

CS 入門 4b クラス 第2回資料 (2012.10.11)

担当: 伊知地 宏

1 BMI を関数で計算

```
def bmi(height, weight)          # 関数名 bmi, 関数引数 height, weight
    weight / ( height / 100.0 ) ** 2 # bmi の計算式
end
```

2 2次方程式を解く (条件分岐)

2次方程式 (quadratic equation) $ax^2+bx+c=0$ を解く関数

```
def det(a,b,c)                  # 判別式計算関数
    b ** 2 - 4 * a * c          # 判別式の計算
end                             # 関数定義の終わり
```

```
def q_eq_solution1(a,b,c)      # 方程式の解 1 を求める関数
    if det(a,b,c) >= 0         # 判別式 det(a,b,c) が 0 以上か?
        (-b + sqrt(det(a,b,c))) / (2 * a) # det(a,b,c) が 0 以上なら計算
    end                       # if 文の終わり
end                           # 関数定義の終わり
```

```
def q_eq_solution2(a,b,c)      # 方程式の解 2 を求める関数 q_eq_solution2
    if det(a,b,c) >= 0
        (-b - sqrt(det(a,b,c))) / (2 * a)
    end
end
```

```
def q_eq(a,b,c)                # 方程式の解を求める関数
    [q_eq_solution1(a,b,c), q_eq_solution2(a,b,c)] # 2 つの解を計算
end
```

```
def q_eq2(a,b,c)               # 方程式の解を求める関数その 2
    if det(a,b,c) > 0          # 判別式 det(a,b,c) が 0 以上のとき
        [q_eq_solution1(a,b,c), q_eq_solution2(a,b,c)] # 2 つの解を計算
    elsif det(a,b,c) == 0      # 判別式 det(a,b,c) が 0 と等しいとき
        [q_eq_solution1(a,b,c)] # 重解を計算
```

```

else                                # それ以外のとき
  nil                               # 解がない
end
end

```

3 最大公約数の計算 (繰り返し)

最大公約数を単純な方法で求める関数

```

def gcd_s(x,y)                      # 関数 gcd_s の定義
  if x > y                           # x, y のうち大きくない方を n に設定
    n = y
  else
    n = x
  end
  k = 1
  while k <= n                       # k が n 以下なら繰り返す:
    if x % k == 0 && y % k == 0      # x と y が k で割り切れるか判定
      ans = k                       # k が x と y の公約数のときに ans に設定
    end
    k = k + 1                       # k の値を 1 増やす
  end
  ans                               # 答えとして ans の値を返す
end

```

4 フィボナッチ数の計算 (配列)

```

def fib(n)
  fib = Array.new(n)  # 長さ n の配列 fib を作る
  fib[0] = 1          # 配列 fib の第 0 要素 (1 番目) に 1 を代入
  fib[1] = 1          # 配列 fib の第 1 要素 (2 番目) に 1 を代入
  for i in 2..n-1     # i を 2 から n-1 まで順に動かして
    fib[i] = fib[i-1] + fib[i-2]
    # 配列 fib の第 i-1 要素と第 i-2 要素を足して第 i 要素に代入
  end
  fib                 # 配列 fib を結果として返す
end

```

[実行例]

```

irb(main):043:0> fib(50)
=> [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584,
4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229,
832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817,

```

```
39088169, 63245986, 102334155, 165580141, 267914296, 433494437, 701408733,  
1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025]
```

5 文字列処理

```
def add_string(str1,str2)  
  s = str1 + str2      # 文字列 str1 と文字列 str2 をつなげて s に代入  
end
```

[実行例]

```
irb(main):033:0> s1 = "Computer "  
=> "Computer "  
irb(main):034:0> s2 = "Science"  
=> "Science "  
irb(main):035:0> add_string(s1,s2)  
=> "Computer Science"
```

次回の予告

次回の授業は情報ネットワーク演習室の第2演習室で授業を行います。今週学んだプログラムの文法を、実際にプログラムを書くことで理解してもらいます。