0,5	0,5	1
38	Визуализация: знакомство с модулем turtle (черепашка для рисования)	2	0,5	0,5	1
39	Модуль turtle: рисуем простые геометрические фигуры	2	0,5	0,5	1
40	Модуль turtle: объединяем рисование и циклы для сложных узоров	2	0,5	0,5	1

41	Словари (dict): как хранить данные в формате "ключ-значение"	2	0,5	0,5	1
42	Добавление и поиск информации в словаре	2	0,5	0,5	1
43	Телефонный справочник с помощью Python	2	0,5	0,5	1
44	Основы работы с файлами: как прочитать текст из файла	2	0,5	0,5	1
45	Основы работы с файлами: как сохранить результат работы в файл	2	0,5	0,5	1
46	Создание простого бота с помощью Python	2	0,5	0,5	1
47	Введение в ООП: классы и объекты	2	0,5	0,5	1
48	Итоговый проект	2	0,5	0,5	1
Итого		96	24	24	48

СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА

Раздел 1. Python для начинающих (48 часов)

Тема 1. Введение в программирование: что такое Python и зачем он нужен (1 час)

Тема 2. Установка языка и запуск первой программы (1 час)

Тема 3. Вводное занятие: работа с компьютером и безопасность (1 час)

Тема 4. Вывод информации на экран: функция print() (1 час)

Тема 5. Числа в Python: целые и дробные (1 час)

Тема 6. Простые математические операции (сложение, вычитание, умножение, деление) (1 час)

Тема 7. Работа с текстом: знакомство со строками (1 час)

Тема 8. Ввод данных от пользователя: функция input() (1 час)

Тема 9. Преобразование данных: как сделать из текста число и наоборот (1 час)

Тема 10. Калькулятор с помощью Python (1 час)

Тема 11. Ошибки в коде: как их читать и исправлять (1 час)

Тема 12. Логика в коде: Истина (True) и Ложь (False) (1 час)

Тема 13. Операторы сравнения: больше, меньше, равно (1 час)

Тема 14. Условный оператор if (1 час)

Тема 15. Альтернативные пути: блоки else и elif (1 час)

Тема 16. Логические and, or, not (1 час)

Тема 17. Создание текстового квеста с помощью ветвлений (1 час)

Тема 18. Введение в циклы (1 час)

Тема 19. Цикл while: повторяем, пока условие истинно (1 час)

Тема 20. Остановка цикла: как использовать break (1 час)

Тема 21. Цикл for: перебор элементов по очереди

Тема 22. Функция random() : генерация случайных чисел, букв, случайного выбора элементов (1 час)

Тема 23. Длина строки и индексы символов (1 час)

Тема 24. Удаление ненужного кода (1 час)

Тема 25. Полезные инструменты для текста: перевод букв в заглавные и строчные (1 час)

Тема 26. Списки (list): как хранить много данных в одной переменной (1 час)

Тема 27. Изменение списков: добавление и удаление элементов (1 час)

Тема 28. Индексы в списках: как достать нужный элемент (1 час)

Тема 29. Перебор элементов списка с помощью цикла for (1 час)

Тема 30. Умный список покупок с помощью Python (1 час)

Тема 31. Введение в функции (1 час)

Тема 32. Создание собственных функций (команда def) (1 час)

Тема 33. Как передавать информацию внутрь функции (аргументы) (1 час)

Тема 34. Как получать результат работы функции (команда return) (1 час)

Тема 35. Знакомство с модулями: встроенные "помощники" Python (1 час)

Тема 36. Модуль random: генерация случайных чисел и случайный выбор (1 час)

Тема 37. Модуль time: как ставить программу на паузу (1 час)

Тема 38. Визуализация: знакомство с модулем turtle (черепашка для рисования) (1 час)

Тема 39. Модуль turtle: рисуем простые геометрические фигуры (1 час)

- Тема 40. Модуль turtle: объединяем рисование и циклы для сложных узоров (1 час)
- Тема 41. Словари (dict): как хранить данные в формате "ключ-значение" (1 час)
- Тема 42. Добавление и поиск информации в словаре (1 час)
- Тема 43. Телефонный справочник с помощью Python (1 час)
- Тема 44. Основы работы с файлами: как прочитать текст из файла (1 час)
- Тема 45. Основы работы с файлами: как сохранить результат работы в файл (1 час)
- Тема 46. Создание простого бота с помощью Python (1 час)
- Тема 47. Введение в ООП: классы и объекты (1 час)
- Тема 48. Итоговый проект (1 час)

