Лабораторная работа №1

Julia. Установка и настройка. Основные принципы.

Доберштейн Алина Сергеевна

Содержание

Список иллюстраций

# 1 Цель работы

Основная цель работы — подготовить рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомиться с основами синтаксиса Julia.

# 2 Задание

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя Jupyter Lab, повторите примеры из раздела лабораторной работы.
3. Выполните задания для самостоятельной работы.

# 3 Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia.

# 4 Выполнение лабораторной работы

Запустила Julia. (рис. 1).

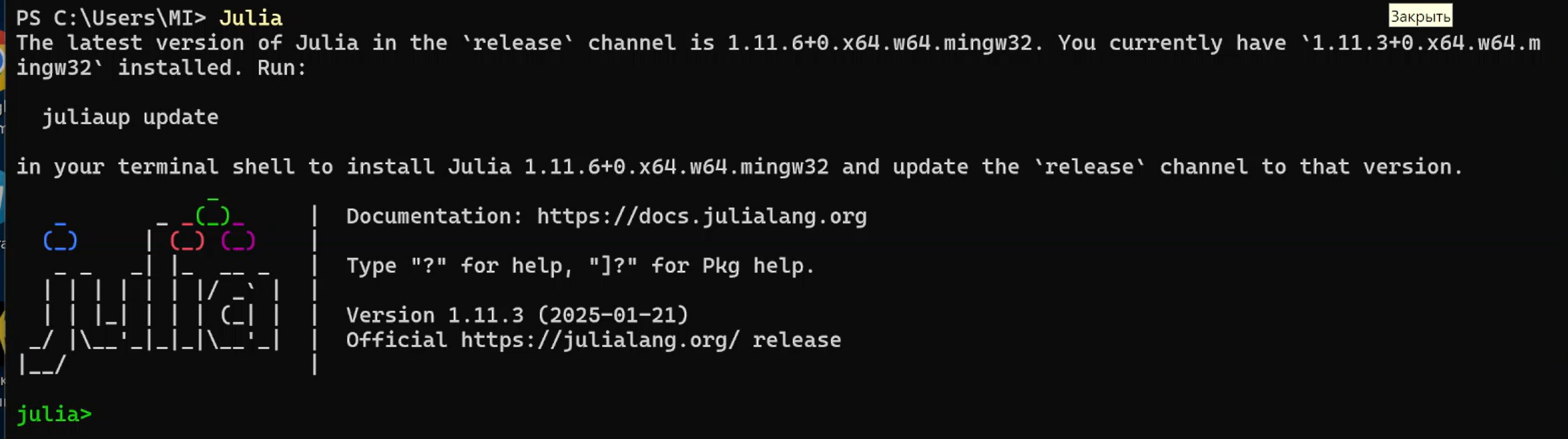


Рис. 1: Julia

# 5 Простейшие операции на языке Julia

Повторила простейшие примеры для знакомства с синтаксисом Julia из лабораторной работы.(рис. 2-8).



