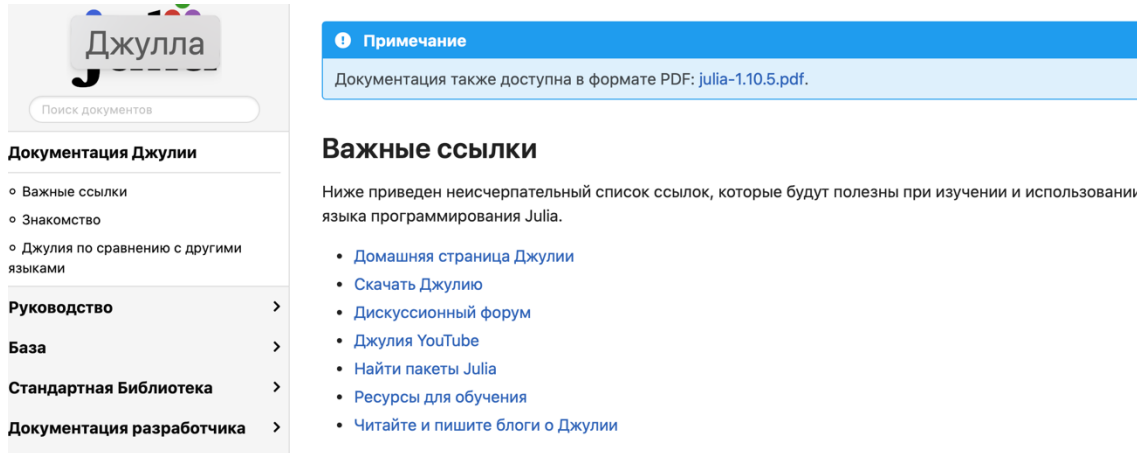


# Язык программирования Julia

<https://docs.julialang.org/en/v1/>

Перейдя по ссылке мы видим удобный и понятный интерфейс.



Сразу приведено множество полезных ссылок по изучению языка.

Слева мы видим, что сайт сразу разделен по темам.

На главной странице рассказывают о возможностях языка, далее идет сравнение Julia с другими языками программирования.

После идет инструкция по установке данного языка.

И в последствии можно приступить к изучению, данный сайт подойдет как для новичков, которые хотят изучить данный язык, так и для опытных программистов, ведь список тем и материалов очень велик.

## Модули

Modules in Julia help organize code into coherent units. They are delimited syntactically inside `module` `NameOfModule` ... `end`, and have the following features:

1. Модули - это отдельные пространства имен, каждое из которых представляет новую глобальную область. Это полезно, потому что позволяет использовать одно и то же имя для разных функций или глобальных переменных без конфликтов, если они находятся в отдельных модулях.
2. Modules have facilities for detailed namespace management: each defines a set of names it exports, and can import names from other modules with `using` and `import` (we explain these below).
3. Модули могут быть предварительно скомпилированы для более быстрой загрузки и могут содержать код для инициализации во время выполнения.

Как правило, в больших пакетах Julia вы увидите код модуля, организованный в файлы, например

```
module SomeModule

# export, using, import statements are usually here; we discuss these below

include("file1.jl")
include("file2.jl")

end
```

Files and file names are mostly unrelated to modules; modules are associated only with module expressions. One can have multiple files per module, and multiple modules per file. `include` behaves as if the contents of the source file were evaluated in the global scope of the including module. In this chapter, we use short and simplified examples, so we won't use `include`.

Так расписано множество тем, сначала идет объяснение темы как таковой, после приведены легкие примеры написания кода по этой теме и дальше идет по увеличению сложности.

<https://docs.juliaplots.org/stable/>

"Визуализация данных в Julia" - ресурс, посвященный библиотекам и инструментам для визуализации данных на языке Julia.

Сайт внешне похож на предыдущий, все те же удобства и практичность.

Но в данном сайте упор идет больше на обучение визуала в Julia.

