設問 1 IRBで `100` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"100"`
2. ◉ `100`
3. ○ `Integer`
4. ○ `nil`

設問 2 IRBで `3.14` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"3.14"`
2. ○ `3`
3. ◉ `3.14`
4. ○ `Float`

設問 3 IRBで `"Hello Ruby"` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `Hello Ruby`
2. ○ `String`
3. ◉ `"Hello Ruby"`
4. ○ `nil`

設問 4 IRBで `42.class` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `Fixnum`
2. ◉ `Integer`
3. ○ `Numeric`
4. ○ `42`

設問 5 IRBで `'Ruby'.class` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ Text
2. ○ Character
3. ◉ `String`
4. ○ `'Ruby'`

設問 6 IRBで `"Ruby" + " " + "World"` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"Ruby+ +World"`
2. ◉ `"Ruby World"`
3. ○ `["Ruby", " ", "World"]`
4. ○ エラー

設問 7 IRBで `5 * (2 + 3)` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `13`
2. ○ `"5 * (2 + 3)"`
3. ◉ `25`
4. ○ エラー

設問 8 IRBで `true.class` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `Boolean`
2. ◉ `TrueClass`
3. ○ `true`
4. ○ `Object`

設問 9 IRBで `nil.class` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `Null`
2. ○ `Nothing`
3. ◉ `NilClass`
4. ○ `Object`

設問 10 IRBで `num = 10` と入力した後、続けて `"結果: #{num}"` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"結果: #{num}"`
2. ○ `"結果: num"`
3. ◉ `"結果: 10"`
4. ○ エラー

設問 11 Rubyでシンボルを表す正しい書き方はどれですか？

1. ○ `"user_name"`
2. ○ `user_name`
3. ◉ `:user_name`
4. ○ `@user_name`

設問 12 IRBで `:success` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"success"`
2. ○ `success`
3. ◉ `:success`
4. ○ `Symbol`

設問 13 IRBで `:status.class` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `String`
2. ○ `Identifier`
3. ◉ `Symbol`
4. ○ `:status`

設問 14 IRBで `"apple".to_sym` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `"apple"`
2. ◉ `:apple`
3. ○ `Symbol`
4. ○ エラー

