 Государственное бюджетное профессиональное образовательное учреждение Краснодарского края

«Краснодарский торгово – экономический колледж»

Жоха В. Н.

**Методическая разработка**

**открытого урока**

по теме: «Основы языка Python»

(ОУД 05. Информатика)

Краснодар, 2024

Разработчик: Жоха В. Н. – преподаватель математики и информатики ГБПОУ КК «КТЭК»

Рецензенты:

|  |  |
| --- | --- |
|  |  |
|  |  |

Методическая разработка открытого урока по теме: «Основы языка Python» в рамках учебного предмета ОУД 05. Информатика предназначена для преподавателей в целях обмена опытом работы в проведении уроков с использованием технологии комбинированного обучения по специальностям:

* Экономика и бухучет (по отраслям);
* Банковское дело;
* ПРОФЕССИОНАЛИТЕТ: Сервис на транспорте;
* ПРОФЕССИОНАЛИТЕТ: Туризм и гостиничное дело;

Утверждено на заседании цикловой методической комиссией ЦМК ПП МО №1

Протокол № \_\_ от «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

Председатель ЦМК \_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Пояснительная записка | 4 |
| План урока | 7 |
| Список используемой литературы | 16 |

Пояснительная записка

Формирование у студентов профессиональных компетенций является обязательным требованием, в связи с переходом учебных заведений на государственные стандарты нового поколения. Профессиональные компетенции обеспечат будущим специалистам более полную подготовку в своей будущей профессиональной деятельности.

В наше время, когда современные технологии становятся все более важными и влияют на все сферы нашей жизни, умение программировать становится неотъемлемым навыком. Программирование – это язык будущего, с помощью которого создаются новые, инновационные и полезные продукты и сервисы для современного общества.

Освоение программирования полезно не только для профессионалов в этой области, но и для каждого человека. Навыки программирования помогают развивать логическое мышление, улучшать критическое мышление и способствуют развитию аналитических навыков.

Еще одно преимущество обучения программированию – это рост возможностей для карьеры. В настоящее время все больше компаний нуждаются в специалистах, которые могут разрабатывать и поддерживать программное обеспечение, создавать приложения для своей работы при выполнении задач, входящих в круг специалиста. Умение программировать делает такого специалиста конкурентоспособным на рынке труда и открывает двери к новым карьерным возможностям. Ведь в будущем практически все сферы деятельности будут связаны с информационными технологиями.

Таким образом, освоение основ программирования является важным и полезным навыком для каждого человека, умение программировать поможет развивать важные навыки и повышать свою конкурентоспособность на рынке труда. Эти навыки открывают перед молодыми специалистами новые возможности и позволят, в частности, вносить вклад в современное общество с помощью информационных технологий.

Методическая разработка состоит из основных разделов:

* подготовительный этап, в котором планируется порядок проведения урока;
* структура занятия состоит в подробном распределении элементов занятия и изучаемых вопросов, а также методов обучения и времени, необходимого для изучения темы;
* методика проведения занятия включает в себя подробное описание каждого элемента занятия и его цели;
* заключительный этап анализирует результаты работы и подводит итоги проведенного урока.

**План урока**

**Тема:** «Основы языка Python»

**Методическая цель урока:** использование технологий комплексного обучения для активизации профессиональной деятельности студентов.

**Дидактические цели урока:**

*- образовательные* – сформировать знания об одном популярном языке программирования Python и показать профессиональную значимость полученных знаний;

- *развивающие* – развивать познавательную активность, учебную самостоятельность, профессиональное внимание, умение использовать полученные знания в практической деятельности.

*- воспитательные* – содействовать воспитанию положительных мотивов учебно-познавательной, трудовой деятельности, прививать навыки коллективного труда, интерес к будущей профессии.

**Тип урока:** урок изучения нового языка программирования, новой среды программирования, новых понятий программирования.

**Вид учебного занятия:** урок-практическое занятие, с элементами беседы, показа примеров и решения заданий в среде Google Colab.

**Форма организации:** объяснение, решение примеров и выполнение практических примеров.

**Средства обучения:**

-Интерактивная среда программирования Google Colab;

- обучающие конспекты.

**Междисциплинарные связи:** информатика

**Студент должен знать:**

- основные понятия информатики, файл, запись в файл, сохранение, типы данных, синтаксис при построении арифметических операций.

**Студент должен уметь:**

* запускать интерактивную среду программирования Google Colab;
* подключать используемые в коде библиотеки Python, используя директиву import;
* объявлять переменные различных типов;
* строить на языке программирования арифметические выражения с использованием арифметических операторов: +;-; \*; /; //; %.
* Вычислять значения арифметических выражений по заданным в коде частным числовым значениям;
* Строить алгебраические выражения с арифметическими операторами без использования библиотеки math;
* Строить алгебраические выражения с использованием простейших функций из библиотеки math;
* Строить алгебраические выражения, в которых значения переменных вводятся из среды программирования;
* Оформлять написанный код в соответствии с требованиями внутреннего кодекса PEP-8.