Рис. 2: Простейшие арифметические операции

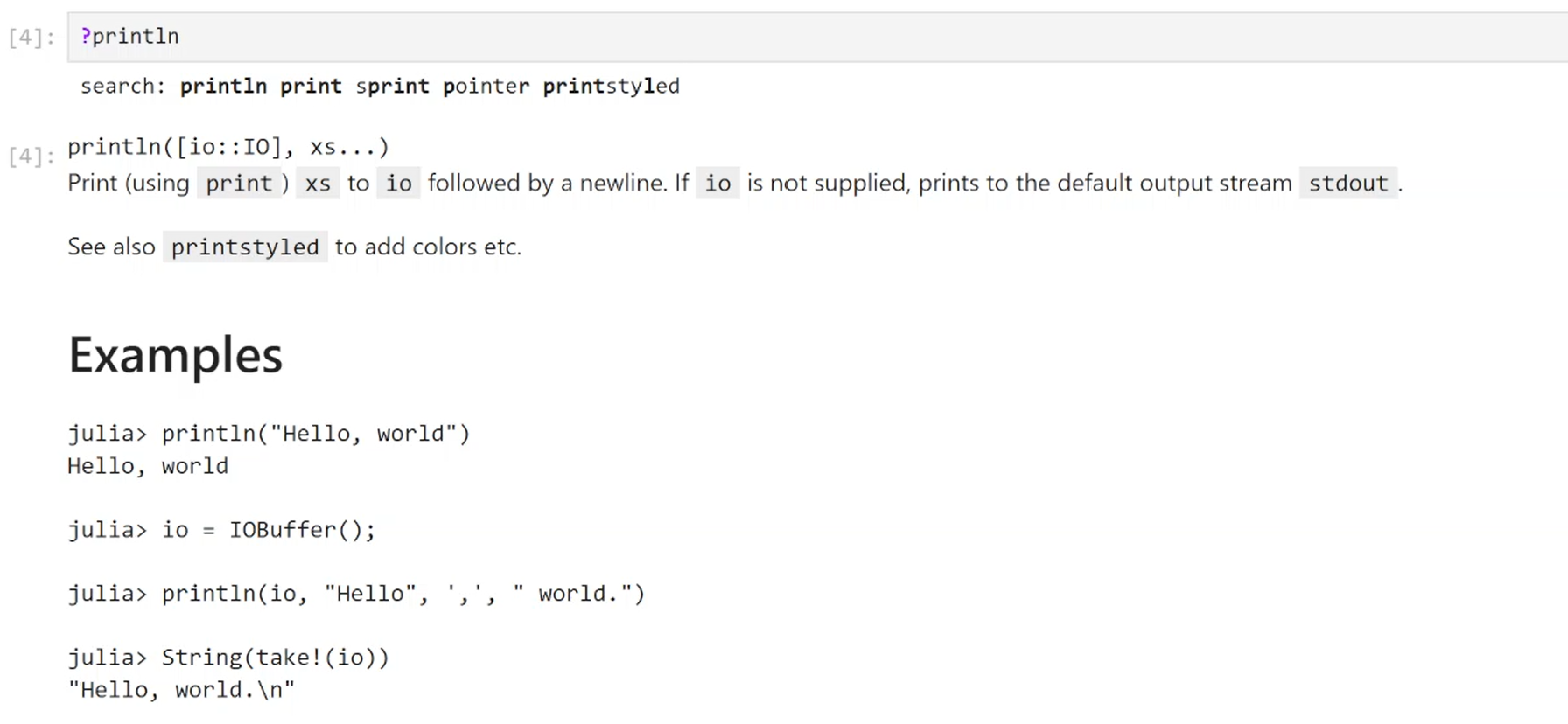


Рис. 3: Пример получения информации по функции



Рис. 4: Примеры определения типа числовых величин

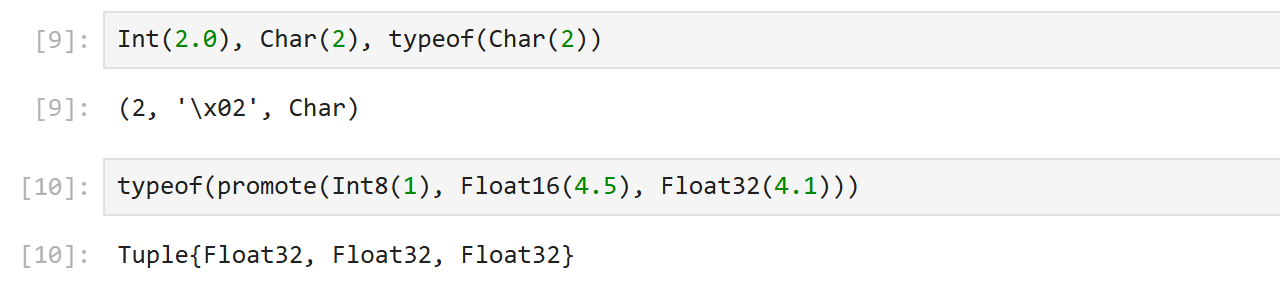


Рис. 5: Примеры приведения аргументов к одному типу



Рис. 6: Примеры определения функций

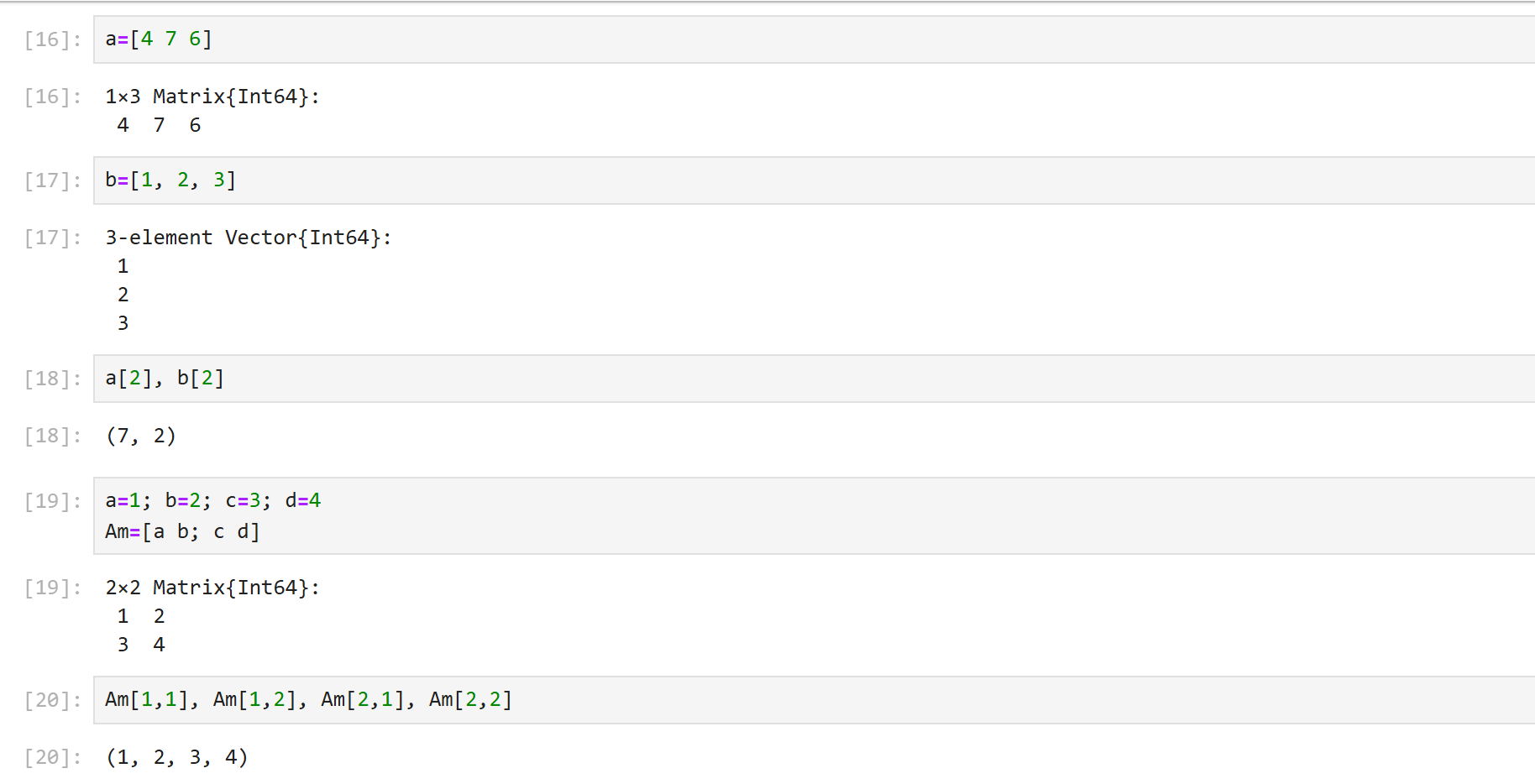


Рис. 7: Примеры работы с массивами

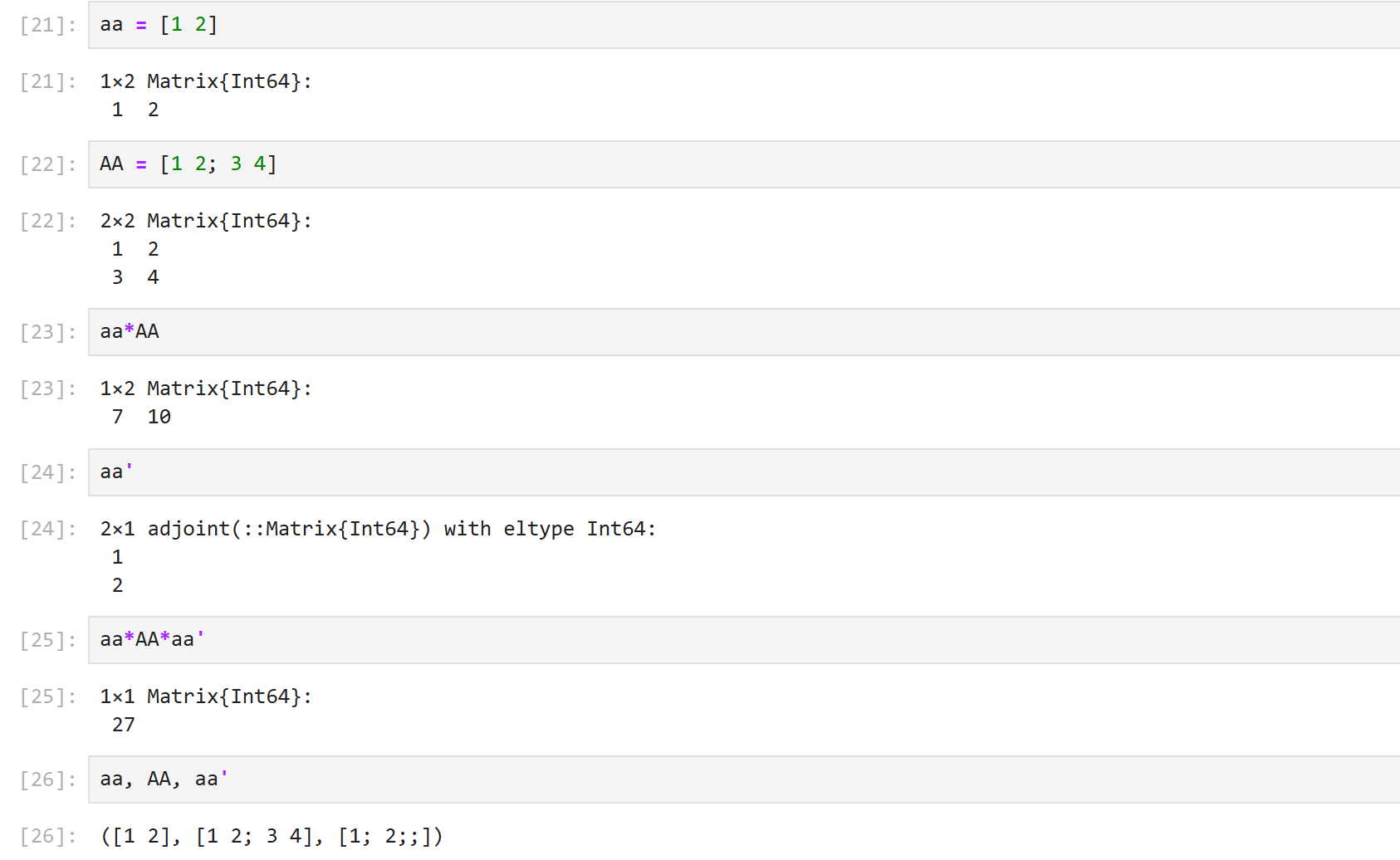


Рис. 8: Примеры работы с массивами

## 5.1 Основные функции Julia

Изучила документацию по функции read(). (рис. 9).



Рис. 9: Функция read

Примеры использования функции read(). (рис. 10).

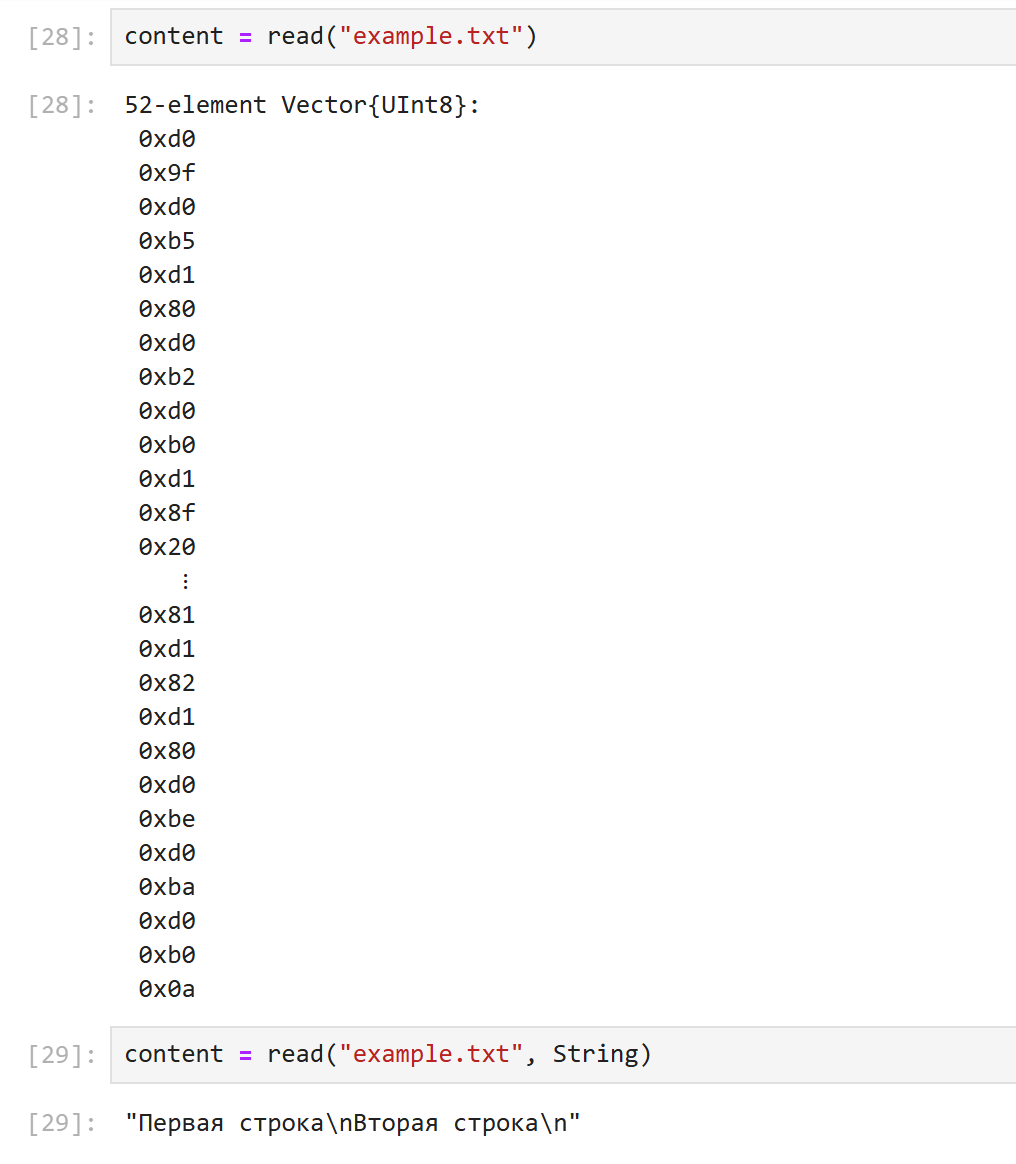


Рис. 10: Примеры с функцией read

Функция read() читает содержимое файла или потока целиком в виде массива байт или другого типа, если его задать. Читает содержимое в одну строку, без переноса, с разделителями.

Изучила документацию по функции readline(). (рис. 11).

