

mruby ver3.0 bytecode sample.

mruby code

```
puts "ABC", 3.14159265358979

def func1
  puts "This is func1"
end

class Class1
  def method1
    puts "This is Class1"
  end
  def method2
  end
end

begin
  raise
rescue
  puts
end
```

RITE binary header (File header)

```
00000000: 52 49 54 45 30 32 30 30 00 00 01 a8 4d 41 54 5a RITE0200...MATZ
            ident      major/minor size      compiler name
00000010: 30 30 30 30
            compiler ver      0000
```

RITE Section header

```
            49 52 45 50 00 00 01 77      IREP...0
            ident      size
IREP record
            30 33 30 30      0300
            rite version
```

IREP record SEQ 1

```
00000020: 00 00 00 a3 00 01 00 05 00 02 00 01 00 44 14 01 .....D..
            record size #local #reg #child #clen ilen
00000030: 51 02 00 02 ...(ilen bytes)
      Catch handler 1 (#clen)
00000070: 00 00 00 00 1e 00 00 00 24 00 00 00 27 .....$...'.
            type begin      end      target

      POOL 1
00000070: 00
00000080: 02 00 00 03 41 42 43 00 05 11 2d 44 54 fb 21 09 ....ABC...-DT.!. 長さは、TT により変化
            #pool TT len      ("ABC")      TT (double)
00000090: 40
            (double)      @

      SYMS 1
00000090: 00 05 00 04 70 75 74 73 00 00 05 66 75 6e 63 ....puts...func
            #slen len      data1\0      len      data2\0
```

IREP record SEQ 2

```
000000c0: 00 00 00 39 00 02 00 05 00 00 00 00 00 .....9.....
            record size #local #reg #child #clen ilen
000000d0: 0f 35 00 00 00 14 02 51 03 00 2f 02 00 01 39 02 .5....Q../...9.
      POOL 2
000000e0: 01 00 00 0d 54 68 69 73 20 69 73 20 66 75 6e .....This is fun
      SYMS 2
000000f0: 00 01 00 04 70 75 74 73 00 .....puts.
```

```

000000f0:                                00 00 00 3b                ....;
00000100: 00 01 00 03 00 02 00 00 00 15 68 01 5b 02 00 64  ....h.[..d
        #local #reg #child #clen  ilen
        POOL 3
00000110:                                00                .
00000120: 00
        #pool
        SYMS 3
00000120: 00 02 00 07 6d 65 74 68 6f 64 31 00 00 07 6d  ....method1...m
        #slen len      data1\0                len

```

```

00000130:          00 00 00 3a 00 02 00 05 00          .....
          record size #local #reg #child

00000140: 00 00 00 00 0f 35 00 00 00 14 02 51 03 00 2f 02  ....5....Q../.
          #clen ilen

POOL 4
-----
00000150:          00 01 00 00 0e 54 68 69 73 20 69 73  ..9.....This is
          #pool TT len      data

SYMS 4
-----
00000160:          00 01 00 04 70 75 74 73          ....puts
          #slen len      data

```

```

00000170:  00 00 00 1a 00 02 00 03 00 00 00 00 00 08 35  .....5
               record size #local #reg #child #clen ilen

    POOL 5
00000180:  00 00
               #pool

    SYMS 5
00000180:  00 00
               #slen

```

```

00000180:          4c 56 41 52 00 .....9....LVAR.
          ident      size
00000190: 00 00 15 00 00 00 01 00 01 26 00 00 00 00 00 .....&.....
          size

```

```
000001a0: 45 4e 44 00 00 00 00 08          END.....
               ident      size
```