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

№ темы	Наименование практического занятия
1	Практическая работа. Найти 3 сайта или приложения, написанных на Python, и описать их функции.
2	Практическая работа. Настройка среды разработки. Вывод фразы «Программа запущена!» в консоль.
3	Практическая работа. Вывод графического рисунка (робота или домика) с помощью символов и текста.
4	Практическая работа. Форматирование текста: вывод слов через разные разделители (тире, точки, смайлики).
5	Практическая работа. Создание карточки «человека»: объявление переменных для имени, возраста и профессии.
6	Практическая работа. Исправление «сломанного» кода, где имена переменных написаны с ошибками.
7	Практическая работа. Программа «Весы»: сложение целых и дробных чисел.
8	Практическая работа. Расчет разницы в возрасте между пользователем и его родителями
9	Практическая работа. Программа для дележа конфет: сколько достанется каждому и сколько останется в остатке.
10	Практическая работа. Склеивание (конкатенация) нескольких слов в одну осмысленную фразу.
11	Практическая работа. Создание диалога с компьютером: программа спрашивает имя и отвечает персональным приветствием.
12	Практическая работа. Исправление ошибки, когда программа пытается сложить текст вместо цифр.
13	Практическая работа. Написание полноценного калькулятора для двух чисел с четырьмя действиями.
14	Практическая работа. Поиск и устранение 5 намеренных ошибок в готовом коде (отладка).
15	Практическая работа. Проверка утверждений (например, $5 > 10$) и вывод логического результата.
16	Практическая работа. Программа, которая сравнивает данные двух пользователей и определяет победителя.

17	Практическая работа. Проверка пароля: если пароль верный, программа «открывает дверь».
18	Практическая работа. Программа «Детектор температуры»: холодно, тепло или жарко на улице.
19	Практическая работа. Сложная проверка: пускать на аттракцион, только если рост > 140 И возраст > 12.
20	Практическая работа. Мини-игра с выбором пути (налево, направо, прямо) и разными финалами.
21	Практическая работа. Сравнение: вывести фразу 100 раз вручную и с помощью 2 строк кода.
22	Практическая работа. Программа «Счетчик секунд», которая работает до определенного числа.
23	Практическая работа. Программа, которая запрашивает слова, пока пользователь не скажет «хватит».
24	Практическая работа. Вывод каждой буквы имени пользователя по отдельности в столбик.
25	Практическая работа. Вывод всех четных чисел от 2 до 20 с использованием шага.
26	Практическая работа. Игра «Бросок кубика»: генерация случайного числа от 1 до 6.
27	Практическая работа. Программа «Генератор имен»: сборка имени персонажа из случайных слогов.
28	Практическая работа. Программа, которая находит секретную букву в слове по её номеру (индексу).
29	Практическая работа. Оптимизация длинной и запутанной программы: сокращение строк без потери смысла.
30	Практическая работа. Создание программы, которая исправляет имена (делает первую букву большой).
31	Практическая работа. Создание списка «Список дел на завтра» из 5 пунктов.
32	Практическая работа. Программа для управления рюкзаком: подбор нового предмета и выбрасывание старого.
33	Практическая работа. Программа «Рейтинг»: вывод 1-го, 3-го и последнего места из победителей «турнира».
34	Практическая работа. Автоматическая рассылка приветов всем друзьям из списка контактов.
35	Практическая работа. Программа, которая считает количество товаров и выводит их красивым списком.

36	Практическая работа. Анализ программы и выделение повторяющихся действий в отдельный блок.
37	Практическая работа. Написание функции, которая рисует «рамку» из символов вокруг любого текста.
38	Практическая работа. Функция «Машина», которая принимает название марки и «едет» (выводит текст).
39	Практическая работа. Функция-калькулятор, которая возвращает результат вычислений для дальнейшего использования.
40	Практическая работа. Импорт модуля math и расчет квадратного корня числа.
41	Практическая работа. Игра «Угадай число»: компьютер загадывает, а пользователь пытается отгадать за 5 попыток.
42	Практическая работа. Создание цифрового таймера с задержкой в 1 секунду между цифрами.
43	Практическая работа. Управление черепашкой: нарисовать первую линию и повернуть её.
44	Практическая работа. Написание программы для рисования квадрата, треугольника и круга.
45	Практическая работа. Создание сложных узоров и спиралей с помощью черепашки и циклов for.
46	Практическая работа. Создание «Переводчика»: словарь со словами и сохранение базы слов в файл.
47	Практическая работа. Создание класса «Робот» с именем и цветом. Создание нескольких объектов-роботов.
48	Практическая работа. Разработка собственного проекта: чат-бота, игры или графического рисунка.

ФОРМЫ АТТЕСТАЦИИ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа предусматривает аттестацию обучающихся, которая направлена на выявление исходного, текущего, промежуточного и итогового уровня теоретических знаний, развития практических умений и навыков, сформированных компетенций.

- **текущий контроль** позволяет судить об успешности овладения знаниями, о процессе становления и развития критического восприятия. Этот контроль должен быть регулярным и направленным на проверку усвоения учащимися определенной части учебного материала;
- **промежуточный контроль** проводится по завершении крупного раздела программы. Он позволяет судить об эффективности овладения разделом программного материала;
- **итоговый контроль** направлен на установление уровня владения материалом и проводится в конце прохождения курса.

ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Все разделы программы обеспечены необходимыми методическими и дидактическими материалами.

Методическое обеспечение реализации программы

Методика обучения по программе состоит из сочетания словесного изложения теоретического материала с показом иллюстрирующего материала и приемов решения практических и расчетных задач.

В работе используются следующие методы:

- **словесные:** объяснение нового материала, беседа, анализ цифровых понятий и принципов работы устройств, разбор типичных ошибок при выполнении заданий на компьютере;
- **наглядные:** использование схем, таблиц, алгоритмов действий, наглядных изображений с устройством компьютера, мультимедийных презентаций, обучающих видеороликов и интерактивных демонстраций;
- **практические:** выполнение тренировочных упражнений на компьютере, отработка навыков работы с мышью и клавиатурой, создание простых цифровых продуктов, выполнение игровых и творческих заданий в детских программах;
- **проблемно-поисковые:** анализ проблемных ситуаций при работе с компьютером, выдвижение предположений о способах решения задачи, объяснение причин возникновения ошибок, сравнение различных способов выполнения одного и того же действия;
- **рефлексивные:** самооценка результатов выполненной работы, взаимопроверка, анализ собственных ошибок и затруднений при работе с цифровыми инструментами.

Формы занятий:

- практическая работа (написание программ, рефакторинг, исправление кода, разработка проектов);
- самостоятельная работа.

Дидактическое обеспечение предполагает наличие текстов разноуровневых заданий, тематических заданий по каждому разделу и теме. В качестве материала для работы служат контрольно-измерительные материалы.

Материально-технические условия реализации Программы

- Доски в Miro
- Компьютер
- Internet, программа MTS-link.
- Компьютерная программа Visual Studio Code
- Компьютерная программа PyCharm

Перечень инструментов, принадлежностей и материалов для обучающегося:

- Доски в Miro

- Компьютер
- Платформа для видеоконференций для дистанционного формата (Livedigital и аналогичные платформы);
- Компьютерная программа Visual Studio Code
- Компьютерная программа PyCharm

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Литература

- 1.1. Джейсон Бриггс. Python для детей. Самоучитель по программированию.
- 1.2. Карл Бирнс. Программирование на Python для начинающих. — М.: Эксмо.
- 1.3. Луис Хэндикемп. Python для начинающих. Учимся программировать с нуля
- 1.4. Эл Свейгарт. Учим Python, делая крутые игры. — М.: Лаборатория знаний.

2. Интернет-ресурсы

- 2.1. Python.org — официальный сайт языка программирования Python и полная документация — <https://www.python.org>
- 2.2. Replit / PyCharm — онлайн и офлайн среды разработки для написания, отладки и запуска кода — <https://replit.com>