

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ - СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ**

ШКОЛА №1 р.п. СТЕПНОЕ

СОВЕТСКОГО РАЙОНА САРАТОВСКОЙ ОБЛАСТИ

Рассмотрено и рекомендовано на
заседании педагогического совета
Протокол № 1
от «26 » августа 2025 г.

«Утверждаю»
Директор
_____ Исакина Н.Ю.
Приказ № 129
от «29» августа 2025 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«СТАРТОВЫЙ УРОВЕНЬ: ПРОГРАММИРОВАНИЕ НА PYTHON»**

КУБ «ПРОГРАММИРОВАНИЕ НА PYTHON»

Направленность: техническая

Возраст обучающихся: 10 - 14 лет

Срок реализации: 9 месяцев

Составитель программы:
Здоров Андрей Николаевич,
педагог дополнительного образования

р.п.Степное,
2025г.

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК

ПРОГРАММЫ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Стартовый уровень: Программирование на Python» разработана с учётом возрастных особенностей обучающихся и Положения о дополнительной общеобразовательной (общеразвивающей) программе МБОУ-СОШ № 1 р.п. Степное Советского района Саратовской области (утв. Приказом директора МБОУ-СОШ №1 р.п. Степное от 31.08.2021 г. № 215)

Направленность программы – техническая.

Аннотация к программе:

Python используется в различных областях, включая разработку веб-приложений, научное моделирование, анализ данных, искусственный интеллект, машинное обучение, автоматизацию задач и многое другое. Россия имеет сильное присутствие во многих из этих областей, и изучение Python позволит российским специалистам оставаться конкурентоспособными на мировой арене. В России существует большой спрос на IT-специалистов, в том числе на программистов, владеющих Python. Изучение Python может увеличить шансы на рынке труда и открыть двери к новым возможностям карьерного роста и развития.

Таким образом, изучение Python имеет большую актуальность для России. Этот язык программирования может быть полезен в различных сферах деятельности и обеспечить успешное будущее в IT-индустрии.

Данная программа позволяет обучающимся с разным уровнем знания информатики освоить основы программирования научиться проектировать и разрабатывать приложения на языке программирования Python.

Программа «Стартовый уровень: Программирование на Python» реализуется в сетевой форме взаимодействия с образовательными учреждениями Советского района.

Педагогическая целесообразность программы «Базовый курс. Программирование на языке Python» основана на применении технологий индивидуализации обучения, дифференцированного и развивающего обучения.

Отличительная черта программы:

Обучающиеся применяют полученные знания на практике для создания проектов с применением языка программирования, разрабатывают работы и проекты в командах, что способствует развитию коммуникативных навыков и возможностью в дальнейшем работать в коллективе.

Адресат программы – дети в возрасте 10-14 лет.

Возрастные особенности учащихся 10-14 лет. Обучающиеся данного

возраста способны на высоком уровне усваивать разнообразную информацию в области мобильных разработок. На занятиях предусматривается деятельность, создающая условия для творческого развития воспитанников различных возрастных категорий и учитывается дифференцированный подход, зависящий от степени одаренности и возраста воспитанников.

Срок реализации программы – 9 месяцев

Численность детей в группе – 12 человек.

Объем программы - 72 часа.

Уровень освоения программы – стартовый.

Форма обучения – очная.

Форма организации занятий – индивидуально-групповая.

Режим занятий – 1 раз в неделю по 2 часа.

Цель программы – формирование и развитие у обучающихся знаний, умений и навыков в области программирования на языке Python для решения практических и образовательных задач.

Задачи:

Обучающие:

- Систематизировать и пополнить знания в области алгоритмизации;
- дать базовые знания, умения и навыки о принципах и методах функционального программирования;
- дать базовые знания, умения и навыки о принципах и методах объектно-ориентированного программирования;
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- сформировать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.

Развивающие:

- сформировать навыки поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- способствовать развитию самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- познакомить с навыками проектной деятельности

Воспитательные:

- совершенствовать коммуникативные навыки при работе в паре, коллективе;
- воспитать самостоятельность при решении задач и умение работать в команде.

Планируемые результаты

Образовательные:

- приобретут знания в области алгоритмизации;
- приобретут базовые знания, умения и навыки о принципах и методах функционального программирования;
- приобретут базовые знания, умения и навыки о принципах и методах объектно-ориентированного программирования;
- научатся ориентироваться в интерфейсе и функциональных возможностях среды разработки на языке Python;
- приобретут навыки разработки программ на языке программирования Python.

Метапредметные:

- приобретут навыки эффективного поиска, анализа и применения информации;
- приобретут навыки применения творческого подхода для решения практических и учебных задач;
- познакомятся с навыками проектной деятельности в формате разработки, реализации и защиты группового проекта.

