

R 프로그래밍

(2주차)

2016. 03. 12(토)

장운호

(ADP 002-0004)

목차

※ 인코딩 개념 소개

I. 변수

II. 함수

III. 객체

Wrap-up. R의 작동방식 정리



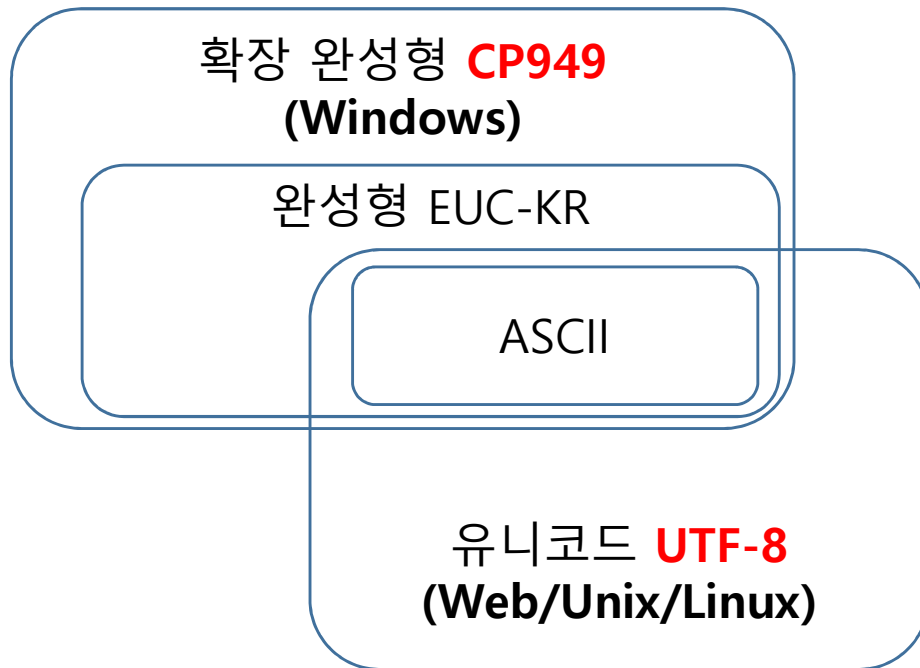
※. 인코딩 개념 소개

인코딩(encoding)

컴퓨터에서 문자를 처리하는 체계를 일컬음.

- 텍스트가 깨져서 나오는 경우는 인코딩이 맞지 않았기 때문임.

[한글 인코딩의 구분법]



[주요 인코딩 개요]

ASCII	영문과 숫자를 처리하기 위한 세계표준 영문 및 숫자 인코딩 체계
CP949	완성형으로 한글을 표현하기 위해 6만여의 한글에 각각의 코드를 부여할 수 있도록 Microsoft에서 개발한 한글인코딩 체계
UTF-8	세계 전체의 모든 문자를 수용하여 일관된 체계로 관리가 가능하도록 정리한 문자 인코딩 체계

자료) mindscale.kr (웹크롤링 강좌)

