

Вариативное задание 2.2

Практика

аннотированный список статей про Julia

Julia – высокоуровневый язык, который разработан для научного программирования. Язык поддерживает широкий функционал для математических вычислений и работы с большими массивами данных. Это язык с динамической компиляцией, программы на Julia компилируются в быстрый нативный код для таких платформ, как Windows, macOS и Linux. Приложения на языке Julia работают так же быстро, как приложения, которые написаны на быстрых низкоуровневых языках, таких как C или C++. Синтаксически Julia похож на Python и MATLAB, но эти языки – интерпретируемые, поэтому приложения на Julia работают быстрее. Ниже приведён аннотированный список статей про Julia.

1. Шесть причин выучить Julia в 2024 году

Автор: Максим Сидоров

Шесть причин выучить Julia в 2024 году // Habr URL:

https://habr.com/ru/companies/etmc_exponenta/articles/831670/ (дата обращения: 23.09.2024).

Аннотация

Статья на Habr рассматривает шесть ключевых причин, почему язык программирования Julia становится все более популярным в 2024 году. Автор акцентирует внимание на его высоких показателях производительности, простоте синтаксиса и широком применении в научных вычислениях и машинном обучении. Также обсуждаются перспективы развития языка и его сообщество.

2. Кратко про FP в Julia

Автор: bascasedaily1

Кратко про FP в Julia // Habr URL:

<https://habr.com/ru/companies/otus/articles/794819/> (дата обращения: 23.09.2024).

Аннотация

В этой статье на Habr представлена информация о функциональном программировании (FP) в языке Julia. Рассматриваются основные концепции

FP, такие как функции высшего порядка и неизменяемость данных, а также преимущества использования функционального подхода для решения задач. Приведены примеры кода и рекомендации по использованию FP в Julia.

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

println(factorial(5)) # 120
```

```
# неизменяемая структура данных
struct ImmutableVector
    data::Vector{Int}
end

# функция, создающая новый ImmutableVector с добавленным элементом
function add_element(ivec::ImmutableVector, element::Int)
    new_data = copy(ivec.data)
    push!(new_data, element)
    return ImmutableVector(new_data)
end

ivec = ImmutableVector([1, 2, 3])
ivec_new = add_element(ivec, 4)
```

3. Julia. Знакомство

Автор: Yermack

Julia. Знакомство // Habr URL: <https://habr.com/ru/articles/423811/> (дата обращения: 23.09.2024).

Аннотация

Статья представляет собой вводное руководство по языку Julia, описывающее его особенности и преимущества. Рассматриваются основные синтаксические конструкции, инструменты разработки и библиотеки,

доступные для пользователей. Цель статьи — помочь новичкам быстро освоиться с языком и начать писать свои первые программы.

```
x = 5+8
2x - 3x + 2x^2
Out: 325
```

```
A = [1 2 3; 6 5 4; 7 8 9]
Out: 3x3 Array{Int64,2}:
 1  2  3
 6  5  4
 7  8  9
A[2,1]
Out: 6
A[end]
Out: 9
size(A)
Out: (3, 3)
```

```
y = 5
y += 2
4 <= y < 8
Out: true
```

```
function myabs(x)
    if x >= 0
        return x
    else
        return -x
    end
end
myabs(-12)
Out: 12
```

4. Что такое Julia

Автор: Работник Workspace

Что такое Julia // Workspace URL: <https://workspace.ru/tools/language/julia/>
(дата обращения: 23.09.2024).

Аннотация

На сайте Workspace представлена краткая информация о языке программирования Julia, его назначении и области применения. Описываются ключевые характеристики языка, такие как высокая скорость выполнения кода и возможность работы с большими объемами данных. Также упоминаются сферы, где Julia находит активное применение, включая научные исследования и анализ данных.

5. О языке программирования Julia

Автор: Дмитрий Беляев

О языке программирования Julia // Gitbooks URL:
<https://closescreen.gitbooks.io/julia-lang-ru/content/> (дата обращения:
23.09.2024).

Аннотация

Этот ресурс на Gitbooks предлагает подробное введение в язык Julia, охватывающее его историю, архитектуру и особенности. В статье рассматриваются основные библиотеки и инструменты, которые делают разработку на Julia удобной и эффективной. Также обсуждаются примеры использования языка в различных областях науки и техники.

```
function fib(n)
    if n == 0 return 0 end
    if n == 1 return 1 end
    return fib(n-1) + fib(n-2)
end
```

6. Julia 1.6 Documentation

Автор: Разработчики Julia
Julia 1.6 Documentation // Julialang URL: <https://docs.julialang.org/en/v1.6/> (дата обращения: 23.09.2024).

Аннотация

Официальная документация языка, где можно найти полное описание синтаксиса, стандартной библиотеки и примеры использования различных функций. Рассматриваются основные концепции, такие как динамическая типизация и высокопроизводительность.