Личностные:

- научатся выстраивать эффективные коммуникации при работе в паре, коллективе;
- научатся принимать самостоятельные решения при выполнении задач.

1.2. Содержание программы

1.2.1. Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	Раздел 1. Введение в Python	4	3	1	Тест по модулю
	1.1. Инструкции и структура программы.	2	2	0	Устный опрос
	1.2. Установка Python. Ввод и вывод данных.	2	1	1	Решение задач
2.	Раздел 2. Типы данных и операции	12	6	6	Тест по модулю
	2.1. Переменные. Операторы.	4	2	2	Тест по теме
	2.2 Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	8	4	4	Решение задач

3.	Раздел 3. Инструкции и синтаксис	14	7	7	Тест по модулю
	3.1. Условный оператор if, Альтернативный условный оператор elif, else.	4	2	2	Решение задач
	3.2. Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.	10	5	5	Решение задач с использованием циклов
4.	Раздел 4. Функции и модули в программировании	18	9	9	Тест по модулю
	4.1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	6	3	3	Тест по теме
	4.2. Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	12	6	6	Решение задач с использованием строк
5.	Раздел 5. Сложные типы данных	14	7	9	Тест по модулю
	5.1. Списки. Срезы списков. Решение задач со списками	8	3	5	Решение задач
	5.2. Матрицы	2	1	1	Решение задач
	5.3. Словари	2	1	1	Решение задач
	5.4. Множества в языке Python	2	1	1	Решение задач
	Раздел 6. Написание и отладка программ. Самостоятельная работа	12	0	12	Защита проекта
	Итого	72	15	57	

1.2.2. Содержание программы

Раздел 1. Введение в Python – 4 ч.

1.1. Инструкции и структура программы.

Теория: техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка- 2 ч.

1.2. Установка Python. Ввод и вывод данных – 2ч.

Теория: Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации – 1 ч.

Практика: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран – 1 ч.

Раздел 2. Типы данных и операции – 12 ч.

2.1. Переменные. Операторы – 4 ч.

Теория: 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 – 2 ч.

Практика: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор – 2 ч.

2.2 Типы данных. Базовые функции для работы с различными типами данных – 8 ч.

Теория: Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math) – 4 ч.

Практика: Решение математических задач с использованием функции import math – 4 ч.

Раздел 3. Инструкции и синтаксис – 14 ч.

3.1. Условные операторы if, elif, else – 4 ч.

Теория: 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 – 2 ч.

Практика: Практическое закрепление знаний по условным операторам. Создание программ – 2 ч.

3.2. Циклы while, for. Обработка исключений. Случайные числа – 10 ч.

Теория: 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 – 5 ч.

Практика: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного – 5 ч.

Раздел 4. Функции и модули в программировании – 18 ч.

4.1. Встроенные и пользовательские функции – 6 ч.

Теория: 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 – 3 ч.

Практика: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций – 3 ч.

4.2. Строки - последовательности символов – 12 ч.

Теория: 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 – 6 ч.

Практика: Написание программ. Работа со строками – 6 ч.

Раздел 5. Сложные типы данных – 14 ч.

5.1. Списки. Срезы списков – 8 ч.

Теория: 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 - 3 ч. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач – 3 ч..

Практика: Написание программ. Работа со списками – 5 ч.

5.2. Матрицы – 2 ч.

Теория: Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python – 1 ч.

Практика: Создание программ. Практическое закрепление полученных знаний – 1 ч.

5.3. Словари – 2 ч.

Теория: Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы – 1 ч.

Практика: Создание программ. Практическое закрепление полученных знаний – 1 ч.

5.4. Множества в языке Python – 2 ч.

Теория: Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству – 1 ч.

Практика: Создание программ. Практическое закрепление полученных знаний – 1 ч.

Раздел 6. Написание и отладка программ. Самостоятельная работа – 12 ч.

Практика: Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python».

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1. Методическое обеспечение программы

Методы обучения - словесный, наглядный, объяснительно-иллюстративный, практический, проектный.

Методы обучения – словесный, наглядный, объяснительно-иллюстративный, практический, проектный.

Формы организации образовательного процесса – в группах до 12 человек.

Дифференциация обучения – объединение в группу детей по

принципу учета состояния здоровья. Заключается в организации работы различной по содержанию, объёму, сложности, методам, приёмам и средствам в зависимости от психофизических возможностей ребенка (Л. А. Дружинина).

Индивидуальный подход – гибкое использование педагогом различных форм и методов педагогического воздействия с целью достижения оптимальных результатов образовательного процесса по отношению к каждому ребенку.

2.2. Условия реализации программы

Материально-техническое обеспечение:

Площадка проведения занятий оснащена спектром оборудования, средств обучения и воспитания для развития проектной деятельности обучающихся.