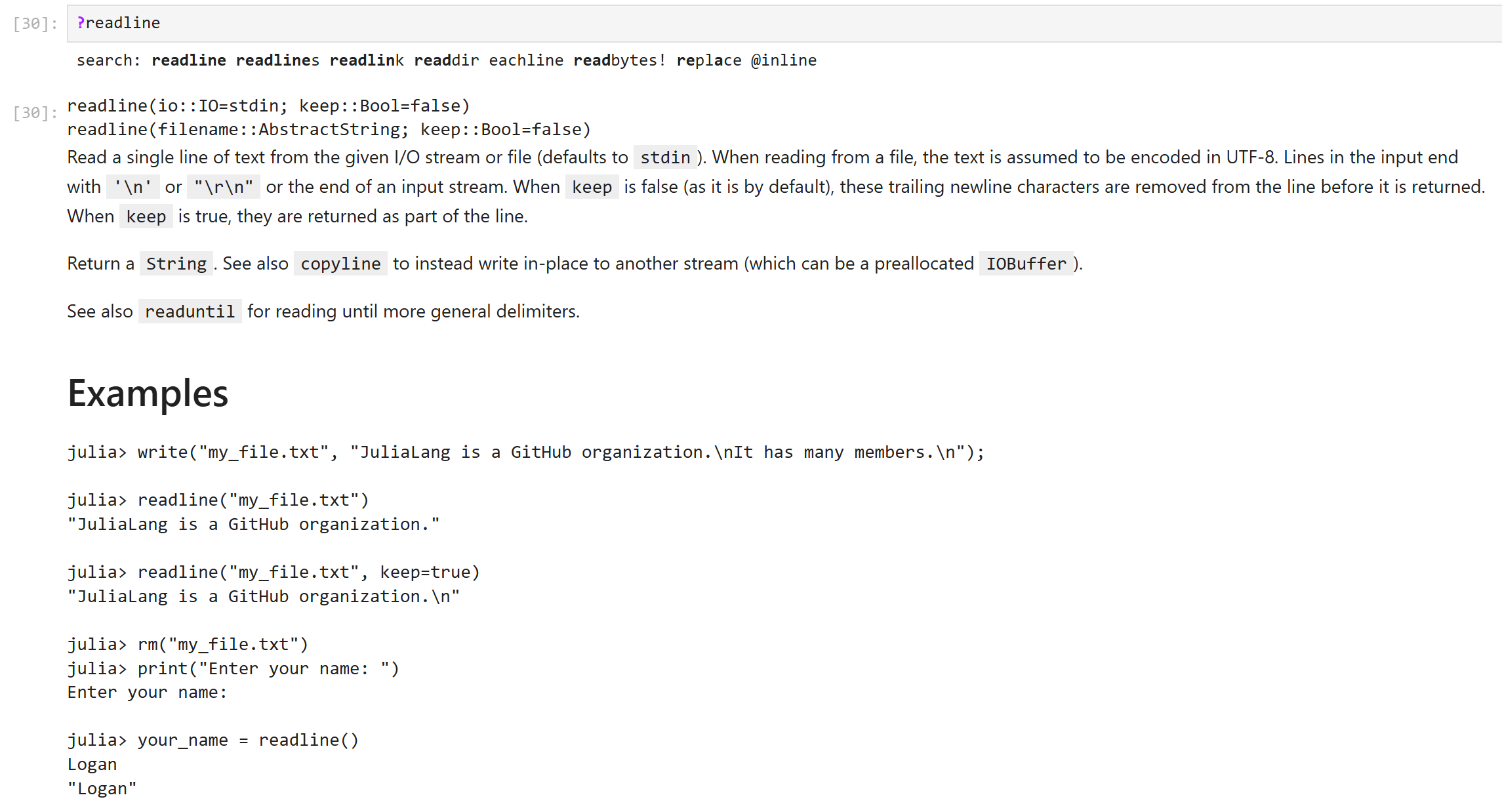


Рис. 11: Функция readline

Примеры использования функции readline(). (рис. 12).

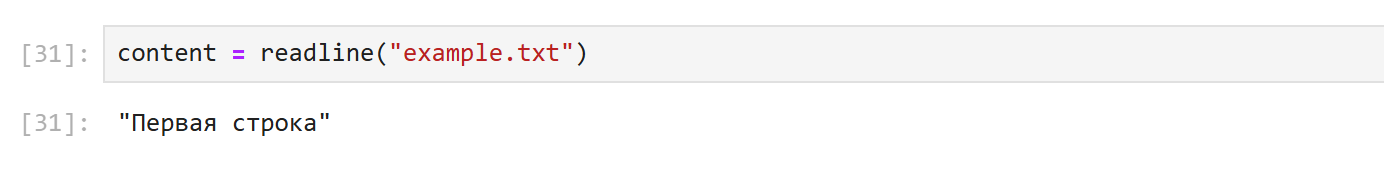


Рис. 12: Примеры с функцией readline

Эта функция читает одну строку из файла или потока. При достижении конца файла выбрасывает исключение.

Изучила документацию по функции readlines(). Примеры использования функции readlines(). (рис. 13).

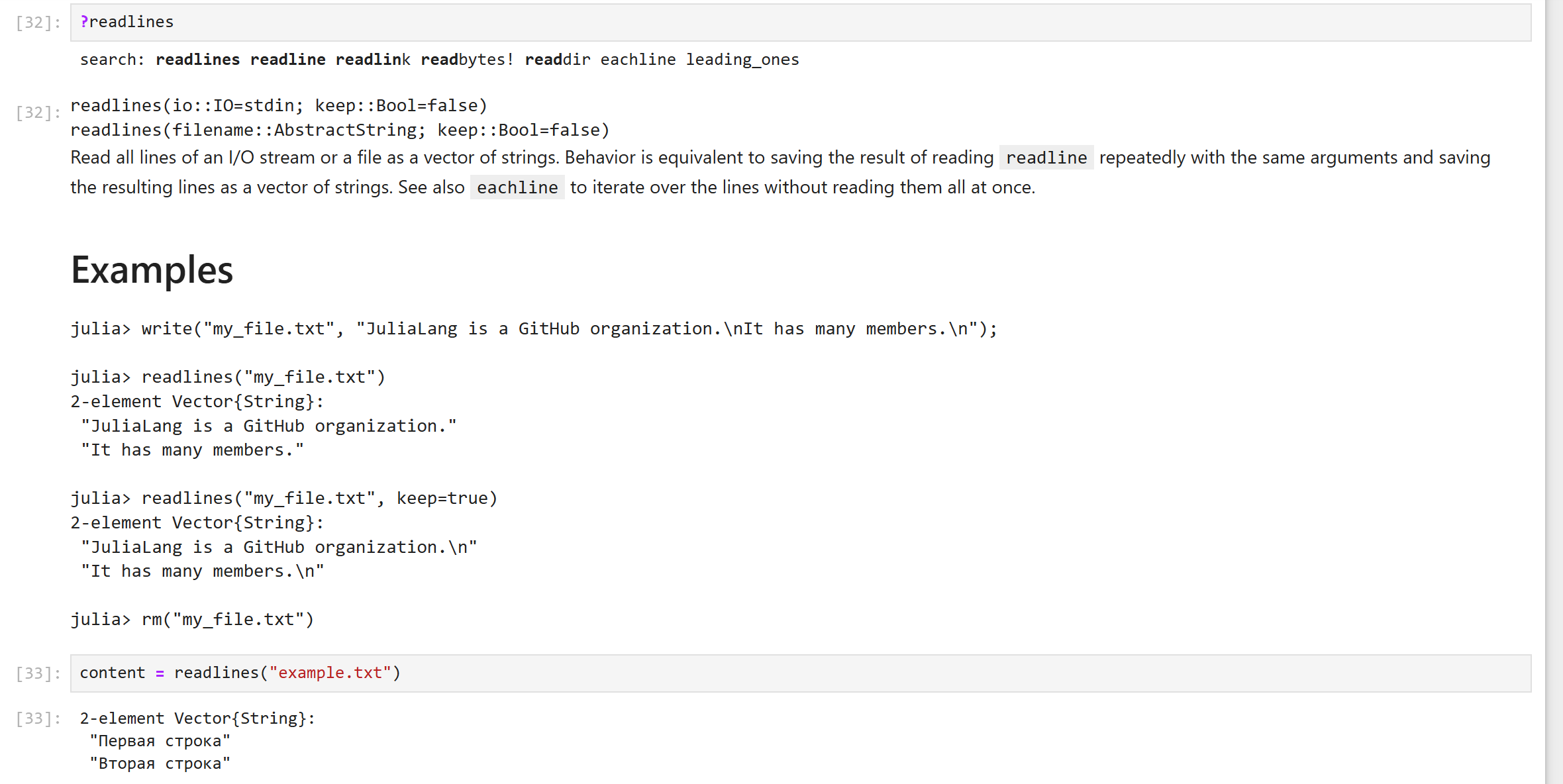


Рис. 13: Функция readlines

Эта функция читает все строки из файла и возвращает массив строк.

Изучила документацию по функции readdlm(). (рис. 14).

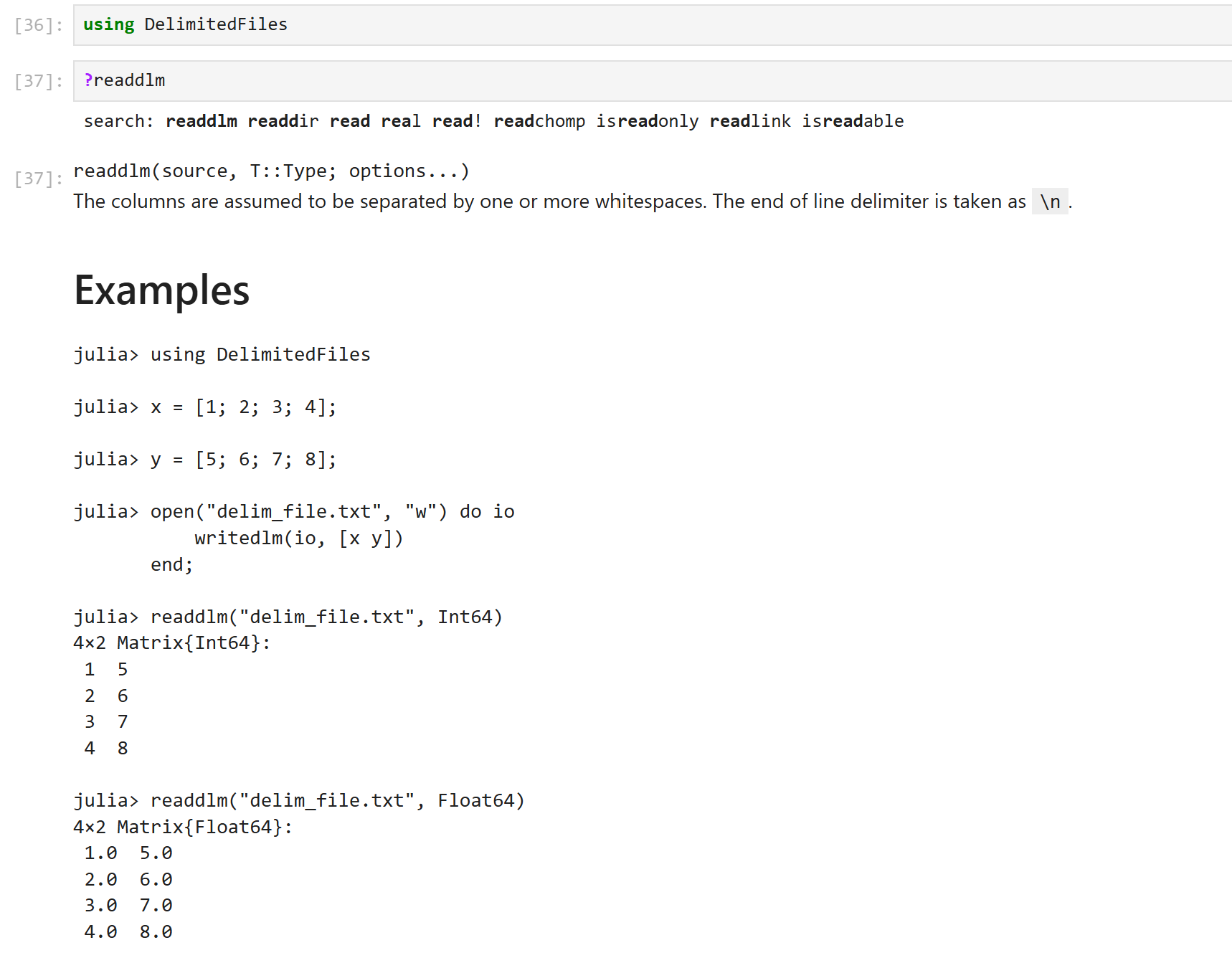


Рис. 14: Функция readdlm

Примеры использования функции readdlm(). (рис. 15).

