

Общество с ограниченной ответственностью

«Проектная школа - Самара»

443015 г. Самара, ул. Главная, д.3, оф.20, тел. 8(927)777-09-67

E-mail: mironov@100lines.ru

УТВЕРЖДАЮ

Директор

ООО «Проектная школа — Самара»

Е.Н.Миронов

15 декабря 2021 г.



ДОПОЛНИТЕЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ

(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА

«ПРОГРАММИРОВАНИЕ НА RYTHON (БАЗОВЫЙ)»

Возраст детей: от 11 лет

Срок обучения: 72 академических часа

Автор-составитель:

Мальцев С.А., педагог ДО

г.о. Самара, 2021

Введение

Программа «Программирование на Python (базовый)» разработана в соответствии с требованиями к дополнительным общеобразовательным общеразвивающим программам технической направленности и направлена на формирование у обучающихся подростков основ алгоритмического мышления и навыков работы с реальным текстовым языком программирования.

Python — один из самых популярных и востребованных языков программирования в мире, применяемый в веб-разработке, анализе данных, машинном обучении, автоматизации и научных исследованиях. Его лаконичный синтаксис, читаемый код и огромная экосистема библиотек делают Python идеальным языком для первого знакомства с профессиональным программированием.

Актуальность программы обусловлена растущим спросом на IT-специалистов и необходимостью формирования у подростков устойчивых навыков программирования на реальном языке, а не только в визуальных средах. Программа обеспечивает плавный переход от блочного программирования (Scratch) к текстовому коду и закладывает фундамент для дальнейшего изучения современных технологий.

1. Пояснительная записка

Актуальность программы. В условиях цифровой трансформации экономики знание языка программирования становится такой же базовой компетенцией, как математика или иностранный язык. Python занимает лидирующие позиции в рейтингах популярности языков (Top-1 по версии IEEE, Stack Overflow) и является стандартом де-факто в области Data Science, искусственного интеллекта и веб-разработки. Программа актуальна, так позволяет обучающимся освоить универсальный инструмент, востребованный в самых разных сферах — от создания собственных игр и Telegram-ботов до анализа данных и автоматизации рутинных задач.

Отличительные особенности программы. Программа строится на принципах проектного обучения и постепенного погружения в профессиональную среду. В отличие от визуальных сред, здесь обучающиеся работают с настоящим текстовым кодом, учатся читать документацию, использовать IDE (интегрированную среду разработки), работать с терминалом и системой контроля версий Git на базовом уровне. Особое внимание уделяется формированию культуры написания чистого кода (PEP 8), отладке и умению самостоятельно находить решения через официальные источники.

Учебные дисциплины:

Основы алгоритмизации и логики (линейные, разветвляющиеся и циклические структуры). Синтаксис и семантика языка Python (типы данных, операторы, функции). Структуры данных (списки, кортежи, словари, множества). Основы объектно-ориентированного программирования. Работа с файловой системой и внешними библиотеками.

Практические дисциплины:

Написание консольных приложений и утилит. Создание текстовых игр и интерактивных сценариев. Визуализация данных с помощью библиотеки Turtle. Проектная деятельность (разработка, тестирование и защита собственных IT-проектов).

Цель образовательной программы: формирование у обучающихся устойчивых навыков программирования на языке Python, развитие алгоритмического и логического мышления, способности самостоятельно решать прикладные задачи и создавать собственные программные продукты.

Задачи программы:

Обучающие:

- познакомить с синтаксисом языка Python и средой разработки (IDE);
- научить составлять линейные, циклические и разветвляющиеся алгоритмы на языке Python;
- сформировать навыки работы с основными структурами данных: переменными, списками, словарями, кортежами;
- обучить созданию и использованию функций, работе с модулями и библиотеками;
- познакомить с основами объектно-ориентированного программирования (классы и объекты);
- обучить основам отладки кода и работе с исключениями.

Воспитывающие:

- воспитывать культуру написания чистого, читаемого кода (стандарт PEP 8);
- формировать уважение к интеллектуальной собственности, умение корректно использовать чужой код с указанием авторства и соблюдением лицензий;
- воспитывать настойчивость в преодолении трудностей, умение работать с документацией и самостоятельно искать решения.

Развивающие:

- развивать логическое, алгоритмическое и абстрактное мышление;
- развивать способность к декомпозиции сложных задач на простые подзадачи;
- развивать навыки самопрезентации, публичных выступлений и аргументации принятых технических решений.

2. Принципы, формы и методы обучения

Основными **принципами обучения** являются:

1. **Научность.** Изучаются фундаментальные концепции программирования, актуальные для любых языков (переменные, циклы, условия, функции, ООП), что создаёт прочный фундамент для дальнейшего профессионального роста.
2. **Доступность.** Сложные концепции объясняются через практические задачи и жизненные аналогии. Python обладает лаконичным синтаксисом, что снижает порог входа по сравнению с другими языками.
3. **Связь теории с практикой.** Каждая изученная тема немедленно закрепляется на практических задачах и мини-проектах.
4. **Воспитательный характер обучения.** Формируется культура работы с кодом, терпение при отладке, умение доводить начатое дело до конца.
5. **Активность обучения.** Обучающиеся не копируют код за педагогом, а понимают логику работы программы и предлагают собственные решения.
6. **Наглядность.** Использование интерактивной среды, визуализация алгоритмов, демонстрация готовых проектов.
7. **Систематичность и последовательность.** Материал подаётся от простого к сложному: от вывода текста на экран — к созданию полноценных приложений.
8. **Индивидуальный подход.** Педагог учитывает темп работы, уровень подготовки и интересы каждого обучающегося, предлагая задачи разного уровня сложности.

Формы организации образовательного процесса: сочетание фронтально-групповой и индивидуальной форм (групповое обсуждение идей, индивидуальное выполнение практических заданий, коллективное тестирование и код-ревью).

Методы обучения:

1. **Объяснительно-иллюстративный** — демонстрация готовых проектов, разбор примеров кода, работа с документацией.
2. **Эвристический** — создание собственных алгоритмов и решений.
3. **Проблемный** — постановка задачи («как реализовать игру «Угадай число» с подсчётом попыток?») и самостоятельный поиск решения.
4. **Программированный** — выполнение практических работ по пошаговым алгоритмическим картам.
5. **Репродуктивный** — написание кода по образцу с последующей модификацией.
6. **Поисковый** — решение задач с использованием официальной документации и справочных ресурсов.

7. Метод проектов — основной метод, включающий этапы: постановка задачи → проектирование → написание кода → тестирование → защита.

3. Механизм отслеживания результатов

Контроль освоения учебного материала проходит в три этапа:

1. Входной мониторинг — диагностика базовой компьютерной грамотности и опыта программирования (устный опрос, мини-тест).
2. Промежуточный контроль — текущие практические работы, защита мини-проектов (например, защита игры «Виселица» после изучения списков и циклов).
3. Итоговый мониторинг — создание и публичная защита собственного итогового проекта на основе пройденного материала.

В конце курса проводится анализ качества программы и при необходимости её коррекция для следующего набора.

4. Прогнозируемый результат

В конце обучения по данному курсу ученики должны знать:

- основы синтаксиса языка Python (переменные, типы данных, операторы, отступы);
- принципы работы с основными структурами данных (списки, кортежи, словари, строки);
- назначение и принципы работы циклов (for, while) и условных конструкций (if-elif-else);
- принципы создания и использования функций, область видимости переменных;
- основы работы с модулями и стандартными библиотеками (math, random, turtle);
- принципы обработки исключений (try-except);
- основы работы с файлами (чтение, запись);
- основы объектно-ориентированного программирования (классы, объекты, атрибуты, методы);
- правила написания чистого кода (стандарт PEP 8);
- правила цифровой гигиены и техники безопасности при работе за ПК.

В конце обучения по данному курсу ученики должны уметь:

- самостоятельно создавать консольные приложения на языке Python;
- писать алгоритмы с использованием циклов, условий и функций;
- работать со списками, словарями и строками (создание, изменение, перебор, срезы);

- создавать и применять собственные функции;
- подключать и использовать сторонние библиотеки;
- читать и понимать чужой код, работать с документацией;
- находить и исправлять синтаксические и логические ошибки (отладка);
- презентовать свою работу, аргументировать принятые проектные решения.

5. Учебно-тематический план обучения

№ п/п	Тема	часы		
		всего	теория	практ.
1	Тема 1. Установка Django. Создание проекта. Запуск сервера.	2	0,5	1,5
2	Тема 2. Приложения сайта. Структура проекта.	2	0,5	1,5
3	Тема 3. Создание HTML-страницы. Изучение первых тегов.	2	0,5	1,5
4	Тема 4. Продолжение работы с HTML. Структура документа.	2	0,5	1,5
5	Тема 5. Тег div и создание меню навигации.	2	0,5	1,5
6	Тема 6. Верстка страницы. Основы CSS и дизайна.	2	0,5	1,5
7	Тема 7. Продолжение верстки. Стилизация элементов.	2	0,5	1,5
8	Тема 8. Шаблонизатор Jinja2. Наследование шаблонов.	2	0,5	1,5
9	Тема 9. Jinja2 продолжение. Переменные в шаблонах.	2	0,5	1,5
10	Тема 10. Статические файлы. Подключение CSS и JS.	2	0,5	1,5
11	Тема 11. База данных. Модели Django. Миграции.	2	0,5	1,5
12	Тема 12. Публикуем новости на сайте. Админ-панель.	2	0,5	1,5
13	Тема 13. Добавление записей в базу данных через формы.	2	0,5	1,5
14	Тема 14. Переход по новости. Динамические URL.	2	0,5	1,5
15	Тема 15. Редактирование форм. Валидация данных.	2	0,5	1,5
16	Тема 16. Загрузка изображений в новости. Media files.	2	0,5	1,5
17	Тема 17. Алгоритмическая разминка. Треугольники и типы.	2	0,2	1,8
18	Тема 18. Списки и цикл for. Углубленная практика.	2	0,5	1,5
19	Тема 19. Словари. Методы и применение.	2	0,5	1,5
20	Тема 20. Операции со списками и словарями.	2	0,5	1,5

21	Тема 21. Решение алгоритмических задач.	2	0,2	1,8
22	Тема 22. Введение в анализ данных. Библиотека NumPy.	2	0,5	1,5
23	Тема 23. NumPy: практика работы с массивами.	2	0,5	1,5
24	Тема 24. Введение в Pandas. Объект DataFrame.	2	0,5	1,5
25	Тема 25. Pandas продолжение. Фильтрация и группировка.	2	0,5	1,5
26	Тема 26. Базы данных: повторение SQL запросов.	2	0,5	1,5
27	Тема 27. Интеграция Pandas и БД. Чтение/запись данных.	2	0,5	1,5
28	Тема 28. Введение во Flask. Маршрутизация.	2	0,5	1,5
29	Тема 29. Flask: шаблоны и обработка запросов.	2	0,5	1,5
30	Тема 30. Интерактивный квиз. Закрепление материала Python.	2	0,5	1,5
31	Тема 31. Завершаем проект на Flask. Деплой.	2	0,5	1,5
32	Тема 32. Введение в OpenCV. Работа с изображениями.	2	0,5	1,5
33	Тема 33. OpenCV. Поиск лиц на изображении.	2	0,5	1,5
34	Тема 34. Обучение модели для распознавания лица.	2	0,5	1,5
35	Тема 35. Реализация распознавания в реальном времени.	2	1,2	1,8
36	Тема 36. Итоговая защита проектов. Рефлексия курса.	2	0,0	2,0
	ИТОГО	72	16,6	55,4

6. Обеспечение программы

Для организации образовательного процесса по программе необходимы следующие ресурсы.

Кадровые: Специалист, имеющий высшее или среднее профессиональное образование (педагогическое, техническое или IT-направление), владеющий языком Python, знаниями и методикой преподавания курса, понимающий возрастную психологию подростков.

Материально-технические: Компьютерный класс, оснащённый персональными компьютерами или ноутбуками (1 место на 1 учащегося); настенная доска (маркерная или интерактивная); мультимедийный проектор и экран; система стабилизированного электроснабжения; доступ к сети Интернет. Помещение должно соответствовать санитарно-эпидемиологическим требованиям (СанПиН) к организации работы с ЭВМ.

Программно-методические:

1. Интерпретатор Python 3.x, установленный на каждый ПК.
2. Интегрированная среда разработки (PyCharm Educational, Thonny или VS Code).
3. Веб-браузеры с настроенной системой фильтрации контента для доступа к официальным ресурсам (python.org, metanit.com и др.).
4. Методическое пособие (поурочные планы, конспекты занятий).
5. Раздаточный материал (алгоритмические карты, шпаргалки по синтаксису, шаблоны проектов).
6. Мультимедийные презентации к каждому занятию.

7. Литература

Нормативно-правовая база:

1. Федеральный закон РФ «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
3. Концепция развития дополнительного образования детей до 2030 года.

Для педагога (учебная и методическая литература):

4. Степнов С. Python. Курс разработки. — СПб.: БХВ-Петербург, 2022.
5. Лутц М. Изучаем Python. — М.: Вильямс, 2021.
6. Яромко Т.В. Программирование на языке Python. — СПб.: БХВ-Петербург, 2022.
7. Гаврилов М.В. Самоучитель Python. — СПб.: Наука и техника, 2023.

Для обучающихся:

8. Маттисс Э. Python для детей. — М.: Манн, Иванов и Фербер, 2022.
9. Свейгарт Э. Автоматизация рутинных задач с помощью Python. — М.: Вильямс, 2022.
10. Доусон М. Програмируем на Python. — М.: Эксмо, 2023.

Электронные ресурсы:

11. Официальный сайт Python: <https://www.python.org>
12. Документация Python на русском: <https://docs-python.ru>
13. Metanit — учебник по Python: <https://metanit.com/python>

14. Stepik — курсы по программированию: <https://stepik.org> 15. GitHub — платформа для размещения проектов: <https://github.com>