※ 아스키(ASCII) 코드표

10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자	10	HEX	문자
0	0x00	NULL	22	0x16	STN	44	0x2C	,	66	0x42	B	88	0x58	X	110	0x6E	n
1	0x01	SOH	23	0x17	ETB	45	0x2D	-	67	0x43	C	89	0x59	Y	111	0x6F	o
2	0x02	STX	24	0x18	CAN	46	0x2E	.	68	0x44	D	90	0x5A	Z	112	0x70	p
3	0x03	ETX	25	0x19	EM	47	0x2F	/	69	0x45	E	91	0x5B	[113	0x71	q
4	0x04	EOT	26	0x1A	SUB	48	0x30	0	70	0x46	F	92	0x5C	\	114	0x72	r
5	0x05	ENQ	27	0x1B	ESC	49	0x31	1	71	0x47	G	93	0x5D]	115	0x73	s
6	0x06	ACK	28	0x1C	FS	50	0x32	2	72	0x48	H	94	0x5E	^	116	0x74	t
7	0x07	BEL	29	0x1D	GS	51	0x33	3	73	0x49	I	95	0x5F	_	117	0x75	u
8	0x08	BS	30	0x1E	RS	52	0x34	4	74	0x4A	J	96	0x60	`	118	0x76	v
9	0x09	HT	31	0x1F	US	53	0x35	5	75	0x4B	K	97	0x61	a	119	0x77	w
10	0x0A	Wn	32	0x20	SP	54	0x36	6	76	0x4C	L	98	0x62	b	120	0x78	x
11	0x0B	VT	33	0x21	!	55	0x37	7	77	0x4D	M	99	0x63	c	121	0x79	y
12	0x0C	FF	34	0x22	"	56	0x38	8	78	0x4E	N	100	0x64	d	122	0x7A	z
13	0x0D	Wr	35	0x23	#	57	0x39	9	79	0x4F	O	101	0x65	e	123	0x7B	{
14	0x0E	SO	36	0x24	\$	58	0x3A	:	80	0x50	P	102	0x66	f	124	0x7C	
15	0x0F	SI	37	0x25	%	59	0x3B	;	81	0x51	Q	103	0x67	g	125	0x7D	}
16	0x10	DLE	38	0x26	&	60	0x3C	<	82	0x52	R	104	0x68	h	126	0x7E	~
17	0x11	DC1	39	0x27	'	61	0x3D	=	83	0x53	S	105	0x69	i	127	0x7F	DEL
18	0x12	DC2	40	0x28	(62	0x3E	>	84	0x54	T	106	0x6A	j			
19	0x13	DC3	41	0x29)	63	0x3F	?	85	0x55	U	107	0x6B	k			
20	0x14	DC4	42	0x2A	*	64	0x40	@	86	0x56	V	108	0x6C	l			
21	0x15	NAK	43	0x2B	+	65	0x41	A	87	0x57	W	109	0x6D	m			



I. 변수

1. 변수란?

“값”을 저장할 수 있는 메모리 공간에 붙여진 이름을 일컬음.

- 변수의 이름을 활용하여 계산된 값을 저장할 수 도 있고, 저장된 값을 참조할 수도 있음.

[변수의 선언]

일반적인
프로그램
언어에서
사용되는
의미

변수라는 것을 선언하면
메모리 공간이 할당되고,
할당된 메모리 공간에
이름이 붙는다.

R에서의
의미

변수는 초기값 할당과 동시에
자동으로 선언된다.

$x = 3$ 또는 $x \leftarrow 3$
(Assignment Operator 활용)

[변수의 이름]

의미

변수의 이름을 통해서
할당된 메모리공간에
접근이 가능하다.

명명
규칙

- 1) 반드시 문자, 숫자, 마침표 또는 Underscore(_)로 만들어야 함.
- 2) 반드시 문자 또는 마침표로 시작
※ 마침표로 시작시 숫자가 바로 뒤따라 올 수 없음
- 3) 예약어는 변수명이 될 수 없음

2. 변수 사용법

변수는 “저장하는 방법 ” 과 “읽는 방법”, 두 가지만 익히면 사용 가능함.

[변수 저장 방법]

```
myBestNumber <- 7
```

```
herBestNumber <- 8
```

```
myBestBook <- "칼의 노래"
```

```
herBestDrama <- "signal"
```

```
sum_of_our_best_number <-  
  myBestNumber + herBestNumber
```

[변수 읽는 방법]

```
Print(myBestNumber)
```

```
myBestNumber
```

```
Sum_of_our_best_number
```

```
myBestNumber + herBestNumber
```

```
myBestBook + herBestDrama
```


3. 예약어 (Reserved Words)와 상수(Constants)

사전에 프로그램에서 특정한 기능이 정해져 있는 예약어들은 변수로 사용이 불가능함.

- 상수 중에서 내장상수들은 값의 변경이 가능함.

[예약어]

> ?reserved

if	else	repeat	while	function
for	in	next	break	TRUE
FALSE	NULL	Inf	NaN	NA 등

[상수]

의미

값을 변경시킬 수 없거나,
값이 미리 지정되어 있는 객체

종류

- 1) 숫자(1~9), 문자(a-z/A-Z/가-힣)
 - 2) 내장 상수
 - LETTERS
 - letters
 - pi
 - month.name
 - month.abb
- ※ 내장상수는 값 변경이 가능함.