設問 15 IRBで `:key == "key"` と入力してエンターキーを押すと、通常どのように表示されますか？

1. ○ `true`
2. ◉ `false`
3. ○ `nil`
4. ○ エラー

設問 16 配列の作成方法として正しいものはどれでしょうか？

1. ○ Array.new
2. ○ Array()
3. ◉ []
4. ○ {}

設問 17 次のコードの結果は何になりますか？ `arr = [1, 2, 3]; arr[1]`

1. ○ 1
2. ◉ 2
3. ○ 3
4. ○ nil

設問 18 `arr = [1, 2]; arr.push(3)`を実行した後のarrの内容はどれでしょうか？

1. ○ `[1]`
2. ○ `[1, 2]`
3. ○ `[3, 2, 1]`
4. ◉ `[1, 2, 3]`

設問 19 配列から最後の要素を削除するメソッドはどれでしょうか？

1. ○ shift
2. ◉ pop
3. ○ delete
4. ○ last

設問 20 `arr = [1, 2, 3]; arr.length`の戻り値は何になりますか？

1. ◉ 3
2. ○ 2
3. ○ 1
4. ○ nil

設問 21 `arr = (1..5).to_a`を実行した場合、arrの内容はどれでしょうか？

1. ○ `[1]`
2. ○ `[5]`
3. ◉ `[1, 2, 3, 4, 5]`
4. ○ `[2, 3, 4]`

設問 22 `(1...5).to_a`を使用して配列を作成すると、その内容はどうなりますか？

1. ○ `[1, 2, 3, 4, 5]`
2. ◉ `[1, 2, 3, 4]`

3. ○ `[1, 2]`
4. ○ `[5]`

設問 23 次のコードはどのような値を返しますか？ `arr = (10..15).to_a; arr[0]`

1. ◉ 10
2. ○ 11
3. ○ 15
4. ○ nil

設問 24 次のコードを実行した場合、結果はどうなりますか？ `arr = (1..3).to_a; arr.include?(3)`

1. ○ `false`
2. ○ `nil`
3. ○ `null`
4. ◉ `true`

設問 25 `(1..5).to_a.select { |x| x.even? }`を実行した結果はどれでしょうか？

1. ○ `[1, 3, 5]`
2. ○ `[5]`
3. ◉ `[2, 4]`
4. ○ `[1, 2, 3]`

設問 26 次のコードを実行した場合の結果はどれでしょうか？ `h = {a: 1 b: 2}; h[:a]`

1. ○ 1
2. ○ 2
3. ○ nil
4. ◉ エラー

設問 27 次のコードを実行した場合、`h = {a: 1, b: 2}; h.keys`の戻り値はどれでしょうか？

1. ○ `[1, 2]`
2. ○ `[:b]`
3. ◉ `[:a, :b]`
4. ○ `{a, b}`

設問 28 ハッシュに新しいキーと値を追加する正しい方法はどれですか？

1. ○ `h.push(:c => 3)`
2. ◉ `h[:c] = 3`
3. ○ `h.insert(:c, 3)`
4. ○ `h.add(:c, 3)`

設問 29 次のコードを実行すると、ハッシュの内容はどうなりますか？ `h = {a: 1, b: 2}; h.delete(:b)`

1. ○ `{b: 2}`
2. ○ `{}`
3. ○ `{a: 1, b: 2}`
4. ◉ `{a: 1}`

設問 30 `h = {a: 1, b: 2}; h.value?(2)`の結果は何になりますか？

1. ◉ `true`
2. ○ `false`
3. ○ `nil`
4. ○ エラー

設問 31 `Proc.new { |x| x * 2 }`を使うと、どのようなProcオブジェクトが作成されますか？

1. ◉ 引数を2倍するProc
2. ○ 引数を文字列に変換するProc
3. ○ 引数を出力するProc
4. ○ Procオブジェクトが作成されない

設問 32 `proc = Proc.new { |x| x + 1 }; proc.call(5)`を実行すると何が返されますか？

1. ○ 5
2. ◉ 6
3. ○ nil
4. ○ エラー

設問 33 `proc = Proc.new { puts "Hello" }; proc.call`の結果はどれでしょうか？

1. ○ 何も表示されない
2. ○ `"Hello"`
3. ◉ 標準出力に`Hello`が表示される
4. ○ エラーが発生する

設問 34 `proc = Proc.new { |x| x * x }; proc.call(3)`を実行した場合の結果は何になりますか？

1. ○ 6
2. ○ nil
3. ○ 3
4. ◉ 9

設問 35 `lambda = ->(x) { x * 2 }; lambda.call(4)`を実行すると何が返されますか？

1. ◉ 8
2. ○ 4
3. ○ nil
4. ○ エラー

設問 36 次のコードの実行結果は何ですか？ `if 5 > 3 then "Yes" else "No" end`

1. ◉ Yes
2. ○ No
3. ○ nil
4. ○ エラー

設問 37 次のコードはどのように書き換えられますか？ `"Yes" if 10 > 2`

1. ○ `if 10 > 2 "Yes" end`
2. ○ `"Yes" unless 10 > 2`
3. ◉ `if 10 > 2 then "Yes" end`
4. ○ `"Yes" while 10 > 2`

設問 38 次のコードを実行すると何が出力されますか？ `x = 0; if x == 0; puts "Zero"; else; puts "Not Zero"; end`

1. ○ Not Zero
2. ○ エラー
3. ○ 空の文字列
4. ◉ Zero

設問 39 `unless`文を使った場合、`unless false then "A" else "B" end`の結果はどうなりますか？

1. ○ B
2. ◉ A
3. ○ エラー
4. ○ nil

設問 40 次のコードは正しいですか？ `if 1 > 0 then puts "Positive" end`

1. ◉ はい
2. ○ いいえ
3. ○ エラーが発生する
4. ○ 不明

設問 41 次のコードを実行した場合の結果は何ですか？ `case 2; when 1 then "One"; when 2 then "Two"; else "Other"; end`

1. ◉ Two
2. ○ One
3. ○ エラー
4. Other

設問 42 次のコードを実行すると、何が出力されますか？ `value = "b"; case value; when "a" then 1; when "b" then 2; else 3; end`

1. ○ 1
2. ◉ 2
3. ○ 3
4. ○ エラー

設問 43 次のコードの動作はどのようになりますか？ `case 10; when 1..5 then "Low"; when 6..10 then "Medium"; else "High"; end`

1. ○ Low
2. ○ High
3. ◉ Medium

4. ○ エラー

設問 44 次のコードを実行するとどうなりますか？ `case nil; when false then "False"; when nil then "Nil"; else "Other"; end`

