

BCP 2.2

Подборка материалов, включающую аннотированные статьи и ресурсы по языку программирования Julia, а также примеры решения различных задач на этом языке

1. <https://xakep.ru/2019/08/05/julia/>

- Julia - молодой язык программирования для научных вычислений.
- Создатели хотели совместить удобство R и Python с производительностью C.
- Julia разрабатывается в Массачусетском технологическом институте с 2009 года.
- Язык имеет открытый исходный код и либеральную лицензию.
- Julia обладает скоростью C и динамизмом Ruby.
- Язык поддерживает динамическую типизацию и параллелизм.
- Julia имеет богатый язык описательных типов данных и опциональность.
- Синтаксис Julia лучше, чем у Matlab, но пока не сравним с Python.

2. <https://julialang.org/>

- Язык программирования Julia - это высокоуровневый, динамически типизированный язык, ориентированный на научные вычисления и численные расчеты.
- Julia разработана с целью сочетания производительности низкоуровневых языков, таких как C или Fortran, с удобством высокоуровневых языков, как Python или R.
- На сайте представлена информация об особенностях языка, его экосистеме (библиотеки, пакеты), инструментах разработки (IDE, редакторы), а также руководства по установке и использованию.
- Сайт содержит подробную документацию, включая учебные материалы, руководства, примеры кода и статьи. Это помогает новым пользователям быстро освоить язык.
- Особое внимание уделено применению Julia в различных областях: научные вычисления, машинное обучение, анализ данных, веб-разработка и др.
- Сайт также информирует о сообществе разработчиков Julia, конференциях и других ресурсах для взаимодействия с ними.
- В целом, сайт Julia Lang предоставляет всестороннее и структурированное представление о языке программирования Julia и его экосистеме.

3. <https://workspace.ru/tools/language/c/>

- Julia — высокоуровневый язык программирования для научных вычислений.
- Поддерживает динамическую компиляцию, математические вычисления и работу с массивами данных.
- Синтаксис похож на Python и MATLAB.
- Разрабатывается в IDE Juno, интегрированной с Atom.
- Поддерживает визуализацию, JIT-компиляцию и облачные вычисления.
- Работает с большими массивами данных и поддерживает параллельные вычисления.
- Поддерживает метапрограммирование и взаимодействие с C, Python и R.
- Применяется в математике, анализе данных, машинном обучении и веб-разработке.
- Преимущества: простой синтаксис, хорошая производительность, регулярные обновления.
- Недостатки: малое количество специалистов, возможные баги.
- Примеры готовых работ на GitHub: github.com/search?q=julia

4. <https://appttractor.ru/info/articles/yazyk-programmirovaniya-julia.html>

- Julia - высокоуровневый язык программирования для научных вычислений и технических задач.
- Julia сочетает простоту и читаемость с высокой производительностью.
- Язык обладает превосходной производительностью, сравнимой с C и Fortran.
- Julia имеет динамическую типизацию, упрощающую разработку и ускоряющую процесс программирования.
- Платформа имеет обширную библиотеку для различных научных и инженерных задач.
- Julia обладает интерактивной средой с REPL для быстрого тестирования и взаимодействия с языком.
- Язык поддерживает многопоточность и параллельные вычисления.
- Julia является проектом с открытым исходным кодом и активным сообществом.
- Синтаксис Julia похож на Python, что облегчает изучение для тех, кто знаком с программированием.

5. <https://www.litres.ru/book/vadim-nikitin-32700223/julia-yazyk-programmirovaniya-bystryy-start-69596290/chitat-onlayn/>

- Для работы с Julia необходимо загрузить и установить дистрибутив на Windows, macOS, Linux, FreeBSD.
- Julia имеет встроенную справочную систему и менеджер пакетов для работы с пакетами.
- Пакеты расширяют функциональность Julia.
- Переменные в Julia - это имена, привязанные к значениям.
- Имена переменных могут быть любой длины и содержать символы Unicode, кроме цифр.
- Имена переменных чувствительны к регистру, кроме ключевых слов.

Примеры кодов на Julia

1. Основные математические операции

```

1 a = 15
2 b = 10
3 println("Сложение: $(a + b)")
4 println("Вычитание: $(a - b)")
5 println("Умножение: $(a * b)")
6 println("Деление: $(a / b)")
7

```

```

Сложение: 25Вычитание: 5
Умножение: 150
Деление: 1.5

```

2. Нахождение факториала числа

```

1 function factorial(n)
2     if n == 0
3         return 1
4     else
5         return n * factorial(n-1)
6     end
7 end
8
9 println("Факториал равен: $(factorial(5))")
10

```

```

Факториал равен: 120

```

3. Поиск и вывод первых 10 чисел фибоначчи

```

1 function fibonacci(n)
2     if n <= 1
3         return n
4     else
5         return fibonacci(n-1) + fibonacci(n-2)
6     end
7 end
8
9 println("Первые 10 чисел Фибоначчи:")
10 for i = 0:9
11     print("$(fibonacci(i)) ")
12 end
13 println()

```

```

Первые 10 чисел Фибоначчи:0 1 1 2 3 5 8 13 21 34

```

4. Сортировка элементов массива

```
1 numbers = [5, 2, 8, 1, 9]
2 sorted_numbers = sort(numbers)
3 println("отсортированный массив :
    $sorted_numbers")
```

отсортированный массив : [1, 2, 5, 8, 9]

5. Генерация случайного числа

```
1 using Random
2 rand_num = rand(1:100)
3 println("Случайное число:
    $rand_num")
4
```

Случайное число: 91