**Ход урока**

1. *Организационный момент (3 мин):*

* проверка присутствующих, готовности студентов к уроку;
* сообщение темы и цели урока;

1. *Актуализация знаний, обучающихся методом фронтального опроса (5 мин):*
2. Что такое программирование?
3. Назовите несколько языков программирования!
4. Что такое программа?
5. Что такое алгоритм?
6. Какие виды алгоритмов вы знаете?
7. Что такое данные и чем они отличаются от значений вычисленных данных?

*3.Мотивация учебной деятельности (2 мин)*

Уважаемые студенты!

Вы на этом занятии познакомитесь с основами программирования на одном из самых популярных языков программирования соверменности - языком Python (Питон), на котором будем учиться программировать в среде Google Colab, закрепите полученные знания.

Вы познакомитесь с принципами и правилами объявления переменных различных типов, с объявлением – декларированием подключаемых библиотек, с формированием арифметических выражений и алгебраических выражений как для простейших выражений, так и более сложных алгебраических выражений, требующих использования библиотечных функций, а также познакомитесь с техникой ввода данных в программу.

*4. Изучение нового материала*

**Из истории создания языка Питон**

 В феврале 1991 года сотрудник центра математики и информатики в Нидерландах, **Гвидо ван Россум**, опубликовал исходный код нового на тот момент языка программирования — **Python**.

Создание языка было его личной инициативой. Изначально у проекта не было официального бюджета и все делалось только на энтузиазме.

Поэтому Гвидо Ван Россум стремился экономить время и определил для себя несколько правил, одно из которых — это «заимствовать идеи отовсюду, откуда это имеет смысл».

Язык программирования Python - это интерпретируемый, объектно-ориентированный язык программирования высокого уровня, предназначенный для решения самого широкого круга задач.

С его помощью можно обрабатывать числовую и текстовую информацию, создавать изображения, работать с базами данных, разрабатывать Web-сайты и приложения с графическим интерфейсом.

Python - язык кроссплатформенный, он позволяет создавать программы, которые будут работать во всех операционных системах.

Программа на языке Python представляет собой обычный текстовый файл с расширением \*.py (консольная программа) или \*.pyw (программа с графическим интерфейсом). При работе среде программирования Google Colab файл создается с расширением \*.jpynb. Все инструкции из этого файла выполняются интерпретатором построчно.

Python относится к категории языков объектно-ориентированных. Это означает, что практически все данные в нем являются объектами, даже значения, относящиеся к элементарным типам, наподобие чисел и строк, а также сами типы данных. В переменной всегда сохраняется только ссылка на объект, а не сам объект. Например, можно создать функцию, сохранить ссылку на нее в переменной, а затем вызвать функцию через эту переменную. Такое обстоятельство делает язык Python идеальным инструментом для создания программ, использующих функции обратного вызова, - например, при разработке графического интерфейса. Тот факт, что язык является объектно-ориентированным, отнюдь не означает, что и объектно-ориентированный стиль программирования (ООП) является при его использовании обязательным. На языке Python можно писать программы как в стиле ООП, так и в процедурном стиле, - как того требует конкретная ситуация или как

предпочитает программист. Python не допускает двоякого написания кода: код можно написать только одним способом; в нем отсутствуют лишние конструкции. Все программисты должны придерживаться стандарта РЕР-8, описанного в документе https://www.python.org/dev/peps/pep-0008/.

**Интерактивная среда Google Colab**

Google Colab - это среда для разработки и выполнения программного кода в облаке. Она предоставляет возможность писать и запускать код на языке Python, используя только браузер, без установки специальных программ на компьютер.

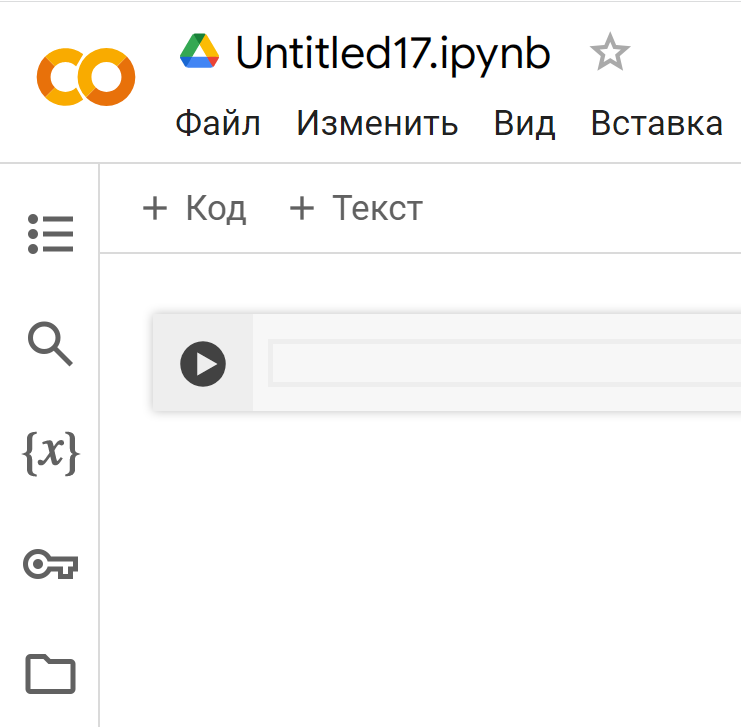


Рис.1 Среда программирования Google Colab

Создадим блокнот в Google Colab:

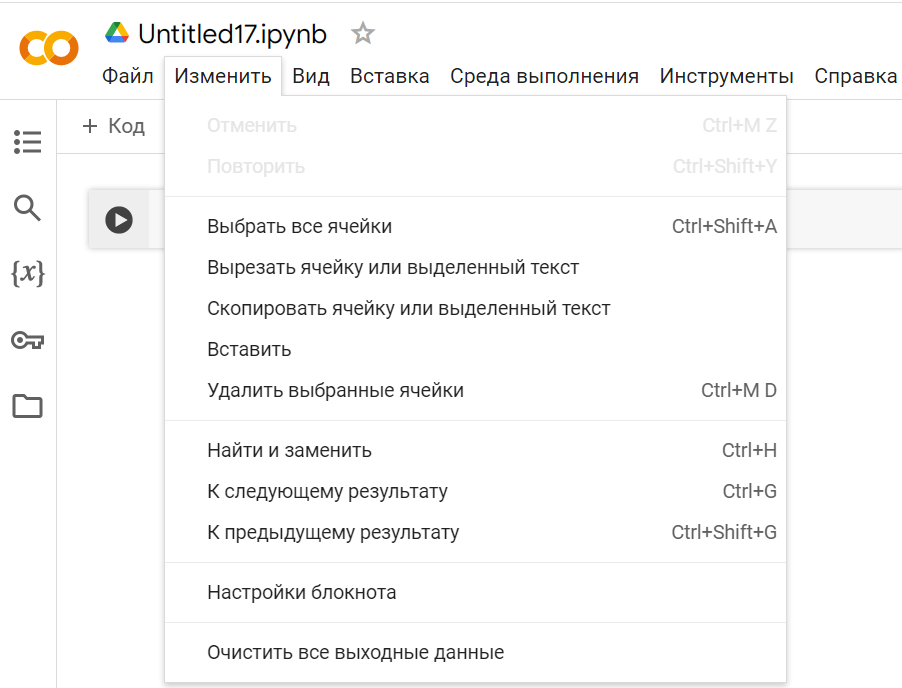


Рис. 2 Меню с командами для выполнения в среде программирования

Переименование и сохранение файла блокнота на диск:

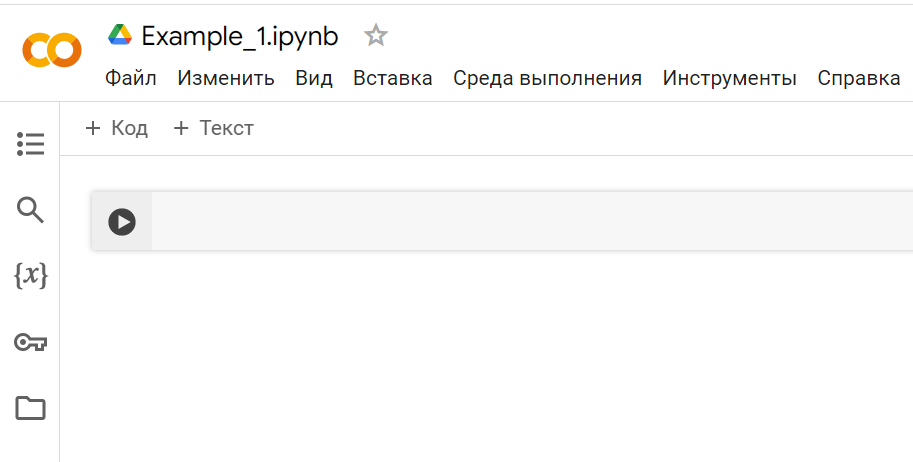


Рис. 3. Сохранение файла блокнота на диск

**Часть 1. Арифметические операции с числами в Python:**

Пример 1: прибавим числа:

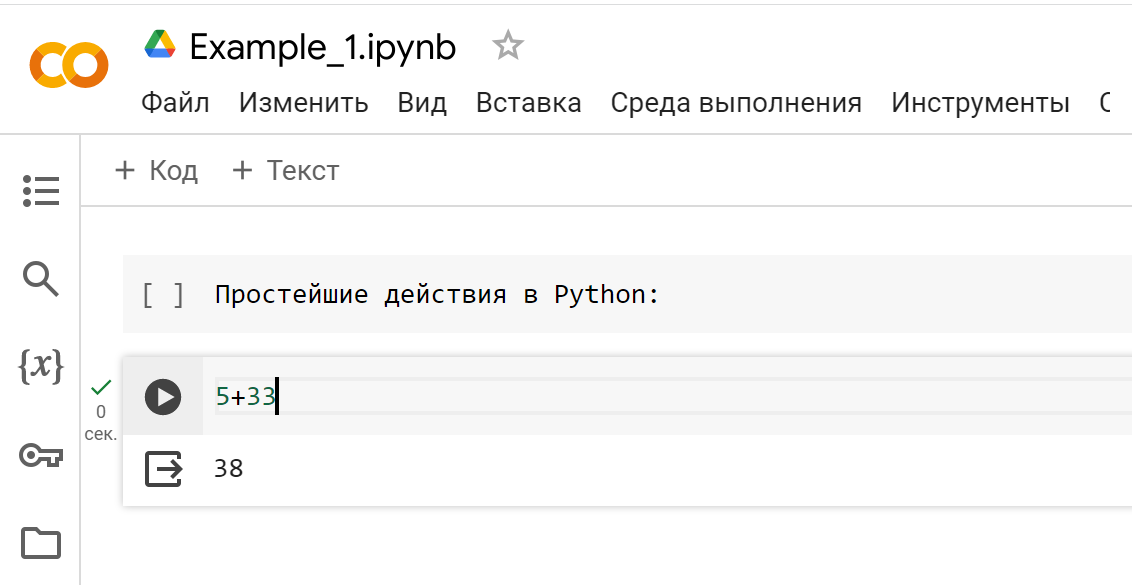


Рис. 4. Суммирование чисел в блокноте.

Задание 1: Прибавьте 39890 к 102895.

Пример 2: Вычтем число от числа:

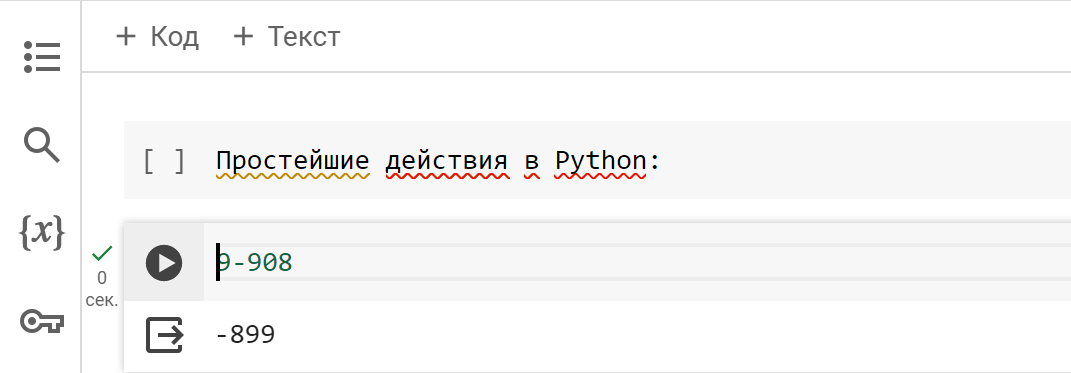


Рис. 5. Вычитание чисел в блокноте.

Задание 2: Отнимите 80887 от 102895.

Пример 3: Умножим числа:

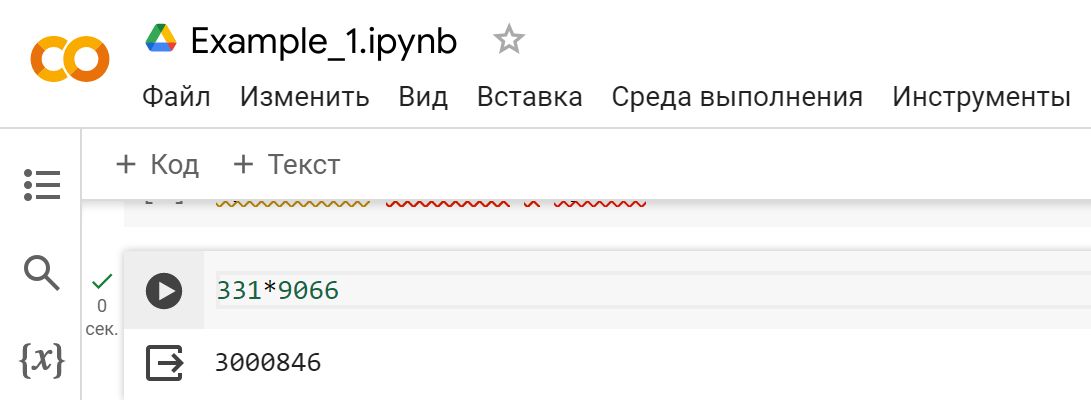


Рис. 6. Умножение чисел в блокноте.

Задание 3: умножьте 39890 на 1002895.

Пример 4: возведем число в степень:

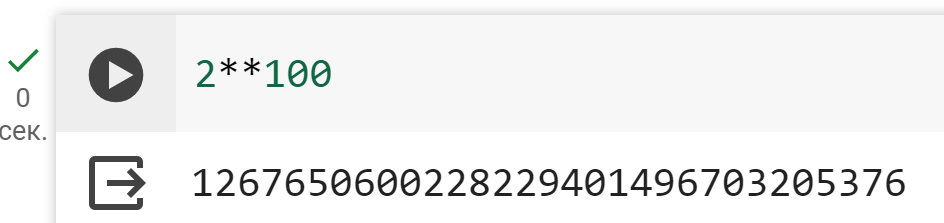


Рис. 7. Возведение числа в степень.

Пример 5: В Python можно с легкостью вычислять очень большие числа, например, возведем число 2 в 300-ю степень:

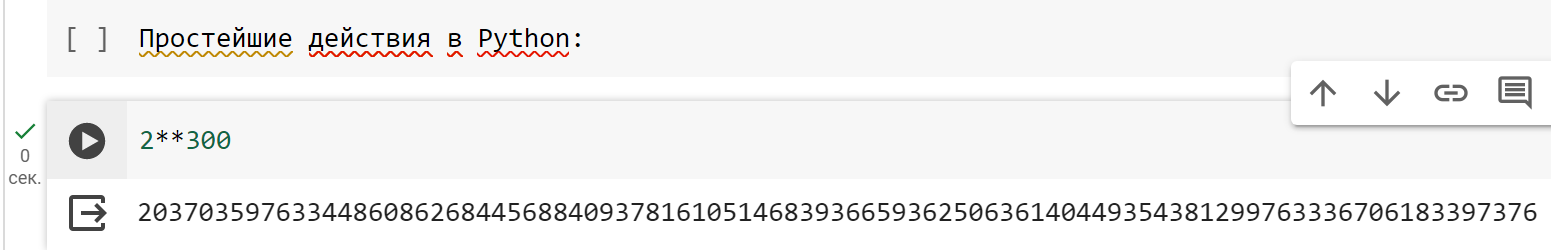


Рис. 8. Возведение числа 2 в степень 300.

Но и это не предел!

Задание 4: Возведите число 2 в 800-ую степень.

Пример 6: Возведем число 2 в дробную степень 1,5988:

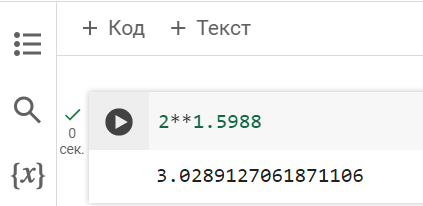


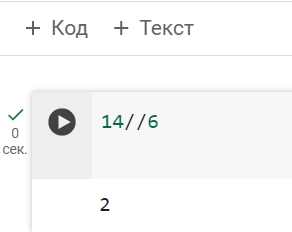
Рис. 9. Возведение числа 2 в дробную степень.

Задание 5: Возведите 1,3387 в степень 3,8999

Примеры на деление чисел:

Примеры.

1. Деление любых чисел нацело: примеры 7, 8.

Примеры 7 и 8:

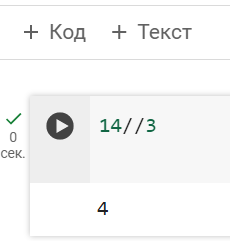


Рис. 10. Деление нацело.

Задание 6. Выполните деление 1338799 на 144009 нацело.

2. Пример 9: деление "стандартное":

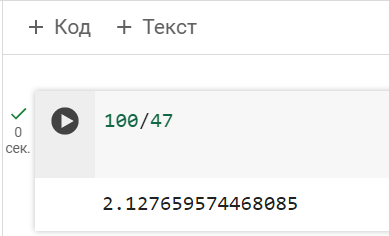


Рис. 11. Деление чисел.

Задание 7: Поделите 9999 на 999.

Пример 10: Получение остатка от деления:

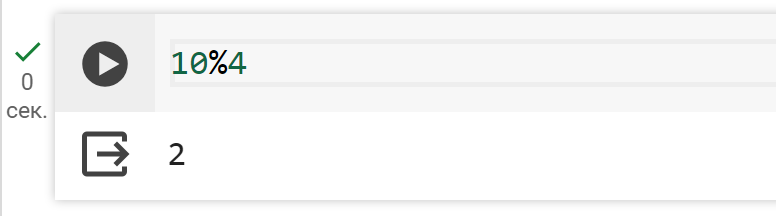


Рис. 12. Получение остатка от деление чисел.

Задание 8. Получите остаток от деления числа 123456789 на 6789.

**Часть 2. Переменные в Python, функция print() и типы данных**

Переменные предназначены для хранения данных. Название переменной в Python должно начинаться с алфавитного символа или со знака подчеркивания и может содержать алфавитно-цифровые символы и знак подчеркивания.

Пример 10:

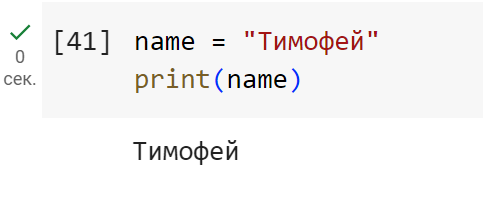


Рис. 13. Задание переменной и отображение ее значения с помощью функции print().

Задание 9: Создайте переменную my\_name = Ваше\_имя и c помощью функции print() отобразите ее в блокноте.

Пример 11: Задание переменной для числового значения:

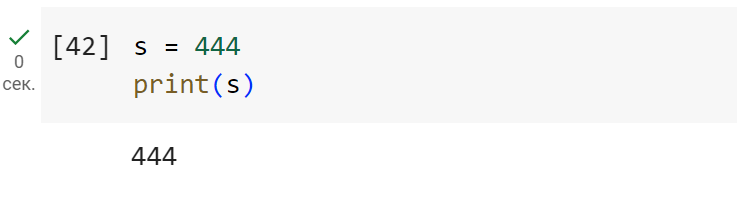


Рис. 14. Задание переменной с фиксированным ее значением и отображение ее с помощью функции print().

Задание 10: Создайте две переменные first\_num, second\_num типа int (целое число) и задав для них любые числовые значения в блокноте с помощью функции print(first\_num, second\_num) отобразить значения этих чисел.

Пример 12. Задание переменных и вычисление арифметических операций над ними:

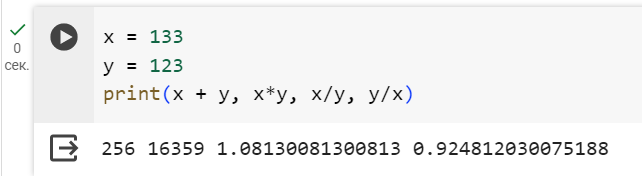


Рис. 15. Задание переменных с фиксированным их значением и отображение с помощью функции print() их значений с результатами арифметических операций над ними.

Задания 11.

Найти значения следующих арифметических выражений:

1.

2.

3.

Пример 13. Вычисление значения арифметического выражения с заданными числовыми значениями:

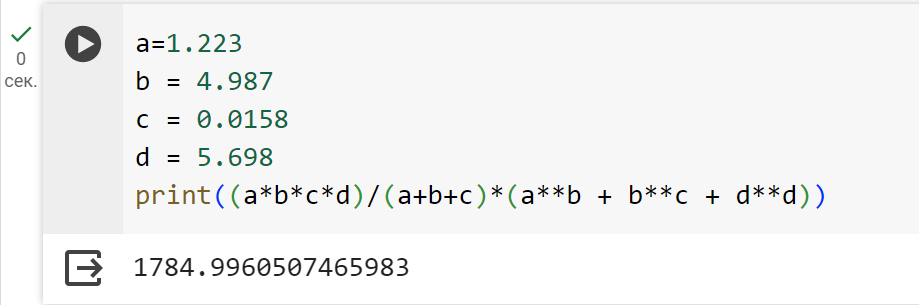


Рис. 16. Код примера вычисления значения арифметического выражения с наперед заданными значениями объявленных переменных.

Задание 14. Объявить 4 переменных L, O, N, T и значения для них и произвести с ними арифметические действия:

**Часть 3. Более сложные математические операции.**

Для более сложных математических вычислений в программу импортируют методы с помощью директивы import.

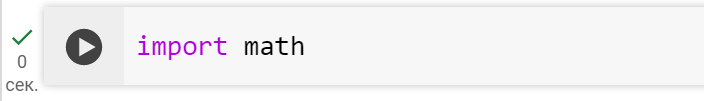


Рис. 17. Включение библиотеки в код программы

Модуль math – один из наиважнейших в Python. Этот модуль предоставляет обширный функционал для работы с числами. Ниже приведен перечень наиболее частоиспользуемых функций.

**math.ceil**(X) – округление до ближайшего большего числа.

**math.copysign**(X, Y) - возвращает число, имеющее модуль такой же, как и у числа X, а знак - как у числа Y.

**math.fabs**(X) - модуль X.

**math.factorial**(X) - факториал числа X.

**math.floor**(X) - округление вниз.

**math.fmod**(X, Y) - остаток от деления X на Y.

**math.frexp**(X) - возвращает мантиссу и экспоненту числа.

**math.ldexp**(X, I) - X \* 2i. Функция, обратная функции math.frexp().

**math.fsum**(последовательность) - сумма всех членов последовательности. Эквивалент встроенной функции sum(), но math.fsum() более точна для чисел с плавающей точкой.

