

# Язык программирования Ruby

CS 212

20 февраля 2026 г.

# Что разберём сегодня

1. Типы
2. Функции, аргументы и блоки
3. Управляющие конструкции
4. Литералы и строки
5. Массивы
6. Хэши
7. Даты
8. Файлы

# Типы данных в Ruby

```
1 puts 42.class      # Integer
2 puts 3.14.class   # Float
3 puts true.class   # TrueClass
4 puts nil.class    # NilClass
5 puts :ok.class    # Symbol
6 puts 'ruby'.class # String
7 puts [1, 2].class # Array
8 puts({a: 1}.class) # Hash
```

Листинг 1: Все – объекты

# Переменные и константы

```
1 site = 'sfedu.ru'      # local
2 @user = 'alice'       # instance
3 @@count = 0           # class
4 $env = 'dev'           # global
5
6 APP_NAME = 'Ruby L2'  # constant
```

Листинг 2: Имена и области

# Методы и аргументы

```
1 def greet(name, lang: 'ru')
2   lang == 'ru' ? "Privet, #{name}!" : "Hi,
3     #{name}!"
4
5 puts greet('World')
6 puts greet('Bob', lang: 'en')
```

Листинг 3: Позиционные и именованные аргументы

# Значения по умолчанию и splat

```
1 def sum(prefix = 'sum', *nums)
2   "#{prefix}: #{nums.sum}"
3 end
4
5 puts sum('total', 1, 2, 3)
6 puts sum('empty')
```

Листинг 4: Гибкая сигнатура

## Блоки

```
1 [1, 2, 3].each { |n| puts n * 10 }
2
3 def around
4   puts 'start'
5   yield if block_given?
6   puts 'finish'
7 end
8
9 around { puts 'work' }
```

Листинг 5: each и yield

## if / elsif / unless

```
1 score = 74
2
3 if score >= 90
4   grade = 'A'
5 elsif score >= 70
6   grade = 'B'
7 else
8   grade = 'C'
9 end
10
11 puts 'retry' unless score > 0
```

Листинг 6: Условия

## case и циклы

```
1  role = :editor
2  puts case role
3      when :admin then 'all'
4      when :editor then 'write'
5      else 'read'
6      end
7
8  i = 0
9  while i < 3
10     puts i
11     i += 1
12 end
```

Листинг 7: case + while + each

# Литералы Ruby

```
1 num = 1_000_000
2 pi = 3.14
3 sym = :active
4 range = 1..5
5 words = %w[ruby is fun]
```

Листинг 8: Числа, символы, диапазоны

# Одинарные и двойные кавычки

```
1 name = 'Ruby'
2
3 single = 'Hello, #{name}\n' # no interpolation
4 double = "Hello, #{name}\n" # interpolation + \n
5
6 puts single
7 puts double
```

Листинг 9: Интерполяция и escape-последовательности

## frozen\_string\_literal

```
1 # frozen_string_literal: true
2
3 s = 'ruby'
4 # s << '!'      # FrozenError
5 t = "#{s} lang" # new string object
6
7 puts t
```

Листинг 10: Заморозка строковых литералов

# Строки: базовые методы

```
1 name = 'Ruby'
2 msg = "Hello, #{name}!"
3
4 puts msg.upcase
5 puts ' trim '.strip
6 puts 'abcabc'.gsub('a', 'A')
7 puts '42'.to_i + 8
```

Листинг 11: Полезные операции

# Строки: поиск и срезы

```
1 s = 'programming'
2
3 puts s.include?('gram') # true
4 puts s.start_with?('pro') # true
5 puts s.end_with?('ing') # true
6 puts s.index('g')      # 3
7
8 puts s[0, 4]   # prog
9 puts s[3..6]  # gram
10 puts s[-3..-1] # ing
```

Листинг 12: include?, index, slice, ranges

# Массивы: создание и доступ

```
1 arr = [1, 2, 3]
2 arr << 4
3 arr.unshift(0)
4
5 p arr[0]      # 0
6 p arr[-1]    # 4
7 p arr.take(3) # [0, 1, 2]
8 p arr.drop(2) # [2, 3, 4]
```