참조) <http://www.programiz.com/r-programming/>

4. “값”의 종류

R에는 5가지 종류의 데이터형식(값)이 있고, 상호간의 형 변환이 가능함.

- R에서는 단일 수치값(Scaler)은 존재하지 않음.
※ 단일 숫자들도 R 내부적으로는 원소가 1개인 데이터집합(Vector)으로 처리됨.

[“값”의 종류]

형식 mode	범례
정수형	Integer
숫자형	numeric (실수형)
문자형	character
논리형 boolean	logical (TRUE / FALSE) ※ 숫자로도 표현가능 TRUE(0이 아닌 모든 값) / FALSE(0)
허수형	complex

[데이터형식 변환 방법]

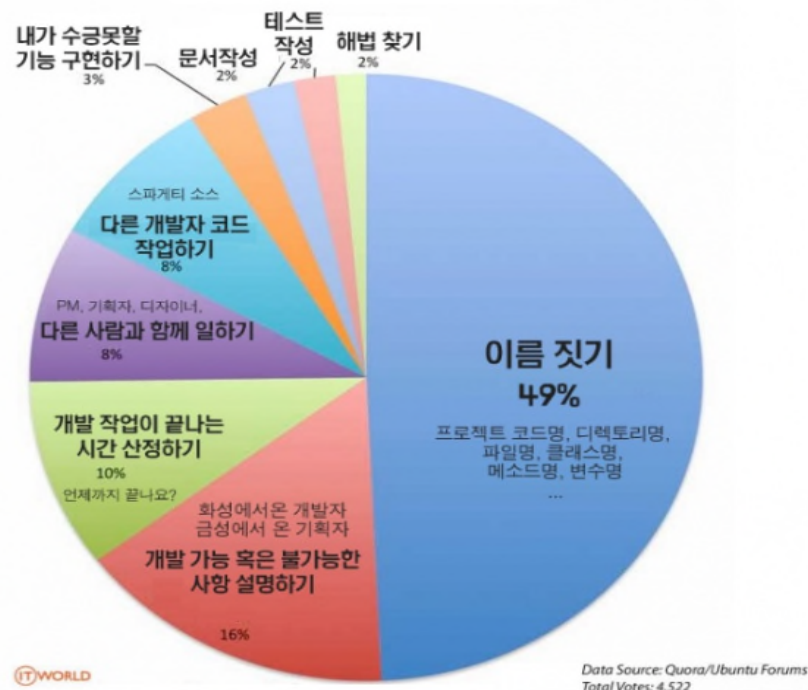
숫자 → 문자	as.character(7)
문자 → 숫자	as.numeric ("7")
실수 → 정수	as.integer (7.5)
숫자 → 논리값	as.logical (3) as.logical (0)

5. 변수 작성시 주의점

가능한 한 최대한 의미있게 이름을 부여하는 것이 중요함.

- 직관적으로 잘 지은 변수명은 담당자가 바뀌어도 알아보기 쉬워, 관리가 용이함.

[프로그래머가 가장 힘들어 하는 일]



[변수명의 유형]

