

Язык программирования Julia

Материалы для изучения

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

Данный ресурс является официальной документацией по языку программирования Julia. Здесь представлена вся необходимая информация для программиста, который хочет изучить новый язык. Сначала идет знакомство с Julia, сравнение с другими языками. Далее представлена инструкция по скачиванию и установке. После скачивая можно начинать изучать язык. На данном сайте все темы удобно разделены, что поможет структурировано изучать материал.

http://www.unn.ru/books/met_files/JULIA_tutorial.pdf

Учебно-методическое пособие представляет собой базовое руководство по языку Julia. Пособие содержит сведения по установке интерпретатора языка Julia, принципах работы, основных конструкциях и возможностях языка.

https://cmp.phys.msu.ru/sites/default/files/VA_Antonyk_Julia_2019.pdf

Пособие знакомит читателей с новым языком программирования Julia. Показывает возможности и особенности, а также на простых примерах иллюстрируются основные идеи в реализации языка Julia

https://exponenta.ru/storage/app/media/Conf_2023/Презентации_с_конференции_2023/Моделирование%20в%20инженерном%20деле_5.04.2023/khirulin_kamil.pdf

Презентация содержит большое количество полезной информации о языке программирования Julia. Описаны особенности языка, средства разработки. Также представлен синтаксис языка и программы.

Примеры кода

1. Функция для вычисления факториала

```
function factorial(n::Int)
    if n == 0
        return 1
    else
        return n * factorial(n - 1)
    end
end

println(factorial(5))
```

[1] ✓ 4.5s

... 120

В этом коде создается функция `factorial`, которая с помощью рекурсии вычисляет факториал числа `n`. Если число равно нулю, то сразу выводится единица.

2. Работа с массивами

```
arr = [1, 2, 3, 4, 5]
doubled = [x * 2 for x in arr]
println(doubled)
```

[2] ✓ 0.4s

... [2, 4, 6, 8, 10]

Сначала создается массив `arr`, далее каждый его элемент умножается на два и выводится новый массив.

3. Простейшие вычисления

```
5+9921
```

[2] ✓ 0.0s

... 9926

Julia очень легко справляется с обычными математическими вычислениями, при этом не требуется использовать синтаксис, достаточно просто ввести выражение как на калькуляторе

4. Циклы

```
dream = ["Please", "buy", "me", "a", "mercedes"]
println(dream)

for word in dream
    print(word, " ")
end

[16] ✓ 0.0s

... ["Please", "buy", "me", "a", "mercedes"]
    Please buy me a mercedes
```

В этой программе слова сначала выведены в виде массива, а дальше, цикл проходит по каждому элементу массива и разделяя их пробелами выводит нам предложение в нормальном виде

5. Решение системы линейных уравнений

```
using LinearAlgebra

A = [4 8; 2 2]
B = [5; 4]

x = A \ B

println(x)

[23] ✓ 0.0s

... [2.75, -0.75]
```

В первой строке мы подключаем пакет для линейной алгебры. Далее на основе теории по алгебре, мы создаем два массива. Массив A содержит коэффициенты системы, массив B содержит свободные члены системы. Далее используя формулу для нахождения неизвестных, программа выводит нам ответ