pr7 함수

강현승

2022-10-18

함수와 사용자 정의 함수

함수

- 특정 목적에 맞게 생성된 연산과정의 집합
- ex) `mean` 함수: 모든 원소의 합을 원소의 개수로 나눔

사용자 정의 함수

- 사용자의 편의에 따라 직접 작성하여 사용하는 함수
- 함수명 = function(인수){연산과정} 형태로 작성 (한 가지 연산만 할 경우 {}로 묶지 않아도 됨)
- 연산과정으로 나오는 결과 값을 return, print, cat 등으로 반환하는 형태가 이상적

예시 1. 두 숫자를 비교해 더 큰 수를 반환하는 함수

```
# 2 개의 숫자를 인수로 받아서 더 큰 수를 반환하는 함수
compare = function(x, y) {
  if (x > y)
    cat(x)
  else
    cat(y)
}

compare(10,20)
```

```
## 20
```

예시 2. 평균 값과 표준 오차를 계산하는 함수

```
# 표준오차 = 표준편차 / 표본의 크기

se = function(x) {
  tmp.sd = sd(x)#표준편차
  tmp.N = length(x)#표본크기
  tmp.se = tmp.sd / sqrt(tmp.N)#평균의표준오차
  return(tmp.se)
}

A = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
se(A)
```

```
## [1] 0.9574271
```

예시 3. 데이터 프레임의 앞 뒤 3 개의 데이터를 리스트로 보여주는 함수

```
head_tail = function(x) {
  front = head(x, 3) # 데이터 프레임 상위 3 개 데이터 프래임 생성
  rear = tail(x, 3) # 데이터 프레임 하위 3 개 데이터 프레임 생성
  F_R = list(front, rear) # 2 개 데이터 프레임 리스트로 묶음
  return(F_R) # 묶은 리스트 반환
}

head_tail(mtcars)
```

```
## [[1]]
##                mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
##
## [[2]]
##                mpg cyl disp  hp drat   wt qsec vs am gear carb
## Ferrari Dino  19.7   6  145 175 3.62 2.77 15.5  0  1    5    6
## Maserati Bora 15.0   8  301 335 3.54 3.57 14.6  0  1    5    8
## Volvo 142E    21.4   4  121 109 4.11 2.78 18.6  1  1    4    2
```

예시 4. 홀수 판별 함수

```
oddnum = function(x) {
  if (x %% 2 == 1) {
    # 2로 나눈 나머지가 1이면
    return(T) # T를 반환
  } else{
    # 그렇지 않으면
    return(F) # F를 반환
  }
}
```

scope of variable

- 함수 바깥에서 생성된 변수는 같은 함수 안에서는 언제나 사용 가능
- 함수 안에서 생성된 변수는 함수가 종료되면 사라짐
- 함수 내애서 생성된 변수가 사라지지 않게 하려면 “<←”을 할당 연산자로 사용

```
# <- 할당 연산자 사용
scopetest = function(x) {
  a = 10
  print(a)
  print(x)
}

scopetest(9)
```

```
## [1] 10
## [1] 9
```

```
#print(a) # 주석 제거 후 함수 실행하여 메세지 확인할것

# 할당 연산자 사용
scopetest = function(x) {
  a <<- 10
  print(a)
  print(x)
}

scopetest(9)
```

```
## [1] 10
## [1] 9
```

```
print(a)
```

```
## [1] 10
```

함수의 default 값 설정

- 인수를 입력하지 않았을 때 자동으로 적용되는 값을 default라고 함.
- 함수 작성 시 “인수=T 또는 인수=10” 이런 식으로 미리 인수에 적용될 값을 입력

```
add10 = function(x = 10)
  x + 10
add10()
```

```
## [1] 20
```

```
add10(20)
```

```
## [1] 30
```