나쁜
사례

```
aaa  
a001  
b20160312_1
```

좋은
사례

```
dummyVar  
weight_by_term_frequency  
wordCloudByTfidf
```

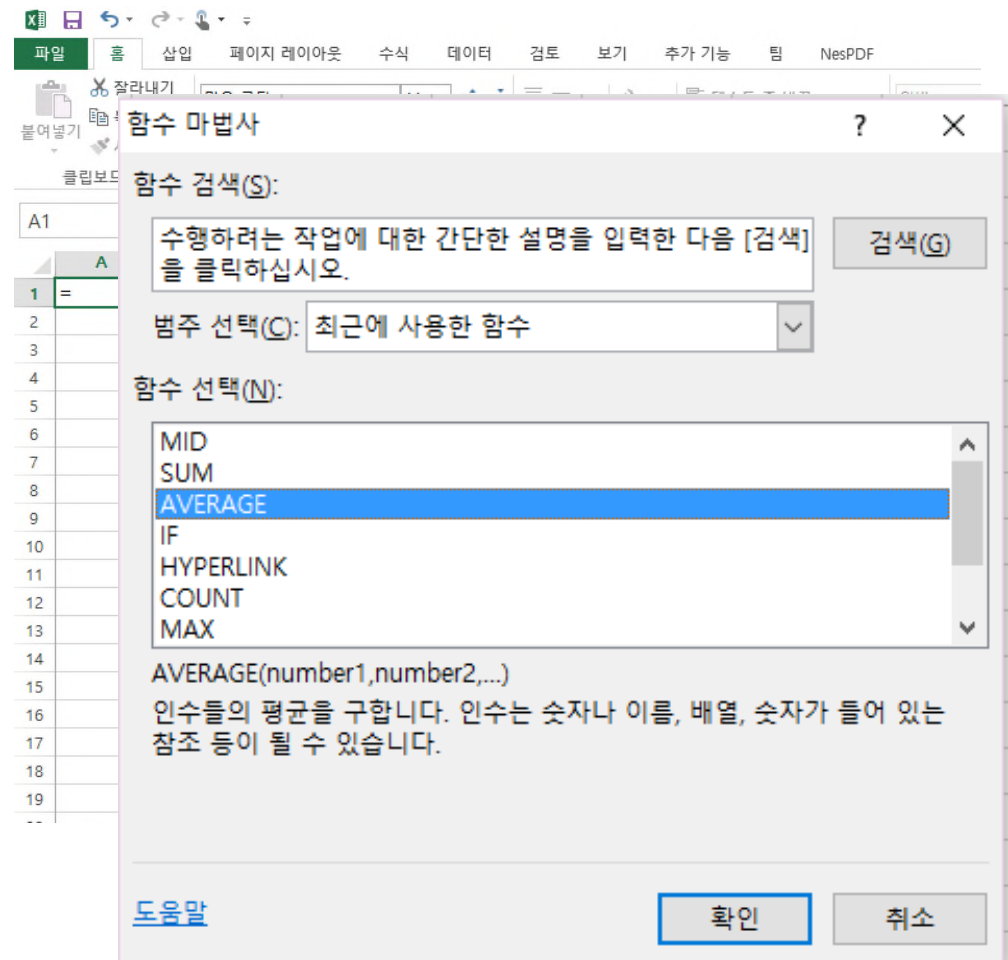
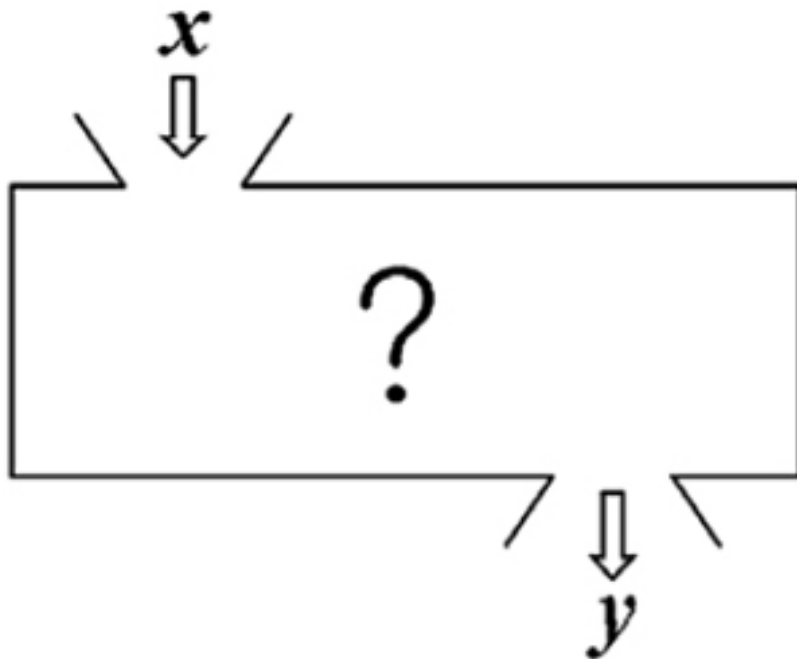


II. 함수

1. 함수란?

