

Муниципальное общеобразовательное учреждение –
средняя общеобразовательная школа №3 им. Л.Г. Венедиктовой
г. Маркса Саратовской области

Центр образования естественнонаучного и технологического профилей
«Точка роста»

ПРИНЯТА на заседании педагогического совета « <u>29</u> » <u>августа</u> 2025 г. Протокол № <u>1</u>	УТВЕРЖДАЮ Директор МОУ СОШ №3 Хорина О.В. « <u>29</u> » <u>августа</u> 2025 г.
--	--

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Основы программирования на Python»**

Направленность программы: техническая

Возраст обучающихся: 14-16 лет

Срок реализации программы: 1 год

Автор-составитель:
педагог дополнительного
образования, реализующий
программу
Патракова Светлана Александровна

Маркс
2025 год

1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

1.1. Пояснительная записка

Направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность.

Актуальность программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирование невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

Педагогическая целесообразность программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Отличительной особенностью программы является её направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Данная программа разработана согласно Положению о структуре, порядке разработки и утверждению дополнительных общеобразовательных общеразвивающих программ МОУ-СОШ №3 г. Маркса. Приказ №210 от 29.08.2023 г.

Адресат программы:

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 14-16 лет.

Возрастные особенности. Подростковый возраст часто называется переходным, так как это этап перехода от детства к юности. Подростки характеризуются значительными сдвигами в мышлении и познавательной деятельности. В связи с этим необходимо придавать процессу обучения проблемный характер, учить подростков самостоятельному нахождению и формулированию проблем, выработке у них аналитико-синтетических умений, способности делать теоретические обобщения. Не менее важно развивать навыки самостоятельной учебной работы.

Количество обучающихся в группе: 10-15 человек

Срок реализации программы: 9 месяцев

Объем программы – 34 часа

Режим занятий: 1 раз в неделю по 1 часу

Форма обучения. Очная

Режим занятий. Продолжительность одного академического часа - 40 мин. Общее количество часов в неделю – 1 час.

1.2. Цель и задачи программы

Цель программы: Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи программы:

Обучающие:

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

Развивающие:

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;

Воспитательные:

- повышение общекультурного уровня;
- развитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

1.3. Планируемые результаты освоения программы

Предметные:

- Знают основные алгоритмические конструкции и правила их записи, с основными способами организации данных.
- Умеют составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Распознают необходимость применения той или иной алгоритмической конструкции при решении задач.
- Умеют организовывать данные для эффективной алгоритмической обработки.
- Умеют разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Могут осуществлять отладку и тестирование программы.

Метапредметные:

- владеют умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- определяют последовательность промежуточных целей с учетом конечного результата, разбивают задачи на подзадачи, разрабатывают последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;

Личностные:

- Повышен общекультурный уровень;
- владеют навыками сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- умеют проявлять инициативность и настойчивы в преодолении трудностей.

**1.4. Содержание программы.
Учебный план**

№ п /п	Тема занятий	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практик а	
1	Раздел 1. Знакомство с языком Python	3	1	2	Опрос Тестирование, решение практических задач
1.1	Инструктаж по технике безопасности в компьютерном классе Общие сведения о языке Практическая работа: Установка программы Python	2	1	1	
1.2.	Режимы работы Практическая работа: Режимы работы с Python	1		1	
2.	Раздел 2. Переменные и выражения	4	0,5	3,5	Тестирование, решение практических задач
2.1	Переменные и выражения Практическая работа: Работа со справочной системой	1		1	
2.2	Ввод и вывод	1	0,5	0,5	
2.3	Задачи на элементарные действия с числами Практическая работа	1		1	
2.4.	Задачи на элементарные действия с числами	1		1	
3.	Раздел 3. Условные предложения	4	1	3	Тестирование, решение практических задач
3.1	Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1	
3.2	Условный оператор Практическая работа: «Условный оператор»	1		1	
3.3	Множественное ветвление Практическая работа: Множественное ветвление	1		1	
4	Раздел 4. Циклы	4	2	2	Тестирование, решение практических задач
4.1.	Оператор цикла с условием Практическая работа "Числа	1	0,5	0,5	

	Фибоначчи"				
4.2.	Оператор цикла for Практическая работа Решение задачи с циклом for.	1	0,5	0,5	
4.3.	Вложенные циклы Практическая работа: Реализация циклических алгоритмов	1	0,5	0,5	
4.4.	Случайные числа Практическая работа: «Случайные числа»	1	0,5	0,5	
5	Раздел 5. Функции	4	2	2	Тестирование, решение практических задач
5.1.	Создание функций Практическая работа «Создание функций»	1	0,5	0,5	
5.2.	Локальные переменные Практическая работа «Локальные переменные»	1	0,5	0,5	
5.3.	Примеры решения задач с использованием функций Практическая работа «Решение задач с использованием Функций»	1	0,5	0,5	
5.4	Рекурсивные функции Практическая работа Рекурсивные функции	1	0,5	0,5	
6	Раздел 6. Строки - последовательности символов	3	1	2	Решение практических задач
6.1.	Строки Практическая работа: Строки	1		1	
6.2.	Срезы строк	1	1		
6.3	Примеры решения задач со строками Практическая работа: Решение задач со строками.	1		1	
7.	Раздел 7. Сложные типы данных	8	0,5	7,5	Тестирование, решение практических задач
7.1.	Списки Практическая работа: Списки	1	0,5	0,5	