1. ○ False
2. ○ エラー
3. ◉ Nil
4. Other

設問 45 `case "apple"; when /app/ then "Fruit"; else "Other"; end`の結果は何ですか？

1. ◉ Fruit
2. ○ nil
3. ○ エラー
4. Other

設問 46 `while`文で正しい構文はどれですか？

1. ○ `while true puts "loop" end`
2. ◉ `while true do puts "loop" end`
3. ○ `while { true } puts "loop" end`
4. ○ `while [true] puts "loop" end`

設問 47 次のコードを実行すると`sum`は最終的にいくつになりますか？ `sum = 0; i = 1; while i <= 3 do sum += i; i += 1; end`

1. ○ 3
2. ○ 4
3. ◉ 6
4. ○ エラー

設問 48 `break`を使うとどうなりますか？ `i = 0; while i < 5 do break if i == 3; puts i; i += 1; end`

1. ○ 0 1 2 3 4
2. ◉ 0 1 2
3. ○ 無限ループ
4. ○ エラー

設問 49 `begin〜end`を用いた`while`ループの正しい使い方はどれですか？

1. ◉ `begin puts "Hello" end while true`
2. ○ `begin while true puts "Hello" end`
3. ○ `begin puts "Hello" while end true`
4. ○ `begin puts "Hello" end until true`

設問 50 `"hello".upcase`を実行すると何が返されますか？

1. ◉ HELLO

2. ○ hello
3. ○ エラー
4. ○ nil

設問 51 次のコードを実行した場合、戻り値は何ですか？ `[1, 2, 3].size`

1. ○ 1
2. ○ 2
3. ◉ 3
4. ○ nil

設問 52 次のコードの結果として正しいものはどれですか？ `obj = "Ruby"; obj.respond_to?(:upcase)`

1. ○ false
2. ◉ true
3. ○ エラー
4. ○ nil

設問 53 `5.times { print "Hi" }`を実行すると何が表示されますか？

1. ○ Hi
2. ○ HiHi
3. ○ HiHiHi
4. ◉ HiHiHiHiHi

設問 54 `[1, 2, 3].include?(2)`を実行するとどのような結果になりますか？

1. ◉ true
2. ○ false
3. ○ nil
4. ○ エラー

設問 55 次のコードを実行した場合、出力される内容は何ですか？ `name = "Alice"; puts "Hello, #{name}!"`

1. ◉ `Hello, Alice!`
2. ○ `#{name}!`
3. ○ `Alice!`
4. ○ エラー

設問 56 `"The sum is #{5 + 3}"`を実行するとどうなりますか？

1. ○ The sum is 5 + 3
2. ◉ The sum is 8
3. ○ 8
4. ○ エラー

設問 57 式展開を利用する正しい方法はどれですか？

1. ○ `'Hello, #{name}'`
2. ○ `"Hello, name"`

3. ◉ `"Hello, #{name}"`
4. ○ `'Hello, name'`

設問 58 次のコードを実行するとどうなりますか？ `puts "#{10 * 2} is twenty"`

1. ◉ 20 is twenty
2. ○ is twenty
3. ○ エラー
4. ○ 10 * 2 is twenty

設問 59 `"#{'Ruby'.upcase}"`の結果は何ですか？

1. ◉ RUBY
2. ○ Ruby
3. ○ エラー
4. ○ nil

設問 60 次のコードを実行した場合の結果は何ですか？ `[1, 2, 3].inspect`

1. ◉ `"[1, 2, 3]"`
2. ○ `"1, 2, 3"`
3. ○ エラー
4. ○ `[1, 2, 3]`

設問 61 `hash = {a: 1, b: 2}; hash.inspect`の実行結果はどれですか？

1. ○ `"[a: 1, b: 2]"`
2. ○ `"{a=>1, b=>2}"`
3. ◉ `"{:a=>1, :b=>2}"`
4. ○ `"a: 1, b: 2"`

設問 62 `nil.inspect`を実行すると、どのような結果が得られますか？

1. ○ 空の文字列
2. ◉ `"nil"`
3. ○ エラー
4. ○ nil

設問 63 次のコードの出力結果は何ですか？ `puts :symbol.inspect`

1. ◉ `":symbol"`
2. ○ `"symbol"`
3. ○ エラー
4. ○ 空の文字列

設問 64 `class MyClass; end; obj = MyClass.new; obj.inspect`を実行すると何が返されますか？ただし、「******」の箇所は実行環境によって変わります。

