**Министерство образования и науки Республики Ингушетия**

**ГАОУ «Гимназия №1 г. Назрань»**

**Федеральный проект «Успех каждого ребенка»**

|  |  |  |
| --- | --- | --- |
| **«Согласовано»**на заседании МО Протокол №1«\_\_\_» \_\_\_\_\_\_\_\_\_\_2020 г. | **«Рассмотрено»**на заседании педагогического советаПротокол №1 «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г. |  **«Утверждаю»** директор гимназии №1 \_\_\_\_\_\_\_\_\_М.Б.Парагульгов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2020 г. |

**Рабочая программа дополнительного образования**

 **«Программирование на языке Python»**

**Направленность:** техническая

**Уровень программы:** стартовый

**Возраст учащихся:** 12-16 лет (7-10 кл.)

**Срок реализации:** 2 года (156 часов)

**Составитель:** Сампиев Ахмет Магомед-Баширович,

педагог дополнительного образования

**2020-2021 учебный год**

**1. Пояснительная записка**

Дополнительная общеобразовательная программа технической направленности составлена на основе:

- Федерального закона № 273-ФЗ от 29.12.2012 «Об образовании в Российской федерации»;

- распоряжения Правительства Российской Федерации № 1726-р от 04.09.2014 «Концепция развития дополнительного образования детей»;

- приказа Министерства образования и науки Российской Федерации № 1008 от 29.08.2013 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- письма Министерства образования и науки Российской Федерации № 09-3242 от 18.11.2015 «Методические рекомендации по проектированию дополнительных общеразвивающих программ»;

- Федерального государственного образовательного стандарта начального общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 6 октября 2009 г. № 373, в ред. приказов от 26.11.2010 № 1241, от 22.09.2011 № 2357, от 18.12.12 № 1060, от 29.12.2014г. № 1643, от 18.05.2015г. № 507);

- Федерального государственного образовательного стандарта основного общего образования (Приказ министерства образования и науки РФ от 17 декабря 2010 года № 1897, в ред. приказа от 29.12.2014г. № 1644).

Актуальность введения дополнительной общеобразовательной программы «Программирование на языке Python» определяется необходимостью построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции − одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных учащихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов − блок-схем, алгоритмов, программ − способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

**Основное направление программы:** научно-техническое. Программа направлена на привлечение учащихся к освоению базовых приемов программирования и алгоритмизации.

**Цель программы:** освоение базовых знаний в области алгоритмизации и программирования; умение развития научно-технического и творческого потенциала личности ребёнка и формированию профессионального самоопределения учащихся в процессе программирования.

**Задачи программы:**

* Познавательная задача:

- развитие познавательного интереса к теории программирования и алгоритмизации, проектированию архитектуры кода, тестированию и предметам естественнонаучного цикла – математика, физика, технология, информатика.

* Образовательная задача:

- формирование умений и навыков: проектирования программного кода; алгоритмизации и написании кода; поиска и устранения ошибок в коде;

- приобретение опыта по проектированию и разработке программного продукта, алгоритмизации, тестированию и отладке программного кода;

- использование возможностей языка Python при составлении функции для решения определенной задачи, используя общепринятые лучшие практики при написании кода.

* Развивающая задача:

- развитие творческой активности, самостоятельности в принятии оптимальных решений в различных ситуациях;

- развитие психофизиологических качеств ученика: память, внимание, способность логически мыслить, анализировать, концентрировать внимание на главном;

- развитие у школьников инженерного мышления, навыков алгоритмизации, программирования и эффективного использования методов тестирования и отладки.

* Воспитывающая задача:

- воспитание ответственности, высокой культуры, дисциплины;

- умений работать в коллективе, эффективно распределять обязанности;

- повышение мотивации учащихся к программированию и созданию собственных программных продуктов.

**2. Организационно-педагогические основы организации учебного курса**

Настоящая программа учебного курса предназначена для учащихся 7-10 классов образовательных учреждений. Занятия проводятся 2 раза в неделю по 2 учебных часа (156 часов) в первый и второй год обучения. Срок реализации программы 2 года.

Форма занятий – групповая, индивидуальная.

Формы работы, используемые на занятиях:

• беседа;

• демонстрация;

• практикумы начинающего программиста, включающего проведение практических, исследовательских работ и прикладного программирования

• творческая работа;

• проектная деятельность.

Формы контроля и оценки образовательных результатов

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения обучающимися практических заданий.