7.2	Срезы списков	1		1	
7.3	Списки: примеры решения задач Практическая работа Решение задач со списками	1		1	
7.4	Матрицы	1		1	
7.5	Кортежи	1		1	
7.6	Введение в словари	1		1	
7.7	Множества в языке Python	2		2	
8	Раздел 8. Стил программирования и отладка программ	4	1	3	Решение практических задач, презентация проекта, рефлексия.
8.1	Стил программирования	1	1		
8.2	Отладка программ	1		1	
8.3	Зачет по курсу «Программирование на языке Python»	2		2	
ВСЕГО		34	9	25	

Содержание учебного плана

Раздел 1. Знакомство с языком Python

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

Практика:

- Практическая работа 1.1. Установка программы Python
- Практическая работа 1.2. Режимы работы с Python

Раздел 2. Переменные и выражения

Теория: Типы данных. Преобразование типов. 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. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практика:

- Практическая работа 2.1. Работа со справочной системой
- Практическая работа 2.2. Переменные и выражения
- Практическая работа 2.4. Задачи на элементарные действия с числами

Раздел 3. Условные предложения

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.

Практика:

- Практическая работа 3.1. Логические выражения
- Практическая работа 3.2. "Условный оператор"
- Практическая работа 3.3. Множественное ветвление

Раздел 4. Циклы

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.

Практика:

- 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

Раздел 5. Функции

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. Числа Фибоначчи.

Практика:

- 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

Раздел 6. Строки - последовательности символов

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.

Практика:

- Практическая работа 6.1. Строки
- Практическая работа 6.2. Решение задач со строками.

Раздел 7. Сложные типы данных

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.

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.

Практика:

- Практическая работа 7.1. Списки.
- Практическая работа 7.2. Решение задач со списками.
- Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Раздел 8. Стиль программирования и отладка программ

Теория: Стиль программирования. Отладка программ.

Практика:

- Отладка программ
- **Зачет** по курсу «Программирование на языке Python»

1.5. Формы аттестации/контроля

Входной контроль осуществляется при комплектовании группы в начале учебного года.

Цель – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

2. Комплекс организационно-педагогических условий

2.1 Методическое обеспечение

Методы обучения:

Выбор метода обучения зависит от содержания занятий, уровня подготовки и опыта обучающихся.

Традиционные методы развития интереса обучающихся к техническому творчеству можно объединить в группы, каждая из которых наиболее продуктивна на том или ином этапе работы.

1. **Словесный:** объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. **Наглядный:** применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. **Практический:** индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. **Интерактивный:** создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Педагогические технологии, применяемые в реализации программы:

1. **Технология проблемного диалога.** Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.
2. **Технология коллективного взаимообучения** («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО)», «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. **Игровая технология.** Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.
4. Элементы **здоровьесберегающих технологий** являются необходимым условием снижения утомляемости и перегрузки учащихся.
5. **Проектная технология** предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.
6. **Информационно-коммуникационные технологии** активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

Формы организации учебного занятия: комбинированное занятие; беседа; игра; викторина, творческая встреча, соревнование, экскурсия и др.

Алгоритм учебного занятия.

I этап - организационный.

Задача: подготовка детей к работе на занятии.

Содержание этапа: организация начала занятия, создание психологического настроя на учебную деятельность и активизация внимания.

II этап - подготовительный (подготовка к восприятию нового содержания).

Задача: Организовать занимающихся, сообщение задач занятия, подготовка к основной части занятия, к предстоящей работе. Содержание этапа: сообщение темы, цели учебного занятия и мотивация учебной деятельности детей (пример, познавательная задача, проблемное задание детям).

III этап - основной.

В качестве основного этапа могут выступать следующие:

1. Усвоение новых знаний и способов действий.

Задача: обеспечение восприятия, осмысления и первичного запоминания связей и отношений в объекте изучения. Целесообразно при усвоении новых знаний использовать задания и вопросы, которые активизируют познавательную деятельность детей.

2. Первичная проверка понимания.

Задача: установление правильности и осознанности усвоения нового учебного материала, выявление неверных представлений, их коррекция. Применяют пробные практические задания, которые сочетаются с объяснением соответствующих правил или обоснованием.

3. Закрепление знаний и способов действий, применяют творческие задания, выполняемые детьми самостоятельно.

4. Обобщение и систематизация знаний.

Задача: формирование целостного представления знаний по теме. Распространенными способами работы являются беседа и практические задания.

IV этап – контрольный.

Задача: выявление качества и уровня овладения знаниями, их коррекция.

Используются виды устного и письменного опроса, вопросы и задания различного уровня сложности (репродуктивного, творческого, поисково-исследовательского). V этап - итоговый.

Задача: дать анализ и оценку успешности достижения цели и наметить перспективу последующей работы.

