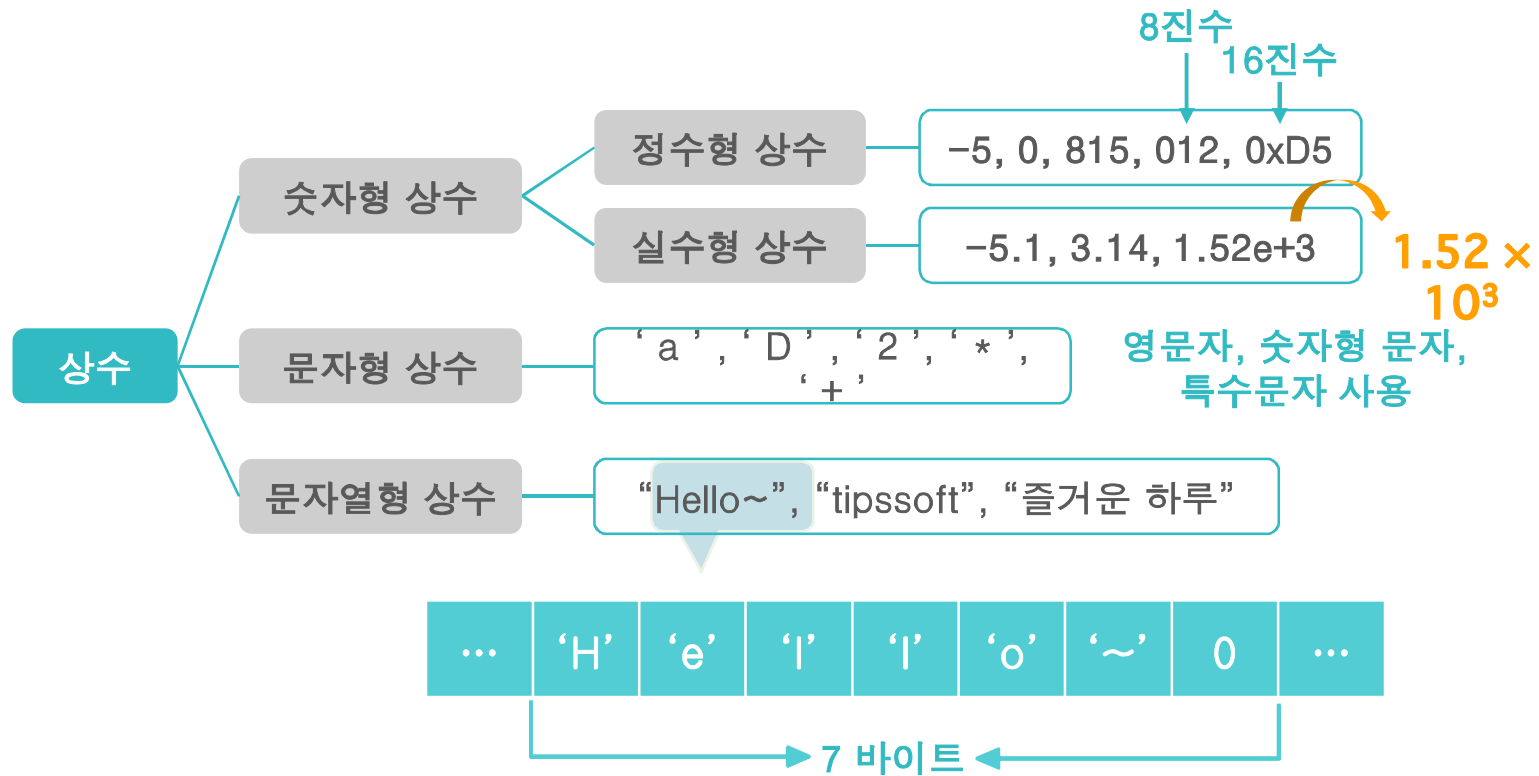

C 언어

- 상수
- 변수

항상 같은 수, 상수

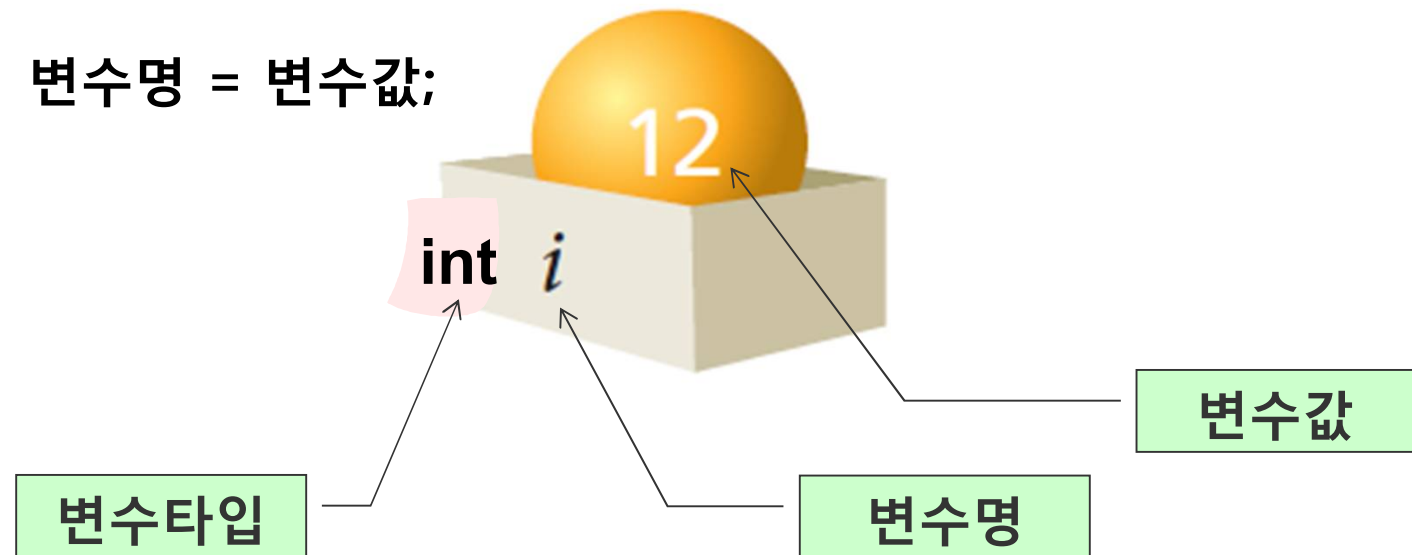
상수(Constant) 는 프로그램을 실행할 때 한 번 값이 결정되면 프로그램이 끝날 때 까지 다른 값으로 바뀌지 않는 정보를 말한다.



변수

- 변수(variable) :
 - 변하는 값이 저장되는 **메모리 공간**

변수타입 변수명 = 변수값;



- 변수명은 데이터가 저장되는 공간(메모리)을 가리키는 별명이다
