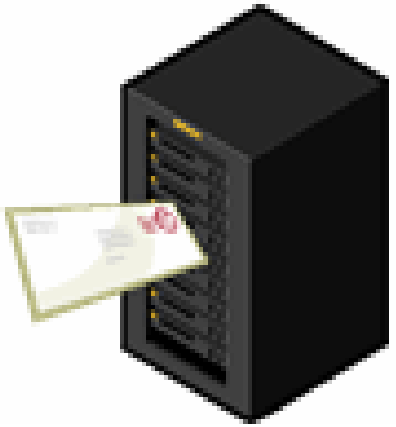
Rで学ぶプログラミングの基礎の基礎

(6) 条件分岐

プログラミングについて

- プログラミングとは
  - 人間がコンピュータに命令をすること
  - R の場合は「ユーザーが R のコマンドをひとつひとつ記述する」作業のこと ⇒ 関数定義！
- R におけるプログラミングのための道具は・・・
  - 変数，ベクトル，関数定義
  - 条件分岐（if）
  - くり返し（for）



本日のメニュー

1. 論理型ベクトルの復習
2. if 文と else 文
3. 複数の条件分岐：if + else if + else 文

論理型ベクトルの復習

- 値同士の大小比較を行った結果は TRUE（真）または FALSE（偽）を要素とする論理型ベクトルとなる

```
> 1 < 2      # 1 は 2 よりも小さい？
[1] TRUE
> 3 < 2      # 3 は 2 よりも小さい？
[1] FALSE
```

- 大小比較を行うための「比較演算子」を紹介する

記号	==	!=	>=	>	<=	<
意味	等しい	≠	≧	＞	≦	＜

論理型ベクトルの復習

- 複数の比較結果を同時に考慮するための「論理演算子」を紹介する

記号	!	&&	||
意味	否定	かつ	または

- 「比較演算子」と「論理演算子」の使用例

```
> x <- 2
> x >= 3              # 変数 x は 3 以上？
[1] FALSE
> x == 2              # 変数 x は 2 と等しい？
[1] TRUE
> 1 < x && x < 3      # 変数 x は 1 より大きく 3 より小さい？
[1] TRUE
```

演習問題①

1. 変数 x に 2 を，変数 y に 3 を代入して下さい．
2. 以下の条件式を作成し，TRUE（真）か FALSE（偽）かを判定して下さい．
   a. x は 1 よりも小さい？
   b. y は 3 未満？
   c. x は y 以下？
3. 以下の条件式を作成し，TRUE（真）か FALSE（偽）かを判定して下さい．
   a. 「x は 1 よりも小さい？」かつ「x は y 以下？」
   b. 「x は 1 よりも小さい？」または「x は y 以下？」
   c. 「x は y 以下？」でない