영어로는 Function으로, 특정기능(목적)을 가진 것을 통칭하는 개념으로, 특정 입력값(Input)에 대해서 예상되는 Output(결과 또는 특정기능, 목적, 동작)을 기대할 수 있는 것임.

[함수(function)의 개념도]



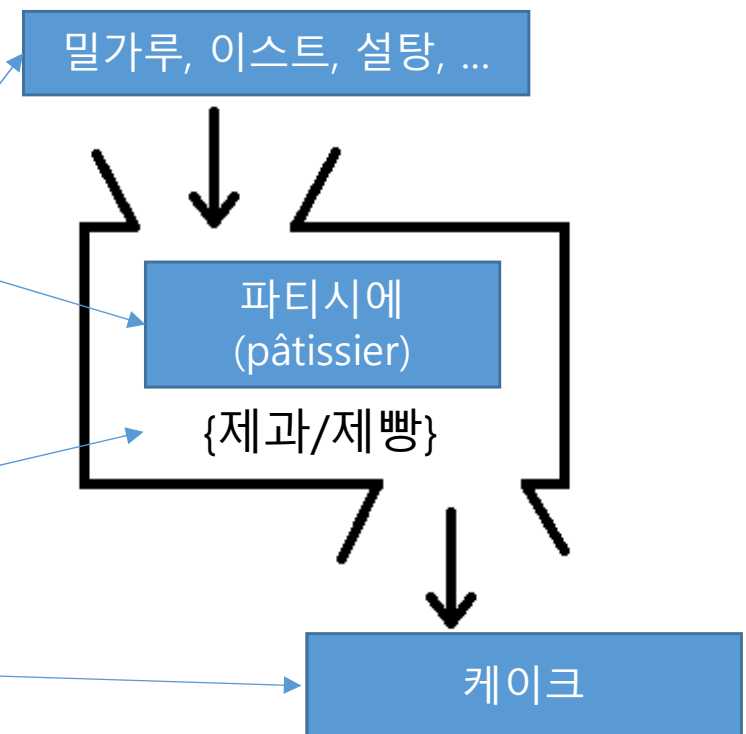
2. 함수의 구성요소

함수는 구별가능(Unique)한 이름(name)을 가지고 구분될 수 있어야 하며, 특정한 입력 값에 대한 처리기능(method)을 가지고 있어야 하고, 마지막으로 이를 일정한 형태의 결과값으로 반환(return)해 줄 수 있어야 함.

[함수(function)의 구성요소]

사례	<code>myVector <- c(1,2,3,4,5)</code>
함수명	<code>c</code> (concatenate/combine의 의미)
입력 (Input)	1,2,3,4,5
기능	입력값들을 엮어 벡터로 만들어 줌
결과 (Return)	1,2,3,4,5를 원소로 하는 벡터 반환

[Bakery의 비유]