1. ○ `#<Object>`
2. ○ `#<Class>`

3. ○ エラー
4. ◉ #<MyClass:0x******>

設問 65 puts "Hello, World!"を実行した場合、何が出力されますか？

1. ◉ Hello, World!
2. ○ Hello, World! (改行なし)
3. ○ エラー
4. ○ 何も出力されない

設問 66 putsとprintの違いは何ですか？

1. ○ どちらも改行を追加しない
2. ◉ putsは改行を追加するが、printは改行を追加しない
3. ○ putsは改行を追加しないが、printは改行を追加する
4. ○ どちらも改行を追加する

設問 67 次のコードを実行すると何が出力されますか？ puts 1, 2, 3

1. ○ 123
2. ◉ 1
2
3
3. ○ エラー
4. ○ 何も出力されない

設問 68 puts nilを実行した場合、何が出力されますか？

1. ○ nil
2. ○ NIL
3. ○ 空文字列 (改行なし)
4. ◉ 空行

設問 69 puts [1, 2, 3]を実行した場合、何が出力されますか？

1. ◉ 1
2
3
2. ○ [1, 2, 3]
3. ○ エラー
4. ○ 1, 2, 3

設問 70 次のコードを実行すると何が出力されますか？ def greet(*names); puts "Hello, #{names.join(', ')}!"; end; greet("Alice", "Bob")

1. ◉ Hello, Alice, Bob!
2. ○ Hello, AliceBob!
3. ○ エラー
4. ○ nil

設問 71 *argsの内容を確認するために正しいコードはどれですか？

1. ○ args.to_s
2. ○ *args.inspect
3. ◉ args.inspect
4. ○ print args

設問 72 次のコードを実行すると何が返されますか？ def add(*numbers); numbers.sum; end; add(1, 2, 3)

1. ○ 1
2. ◉ 6
3. ○ 3
4. ○ エラー

設問 73 次のコードにおいて、putsを使用した結果は何ですか？ def show(*items); items.each { |item| puts item }; end; show("A", "B", "C")

1. ◉ A
B
C
2. ○ A
3. ○ B
4. ○ C

設問 74 次のコードにおいて*argsを使用した場合はどうなりますか？ def describe(*args); puts "You passed #{args.length} arguments."; end; describe(1, 2, 3, 4)

1. ○ You passed 1 argument.
2. ○ You passed 2 argument.
3. ○ You passed 3 argument.
4. ◉ You passed 4 argument.

設問 75 次のコードを実行すると何が表示されますか？ def example; yield; end; example { puts "Hello, Ruby!" }

1. ◉ Hello, Ruby!
2. ○ エラー
3. ○ ブロックが渡されていません
4. ○ nil

設問 76 def greet; yield("Alice"); end; greet { |name| puts "Hello, #{name}!" }を実行すると、何が出力されますか？

1. ○ Alice!
2. ◉ "Hello, Alice!"
3. ○ エラー
4. ○ ブロックがありません

設問 77 次のコードはどのように動作しますか？ def calc; yield(5, 3); end; calc { |a, b| puts a + b }

1. ○ 5

2. ○ 3
3. ○ エラー
4. ◉ 8

設問 78 次のコードを実行すると、エラーが発生する理由は何ですか？ `def example; yield; end; example`

1. ○ メソッドの定義が不正
2. ◉ ブロックが渡されていない
3. ○ エラーは発生しない
4. ○ nil

設問 79 `def safe_yield; yield if block_given?; end; safe_yield { puts "Block executed!" }`の結果は何ですか？

1. ◉ Block executed!
2. ○ エラー
3. ○ 何も表示されない
4. ○ nil

設問 80 次のコードを実行した場合、返される結果は何ですか？ `[1, 2, 3].map { |x| x * 2 }`

1. ◉ `[2, 4, 6]`
2. ○ `[1, 2, 3]`
3. ○ エラー
4. ○ nil

設問 81 `[1, 2, 3].select { |x| x.odd? }`を実行した場合、結果は何になりますか？

1. ○ `[1]`
2. ◉ `[1, 3]`
3. ○ `[2, 3]`
4. ○ エラー

設問 82 `[1, 2, 3, 4].reduce(0) { |sum, x| sum + x }`を実行した結果はどうなりますか？

1. ○ `4`
2. ◉ `10`
3. ○ `1`
4. ○ エラー

設問 83 `[1, 2, 3].any? { |x| x > 2 }`を実行すると結果はどうなりますか？

1. ○ false
2. ◉ true
3. ○ nil
4. ○ エラー

設問 84 `[1, 2, 3].all? { |x| x > 0 }`を実行すると結果はどうなりますか？

1. ◉ true

2. ○ false
3. ○ nil
4. ○ エラー

設問 85 `Person = Struct.new(:name, :age); person = Person.new("Alice", 25); person.name`を実行すると何が返されますか？