Итоговый контроль реализуется в форме соревнований по программированию (региональный этап Джуниоры ВорлдСкилз и Республиканский чемпионат Ing Global Challenge, защите проекта (Региональный этап Балтийского научно-инженерного конкурса).

Способы проверки знаний у обучающихся:

педагогическое наблюдение, опрос, тестирование, самостоятельная работа, анализ творческих работ, участие в конкурсах, выставках и других мероприятиях.

Способы определения результативности заключаются в следующем:

• работы учеников будут зафиксированы на фото и видео в момент демонстрации созданных ими программных продуктов.

• фото и видео материалы по результатам работ учеников будут размещаться на официальном сайте школы.

• фото и видео материалы по результатам работ учеников будут представлены для участия на фестивалях и олимпиадах разного уровня.

Критериями выполнения программы служат:

знания, умения и навыки обучающихся, массовость и активность участия обучающихся в мероприятиях данной направленности.

1. **Содержание курса**
2. **Вводное занятие (2 часа).**

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

### Знакомство с языком Python (6 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест № 1. Знакомство с языком Python

*Учащиеся должны знать / понимать:*

* понятие программы;
* структура программы на Python;
* режимы работы с Python.

*Учащиеся должны уметь:*

* выполнить установку программы;
* выполнить простейшую программу в интерактивной среде;
* написать комментарии в программе.

### Переменные и выражения (11 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. МаРазделтические 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.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами Тест № 2. Выражения и операции.

*Учащиеся должны знать / понимать:*

* общую структуру программы;
* типы данных;
* целые, вещественные типы данных и операции над ними;
* оператор присваивания;

###  Условные предложения (28 часов)

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.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы". Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

*Учащиеся должны знать / понимать:*

* назначение условного оператора;
* способ записи условного оператора;
* логический тип данных;
* логические операторы or, and, not;
* Учащиеся должны уметь:
* использовать условный оператор;
* создавать сложные условия с помощью логических операторов.

### Циклы (46 часов)

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.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

*Учащиеся должны знать / понимать:*

* циклы с условием и их виды;
* правила записи циклов условием;
* назначение и особенности использования цикла с параметром;
* формат записи цикла с параметром;
* примеры использования циклов различных типов.

*Учащиеся должны уметь:*

* 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.

### Функции (20 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач c использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции" Тест № 5. Функции

*Учащиеся должны знать / понимать:*

* понятие функции;
* способы описания функции;
* принципы структурного программирования;
* понятие локальных переменных подпрограмм;
* понятие формальных и фактических параметров подпрограмм;
* способ передачи параметров.

*Учащиеся должны уметь:*

* создавать и использовать функции;
* использовать механизм параметров для передачи значений.

### Строки - последовательности символов (10 часов)

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.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

*Учащиеся должны знать / понимать:*

* назначение строкового типа данных;
* операторы для работы со строками;
* процедуры и функции для работы со строками;
* операции со строками.

*Учащиеся должны уметь:*

* описывать строки;
* соединять строки;
* находить длину строки;
* вырезать часть строки;
* находить подстроку в строке;
* находить количество слов в строке.

### Сложные типы данных (17 часов)

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. 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.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест № 7. Списки

*Учащиеся должны знать / понимать:*

* сложные типы данных;
* способ описания списка;
* способ доступа к элементам списка;
* способ описания кортежа;
* способ описания словаря;
* операции, выполняемые со списками, кортежами и словарями;
* понятие множества;
* способы описания множества;
* операторы работы с множествами.

*Учащиеся должны уметь:*

* 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.

### Стиль программирования и отладка программ (15 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

*Учащиеся должны знать / понимать:*

* что такое стиль программирования;
* правила именования объектов;
* основные рекомендации при написании программ.

*Учащиеся должны уметь:*

* определять вид ошибок и находить ошибки в программе.
* выполнять тестирование и отладку программ.
1. **Тематическое планирование**