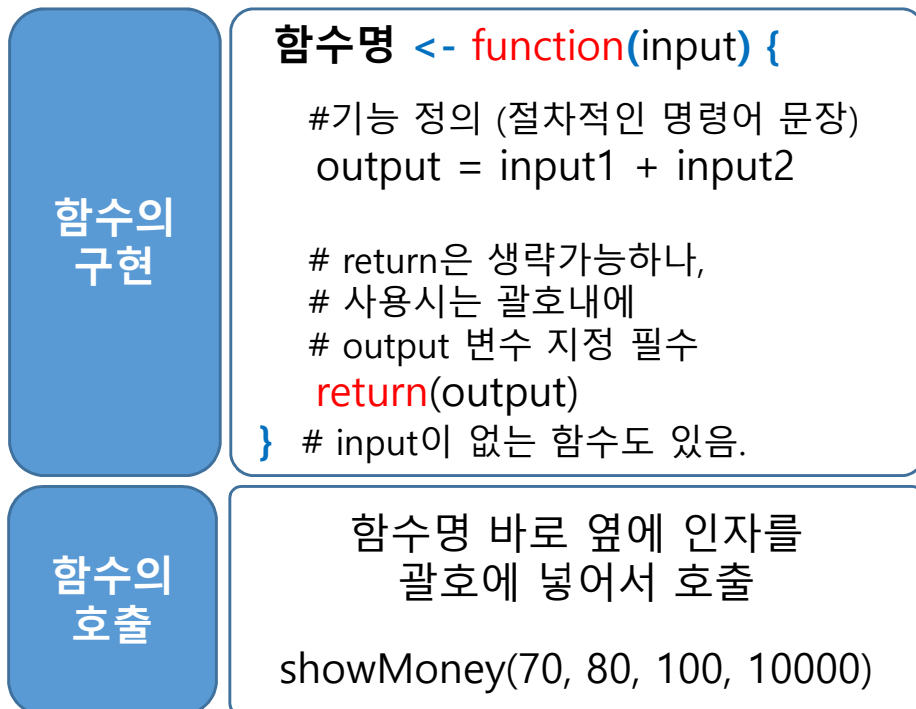


3. 함수의 구현 및 호출

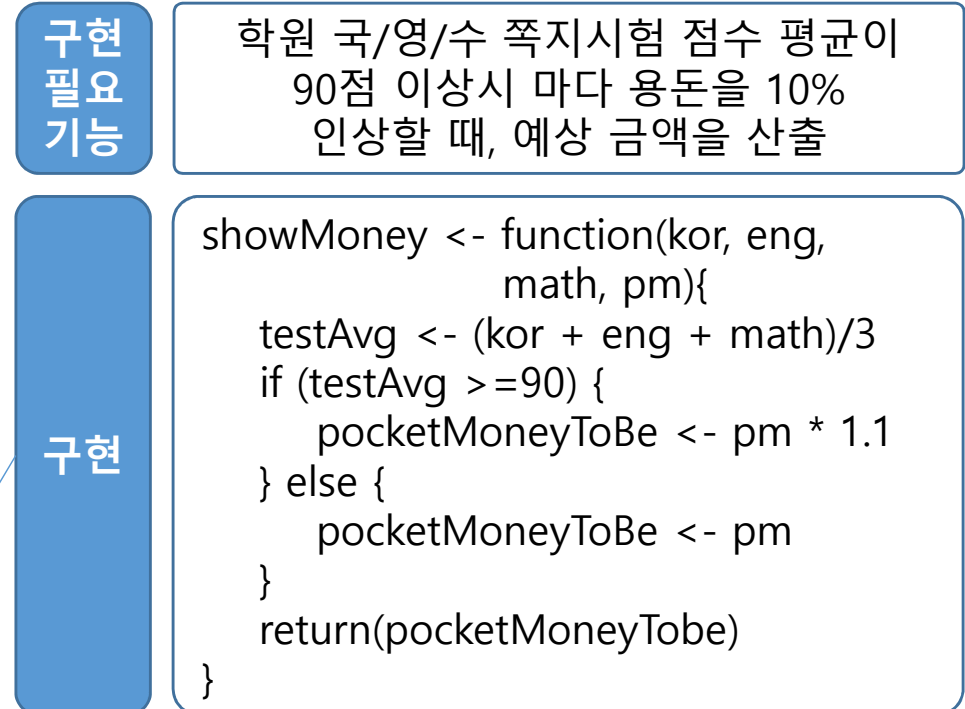
R에는 이미 다양한 기능을 구현해 놓은 함수들이 내장되어 있으나, 만약에 나에게 필요한 기능이 부족하면, 직접 만들어 쓸 수 있음.

- 내장(built-in)함수는 호출만으로 사용가능하고, 사용자 정의 함수는 구현 후 사용 가능

[함수(function)의 구현 및 호출]



[사용자 정의 함수 구현 사례]





Ⅲ. 객체

1. 객체란?

현재 사용중인 컴퓨터 메모리(작업공간)에 특정이름으로 저장되는 모든 것임.



2. 객체의 확인 방법

pryr 패키지의 **address** 함수나 **inspect** 함수를 활용하여 확인 가능

```
install.packages("pryr")  
library(pryr)
```


```
x <- 3  
address(x); address(3) #에러  
address(pi)  
address
```

```
detach("package:pryr", character.only=TRUE)
```

```
mem_add <- function(x) capture.output(.Internal(inspect(x)))  
mem_add("a"); mem_add(1); mem_add("가")
```

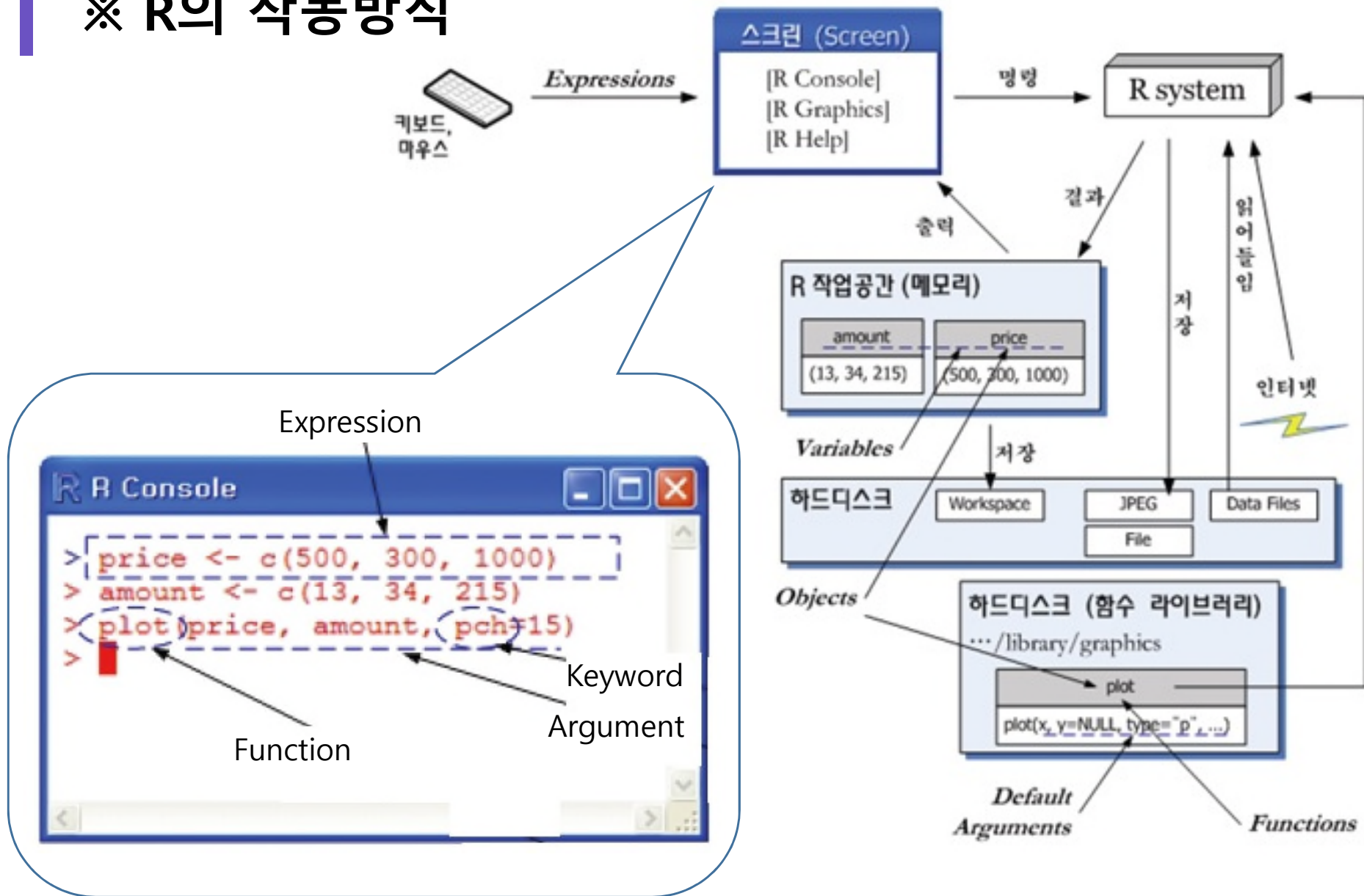
```
.Internal(inspect("a"))
```

```
mem_add(showMoney)
```



Wrap-up. R의 작동방식 정리

※ R의 작동방식



자료) <http://datamining.dongguk.ac.kr/R/R의작동방식및기본사용법.pdf> (일부 편집)

End of Document.

감사합니다.