PR7 연습문제

```
#install.packages('httr')
#install.packages('rvest')
library(httr)
library(rvest)

fetch_books_to_df = function (to_page_num) {
  result = data.frame()
  for (page_num in 1:to_page_num) {
    title = c()
    info = c()
    url <-
      paste0(
        'https://www.aladin.co.kr/shop/wbrowse.aspx?BrowseTarget=List&ViewRowsCoun
t=25&ViewType=Detail&PublishMonth=0&SortOrder=2&page=',
        page_num,
        '&Stockstatus=1&PublishDay=84&CID=437&SearchOption='
      )
    web_source <- GET(url) # GET 함수를 이용해 웹 페이지의 문서를 가져온다.
    my_html <-
      read_html(web_source) # read_html 함수를 이용하여 html 내용을 읽어온다.
    ## 제목 정보
    title_nodes <-
      html_nodes(my_html, 'a.bo3 > b') # 제목 정보가 있는 노드를 가져온다.
    title_text <- html_text(title_nodes) # 제목 텍스트를 저장한다.
    ## 책 정보
    info_nodes <- html_nodes(my_html, 'div.ss_book_list > ul > li')[3]
    info_text <- html_text(info_nodes)

    my_book_info_nodes = html_nodes(my_html, 'div.ss_book_list > ul > li')
    title = append(title, title_text)
    my_book_info_raw_strings = my_book_info_nodes[grep('wSearchResult', my_book_in
fo_nodes)]
    for (rawStrings in my_book_info_raw_strings) {
      info = append(info, html_text(rawStrings))
    }
    result = rbind(result, data.frame(list(title = title, info = info)))
  }
  return(result)
}

str(fetch_books_to_df(3))
```

```
## 'data.frame':    75 obs. of  2 variables:
##  $ title: chr  "코딩 자율학습 나도코딩의 C 언어 입문" "혼자 공부하는 파이썬" "Do it! 점프 투 파
이썬" "이것이 자바다" ...
##  $ info : chr  "나도코딩 (지은이) | 길벗 | 2022년 10월" "윤인성 (지은이) | 한빛미디어 | 2022
년 6월" "박응용 (지은이) | 이지스퍼블리싱 | 2019년 6월" "신용권, 임경균 (지은이) | 한빛미디어 | 2022
년 9월" ...
```

```
head(fetch_books_to_df(3))
```

```
##                                   title
## 1 코딩 자율학습 나도코딩의 C 언어 입문
## 2                   혼자 공부하는 파이썬
## 3                 Do it! 점프 투 파이썬
## 4                          이것이 자바다
## 5       혼자 공부하는 머신러닝 + 딥러닝
## 6        혼자 공부하는 얄팍한 코딩 지식
##                                                 info
## 1             나도코딩 (지은이) | 길벗 | 2022년 10월
## 2          윤인성 (지은이) | 한빛미디어 | 2022년 6월
## 3     박응용 (지은이) | 이지스퍼블리싱 | 2019년 6월
## 4 신용권, 임경균 (지은이) | 한빛미디어 | 2022년 9월
## 5         박해선 (지은이) | 한빛미디어 | 2020년 12월
## 6          고현민 (지은이) | 한빛미디어 | 2022년 5월
```

PR7 도전문제

```
# install.packages('digest')
library(digest)

# 샘플 block 데이타

block1 <- list(
  number = 1,
  timestamp = "2022-09-28",
  data = "세형",
  parent_hash = "0"
)
block1$hash = digest(block1, "sha256")
block2 <- list(
  number = 2,
  timestamp = "2022-09-28",
  data = "재형",
  parent_hash = block1$hash
)
block2$hash = digest(block2, "sha256")
block3 <- list(
  number = 3,
  timestamp = "2022-09-30",
  data = "민철",
  parent_hash = block2$hash
)
block3$hash = digest(block3, "sha256")

# 검증 함수

is_genuine = function(blockchain) {
  genuine_list = logical() # 블록을 검증한 뒤 논리형 벡터에 push
  # 첫 번째 블록의 parent_hash는 쓰레기 값이므로 자신의 데이터를 자신의 hash로 검증
```

```
  currentBlock = blockchain[[1]]
  genuine_list = append(genuine_list, currentBlock$hash == digest(
    list(
      number = currentBlock$number,
      timestamp = currentBlock$timestamp,
      data = currentBlock$data,
      parent_hash = currentBlock$parent_hash
    ),
    "sha256"
  ))
  # 마지막 블록은 자신의 hash를 담은 parent_hash를 가진 블록이 존재할 수 없으므로 자신의 데이타를 자신의 hash로 검증
  currentBlock = blockchain[[length(blockchain)]]
  genuine_list = append(genuine_list, currentBlock$hash == digest(
    list(
      number = currentBlock$number,
      timestamp = currentBlock$timestamp,
      data = currentBlock$data,
      parent_hash = currentBlock$parent_hash
    ),
    "sha256"
  ))
  if (length(blockchain) < 3) {
    return(sum(genuine_list) == 2)
  }
  for (i in 2:length(blockchain)) {
    parentBlock = blockchain[[i - 1]]
    genuine_list = append(genuine_list,
                          blockchain[[i]]$parent_hash == digest(
                            list(
                              number = parentBlock$number,
                              timestamp = parentBlock$timestamp,
                              data = parentBlock$data,
                              parent_hash = parentBlock$parent_hash
                            ),
                            "sha256"
                          ))
  }
  return(sum(genuine_list) == (length(blockchain) + 1))
}

blockchain = list(block1, block2, block3)
is_genuine(blockchain) # 원본으로 확인
```

```
## [1] TRUE
```

```
blockchain = list(block1, block2, block3)
blockchain[[1]]$data = 'a'
is_genuine(blockchain) # 변조된 데이터로 확인(1)
```

```
## [1] FALSE
```

```
blockchain = list(block1, block2, block3)
blockchain[[2]]$data = 'a'
is_genuine(blockchain) # 변조된 데이터로 확인(2)
```

```
## [1] FALSE
```

```
blockchain = list(block1, block2, block3)
blockchain[[3]]$data = 'a'
is_genuine(blockchain) # 변조된 데이터로 확인(3)
```

```
## [1] FALSE
```