|  |  |  |
| --- | --- | --- |
| **№****п/п** | **Наименование тем** | **Количество часов** |
| **Всего** | **Теория** | **Практика** |
| **Инструктаж по технике безопасности в компьютерном классе** | **2** | **2** |  |
| **1.** | **Знакомство с языком Python** | **6** | **2** | **4** |
| 1.1 | Занятие 1. Общие сведения о языкеПрактическая работа: Установка программы Python | 2 | 1 | 1 |
| 1.2. | Занятие 2. Режимы работыПрактическая работа: Режимы работы с Python | 2 | 1 | 1 |
| 1.3 | Тест № 1. Знакомство с языком Python | 2 |  | 2 |
| **2.** | **Переменные и выражения** | **11** | **4** | **7** |
| 2.1 | Занятие 3. ПеременныеПрактическая работа: Работа со справочной системой | 2 | 1 | 1 |
| 2.2 | Практическая работа: Переменные | 2 |  | 2 |
| 2.3 | Занятие 4. ВыраженияПрактическая работа: Выражения | 2 | 1 | 1 |
| 2.4 | Занятие 5. Ввод и вывод | 1 | 1 |  |
| 2.5 | Занятие 6. Задачи на элементарные действия с числамиПрактическая работа 2.5. Задачи на элементарные действия с числами | 2 | 1 | 1 |
| 2.6 | Тест № 2. Выражения и операции. | 2 |  | 2 |
| **3** | **Условные предложения** | **28** | **6** | **22** |
| 3.1 | Занятие 7. Логические выражения и операторы. Практическая работа: Логические выражения | 2 | 1 | 1 |
| 3.2 | Занятие 8. Условный операторПрактическая работа: «Условный оператор» | 4 | 1 | 3 |
| 3.3 | Занятие 9. Множественное ветвлениеПрактическая работа: Множественное ветвление | 4 | 1 | 3 |
| 3.4 | Занятие 10. Реализация ветвления в языке Python.Практическая работа: «Условные операторы» | 2 | 1 | 1 |
| 3.5 | Самостоятельная работа № 1 по теме «Условные операторы". | 10 | 2 | 8 |
| 3.6 | Занятие 11. Зачетная работа № 1. "Составление программ с ветвлением". | **4** |  | 4 |
| 3.7 | Тест № 3. "Условные операторы". | 2 |  | 2 |
| **4** | **Циклы** | **46** | **9** | **37** |
| 4.1. | Занятие 12. Оператор цикла с условиемПрактическая работа "Числа Фибоначчи" | 4 | 1 | 3 |
| 4.2. | Занятие 13. Оператор цикла forПрактическая работа Решение задачи с циклом for. | 4 | 1 | 3 |
| 4.3. | Занятие 14. Вложенные циклыПрактическая работа: Реализация циклических алгоритмов | 4 | 1 | 3 |
| 4.4. | Занятие 15. Случайные числаПрактическая работа: Случайные числа | 4 | 1 | 3 |
| 4.5. | Занятие 16. Примеры решения задач с цикломПрактическая работа: Решение задач с циклом. | 6 | 1 | 5 |
| 4.6 | Самостоятельная работа № 2 "Составление программ с циклом" | 10 | 2 | 8 |
| 4.7 | Тест № 4. Циклы | 2 |  | 2 |
| 4.8 | Занятие 17. Творческая работа № 1. "Циклы" | 12 | 2 | 10 |
| **5** | **Функции** | **20** | **6** | **14** |
| 5.1. | Занятие 18. Создание функцийПрактическая работа Создание функций | 2 | 1 | 1 |
| 5.2. | Занятие 19. Локальные переменныеПрактическая работа Локальные переменные | 2 | 1 | 1 |
| 5.3. | Занятие 20. Примеры решения задач c использованием функцийПрактическая работа Решение задач с использованием функций | 2 | 1 | 1 |
| 5.4 | Самостоятельная работа № 3 по теме "Функции" | 10 | 2 | 8 |
| 5.5 | Занятие 21. Рекурсивные функцииПрактическая работа 5.4. Рекурсивные функции | 2 | 1 | 1 |
| 5.6 | Тест № 5. Функции | 2 |  | 2 |
| **6** | **Строки - последовательности символов** | **10** | **3** | **7** |
| 6.1. | Занятие 22. СтрокиПрактическая работа: Строки | 2 | 1 | 1 |
| 6.2. | Занятие 23. Срезы строк | 2 | 1 | 1 |
| 6.3 | Занятие 24. Примеры решения задач со строкамиПрактическая работа: Решение задач со строками. | 6 | 1 | 5 |
| **7** | **Сложные типы данных** | **17** | **7** | **10** |
| 7.1. | Занятие 25. СпискиПрактическая работа: Списки | 2 | 1 | 1 |
| 7.2 | Занятие 26. Срезы списков | 1 | 1 |  |
| 7.3 | Занятие 27. Списки: примеры решения задачПрактическая работа 7.2. Решение задач со списками | 6 | 1 | 5 |
| 7.4 | Занятие 28. Матрицы | 2 | 1 | 1 |
| 7.5 | Тест № 7. Списки | 2 |  | 2 |
| 7.6 | Занятие 29. Кортежи | 1 | 1 |  |
| 7.7 | Занятие 30. Введение в словари | 1 | 1 |  |
| 7.8 | Занятие 31. Множества в языке Python | 2 | 1 | 1 |
| **8** | **Стиль программирования и отладка программ** | **17** | **7** | **10** |
| 8.1 | Занятие 32. Стиль программирования | 2 | 1 | 1 |
| 8.2 | Занятие 33. Отладка программ | 2 | 1 | 1 |
| 8.3 | Занятие 34. Зачет по курсу «Программирование на языке Python» | 10 | 2 | 8 |
| 8.3 | Занятие 35. Что дальше? | 2 | 2 |  |
|  | **ВСЕГО** | **156** | **45** | **111** |

1. **Планируемые результаты изучения курса «Программирование на языке Python»**

**Личностные результаты**

К личностным результатам освоения курса можно отнести:

* готовность и способность обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
* сформированность целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики.
* осознанное, уважительное и доброжелательное отношение к другому человеку, его мнению, мировоззрению, языку, гражданской позиции.
* готовность и способность вести диалог с другими людьми и достигать в нем взаимопонимания.
* освоенность социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах.
* развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
* начало профессионального самоопределения, ознакомление с миром профессий, связанных с программированием.

**Регулятивные умения**

умение самостоятельно определять цели обучения, ставить и формулировать новые задачи в познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности.

* умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения познавательных задач.
* умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией.
* умение оценивать правильность выполнения познавательной задачи, собственные возможности ее решения.
* владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в познавательной деятельности.

**Познавательные умения**

* умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение и делать выводы.
* умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения познавательных задач.
* осуществлять поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательной организации, в федеральных хранилищах информационных образовательных ресурсов;
* использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач.

**Коммуникативные умения**

умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение.

* умение осознанно использовать речевые средства в соответствии с задачей коммуникации для выражения своих чувств, мыслей и потребностей для планирования и регуляции своей деятельности.
* формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

**Предметные результаты**

В результате работы по программе обучающиеся **научатся:**

* работать с литературой, с журналами, с Интернет-ресурсами (изучать и обрабатывать информацию);
* самостоятельно решать технические задачи в процессе разработки программного продукта (планирование предстоящих действий, самоконтроль, применять полученные знания, приемы и опыт конструирования с использованием специальных элементов и т.д.);
* осуществлять проектирование программных продуктов;
* создавать программы на языке Python под те или иные технические задачи;
* производить тестирование собственных разработанных программных продуктов.

В результате работы по программе курса дети ***получат возможность научиться:***

* алгоритмизировать задачи и разрабатывать архитектуру кода;
* работать в среде программирования и разрабатывать программные продукты;

довести решение задачи до работающей модели.

1. **Методическое и материально-техническое обеспечение образовательной деятельности**

***Литература для учителя:***

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Мэтиз Эрик Изучаем Python. Программирование игр, визуализация данных, веб-приложения. 2-е изд. – СПб.: Питер, 2018
4. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб.: Питер, 2018
5. Плас Дж. Вандер Python для сложных задач: наука о данных и машинное обучение. – СПб.: Питер, 2018

***Литература для родителей и учащихся***

1. Сайт / справочные материалы [Электронный ресурс] − Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] − Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python [Электронный ресурс] − Режим доступа: <https://stepik.org/course/431> , свободный.
4. Сайт / среда разработки для языка Python [Электронный ресурс] − Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.

Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] − Режим доступа: openbookproject.net , свободный.

***Интернет ресурсы***

1. Официальный сайт программы [Электронный ресурс] − Режим доступа: <https://docs.python.org/>, свободный.
2. Сайт, среда разработки для языка Python. [Электронный ресурс] − Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

***Материально-техническое обеспечение:***

* столы для компьютера;
* компьютерные стулья;
* шкафы для дидактических материалов, пособий;
* специальная и научно-популярная литература для педагога и учащихся;
* канцтовары;
* персональный компьютер (на каждого участника):
	+ Процессор не ниже Core i3;
	+ Объем оперативной памяти не ниже 4 Гб, DDR4;
	+ Дисковое пространство на менее 128 Гб;
	+ Монитор диагональю на менее 21”;
* мультимедийный проектор;
* видеоматериалы разной тематики по программе;
* оргтехника;
* выход в сеть Internet;