**math.isfinite**(X) - является ли X числом.

**math.isinf**(X) - является ли X бесконечностью.

**math.isnan**(X) - является ли X NaN (Not a Number - не число).

**math.modf**(X) - возвращает дробную и целую часть числа X. Оба числа имеют тот же знак, что и X.

**math.trunc**(X) - усекает значение X до целого.

**math.exp**(X) - eX.

**math.expm1**(X) - eX - 1. При X → 0 точнее, чем math.exp(X)-1.

**math.log**(X, [base]) - логарифм X по основанию base. Если base не указан, вычисляется натуральный логарифм.

**math.log1p**(X) - натуральный логарифм (1 + X). При X → 0 точнее, чем math.log(1+X).

**math.log10**(X) - логарифм X по основанию 10.

**math.log2**(X) - логарифм X по основанию 2.

**math.pow**(X, Y) - XY.

**math.sqrt**(X) - квадратный корень из X.

**math.acos**(X) - арккосинус X. В радианах.

**math.asin**(X) - арксинус X. В радианах.

**math.atan**(X) - арктангенс X. В радианах.

**math.atan2**(Y, X) - арктангенс Y/X. В радианах. С учетом четверти, в которой находится точка (X, Y).

**math.cos**(X) - косинус X (X указывается в радианах).

**math.sin**(X) - синус X (X указывается в радианах).

**math.tan**(X) - тангенс X (X указывается в радианах).

**math.hypot**(X, Y) - вычисляет гипотенузу треугольника с катетами X и Y (math.sqrt(x \* x + y \* y)).

**math.degrees**(X) - конвертирует радианы в градусы.

**math.radians**(X) - конвертирует градусы в радианы.

**math.cosh**(X) - вычисляет гиперболический косинус.

**math.sinh**(X) - вычисляет гиперболический синус.

**math.tanh**(X) - вычисляет гиперболический тангенс.

**math.acosh**(X) - вычисляет обратный гиперболический косинус.

**math.asinh**(X) - вычисляет обратный гиперболический синус.

**math.atanh**(X) - вычисляет обратный гиперболический тангенс.

**math.erf**(X) - функция ошибок.

**math.erfc**(X) - дополнительная функция ошибок (1 - math.erf(X)).

**math.gamma**(X) - гамма-функция X.

**math.lgamma**(X) - натуральный логарифм гамма-функции X.

**math.pi** - pi = 3,1415926...

**math.e** - e = 2,718281...

Пример 14. Вычислим значение тригонометрической фукнции:

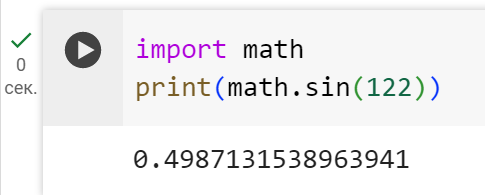


Рис.18. Вычисление значения тригонометрической функции с использованием библиотеки math.

Задание 15. Используя перечень функций в библиотеки math найдите значение квадратного корня из числа 789,377; 198799,3698; 187,9887

Пример 15. Вычислить значение выражения:

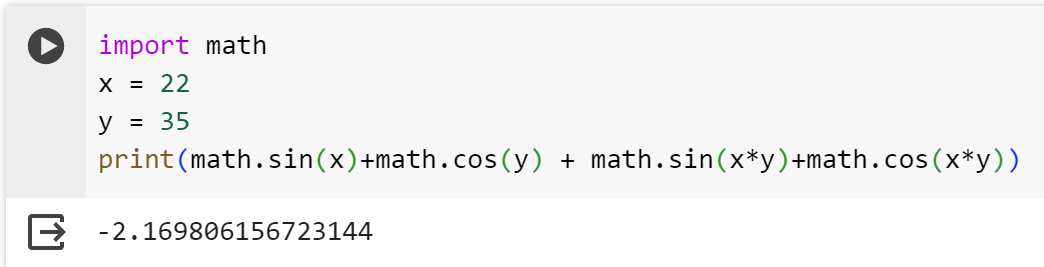


Рис.19. Вычисление значения тригонометрического выражения с использованием библиотеки math.

Задание 16. Вычислить при значение выражения:

Задание 17.

Вычислить значение функции:

Пример 16. Ввод значений объявленных переменных в блокнот:

name = input("Введите свое имя: ")

print(f"Ваше имя: {name}")

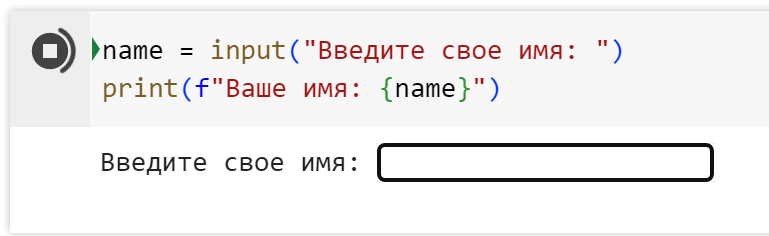


Рис.20. Объявление значения переменной в блокнот.