Для наиболее эффективного усвоения данной образовательной программы, занятия необходимо проводить в светлых помещениях с хорошей вентиляцией. Для того, чтобы работа с проектором была продуктивной, необходимо затемнять зону проектора, а рабочие места обучающихся должны быть достаточно освещены.

Перечень оборудования, необходимого для освоения общеобразовательной программы:

Наименование оборудования	Кол-во, шт.
Стул обучающегося	12
Стул педагога	1
Стол обучающегося	12
Стол педагога	1
Магнитно-маркерная доска	1
Проектор	1
Персональный компьютер обучающегося/планшет для обучающегося	12
Персональный компьютер педагога	1
Программное обеспечение PyCharm	13

Среда PyCharm находится в свободном для скачивания и установки доступе. Среда доступна для установки на Windows. Это значит, что PyCharm может быть установлена на компьютер/ноутбук с ОС Windows. .

Информационное обеспечение:

Для реализации дополнительной общеразвивающей программы

«Базовый курс: Программирование на языке Python» используются следующие материалы:

- учебно-методические пособия;
- конспекты лекций;
- комплект практических работ (Приложение 1);
- презентации;
- примеры программного кода;

Кадровые условия реализации программы

Программа реализуется педагогом дополнительного образования

2.3. Формы аттестации обучающихся

Входной контроль проводится в форме опроса.

Текущий контроль направлен на проверку уровня усвоения нового материала и выявление затруднений на ранней стадии. Текущий контроль проводится в следующих формах: самостоятельная работа, блиц-опрос.

Промежуточная аттестация: проводится в форме тестирования.

Итоговый контроль: проводится в форме представления и защиты проекта. Итоговая работа демонстрирует знания базовых навыков языка программирования Python в среде PyCharm, так же развитие пространственного и творческого мышления для решения поставленной задачи, проектирование дизайн и архитектуру приложения дополненной или виртуальной реальности. Тему итоговой работы определяет педагог в соответствии с уровнем усвоения программы, интересами и личностными особенностями обучающихся. Выполнение итоговой работы оценивается по следующим параметрам:

Набранные баллы	Уровень освоения
0-49	Низкий
50-79	Средний
80-100	Высокий

Описание уровней освоения:

- «Высокий уровень» - обучающийся самостоятельно выполняет все задачи на высоком уровне, его работа отличается оригинальностью идеи, грамотным исполнением и творческим подходом.
- «Средний уровень» - обучающийся справляется с поставленными перед ним задачами, но прибегает к помощи преподавателя. Работа выполнена, но есть незначительные ошибки.
- «Низкий уровень» - обучающийся выполняет задачи, но делает грубые ошибки (по невнимательности или нерадивости). Для завершения работы

необходима постоянная помощь преподавателя.

2.4. Оценочные материалы

Для отслеживания и фиксации результатов предусмотрены следующие формы контроля: опрос, дискуссия, самостоятельная работа, наблюдение.

Опросный лист.

Что больше всего привлекает вас в данном объединении?
(Внимательно прочтите все варианты ответа и выберите не более пяти, наиболее важных для Вас)

1. Возможность получать дополнительные знания.
2. Возможность приобрести навыки работы с различными материалами и инструментами.
3. Интересует данный вид деятельности.
4. Возможность лучше узнать себя, свои способности.
5. Хочу научиться работать самостоятельно.
6. Интересует программа объединения.
7. Хорошая материальная база объединения.
8. Возможность общения по интересу.
9. Здесь можно что-то сделать самому, - «своими руками».
10. Возможность фантазировать, придумывать и реализовывать свои идеи, «задумки».
11. Лучше сориентироваться с выбором профессии.
12. Возможность изобрести и помочь этим нашему обществу.
13. Привлекает личность руководителя объединения, с ним интересно работать.
14. В кружке много увлечённых ребят, с ними интересно общаться

Промежуточная аттестация проводится в форме выполнения самостоятельной работы по изученному материалу.

Аттестация по итогам освоения программы проводится в форме представления и защиты проекта. Итоговая работа демонстрирует навыки программирования, установления причинно-следственных связей, применения алгоритмического подхода, пространственного и творческого мышления для решения поставленной проблемы.

2.5 СПИСОК ЛИТЕРАТУРЫ

Для педагогов

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.
2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.
3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.
1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.
4. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.
5. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-oop/>

для обучающихся

1. Свейгар.Эл. Учим python, делая крутые игры М: Эксмо, 2018. – 416 с.
2. Семакин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2016. - 304 с.
3. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.

Интернет ресурсы

1. Уроки по Python для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>
2. Алгоритмизация. Программирование Python 3 [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/7215/promo>