Содержание этапа: педагог сообщает, как работали обучающиеся на занятии, что нового узнали, какими умениями и навыками овладели.

VI этап - рефлексивный.

Задача: мобилизация детей на самооценку. Может оцениваться работоспособность, психологическое состояние, результативность работы, содержание и полезность учебной работы.

Изложенные этапы могут по-разному комбинироваться, какие-либо из них могут не иметь места в зависимости от педагогических целей и учитывая формы занятия.

2.2. Условия реализации дополнительной общеобразовательной общеразвивающей программы «Программирование на языке Python»

Материально-техническое обеспечение:

Помещение: кабинет для занятий, расположенный по адресу Саратовская область, г. Маркс пр.Строителей 22. Кабинет, соответствующий санитарно-гигиеническим требованиям.

Для реализации образовательной программы используются:

1. столы для компьютера;
2. компьютерные стулья;
3. шкафы для дидактических материалов, пособий;
4. специальная и научно-популярная литература для педагога и учащихся;
5. канцтовары.

Информационно-методическое обеспечение:

1. персональный компьютер;
2. мультимедийный проектор;
3. видеоматериалы разной тематики по программе;
4. оргтехника;
5. выход в сеть Internet;

Аппаратное обеспечение:

1. Процессор не ниже Core2 Duo;
2. Объем оперативной памяти не ниже 4 Гб DDR3;
3. Дисковое пространство на менее 128 Гб;
4. Монитор диагональю не менее 19”;

Программное обеспечение:

1. Операционная система AstraLinux Профессиональная или выше;
2. Интерпретатор Python версии 3.8 и выше;
3. IDE JupyterLab/PyCharm;
4. FoxitReader или другой просмотрщик PDF файлов;
5. Пакет офисных программ;
6. Любой браузер для интернет-серфинга.

Дидактические материалы:

- - таблицы;
- - схемы;

- - памятки;
- - видеозаписи, мультимедийные материалы;
- - упражнения.

2.3. Оценочные материалы

Диагностируются результаты различным образом и на различных этапах деятельности. Для определения уровня знаний и умений обучающихся проводятся: в начале года - входной контроль, в середине года - промежуточный контроль, в конце года — итоговый контроль. Основной способ оценки – самооценка выполнения программ на языке программирования. Эта оценка присутствует на всех этапах. Одной из форм диагностики результатов является зачет или контрольная работа.

Входящий контроль: Тестирование

1. Что выведет следующий код?

```
x = 5
y = 10
print(x + y * 2)
```

1. 20
2. 25
3. 30
4. Ошибка

2. Какой тип данных возвращает функция input() в Python?

1. int
2. float
3. str
4. bool

3. Какой оператор используется для возведения числа в степень в Python?

1. ^
2. **
3. *
4. //

4. Как создать список в Python?

1. list = {1, 2, 3}
2. list = [1, 2, 3]
3. list = (1, 2, 3)
4. list = <1, 2, 3>

5. Какой метод списка добавляет элемент в его конец?

1. insert()
2. append()
3. add()
4. push()

6. Какой результат выведет следующий код?

```
x = [1, 2, 3]
print(x[1])
```

1. 1
2. 2
3. 3
4. Ошибка

7. Какой метод используется для преобразования строки в целое число?

1. int()
2. str()
3. float()

4. chr()
8. Какой результат будет у выражения `5 // 2` в Python?
1. 2.5
 2. 2
 3. 3
 4. 0
9. Какой цикл повторяется, пока условие истинно?
1. for
 2. while
 3. erepeat
 4. foreach

Промежуточный контроль: *Тестирование, решение задач*

Задача 1. Сумма двух чисел

Напишите программу, которая принимает два числа от пользователя и выводит их сумму.

Задача 2. Проверка чётности

Запросите у пользователя число и выведите, чётное оно или нечётное.

Задача 3. Таблица умножения

Составьте таблицу умножения для числа, введённого пользователем. Таблица должна идти от 1 до 10.

Задача 4. Найти факториал

Рассчитайте факториал числа. Факториал — это произведение всех чисел от 1 до заданного.

Задача 5. Переворот строки

Запросите строку у пользователя и выведите её в обратном порядке.

Задача 6. Палиндром

Напишите программу, которая проверяет, является ли введённая строка палиндромом (читается одинаково в обе стороны).

Задача 7. Максимальное из трёх чисел

Пользователь вводит три числа. Программа должна найти и вывести наибольшее из них.

Задача 8. Подсчёт гласных в строке

Введите строку и подсчитайте, сколько в ней гласных букв (а, е, ё, и, о, у, ы, э, ю, я).

Задача 9. Простое ли число

Напишите программу, которая определяет, является ли заданное число простым (делится только на 1 и на само себя).

Задача 10. Генерация списка квадратов

Создайте список квадратов всех чисел от 1 до 10 с помощью list comprehension.

Итоговый контроль: *учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

2.4. Список литературы

Литература для обучающихся и родителей:

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.
5. Сайтпроекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net , свободный.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.

Интернет-ресурсы

Лутц, Марк Python. Справочник. – М.: Вильямс, 2015. 3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный. 4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: