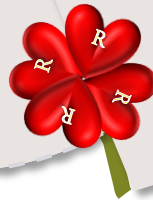


4.리스트(list) & 행렬(matrix)

다양한 데이터를 저장할 수 있는 list 와 여러 행,
열로 구성된 행렬구조에 대해 학습합니다.



1. 리스트(list)가 뭐죠?



내가 좋아하는 과일은 "사과"
내가 좋아하는 친구는 "홍길동"
내가 좋아하는 숫자는 7
내가 좋아하는 알파벳은 'z'
이걸 하나로 묶을 수 있을까?



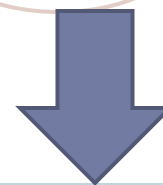
리스트를 이용하면 돼요^^
리스트엔 숫자, 문자 더 넣을 수 있거든요



벡터에선 안되는데, 리스트에서
는 숫자와 문자열을 그렇게 섞
어서 한 곳에 담아도 돼?



리스트의 항목에는 어떤 자료형도 모두
올 수 있어요. 걱정마세요^^
아래처럼 표시하면 돼요.



```
myfavorit <- list(친구='홍길동', 번호=7, 알파벳='z')
```

벡터

15	23	17	홍길동	32
----	----	----	-----	----



리스트

서울	17000	25	34.2	홍길동
----	-------	----	------	-----

리스트(list)

- 숫자형, 문자형 데이터를 섞어서 저장할 수 있으며 벡터와 같이 인덱스 번호로 내부 데이터 접근이 가능

◆ 리스트 만들기

- 리스트 값들도 결국은 변수에 할당
- list() 함수를 이용하여 생성.
- list(키1=값1, 키2=값2..) → 해당 키에 값을 할당

```
> myfavorit<-list(friend='홍길동',mynum=7,myalpha='z')
> myfavorit
$friend<— 키
[1] "홍길동"← 값

$mynum
[1] 7

$myalpha
[1] "z"
```

```
> myfavorit$myalpha ← 리스트의 특정키를 조회하는 경우 변수명$키이름 과 같이 코딩해주면 됩니다.
[1] "z"
> |
```

◆ 리스트에 항목 추가하는 작업

형식 : 변수명\$추가할key←값

```
> myfavorit$mysong<-'hello'
> myfavorit
$friend
[1] "홍길동"

$mynum
[1] 7

$myalpha
[1] "z"

$mysong
[1] "hello"
```

mysong 키 항목에 값 'hello'를 할당

리스트에 추가된 걸 알 수 있어요.

리스트에 벡터형 값도 할당할 수
있어요^^

```
> myfavorit$myfood<-c('월남쌈','쌀국수','부대찌개')
> myfavorit
$friend
[1] "홍길동"

$mynum
[1] 7

$myalpha
[1] "z"

$mysong
[1] "hello"

$myfood
[1] "월남쌈"    "쌀국수"    "부대찌개"
```

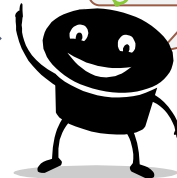
2. 행렬(matrix)

- 2차원 벡터를 표현하기 위한 방법
- `matrix()` 함수를 이용하여 생성

	1열	2열	3열
1행			
2행			
3행			

```
age<-c(25,33,32,37,27,38)
```

25	33	32	37	27	38
----	----	----	----	----	----



벡터는 1행밖에 표시를 할 수 없어요ㅠㅠ



그래ㅠ 나는 2행 3열 구조로 이 데이터들을 처리하고 싶은데..



예^^ 걱정말아요!! 우리에게 행렬이 있어요. `matrix()` 함수로 그림처럼 만들어드리죠^^

	(1,1)	(1,2)	(1,3)
25	25	32	27
33	33	37	38
	(2,1)	(2,2)	(2,3)

◆ 행렬 생성방법

- 행과 열의 인덱스는 1
- `matrix(행렬의 구성원소, nrow=행의갯수, ncol=열의 갯수, dimnames=행열의 이름)`
- 기본적으로 열중심(column-major order)
- 행 중심으로 원소 기록하고자 할 경우 `matrix(행렬의 구성원소, nrow=행의갯수, ncol=열의 갯수, byrow=TRUE)`

```
> age<-matrix(c(25,33,32,37,27,38), nrow=2, ncol=3)
```

```
> age
```

생성할 행, 열의 개수 지정

```
      [,1] [,2] [,3]
[1,]   25   32   27
[2,]   33   37   38
> |
```

ncol=3 을 생략해도 됩니다. 왜냐하면, 행렬의 요소가 6개 이므로, 2행이면 당연히 열의 개수는 3이 되니까요^^

행렬의 특정 열, 행에 접근하는 방법

```
> age[,1]
[1] 25 33
> age[2,]
[1] 33 37 38
> |
```

행 중심으로 원소들을 나열하고자 할 경우 `byrow` 인자 사용

```
> age<-matrix(c(25,33,32,37,27,38), nrow=2, byrow=TRUE)
```

```
> age
```

```
      [,1] [,2] [,3]
[1,]   25   33   32
[2,]   37   27   38
```

◆ 행 과 열 이름 지정하기

➤ 행열에서 행 또는 열에 이름을 지정(vector에서 names())함수 기억나죠?)

➤ dimnames(행렬) ← 값

```
> info<-matrix(c("177cm","68kg","165cm","57kg","160cm","55kg","155cm","50kg"),ncol=2,byrow=TRUE)
> info
```

	[,1]	[,2]
[1,]	"177cm"	"68kg"
[2,]	"165cm"	"57kg"
[3,]	"160cm"	"55kg"
[4,]	"155cm"	"50kg"

1열에는 키, 2열에는 몸무게

각 행은 1번,2번,3번,4번 과 같이 이름을 지정하겠습니다.

참고로, 현재 행렬은 행 중심이므로 행의 순서대로 이름이 할당됩니다.

```
> dimnames(info)<-list(c("1번","2번","3번","4번"),c("키","몸무게"))
> info
```

	키	몸무게
1번	"177cm"	"68kg"
2번	"165cm"	"57kg"
3번	"160cm"	"55kg"
4번	"155cm"	"50kg"

행 이름

열 이름

◆ 행열에 새로운 행/열 추가하는 작업

➤ 행 추가 함수 → `rbind(행렬, 행에 추가할 원소)`➤ 열 추가 함수 → `cbind(행렬, 열에 추가할 원소)`

```
> a1<-c("167cm", "54kg")
```

```
> a2<-c("178cm", "70kg")
```

```
> rbind(info, a1, a2)
```

2개 행 추가

	키	몸무게
1번	"177cm"	"68kg"
2번	"165cm"	"57kg"
3번	"160cm"	"55kg"
4번	"155cm"	"50kg"
a1	"167cm"	"54kg"
a2	"178cm"	"70kg"

```
> cbind(info, c("남", "남", "여", "여"))
```

	키	몸무게	
1번	"177cm"	"68kg"	"남"
2번	"165cm"	"57kg"	"남"
3번	"160cm"	"55kg"	"여"
4번	"155cm"	"50kg"	"여"

```
> |
```

열 추가