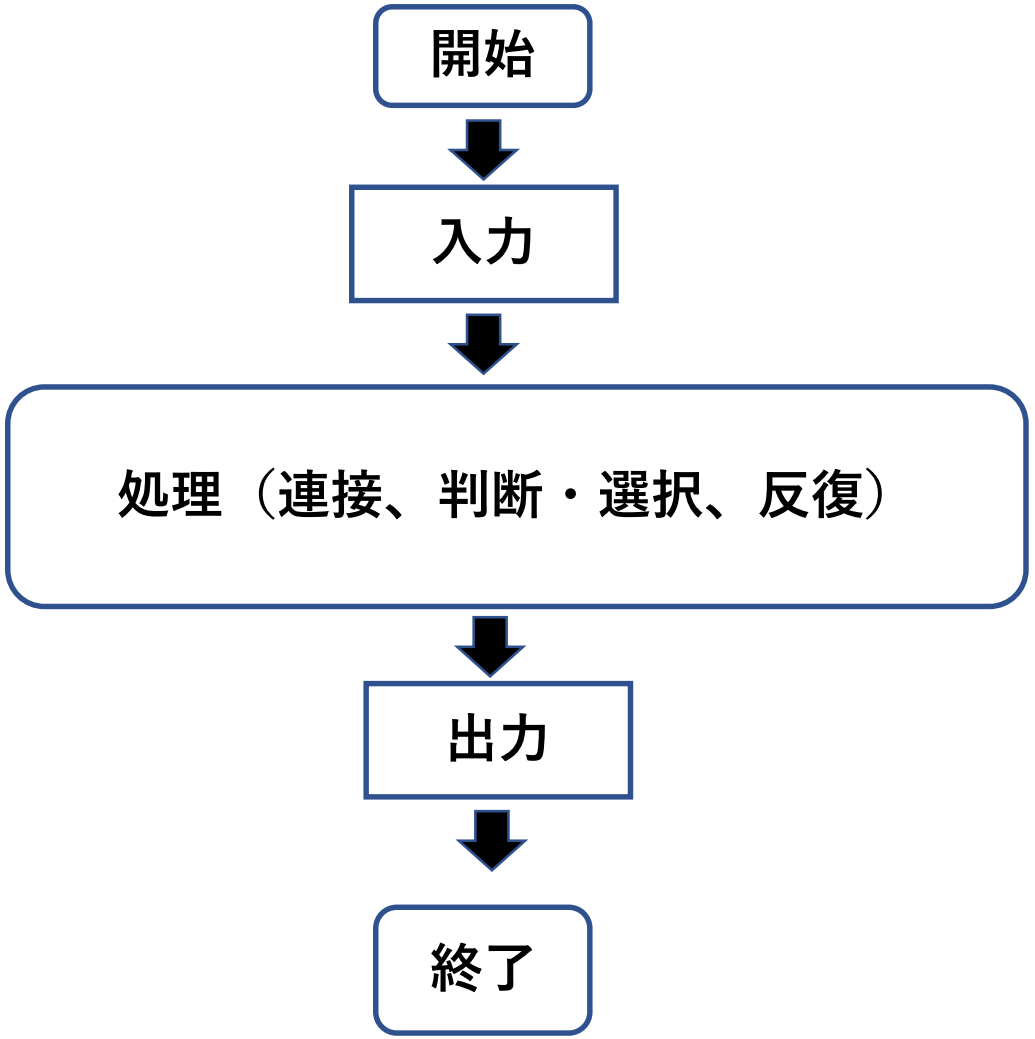
Rプログラミング（関数の作成）

フローチャート（流れ図）

プログラミングの基本



Rにおける関数（ユーザ定義関数）


```
> example01 <- function(x){
 +    y <- x * 2
 +    return(y)
 + }
```

関数を定義する：y=2x

- “＋”は前行の継続で改行で自動入力される
- 返り値（出力値）を出力する
- ｛　｝は上下で対応させる

```
 > example01(3)
[1] 6
```

