

# Ruby

ЛУЧШИЙ ДРУГ ПРОГРАММИСТА

[Скачать](#) [Документация](#) [Библиотеки](#) [Сообщество](#) [Новости](#) [Безопасность](#)

0 Ruby

1

2

3

4

## Ruby за двадцать минут

### Вступление

Здесь представлено небольшое введение в Ruby, которое должно занять не более двадцати минут. Предполагается, что вы уже установили Ruby. (Если на вашем компьютере нет Ruby, [скачайте](#) и установите его перед тем как начать.)

### Интерактивный Ruby

Ruby устанавливается вместе с программой, которая покажет результат любого Ruby кода, который вы напишете в ней. Играть с кодом Ruby интерактивно – отличный способ узнать язык.

Откройте IRB (сокращение для Interactive Ruby).

- Если вы используете **macOS**, откройте `Terminal`, наберите `irb` и нажмите Enter.
- Если вы используете **Linux**, откройте `shell`, наберите `irb` и нажмите Enter.
- Если вы используете **Windows**, откройте `Interactive Ruby` из секции Ruby вашего меню Пуск.

```
irb(main):001:0>
```

Окей, итак он запущен. Что дальше?

Наберите это: `"Hello World"`

```
irb(main):001:0> "Hello World"
=> "Hello World"
```

Ruby повиновался вам!

Начните сейчас, это легко!

[Попробуйте Ruby! \(в своем браузере\)](#)

[Ruby за двадцать минут](#)

[В Ruby из других языков](#)

Исследуйте новый мир...

[Документация](#)

[Научные публикации](#)

[Библиотеки](#)

[Истории успеха](#)

Вступайте в дружелюбное и развивающееся сообщество.

[Почтовые рассылки](#): Разговоры о Ruby в кругу программистов со всего мира.

[Группы пользователей](#):

Познакомьтесь с рубистами рядом с вами.

[Блоги](#): Читайте о том, что происходит в сообществе Ruby прямо сейчас.

Что только что произошло? Неужели мы только что написали кратчайшую в мире программу "Hello World"? Не совсем. Вторая строка, это всего-лишь способ IRB показать нам результат последнего выполненного выражения. Если мы хотим напечатать "Hello World", мы должны написать немного больше:

```
irb(main):002:0> puts "Hello World"
Hello World
=> nil
```

`puts` – основная команда в Ruby, позволяющая напечатать что-нибудь. Но что же такое тогда `=> nil`? Это результат выполнения выражения. `puts` всегда возвращает `nil`, который в Ruby обозначает значение абсолютно-положительного ничего.

## Ваш бесплатный калькулятор

Теперь мы знаем достаточно, чтобы использовать IRB как простейший калькулятор:

```
irb(main):003:0> 3+2
=> 5
```

Три плюс два. Достаточно просто. Что насчет три *умножить* на два? Вы можете набрать сами, это достаточно короткий код, но также вы можете вернуться выше и изменить то, что вы ввели ранее. Попробуйте нажать **стрелку вверх** на клавиатуре и вы увидите строку с `3+2`. Если это так – вы можете использовать стрелку влево, чтобы переместить курсор прямо за `+`, нажать `backspace` и поменять его на `*`.

```
irb(main):004:0> 3*2
=> 6
```

Теперь давайте попробуем возвести три в квадрат:

```
irb(main):005:0> 3**2
=> 9
```

В руби `**` – способ возвести число в степень. Но что если вы хотите пойти в обратном направлении и узнать квадратный корень от чего-либо?

```
irb(main):006:0> Math.sqrt(9)
=> 3.0
```

Окей, погодите, что это было? Если вы думаете "это был способ выяснить квадратный корень из девяти", вы правы. Но давайте приглядимся к деталям. Во-первых, что такое `Math`?

[Ядро Ruby](#): Помощь в полировке последней версии Ruby.

[Решение проблем](#): Сообщайте или помогайте решать проблемы в Ruby. Подписаться

[Последние новости \(RSS\)](#)

## Модули группируют код по темам

`Math` – это встроенный модуль для математических операций. Модули выполняют две роли в Ruby. Наш пример показывает первую роль: сгруппировать похожие методы вместе под знакомым именем. `Math` также содержит методы типа `sin()` и `tan()`.

Следующее – это точка. Что делает точка? Точка – это то, как вы указываете получателя сообщения. Что такое сообщение? В данном случае это `sqrt(9)`, которое означает вызов метода `sqrt`, сокращение от “square root” (квадратный корень), с параметром `9`.

Результат от вызова этого метода – `3.0`. Вы можете заметить, что это не просто `3`. Это потому, что большинство значений квадратного корня от числа не являются целыми, так что метод всегда возвращает число с плавающей точкой.

Что если мы хотим запомнить результат какой-либо математической операции? Присвойте это значение переменной.

```
irb(main):007:0> a = 3 ** 2
=> 9
irb(main):008:0> b = 4 ** 2
=> 16
irb(main):009:0> Math.sqrt(a+b)
=> 5.0
```

Калькулятор это замечательно, но мы отошли от традиционного сообщения `Hello World`, которым начали наше введение и на котором мы должны были сфокусироваться... [так что давайте вернемся к нему.](#)

[Скачать](#) [Документация](#) [Библиотеки](#) [Сообщество](#) [Новости](#) [Безопасность](#) [О Ruby](#)

Этот сайт на других языках: [Български](#), [Deutsch](#), [English](#), [Español](#), [Français](#), [Bahasa Indonesia](#), [Italiano](#), [日本語](#), [한국어](#), [polski](#), [Português](#), [Русский](#), [Türkçe](#), [Tiếng Việt](#), [简体中文](#), [繁體中文](#).

[Этот сайт](#) с гордостью поддерживается членами сообщества Ruby.