Листинг 13: Базовые операции

## Gotcha: инициализация объектом

```
1 a = Array.new(2, {})
2 a[0][:cat] = 'feline'
3
4 p a
5 # [[:cat=>"feline"], [:cat=>"feline"]]
6
7 b = Array.new(2) { {} }
8 b[0][:cat] = 'feline'
9 p b # [[:cat=>"feline"], {}]
```

Листинг 14: Array.new(n, obj)

# Массивы: преобразование

```
1  nums = [1, 1, 2, 3, 4]
2
3  squares = nums.map { |n| n * n }
4  evens = nums.select(&:even?)
5  unique = nums.uniq
6
7  p squares # [1, 1, 4, 9, 16]
8  p evens  # [2, 4]
9  p unique # [1, 2, 3, 4]
```

Листинг 15: map / select / uniq

# Массивы: полезные методы 1

```
1 arr = [1, nil, 2, 2, 3]
2
3 puts arr.fetch(10, 'oops')
4 puts arr.count(2) # 2
5 p arr.compact # [1, 2, 2, 3]
6 p arr.compact.uniq # [1, 2, 3]
```

Листинг 16: fetch, count, compact

## Массивы: полезные методы 2

```
1 nested = [1, [2, 3], [4, [5]]]
2 arr = %w[a b c d e]
3
4 puts arr.first      # a
5 puts arr.last      # e
6 p arr[1, 3]        # ["b", "c", "d"]
7 p arr.values_at(0, 2, 4)
8 p nested.flatten  # [1, 2, 3, 4, 5]
```

Листинг 17: first/last, slice, flatten

## Массивы: полезные методы 3

```
1  nums = [1, 2, 3, 4, 5, 6]
2
3  p nums.group_by { |n| n.even? }
4  puts nums.any? { |n| n > 5 } # true
5  puts nums.all? { |n| n > 0 } # true
```

Листинг 18: `group_by`, `any?`, `all?`

## ХЭШ: ОСНОВЫ

```
1 user = { name: 'Ann', age: 20 }
2
3 puts user[:name]
4 user[:city] = 'Rostov'
5 user[:age] += 1
6
7 p user.keys
8 p user.values
```

Листинг 19: Создание и доступ

## Gotcha: mutable default в Hash.new

```
1 h = Hash.new([])
2 h[:a] << 1
3 h[:b] << 2
4
5 p h[:a] # [1, 2]
6 p h[:b] # [1, 2]
7
8 ok = Hash.new { |hash, key| hash[key] = [] }
9 ok[:a] << 1
10 ok[:b] << 2
11 p ok    # {:a=>[1], :b=>[2]}
```

Листинг 20: Один объект для всех ключей

## Хэш: методы

```
1 h1 = { a: 1, b: 2 }
2 h2 = { b: 9, c: 3 }
3
4 puts h1.fetch(:a)
5 merged = h1.merge(h2)
6 merged.each { |k, v| puts "#{k}=# {v}" }
```

Листинг 21: fetch, merge, each

# Date и DateTime

```
1 require 'date'
2
3 d = Date.new(2026, 2, 20)
4 dt = DateTime.new(2026, 2, 20, 10, 30, 0)
5
6 puts d.year
7 puts dt.strftime('%d.%m.%Y %H:%M')
```

Листинг 22: Создание даты

# Парсинг и арифметика дат

```
1 require 'date'
2
3 d1 = Date.parse('2026-02-20')
4 d2 = Date.strptime('21.02.2026', '%d.%m.%Y')
5
6 puts (d2 - d1).to_i # 1
7 puts (d1 + 7).iso8601 # 2026-02-27
```

Листинг 23: parse / strptime

# Чтение и запись файлов

```
1 File.write('notes.txt', "line 1\nline 2")
2
3 File.open('notes.txt', 'r') do |f|
4   f.each_line { |line| puts line.strip }
5 end
```

Листинг 24: File.open с блоком

## Пути и проверки

```
1 path = File.join('data', 'users.csv')
2
3 puts File.extname(path)      # .csv
4 puts File.basename(path, '.csv')
5 puts File.exist?(path)
```

Листинг 25: Утилиты File

# Итоги

- ▶ База Ruby: типы, методы, блоки, условия, циклы.
- ▶ Коллекции: массивы и хэши.
- ▶ Практика: даты и работа с файлами.