関数を実行する

```
> example02 <- function(x){
+   y <- 10 * (x - mean(x)) / sd(x) + 50
+   return(y)
+ }
```

偏差値を計算する関数の定義

```
> tensuu <- c(83, 63, 72, 33, 94)
> example02(tensuu)
[1] 56.02185 47.41921 51.29040 34.51523 60.75331
```

5個のデータ

5個のデータの偏差値

Rにおける関数（ユーザ定義関数：引数が複数・なしの場合）


```
> example03 <- function(x,y){                    # 引数が２つの関数：z=f(x,y)
+   z <- x^y
+   return(z)
+ }

> example03(2, 10)                               # 返り値はベクトル
[1] 1024


> example04 <- function(){                       #  引数がない関数
+   x <- "Hello World"
+   return(x)
+ }

> example04()
[1] "Hello World"
```

Rにおける関数（ユーザ定義関数）：返り値が複数の場合


```
> example06 <- function(x,y){
+    z1 <- x %/% y                    #   xをyで割った商
+    z2 <- x %% y                     #   xをyで割った余り
+    z <- c(z1, z2)                   #   zをベクトルで定義
+    return(z)
+ }

> example06(17, 6)
[1] 2 5                              # 商が2　余りが5　しかしわかりにくい！
```

Rにおける関数（ユーザ定義関数）：返り値の表示）


```
> example06_2 <- function(x,y){
+ z1 <- x%/%y
+ z2 <- x%%y
+ z <- c(z1,z2)
+ print(z1)
+ print(z2)
+ }

> example06_2(17,6)
[1] 2
[1] 5
```

print（　）：変数の値を表示

```
>example06_4 <- function(x,y){
Z1 <- x%/%y
Z2 <- x%%y
Z <- c(z1,z2)
cat("Q=",z1,"R=",z2,"¥n")
}

Example06_4(17,6)
Q=2  R=5
```

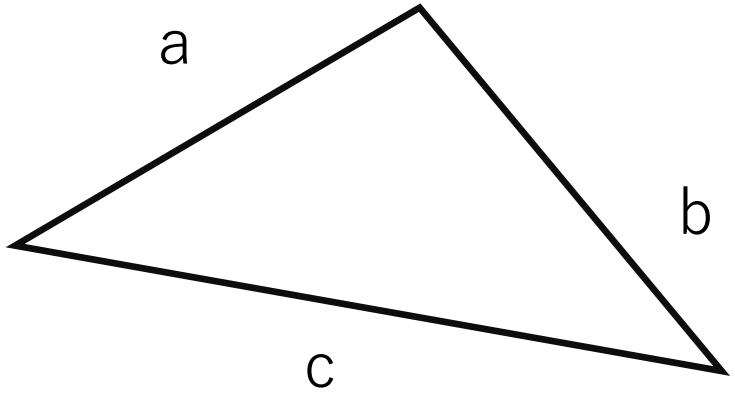
cat（　）：変数の値や文字列を表示。
改行マークをいれる“￥n”

Rにおける関数（ユーザ定義関数）

演習１
3辺の値(a, b, c)を入力すると三角形の面積Sを求めることができる関数を作りなさい。（hint :ヘロンの公式）
ヘロンの公式：

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

ただし$(2s = a + b + c)$




```
heron01 <- function(a,b,c){

      <・・・・・・・・>

  return( )
}
```

1：heron01(2,2,2)を実行してみよう。
2：heron01(1,2,5)を実行してみよう。

Rにおける関数（ユーザ定義関数）

解答例：

```
heron01 <- function(a,b,c){
       s = (a+b+c)/2
       v = sqrt(s*(s-a)*(s-b)*(s-c))
  return(v)
}
```

ヘロンの公式：

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

ただし$(2s = a + b + c)$

1：heron01(2,2,2)を実行してみよう。（正三角形）
2：heron01(1,2,5)を実行してみよう。（エラーメッセージ）

Rにおける関数（ユーザ定義関数）

演習２

各自、数学で学習した内容を参考に、便利な関数をプログラミングしなさい。