1. ◉ Alice
2. ○ 25
3. ○ エラー
4. ○ nil

設問 86 `Person = Struct.new(:name, :age); person = Person.new("Alice", 25); person[:age]`の実行結果はどれですか？

1. ○ Alice
2. ○ nil
3. ◉ 25
4. ○ エラー

設問 87 次のコードを実行するとどうなりますか？ `Person = Struct.new(:name, :age); person = Person.new("Alice", 25); person.each_pair { |key, value| puts "#{key}: #{value}" }`

1. ◉
```
name: Alice
age: 25
```
2. ○ `Alice`
3. ○ `25`
4. ○ エラー

設問 88 次のコードを実行すると何が返されますか？ `Person = Struct.new(:name, :age); Person.members`

1. ○ `[:name]`
2. ◉ `[:name, :age]`
3. ○ `[:age]`
4. ○ エラー

設問 89 `Person = Struct.new(:name, :age); person = Person.new("Alice", 25); person.age = 30; person.age`の結果は何ですか？

1. ◉ `30`
2. ○ `25`
3. ○ エラー
4. ○ nil

設問 90 次のコードを実行すると、どのような結果になりますか？ `class String; def shout; self.upcase + "!"; end; end; "hello".shout`

1. ◉ HELLO!
2. ○ hello!
3. ○ エラー

4. ○ nil

設問 91 `class Integer; def double; self * 2; end; end; 5.double`の結果は何ですか？

1. ○ 5
2. ○ 25
3. ◉ 10
4. ○ エラー

設問 92 モンキーパッチを適用する際に注意すべき点は何ですか？

1. ○ コードの再利用性が高まる
2. ◉ 既存のコードに影響を与える可能性がある
3. ○ コードの実行速度が向上する
4. ○ 問題は特にない

設問 93 `class Array; def second; self[1]; end; end; [1, 2, 3].second`の結果は何ですか？

1. ○ 1
2. ○ 3
3. ○ nil
4. ◉ 2

設問 94 `class Hash; def stringify_keys; self.transform_keys(&:to_s); end; end; {a: 1, b: 2}.stringify_keys`を実行すると何が返されますか？

1. ◉ `{"a" => 1, "b" => 2}`
2. ○ `{:a => 1, :b => 2}`
3. ○ エラー
4. ○ nil

設問 95 定数を定義する正しい方法はどれですか？

1. ◉ `MY_CONSTANT = 10`
2. ○ `my_constant = 10`
3. ○ `My_constant = 10`
4. ○ `myconstant = 10`

設問 96 次のコードを実行した場合、何が出力されますか？ `CONSTANT = 100; CONSTANT = 200; puts CONSTANT`

1. ○ 100
2. ◉ 200
3. ○ エラー
4. ○ 何も出力されない

設問 97 次のコードを実行した場合、何が返されますか？ `module MyModule; MY_CONSTANT = "Hello"; end; puts MyModule::MY_CONSTANT`

1. ○ MY_CONSTANT
2. ◉ Hello

3. ○ エラー
4. ○ nil

設問 98 次のコードを実行するとどうなりますか？ `CONST = 10; Object.send(:remove_const, :CONST); CONST`

1. ◉ エラー
2. ○ `10`
3. ○ nil
4. ○ `CONST`

設問 99 定数を削除する正しい方法はどれですか？

1. ○ `delete_const(:CONST)`
2. ◉ `remove_const(:CONST)`
3. ○ `CONST = nil`
4. ○ `undef(:CONST)`

設問 100 次のコードを実行した場合、何が出力されますか？ `module MyModule; MY_CONST = "Hello"; remove_const :MY_CONST; end; puts MyModule.constants.include?(:MY_CONST)`

1. ○ true
2. ◉ false
3. ○ エラー
4. ○ nil

終了

しおりをつけて閉じる