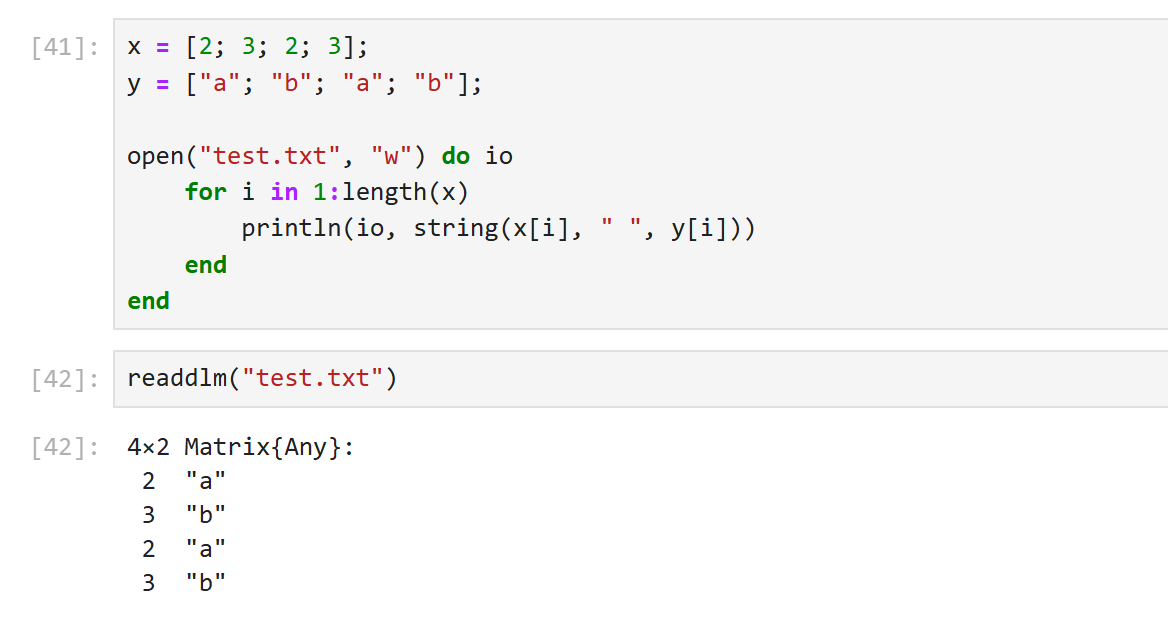


Рис. 15: Примеры с функцией readdlm

Эта функция читает данные из файла с разделителями (delimiter), например CSV или табличные данные. Возвращает матрицу или массив.

Примеры использования функций print() и println(). (рис. 16).

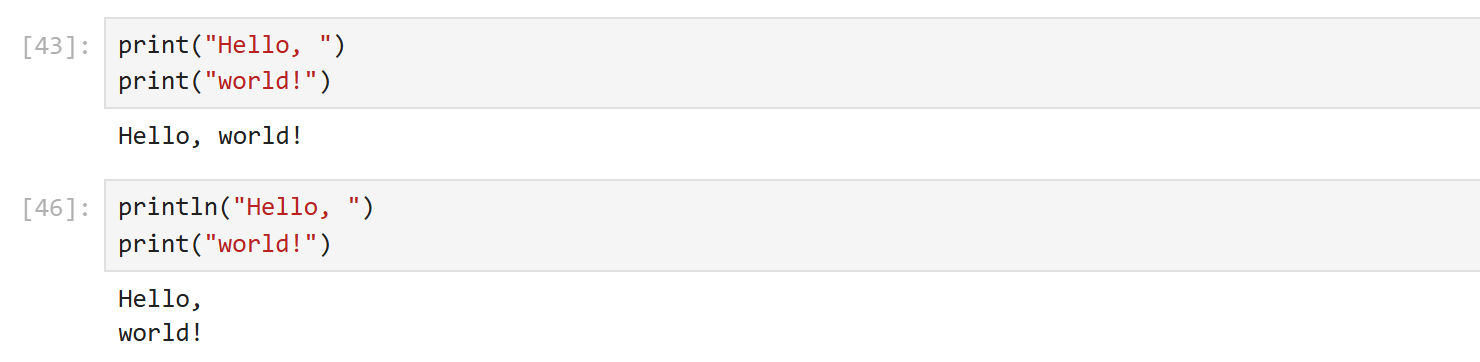


Рис. 16: Примеры с функциями print и println

Функция print() выводит данные на экран без перевода строки. A println() - с переводом в конце строки.

Изучила документацию по функции show(). (рис. 17).

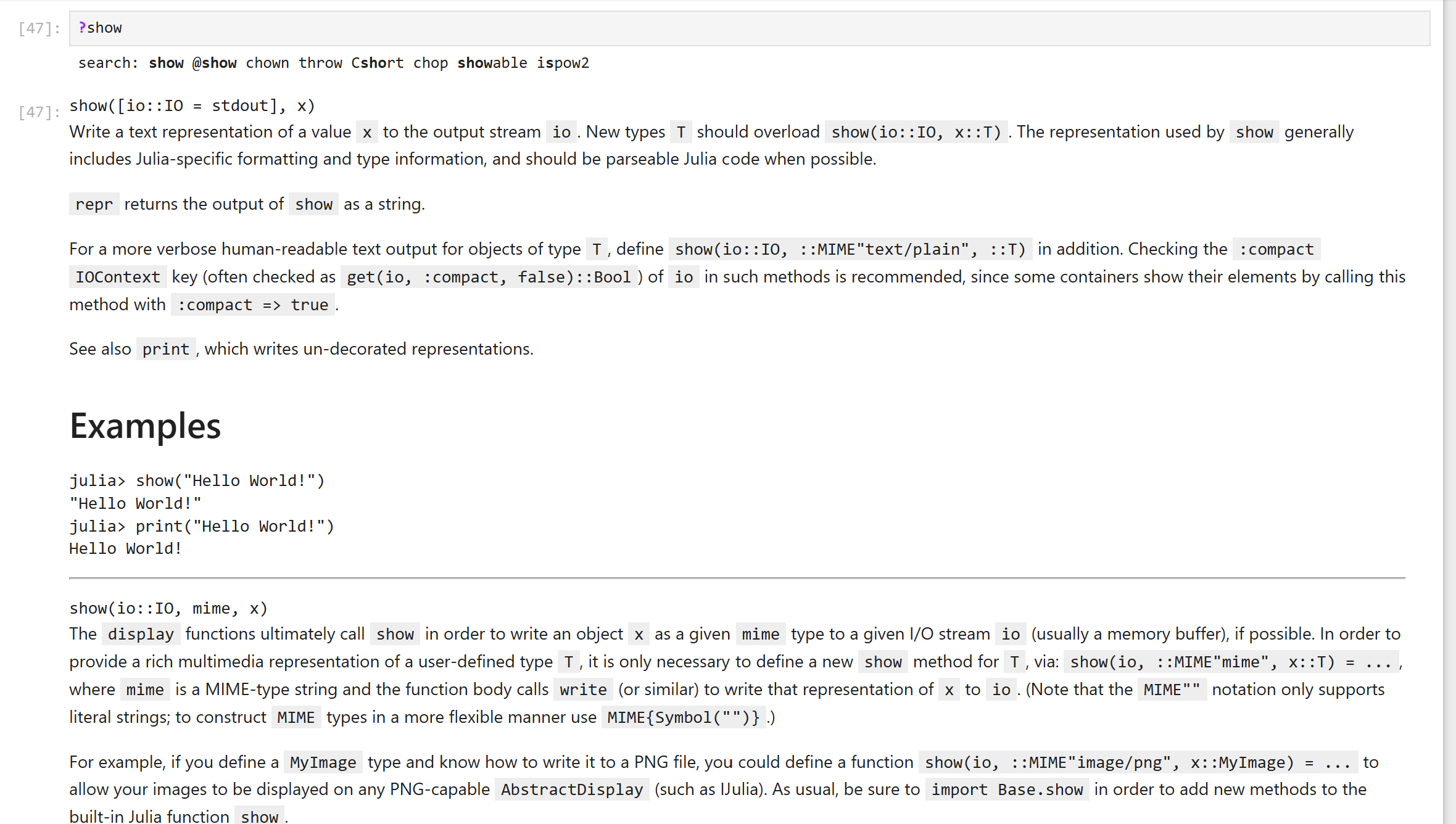


Рис. 17: Функция show

Примеры использования функции show(). (рис. 18).



Рис. 18: Примеры с функцией show

Эта функция выводит данные в более “сыром” или структурированном виде. Часто используется для вывода объектов с отображением их внутреннего представления.

Изучила документацию по функции write(). (рис. 19).

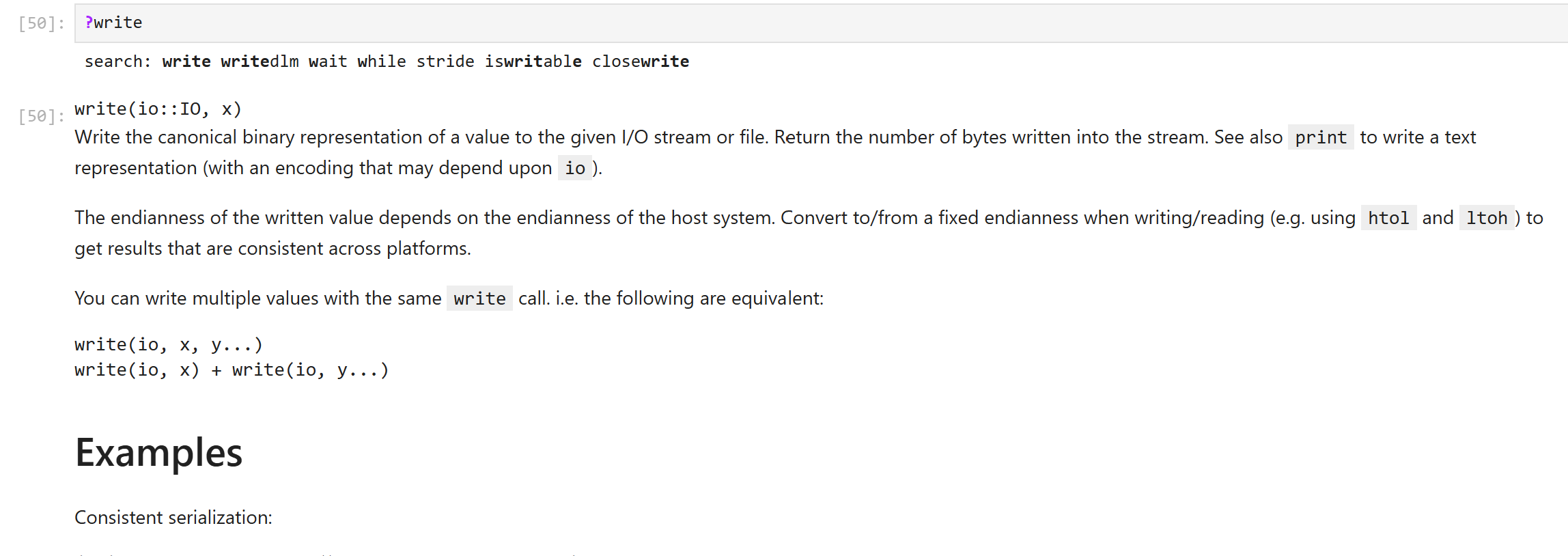


Рис. 19: Функция write

Примеры использования функции write(). (рис. 20-21).