本日のメニュー

1. 論理型ベクトルの復習
2. **if 文と else 文**
3. 複数の条件分岐：if + else if + else 文

条件分岐〔 if 〕

- ある「条件」に合致する場合は処理を実行 ⇒ if を用いる

```
if（条件）{ 条件に合致する場合に実行する処理 }
```

- 「条件」は「比較演算子」を使って判定する

```
> if（x < 0）x <- -x    # x が 0 未満の場合はプラスに
```

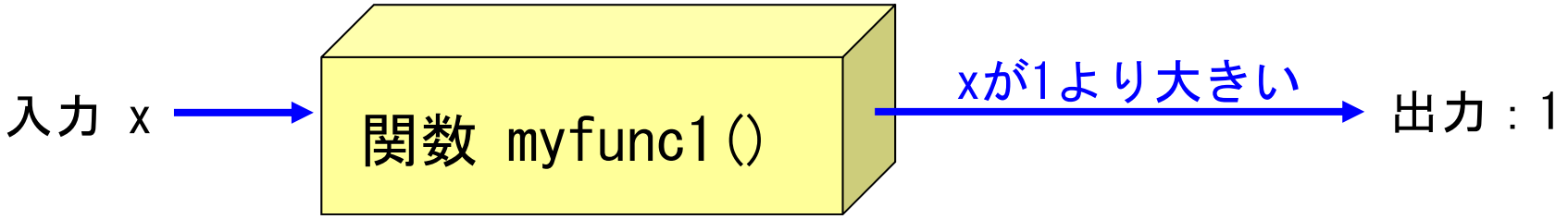
比較演算子

記号	==	!=	>=	>	<=	<
意味	等しい	≠	≧	＞	≦	＜

〔if〕の使用例

- 引数 x が 1 より大きい場合は 1 を出力する

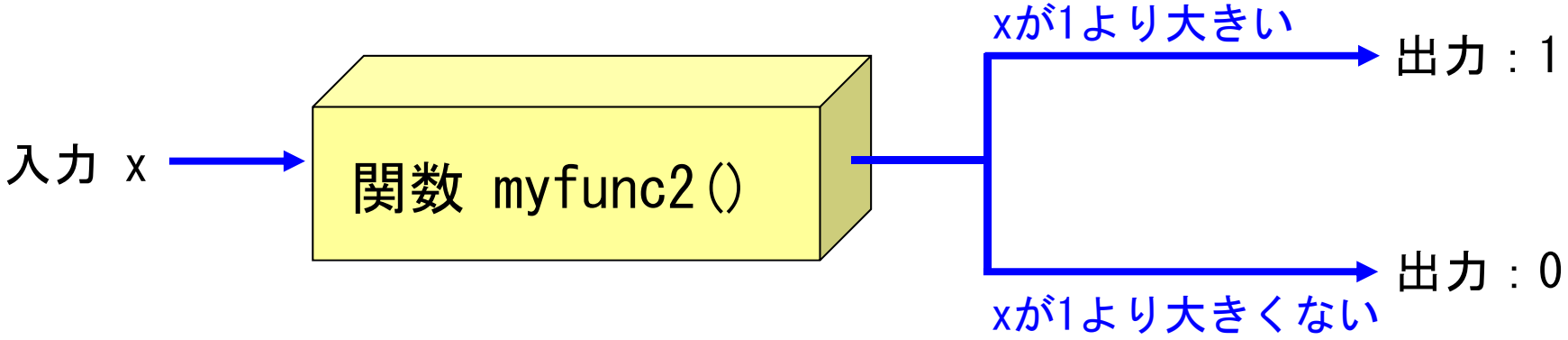
```
> myfunc1 <- function(x) {
+   if (x > 1) return(1)    # 引数 x が 1 よりも大きい場合は 1
+ }
> myfunc1(2)
[1] 1
> myfunc1(0)                # 何も起こらない
```



〔 if + else 〕の使用例

- 引数 x が 1 より大きい場合は 1 を出力し，
  引数 x が 1 より大きくない場合は 0 を出力する

```
> myfunc2 <- function(x) {
+   if (x > 1) return(1)   # 引数 x が 1 よりも大きい場合は 1
+   else       return(0)   # そうでない場合は 0
+ }
> myfunc2(0)
[1] 0
```



演習問題②

1. 引数 x が 1 と等しい場合は 1 を出力し，
   引数 x が 1 と等しくない場合は何もしないような関数 myone()
   を定義して下さい
2. 引数 x が 1 と等しい場合は 1 を出力し，
   引数 x が 1 と等しくない場合は 0 を出力するような関数 myindex()
   を定義して下さい
3. 引数 x と引数 y の比較を行い，x と y の値が等しい場合は x の値を
   返し，等しくない場合は x と y の値の大きい方を返す関数 mymax()
   を定義して下さい

本日のメニュー

1. 論理型ベクトルの復習
2. if 文と else 文
3. 複数の条件分岐：if + else if + else 文

〔if + else if + else〕の使用例

- 引数 x が 1 より大きい場合は 1 を出力し,
  引数 x が 0 ～ 1 の場合は 0 を出力し,
  引数 x が 0 より小さい場合は -1 を出力する

```
> myfunc3 <- function(x) {
+   if      (x >  1) return( 1)   # 引数 x が 1 よりも大きい
+   else if (x >= 0) return( 0)   # 引数 x が 0 ～ 1
+   else             return(-1)   # 上 2 つのどれでもない
+ }
> myfunc3(2)
[1] 1
> myfunc3(0)
[1] 0
> myfunc3(-1)
[1] -1
```

条件分岐の入れ子

- if 文や else 文の中でさらに条件分岐（入れ子）を行うことも出来る

```
> myfunc4 <- function (x) {
+   if (x > 1) {
+                 y <-  1
+   }
+   else {
+     if (x >= 0) y <-  0
+     else        y <- -1
+   }
+   return(y)
+ }
> myfunc4(2)
[1] 1
> myfunc4(0)
[1] 0
> myfunc4(-1)
[1] -1
```

演習問題②

1. 少しネタが古いですが，ユーザーが正の整数を 1 つ指定すると，
   「 3 の倍数」のときは文字「X」を，
   「 3 の倍数でないが 5 の倍数」のときは文字「Y」を，
   「 3 の倍数」でも「 5 の倍数」でもないときはそのままの値を
   返すような関数 nabe1() を作って下さい
2. 1. の仕様を変更します．ユーザーが正の整数を 1 つ指定すると，
   「 3 の倍数」のときは文字「X」を，
   「 5 の倍数」のときは文字「Y」を，
   「 3 の倍数」かつ「 5 の倍数」のときは文字「Z」を，
   上記以外の場合はそのままの値を
   返すような関数 nabe2() を作って下さい

本日のメニュー

1. 論理型ベクトルの復習
2. if 文と else 文
3. 複数の条件分岐：if + else if + else 文

おまけ

- これまで関数の最後は必ず関数 return() で何らかの値を返していたが，必ずしも関数 return() を実行する必要はない
  → 例えば，関数 print() と関数 cat() で何らかの結果を返しても良い

```
> myprint <- function(x) {
+   y <- x
+   print(y)  # 変数 y の値を表示する
+ }
> myprint(777)
[1] 777
> mycat1 <- function(x) {
+   y <- x
+   cat(y)    # 変数 y の値を表示するが，最後に改行されない
+ }
> mycat1(777)
777> |        ← 結果の文字と同じ行から次の命令が始まる・・・
```

おまけ

- 関数 cat() で "¥n"（改行文字）を出力することが出来る

```
> mycat2 <- function(x) {
+   y <- x
+   cat(y)
+   cat("¥n")
+ }
> mycat2(777)
777
>
> mycat3 <- function(x) {
+   y <- x
+   cat("入力された値は ")
+   cat(y)
+   cat(" です. ¥n")
+ }
> mycat3(777)
入力された値は 777 です.
```

おまけ

- 関数 readline() を用いると，ユーザーにデータ入力をしてもらうことが出来る

```
> mytest <- function() {
+   answer <- readline("1+1 ? ")
+   if (answer == "2") cat("Exactly!¥n")
+   else               cat("Wrong...¥n")
+ }
> mytest()
1+1 ? 3
Wrong...
> mytest()
1+1 ? 2
Exactly!
```

"1+1 ?" と表示され待機状態になるので
3 と入力してみる

"1+1 ?" と表示され待機状態になるので
2 と入力してみる

このスライドの内容が書籍になっています

- 舟尾 暢男 『 R で学ぶプログラミングの基礎の基礎（カットシステム）』

  「R と RStudio のインストール」「R の基礎」「変数とベクトル」
  「種々のベクトル」「関数の作成」「条件分岐」「くり返し」
  「複雑なくり返し」「グラフ作成の概要」「問題集」「簡単なゲーム作成」

※ 本資料の演習問題の解答も載っています

Rで学ぶプログラミングの基礎の基礎

終