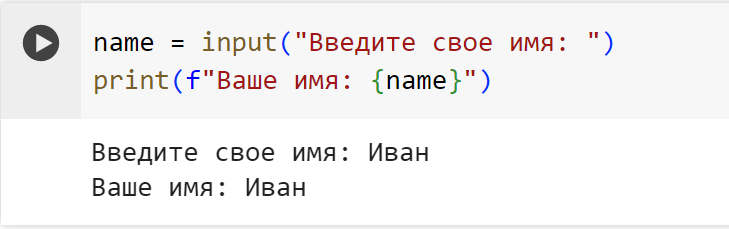


Рис.21. Ввод значения переменной в блокнот.

Задание 17. Напишите код возвращающий ваше имя и фамилию.

Пример 17. В программу вводится три числа и вычисляется значение в виде конкатенации их:

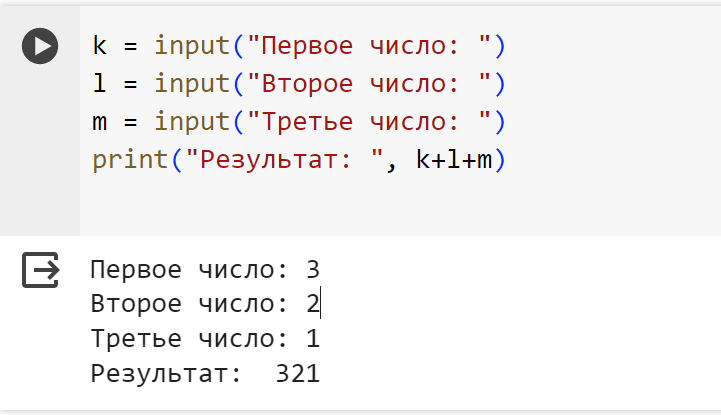


Рис.22. Ввод значений переменных в блокнот. Результат - конкатенация.

Задание 18.

Проведите операцию конкатенации для переменных m='Python - ', u=' язык', j=' программирования'.

Пример 18. Вычисление алгебраического выражения:

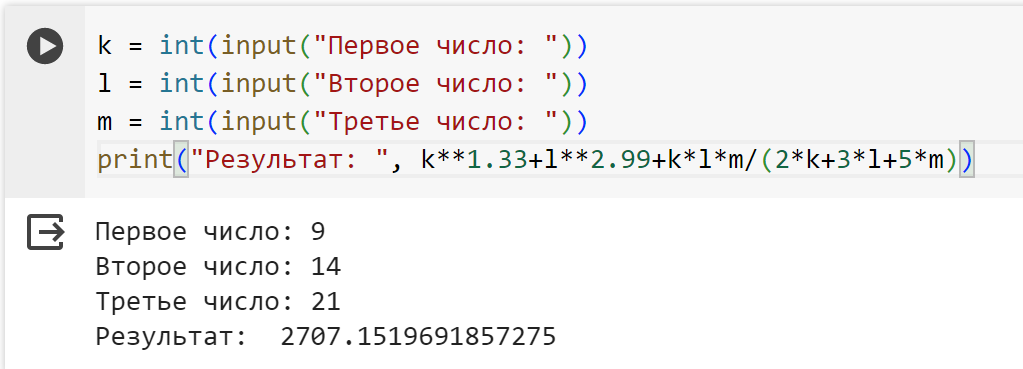


Рис.23. Вычисление алгебраического выражения. Результат - числовое значение.

Задание 19.

Вычислите значения алгебраических выражений для различных значений переменных :

*5. Закрепление полученных знаний (5 мин):* *ответьте на вопросы*:

1. Кто создал язык программирования Python?
2. Для чего служит Google Colab?
3. Перечислите арифметические операции в Python.
4. Для чего служат переменные?
5. Что такое операция конкатенации?
6. В чем отличие операции конкатенации и суммирования?

*6.Итоговая часть занятия (2 мин):*

Выставление и комментирование оценок

*7. Домашнее задание (2 мин):*

1. Вычислите значения алгебраических выражений для различных значений переменных :
2. Используя перечень функций в библиотеки math найдите значение функции в точке :
3. Проведите операцию конкатенации для переменных s='я - ', u=' студент', j=' КТЭК'.

**Использованная литература:**

1. «Изучаем Python», Марк Лутц

2. «Программируем на Python», Майкл Доусон

3.«Изучаем программирование на Python», Пол Бэрри

4. «Легкий способ выучить Python», Зед А. Шоу

**Интернет –ресурсы:**

<https://colab.research.google.com/>

<https://edu.vsu.ru/mod/url/view.php?id=200520>

<https://blog.skillfactory.ru/chto-takoe-google-colaboratory-i-komu-on-nuzhen/>

<https://gb.ru/blog/rabota-v-google-colab/>

<https://rutube.ru/video/abc2f3047d5ed7a180646ac141324f4f/>

<https://sky.pro/wiki/python/chtenie-i-obrabotka-dannykh-iz-google-drive-v-colab/>