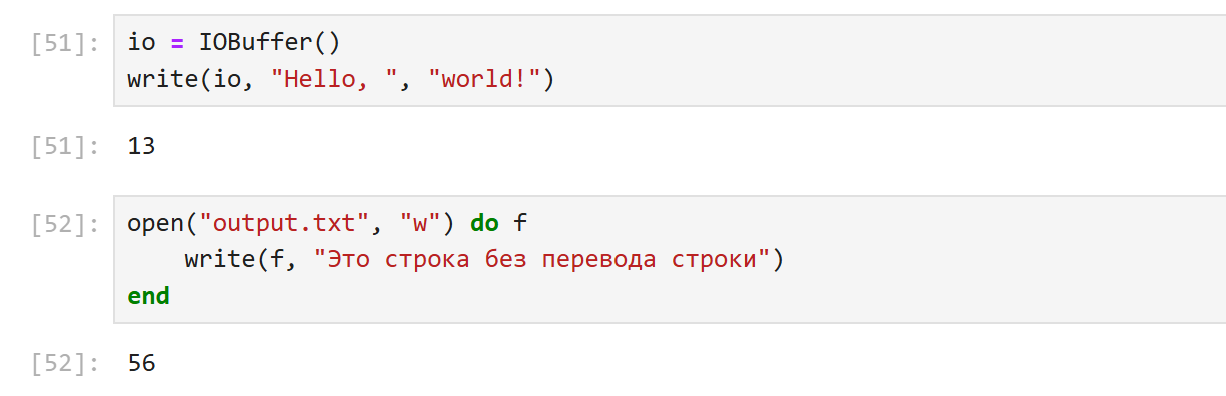


Рис. 20: Примеры с функцией write

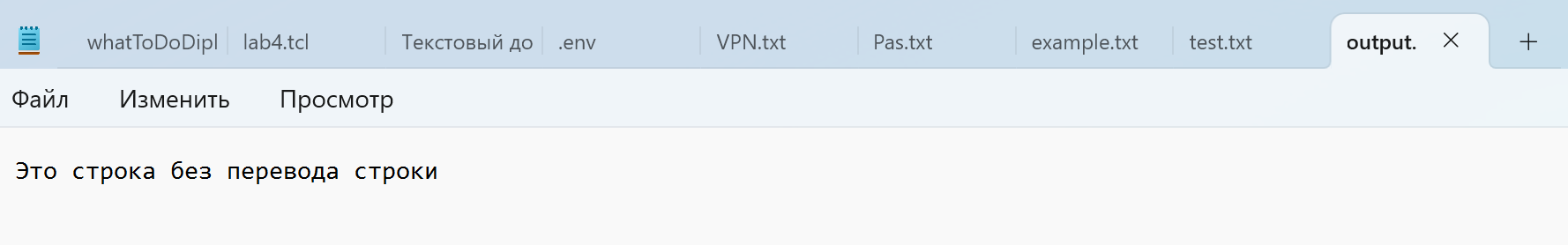


Рис. 21: Примеры с функцией write

Эта функция записывает байты или данные в файл или поток, не добавляет перевод строки, не форматирует данные.

## 5.2 Функция parse()

Изучила документацию функции parse().(рис. 22).

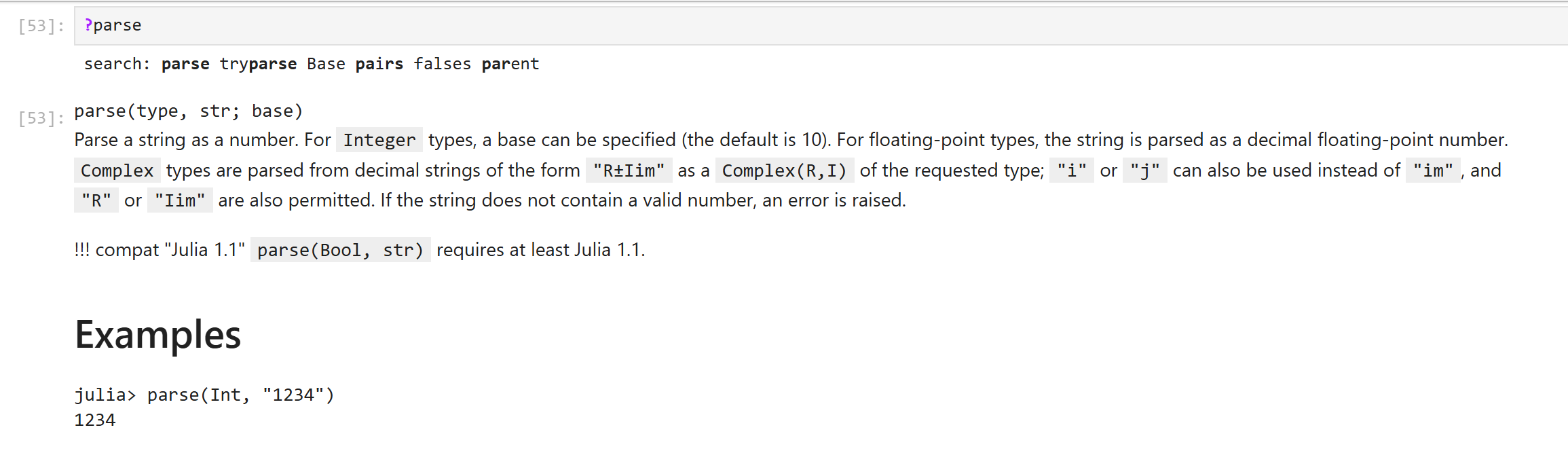


Рис. 22: Функция parse

Привела свои примеры её использования. (рис. 23).



Рис. 23: Использование функции parse

Функция ёparse()ё в Julia используется для преобразования строки (String) в значение указанного типа. Это удобно, когда нужно конвертировать текстовые данные в числа, логические значения и другие типы.

## 5.3 Базовые математические операции

Изучила синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Привела примеры с пояснениями по особенностям их применения: (рис. 24-32)



