

**Конспект занятия на тему:
«Введение в язык программирования Python»**

Возраст обучающихся: 12-16 лет

Год обучения: первый

Разработал: Ковригин Фёдор
Андреевич

Калининград

2025 год

Пояснительная записка

Занятие на тему «Введение в язык программирования Python» является вводным уроком курса программирования на Python. Его цель — познакомить учащихся с основами программирования и объяснить, что такое Python, а также постепенно ввести их в процесс написания кода через изучение основных понятий.

Занятие предназначено для учащихся 12–16 лет, которые обучаются на втором году.

Адресность: конспект занятия предназначен для проведения занятия с детьми от 12-16 лет.

Методические рекомендации:

В процессе работы в среде программирования VS Code необходимо обратить внимание учащихся на то, что файл должен иметь расширение .py в конце.

Если кто-то из учеников справится с заданием быстрее, можно предложить ему повторить выполнение программы, но создать её с использованием других инструментов.

В конце занятия можно предложить каждому учащемуся выразить своё отношение к выполненному заданию и рассказать о ходе его выполнения. После каждого выполненного задания необходимо совместно с учащимися проверить правильность его выполнения.

Предполагаемые результаты применения конспекта как методической продукции:

Данную методическую разработку можно применять систематически в работе с новыми учащимися. Одним из ключевых результатов использования данной методики на занятии будет формирование у учащихся интереса к изучению программирования на языке Python, а также получение базовых навыков написания программ.

Тип занятия: развитие знаний, умений и навыков.

Форма занятия: теоретическое + практическое.

Форма организации работы: групповая либо индивидуальная.

Состав учебной группы: до 10ти обучающихся в возрасте от 12 до 16 лет.

Степень сложности занятия: средний уровень сложности.

Цель: донести что такое программирование и научить писать простые программы

Задачи:

Обучающие задачи:

- Объяснить различия между языками программирования.
- Объяснить, что такое переменные и как они работают.
- Актуализировать знания обучающихся.

Развивающие задачи:

- Развивать навыки написания кода.
- Развивать память, внимание, образное мышление и умение решать задачи.
- Развивать логическое и пространственное мышление.
- Развивать умение работать индивидуально и в команде.

Воспитательные задачи:

- Повышать мотивацию обучающихся за счёт интереса к теме занятия.

Планируемые результаты занятия:

Личностные:

- формирование интереса к общей теме занятия; развитие мотивации в необходимости овладения знаниями и навыками по программированию на Python;
- понимание роли компьютера в жизни современного человека как инструмента для решения задач моделирования процессов и явлений.
- понимание определения переменной.

Методы обучения:

- теоретический (начало в виде лекции)
- практический (работа на компьютере).

Технологии:

- информационно-коммуникационные.

Дидактический материал:

- распечатанная лекция «Урок введение в программирование, Python» (Приложение 1);

Материально-техническое оснащение занятия:

- Интернет и техника для прослушивания у каждого ученика
- Микрофон (обязательно для преподавателя)

План занятия:

1. Организационная часть (5 мин).
2. Теория про языки программирования (10 минут)
3. Ответы на вопросы и обсуждение (10 минут)
4. Установка и настройка программ для работы с кодом (15 минут)
5. Перемена (10 мин)
6. Написание своей программы print(“куркума”) (5 минут)
7. Теория про переменные (10 минут)
8. Ответы на вопросы и обсуждение (10 мин)
9. Написание самостоятельно программы с применением переменных (10 минут)
10. Подведение итогов (5 мин).

Общая продолжительность занятия: 90 минут.

Ход учебного занятия

Этапы занятия	Деятельность педагога	Деятельность обучающихся
Организационная часть	Приветствует ребят Сообщает тему занятия	Рассаживаются за свои компьютеры
Теория	Интересно рассказывает лекцию, раскрывает подробности и отвечает на вопросы	Слушают, а затем задают вопросы и обсуждают рассказанную тему
Выполнение практического задания	Дает ссылки на скачивание, объясняет, как всё скачать и дает практику вместе с теорией	Скачивают необходимые программы, слушают и пишут код
подведение итогов	Отмечает положительные знания и умения обучающихся, дает необходимые рекомендации	Каждый обучающийся запускает свою программу

Список литературы

1. Банкрашков, А.В. Программирование для детей на языке Python / А.В. Банкрашков. - М.: АСТ, 2023. - 288 с.
2. Вордерман, К. Программирование на Python. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. - М.: Манн, Иванов и Фербер, 2022. - 346 с.
3. Воронин, И. Программирование для детей. От основ к созданию роботов / И. Воронин. - СПб.: Питер, 2025. - 192 с.
4. Воронин, И. Программирование для детей. От основ к созданию роботов / И. Воронин, В. Воронина. - СПб.: Питер, 2023. - 304 с.
5. Свейгарт, Э. Программирование для детей. Делай игры и учи язык Scratch! / Э. Свейгарт. - М.: Эксмо, 2025. - 200 с.
6. Торгашева, Ю.В. Программирование для детей. Учимся создавать игры на Scratch / Ю.В. Торгашева. - СПб.: Питер, 2022. - 608 с.
7. Торгашева, Ю.В. Программирование для детей. Учимся создавать игры на Scratch / Ю.В. Торгашева. - СПб.: Питер, 2018. - 128 с.
8. Торгашева, Ю.В. Программирование для детей. Мои первые программы на Scratch / Ю.В. Торгашева. - СПб.: Питер, 2024. - 608 с.
9. Торгашева, Ю.В. Программирование для детей. Мои первые программы на Scratch / Ю.В. Торгашева. - СПб.: Питер, 2024. - 96 с.
10. Уитни, Д. Программирование для детей. Учимся создавать сайты, приложения и игры. HTML, CSS и JavaScript / Д. Уитни. - СПб.: Питер, 2022. - 301 с.
11. Чистый код Роберт Мартин 2008.
12. <https://docs.python.org/3/>

Приложение 1

Урок 1 пайтон.

Тема урока: Пайтон начало.

Каждый сайт и приложение, которым вы пользуетесь - написано на подходящем для него языке программирования.

Все языки можно охарактеризовать по критерию быстрые и медленные, а также - совместимость с определенными платформами (айос, андроид, мак ос, виндовс)

Медленные языки работают медленно, но пишутся быстро, быстрое - наоборот, быстро работают, но долго пишутся (длиннее код, быстрее выполнение кода). На лёгких задачах скорость работы кода скорее всего не будет заметна человеку, но по мере усложнения программы - станет видно задержки.

Исходя из этого мы можем понять, что быстрый с++ подойдёт скорее всего для тяжёлых приложений и игр, где в первую очередь важна скорость работы программы, а не то, как долго придётся писать код. Именно для этой задачи он и используется

Js же, например, медленный язык, но код на нём пишется намного короче. Как думаете, для каких задач подойдёт такой язык?

Для легковесных сайтов, и приложений, при создании которых намного важнее скорость создания, чем скорость самого сайта.

Есть также специализированные языки, которые используются например только для разработки на айфон, или только для андроида.

Пайтон — это универсальный язык программирования. Он медленный(хотя и быстрее чем js) но позволяет писать и сайты, и приложения, и инди игры. А также его основной задачей является создание ИИ, в которой он превосходит всех остальных.

С этим языком мы и научимся работать на нашем курсе. Конечным результатом курса будет создание собственных приложений и сайтов целиком на пайтоне, а также тг ботов.

Для начала работы нам нужно будет скачать vs code и python

Давайте напишем свою первую программу `print("куркума")`

`Print` -, это функция, выводящая что-либо в терминал. Функции в vs коде подсвечены жёлтым. После функций всегда идут круглые скобки, внутри которых мы обычно вводим данные.

Если мы пишем текст в коде - мы должны обернуть его в кавычки. Цифры в кавычки оборачивать не нужно.

В пайтоне есть типы данных. И для него текст(str) и цифры(int) - разные типы. Причём цифры могут быть и в типе str и в типе int - если мы напишем цифру внутри кавычек - она будет работать как текст. Для того, чтоб увидеть различия в работе типов данных давайте взглянем на такие промеры:

`6 + 6 = 12`

"Привет" + "привет" = "приветпривет"

"6" + "6" = 66

В пайтоне с цифрами работают все математические символы умножение, деление, вычитание и тд. С текстом же возможно только умножение и плюс.

Переменная — это слово, которые является тем, чем вы его делаете с помощью =.

Например: num = 5

Print(num) выдаст нам 5, а num * num = 25

В переменные можно добавлять как текст, так и результаты любых вычислений и даже другие переменные

num2 = num + 1

Чему будет равен второй номер?

Сегодня мы познакомились с языками программирования и их назначениями, скачали всё необходимое для написания кода, изучили типы данных и переменные. Всем удачи!