Сделай несколько волн

```
using Plots
default(legend = false)
x = y = range(-5, 5, length = 40)
zs = zeros(0, 40)
n = 100

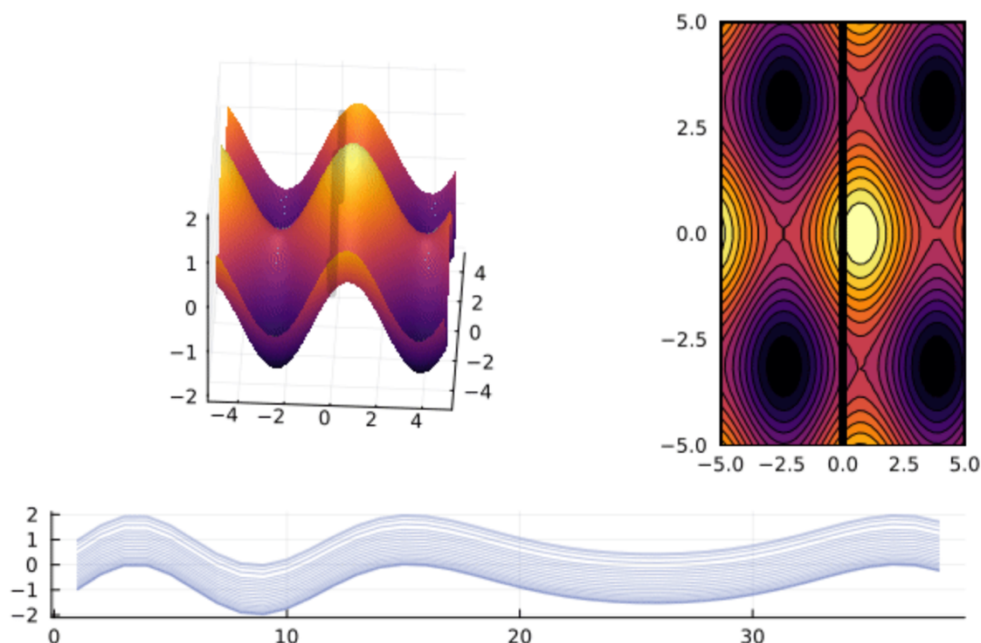
@gif for i in range(0, stop = 2π, length = n)
    f(x, y) = sin(x + 10sin(i)) + cos(y)

    # create a plot with 3 subplots and a custom layout
    l = @layout [a{0.7w} b; c{0.2h}]
    p = plot(x, y, f, st = [:surface, :contourf], layout = l)

    # induce a slight oscillating camera angle sweep, in degrees (azimuth, altitude)
    plot!(p[1], camera = (10 * (1 + cos(i)), 40))

    # add a tracking line
    fixed_x = zeros(40)
    z = map(f, fixed_x, y)
    plot!(p[1], fixed_x, y, z, line = (:black, 5, 0.2))
    vline!(p[2], [0], line = (:black, 5))

    # add to and show the tracked values over time
    global zs = vcat(zs, z')
    plot!(p[3], zs, alpha = 0.2, palette = cgrad(:blues).colors)
end
```



Приведено множество примеров написания кода для построения различных графиков. Все наглядно и просто.

<https://julialang.org>

Вот еще один интересный сайт связанный с изучением языка Julia.

На этом сайте представлены такие разделы как :

«Блог» - где рассказывается о планах по обновлению и развитию языка ,так же показана вся история обновлений.

«Сделай вклад»

Добро пожаловать в руководство участника экосистемы Julia и в сообщество!

Цель этого руководства - **помочь вам начать вносить** свой вклад в экосистему Julia. Он содержит информацию о [преимуществах становления участником Julia](#) и [инструментах](#), необходимых для начала работы. Кроме того, вы можете найти различные [пути вклада](#), которые вы можете выбрать, и как получить [помощь](#) от сообщества.

Сайт дает возможность принять участие в жизни и развитии языка.

Так же рассказывается ,как пользователю получить выгоду и помощь.

Научное программирование на языке Julia:

[https://exponenta.ru/storage/app/media/Conf\\_2023/Презентации\\_с\\_конференции\\_2023/Моделирование\\_в\\_инженерном\\_деле\\_5.04.2023/khirulin\\_kamil.pdf](https://exponenta.ru/storage/app/media/Conf_2023/Презентации_с_конференции_2023/Моделирование_в_инженерном_деле_5.04.2023/khirulin_kamil.pdf)

Презентация очень обширная в плане информации о языке, описывает способности языка.

Примеры написания кода:

1. Простые математические действия:

1	<code>a = 9</code>	10
2	<code>b = 1</code>	
3	<code>c = a + b</code>	
4	<code>println(c)</code>	

Сложение двух чисел.

2. Генерация случайного числа:

1	<code>random_number = rand(1:10)</code>	8
2	<code>println(random_number)</code>	

Функция `random_number` выдает случайное число из указанного диапазона.

3. Нахождение квадратного корня:

1	<code>x = 9</code>	3.0
2	<code>result = sqrt(x)</code>	
3	<code>println(result)</code>	

Функция `sqrt` находит квадратный корень из указанного числа.

4. Нахождение минимального числа из массива:

1	<code>numbers = [1, 5, 3, 9, 2]</code>	1
2	<code>min_value = minimum(numbers)</code>	
3	<code>println(min_value)</code>	

Функция `min_value` находит минимальное число из указанного массива.

5. Сортировка:

1	<code>numbers = [1, 5, 3, 9, 2]</code>	[1, 2, 3, 5, 9]
2	<code>sort_numbers = sort(numbers)</code>	
3	<code>println(sort_numbers)</code>	

Функция `sort_numbers` сортирует числа из заданного массива в порядке возрастания.