Рис. 24: Примеры для сложения



Рис. 25: Примеры для сложения



Рис. 26: Примеры для вычитания

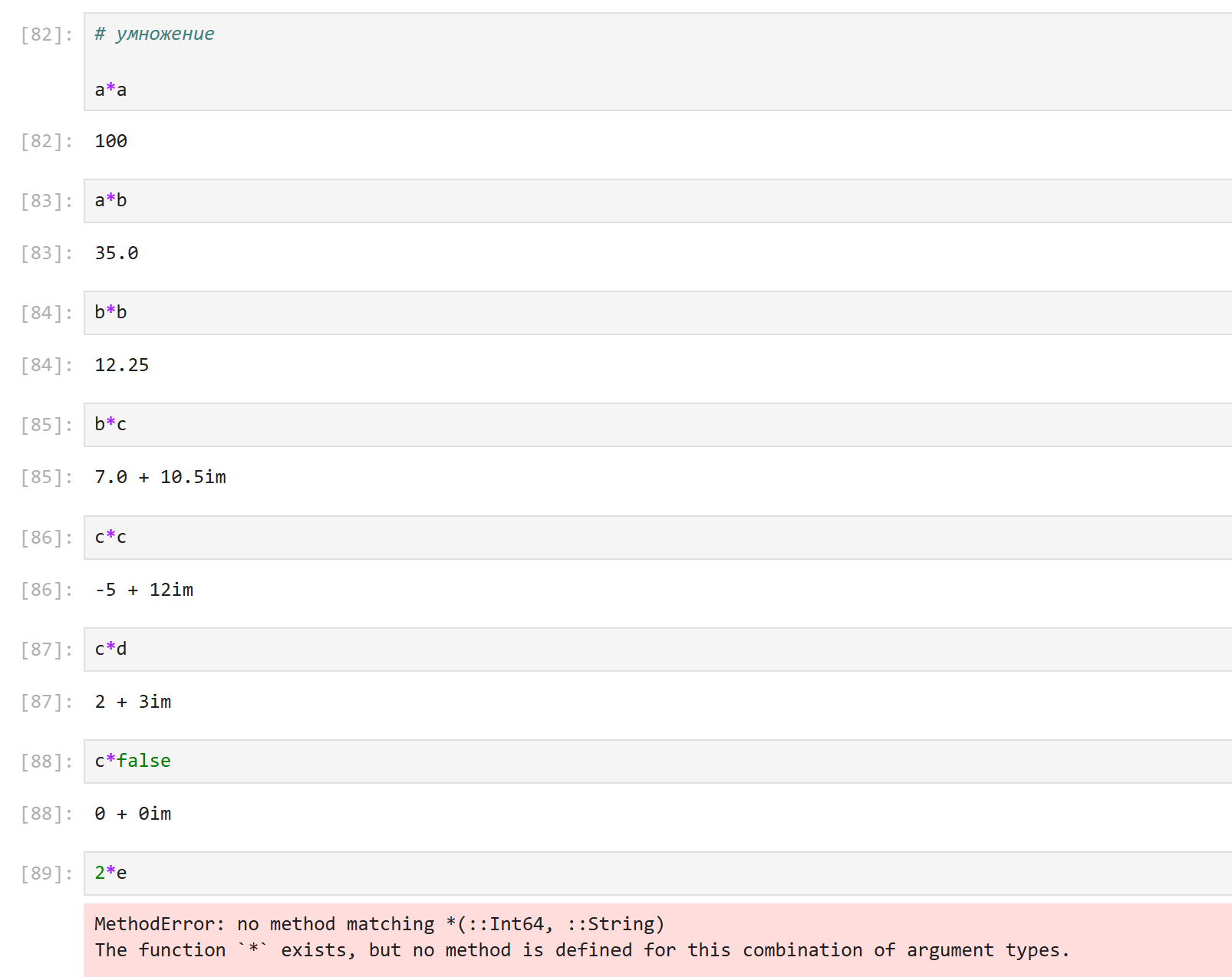


Рис. 27: Примеры для умножения

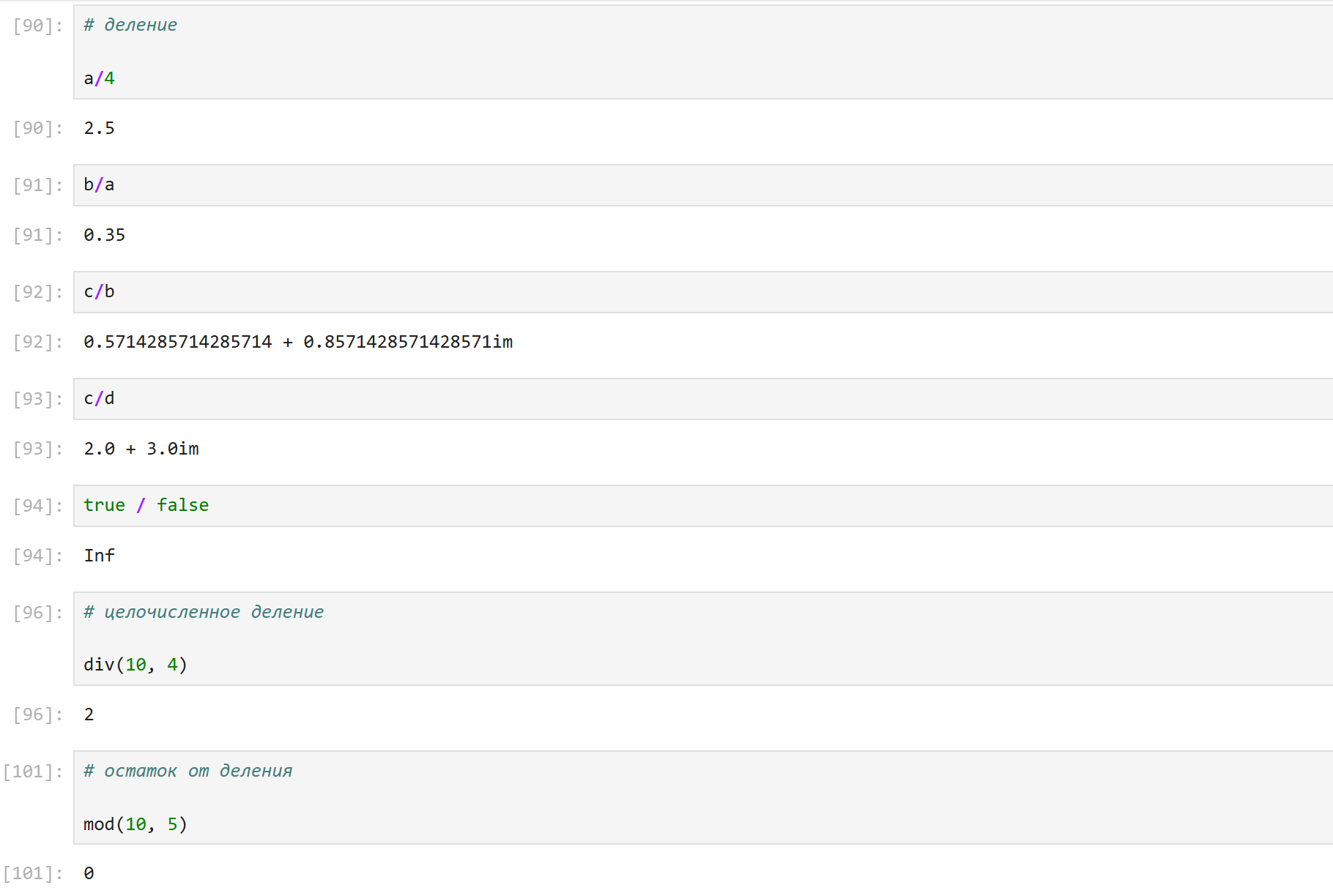


Рис. 28: Примеры для деления

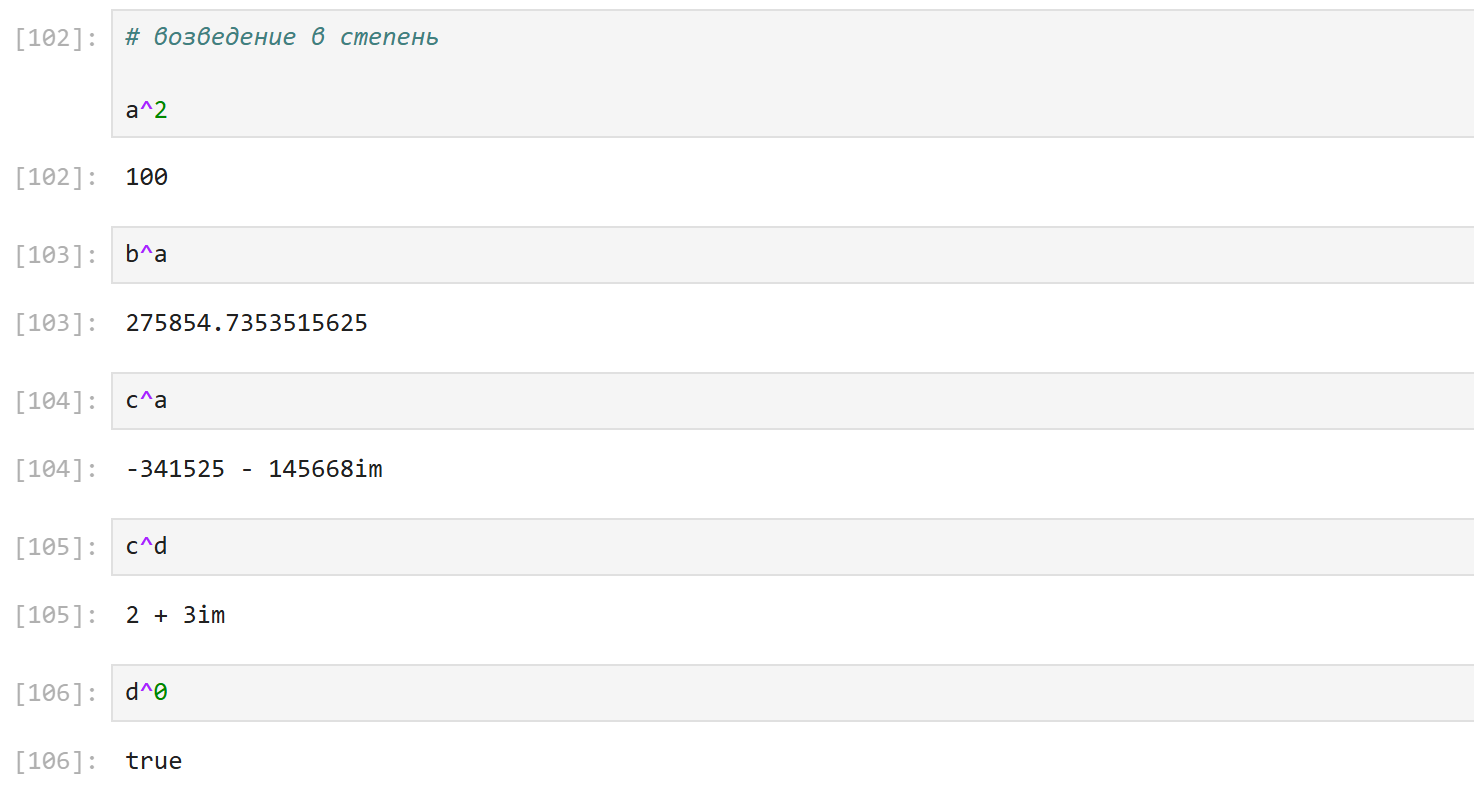


Рис. 29: Примеры для возведения в степень

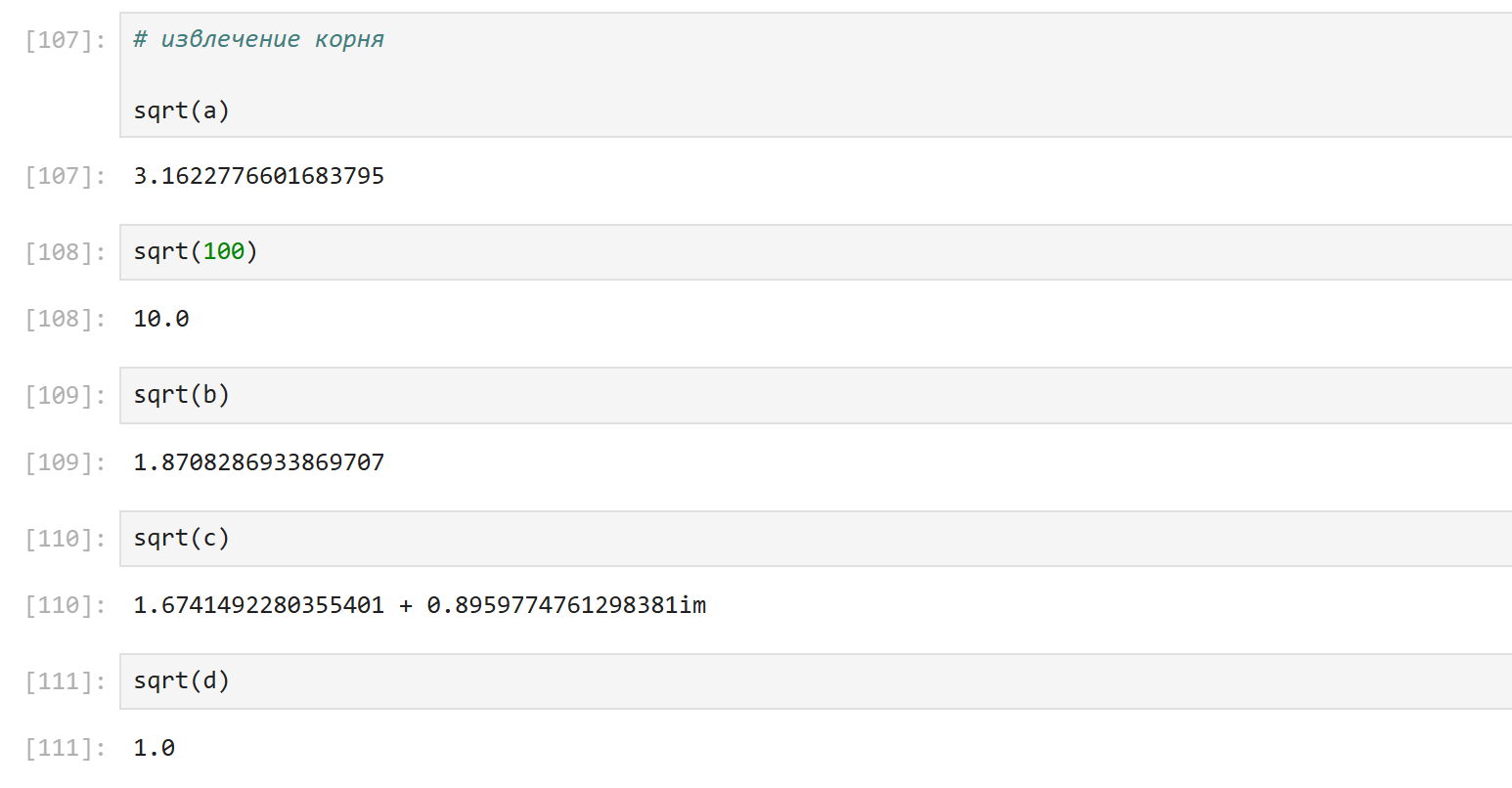


Рис. 30: Примеры для извлечения корня



Рис. 31: Примеры для сравнения



Рис. 32: Примеры для логических операций

## 5.4 Операции над матрицами и векторами

Определила две матрицы и два вектора (вектор-строка и вектор-столбец).(рис. 33)

