

💎 Ruby Interview Questions and Answers (1–50)

1. What is Ruby?

Answer: Ruby is a high-level, open-source, object-oriented programming language known for simplicity and productivity.

2. Who developed Ruby?

Answer: Yukihiro “Matz” Matsumoto in 1995.

3. Is Ruby statically or dynamically typed?

Answer: Dynamically typed.

4. What is IRB in Ruby?

Answer: IRB stands for Interactive Ruby. It is a REPL (Read-Eval-Print Loop) for experimenting with Ruby code.

5. What is the difference between `puts`, `print`, and `p`?

Answer:

- `puts` adds a newline after output
- `print` does not
- `p` returns a more literal output (used for debugging)

6. What are Ruby data types?

Answer:

- Numbers (Integer, Float)
- String
- Symbol
- Array
- Hash
- Boolean
- Nil

7. What is a symbol in Ruby?

Answer: An immutable, unique identifier (starts with `:`), used as lightweight strings or keys in hashes.

8. What is a block in Ruby?

Answer: A chunk of code enclosed in `do...end` or `{...}` that can be passed to methods.

9. What is the difference between a block and a method?

Answer: A method is a reusable function; a block is passed into a method and executed within its context.

10. How do you define a method in Ruby?

Answer:

```
def greet(name)
  "Hello #{name}"
end
```

11. What is `nil` in Ruby?

Answer: Represents “nothing” or “no value”.

12. What is the difference between `==` and `===` in Ruby?

Answer:

- `==` checks value equality
- `===` is used in case statements and class comparisons

13. What is a class in Ruby?

Answer: A blueprint for creating objects with methods and attributes.

14. How do you define a class in Ruby?

Answer:

```
class Person
  def initialize(name)
    @name = name
  end
end
```

15. What is an object in Ruby?

Answer: An instance of a class.

16. What is `self` in Ruby?

Answer: Refers to the current object.

17. What are instance variables?

Answer: Variables that begin with `@`, specific to each object.

18. What is inheritance in Ruby?

Answer: Allows one class to use methods from another class.

```
class Dog < Animal
end
```

19. What is a module?

Answer: A collection of methods and constants. Can't be instantiated but can be mixed into classes.

20. How do you mix in a module?

Answer:

```
include ModuleName
```

21. What is the difference between `include` and `extend`?

Answer:

- `include` adds methods as instance methods
- `extend` adds them as class methods

22. What is mixin in Ruby?

Answer: The inclusion of a module into a class to share reusable code.

23. What is a lambda in Ruby?

Answer: An anonymous function:

```
say = -> { puts "Hi" }
```

24. What is the difference between `lambda` and `proc`?

Answer:

- `lambda` checks arguments strictly
- `proc` is more flexible and returns from the calling method

25. How do you handle exceptions in Ruby?

Answer:

```
begin
  # code
rescue => e
  puts e.message
end
```

end

26. What are the access modifiers in Ruby?

Answer:

- `public`
- `private`
- `protected`

27. What is the default access modifier?

Answer: `public`

28. What are iterators in Ruby?

Answer: Methods like `.each`, `.map`, `.select` used to loop over collections.

29. What is the difference between `.each` and `.map`?

Answer:

- `.each` returns the original array
- `.map` returns a new array with transformed values

30. What is a Hash in Ruby?

Answer: A key-value pair data structure:

```
{ name: "Ruby", year: 1995 }
```

31. How do you create a range in Ruby?

Answer:

```
(1..5) # includes 5  
(1...5) # excludes 5
```

32. What is method overriding?

Answer: A subclass can provide a new implementation of a method already defined in its superclass.

33. What is method overloading in Ruby?

Answer: Ruby does not support traditional method overloading but you can use default or variable arguments.

34. What is a constructor in Ruby?

Answer: The `initialize` method acts as a constructor.

35. What is `attr_accessor`?

Answer: Automatically creates getter and setter methods for instance variables.

36. What is garbage collection in Ruby?

Answer: Ruby automatically frees up memory by destroying unused objects.

37. What are bang (!) methods?

Answer: Methods that modify the object in place (e.g., `sort!`, `upcase!`).

38. What is the `super` keyword?

Answer: Calls the same method from the parent class.

39. What is the use of the `yield` keyword?

Answer: Invokes the block passed to a method.

40. What is monkey patching?

Answer: Changing or adding to existing classes at runtime.

41. What is Duck Typing in Ruby?

Answer: If an object quacks like a duck, it's treated like one — based on behavior, not class.

42. What are global variables?

Answer: Start with `$`, accessible from anywhere. Example: `$name`.

43. What is a singleton method?

Answer: A method defined only on a single object.

44. What are class variables?

Answer: Variables prefixed with `@@`, shared among all instances of a class.

45. How do you define constants in Ruby?

Answer: Capitalized names, e.g., `PI = 3.14`

46. What is method chaining?

Answer: Calling multiple methods in a single line: `user.name.upcase`

47. What is the spaceship operator (<=>)?

Answer: Returns -1, 0, or 1 for comparisons. Used in sorting.

48. How do you comment in Ruby?

Answer:

- Single-line: `# comment`
- Multi-line: `=begin ... =end`

49. What is Rake in Ruby?

Answer: A build automation tool like Make or Ant for Ruby projects.

50. What is RubyGems?

Answer: A package manager for distributing Ruby programs and libraries.