

條件執行：if 指令.

- Textbook reading: Chapter 7.
- 條件執行：if 指令
 - “當條件A 為TRUE 時, 執行命令C”的語法為

```
if ( A ) { C }
```
 - “當條件A 為TRUE 時執行命令C, 否則執行命令D”的語法為

```
if ( A ) { C } else { D }
```
- A simple example.

```
x <- 0.5
y <- 0
if ( x > 3 ) { y <- 2 }
y
## (x>3) is FALSE, so y is 0
if ( x > 3 ) { y <- 2 } else { y <- 10 }
y
## (x>3) is FALSE, so y is 10
```

- Example 1. Write an R function called “add” that asks the user to enter from screen two numbers and enter the sum of the two numbers. If the user enters the correct sum, print “Your answer is correct. Good job!”. If the user fails to enter the right sum, print a correcting message showing the correct answer on the screen.

```
add <- function(){
  x <- readline("Please enter a number: \n")
  y <- readline("Please enter another number: \n")
  question <- paste("Please enter the sum of ", x, " and ", y, ": \n", sep="")
  z <- readline(question)
  x <- as.numeric(x)
  y <- as.numeric(y)
  z <- as.numeric(z)
  if (z==(x+y)) { cat("Your answer is correct. Good job!\n") } else {
    cat( paste("The sum of ", x, " and ", y, " should be ",x+y, "!\n", sep="") )
  }
}

add() #test the function
```

- Be sure that "else" appears at the same line where the "if" part ends (使用 else 時, else 必須和 if 部分結尾的 “}” 在同一行). Run the two sets of commands below to see which one yields an error message.

```
##### First set of commands
x <- 0.5; y <- 0
if (x>3)
{
  y <- 1
}
else
{
  y <- 2
}

##### Second set of commands
x <- 0.5; y <- 0
if (x>3)
{
  y <- 1
} else ### 注意: else 移到了 if 部分的最後一行, 這樣才能執行
{
  y <- 2
}
```

- Example 2. 寫一個函數 `abs0.fun` 計算絕對值.

```
### define abs0.fun
abs0.fun <- function(x){ if (x>0) { return(x) } else { return(-x) } }

### compute |-1| and |1|
abs0.fun(-1); abs0.fun(1)
```

- 在函數運算時, 一旦執行了 `return` 指令, 所有運算就結束了. 因此 Example 2 的函數 `abs0.fun` 也可改寫如下

```
abs1.fun <- function(x){
  if (x>0) { return(x) }
  return(-x)
}
abs1.fun(-1); abs1.fun(1)
```

- Example 3. Suppose that

$$f(x) = \begin{cases} 0 & \text{if } x < 0; \\ x & \text{if } 0 \leq x \leq 1; \\ 1 & \text{if } x > 1. \end{cases}$$

Define a function `f` in R that returns $f(x)$ for a input value x .

Solution.

```
f <- function(x){
  if ( x<0) { return(0) }
  if ( (0 <= x)&(x <= 1) ) { return(x) }
  return(1)
}
f(-1); f(0.5); f(3)
```

- if指令也可以用來檢查function input 是否正常.

Example 4. Write an R function `mean.fun` with input `x`. If `x` is numeric (`is.numeric(x)` is TRUE), the function output is the sample mean of `x`. If `x` is not numeric, the function returns the string:

"The sample mean cannot be computed since x is not numeric."

Sol.

```
mean.fun <- function(x){
  a <- "The sample mean cannot be computed since x is not numeric."
  if (!is.numeric(x)) { return(a) }
  return(mean(x))
}

##test
mean.fun("hi")
mean.fun(1:5)
```