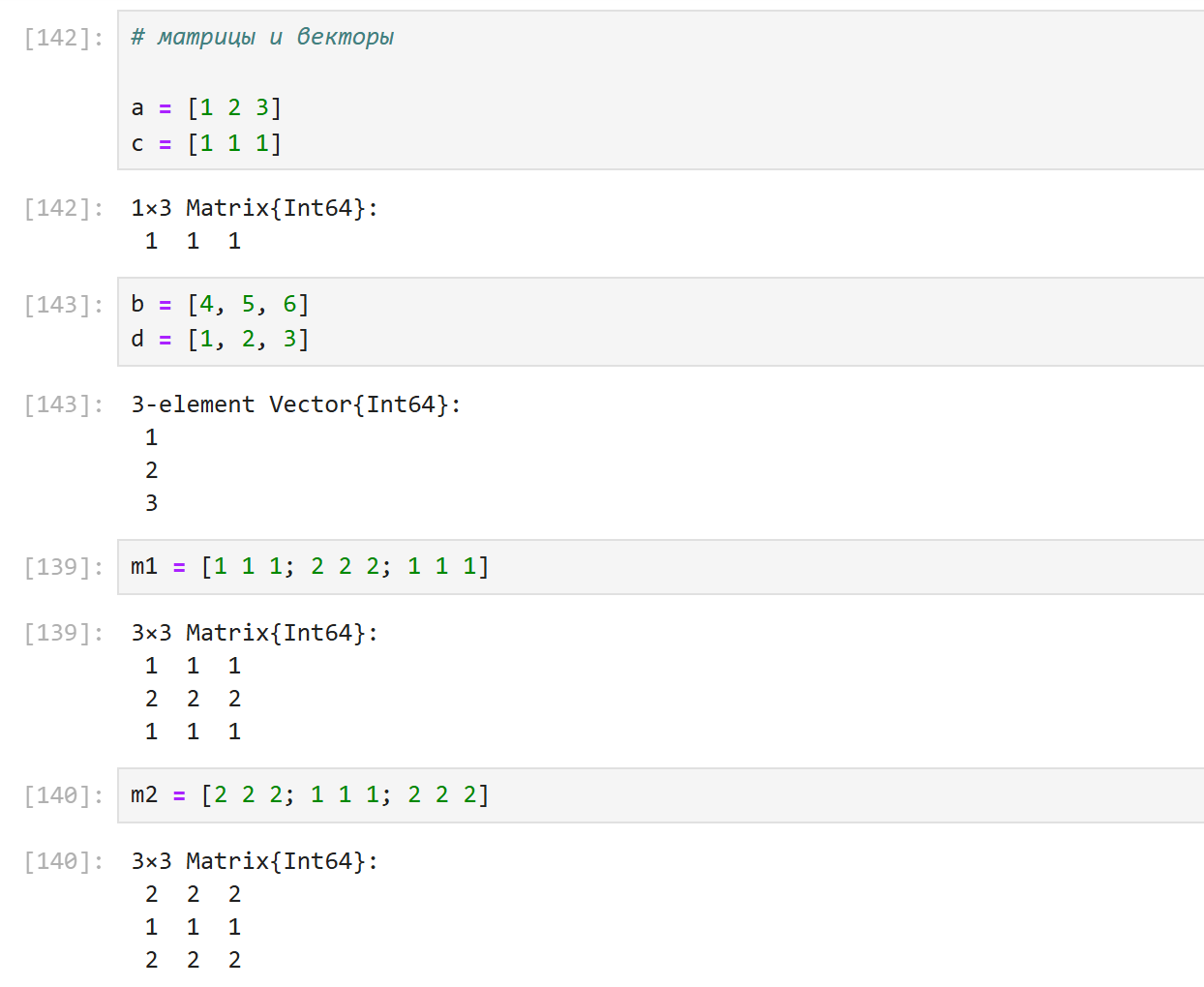


Рис. 33: Определение матриц и векторов

Привела примеры с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр (рис. 34-39)



Рис. 34: Примеры для сложения векторов и матриц

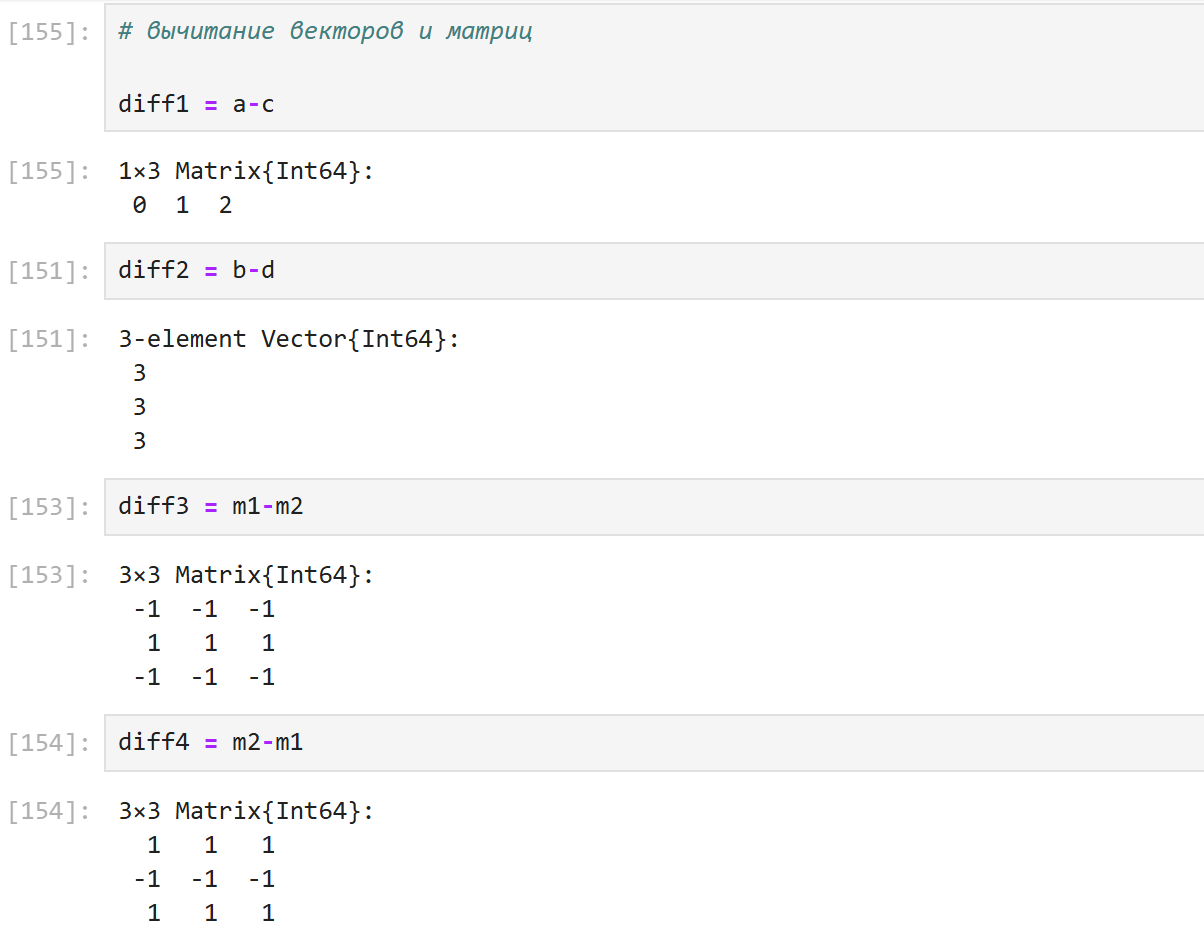


Рис. 35: Примеры для вычитания векторов и матриц

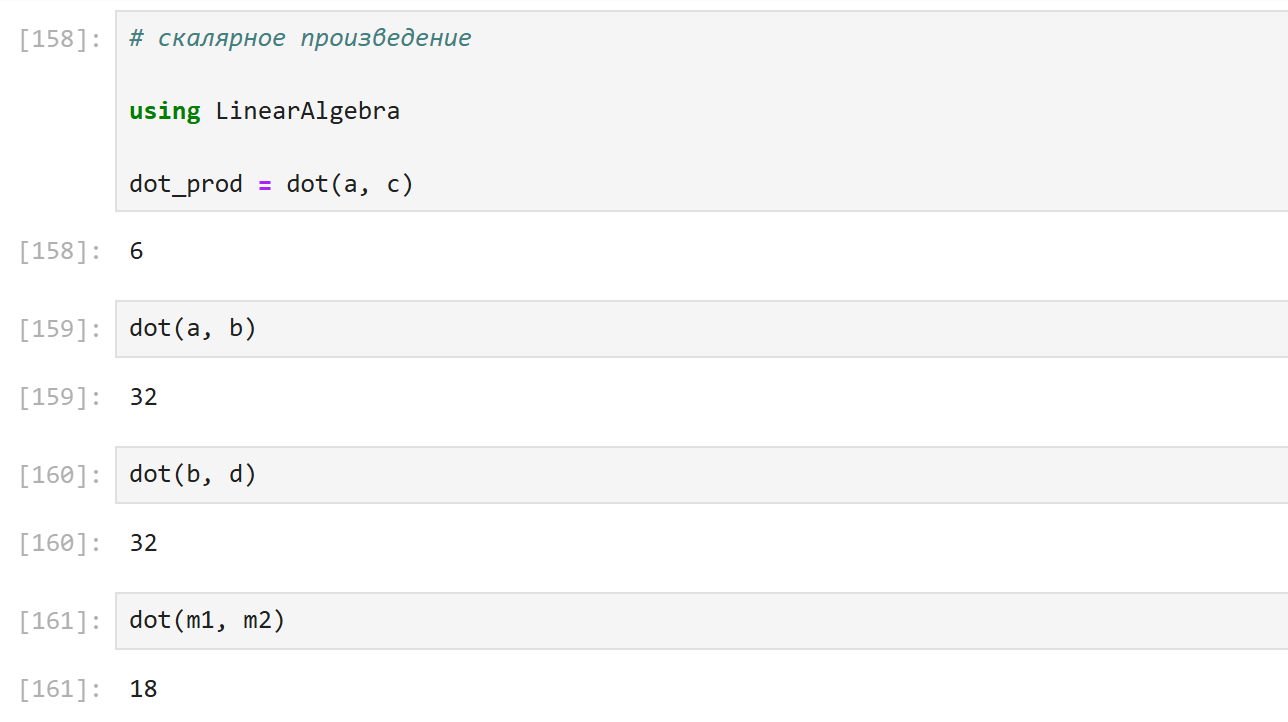


Рис. 36: Примеры для скалярного произведения

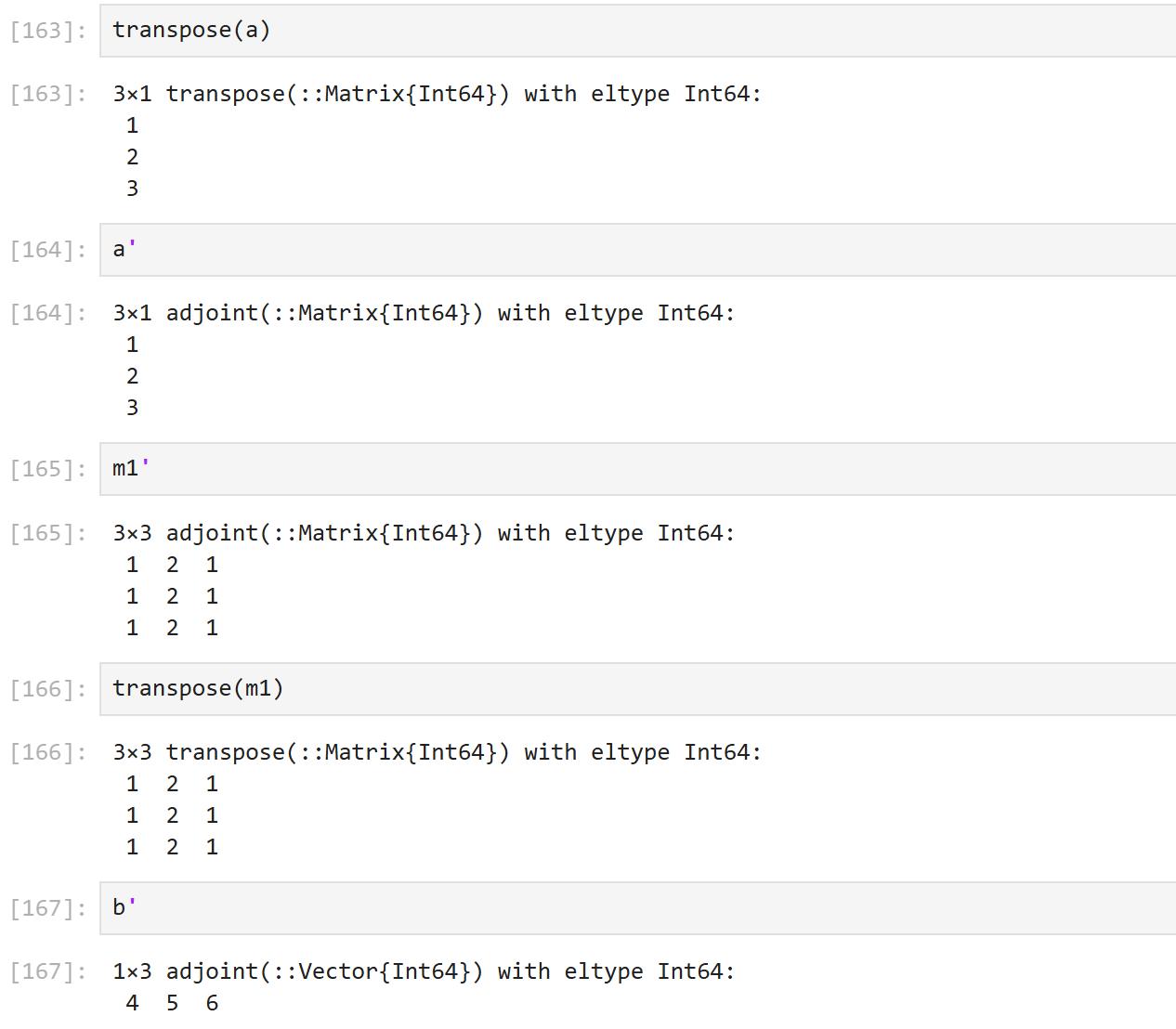


Рис. 37: Примеры для сложения транспонирования

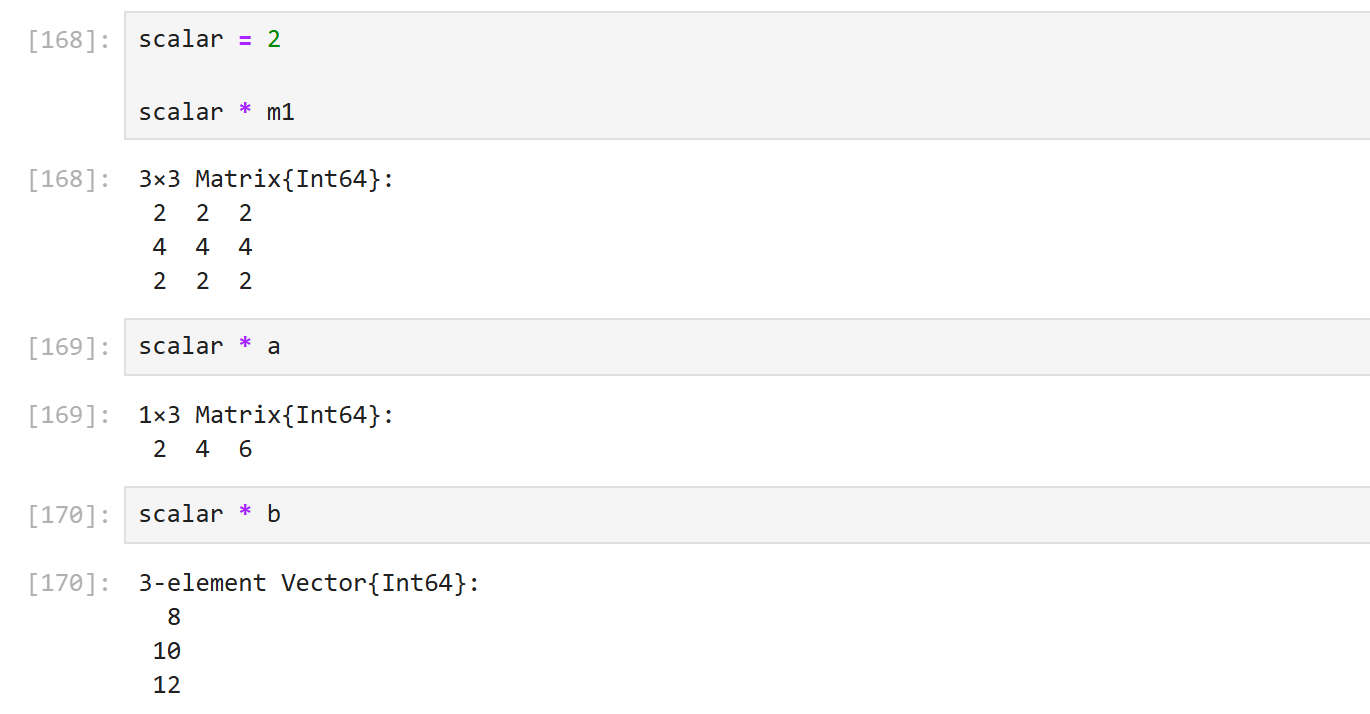


Рис. 38: Примеры для умножения на скаляр

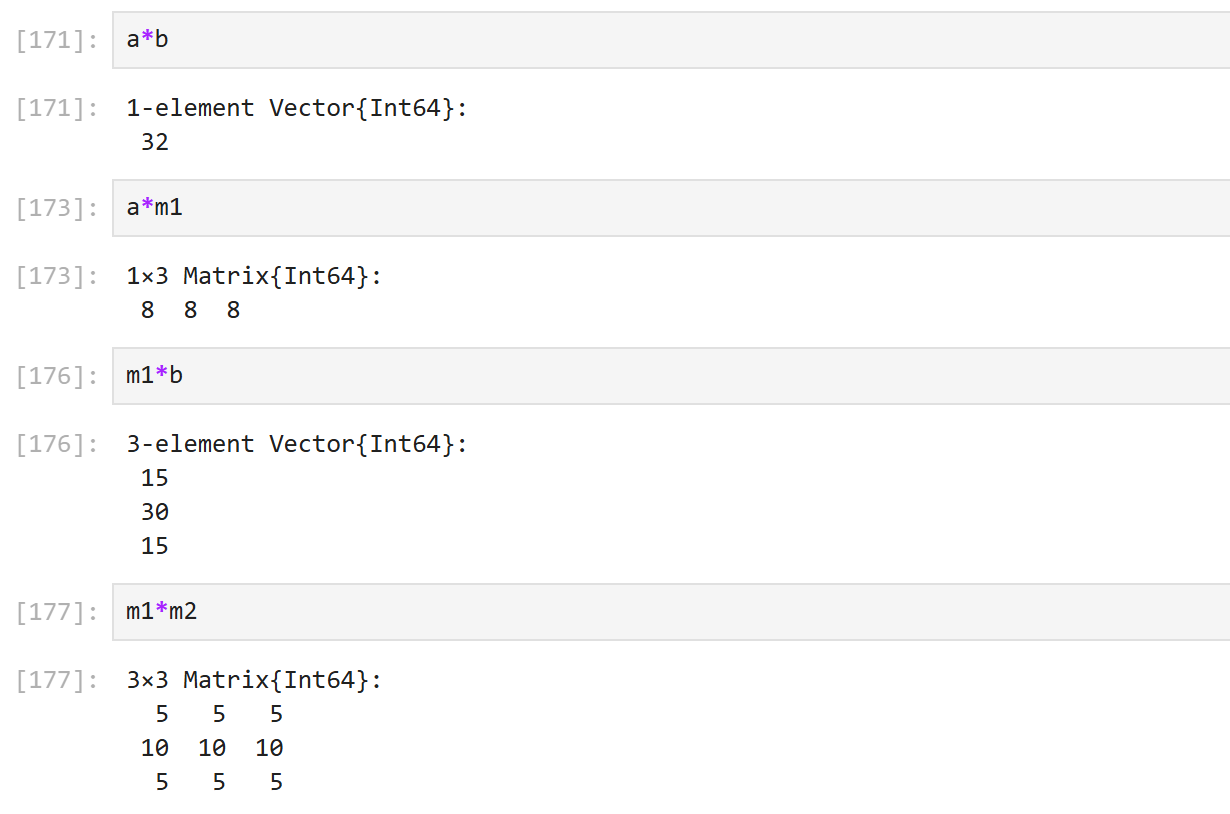


Рис. 39: Примеры для скалярного произведения

# 6 Выводы

В результате выполнения данной лабораторной работы я подготовила рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомилась с основами синтаксиса Julia

# Список литературы