-

변수의 선언

- 변수를 선언하는 것은 저장 공간을 할당하는 것이다.
- 변수타입 변수명 ;

int

자료형

num

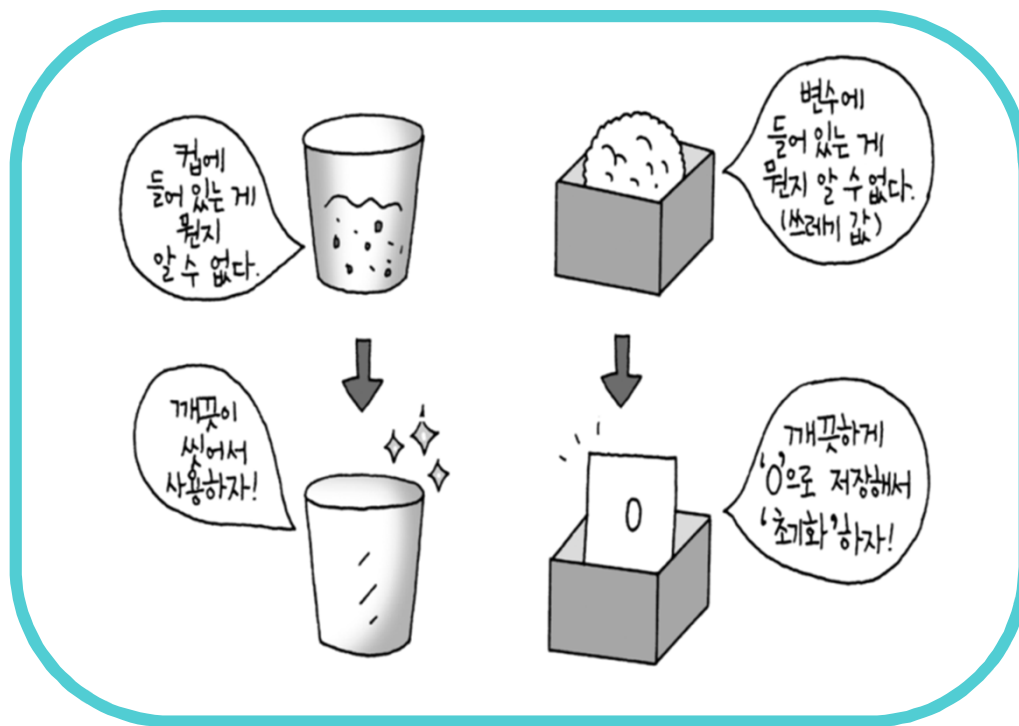
변수 이름

;

구분자

```
bool    fact;
short   maxItems;           // Word separator: Capital
long    long national_debt; // Word separator: underscore
float   payRate;           // Word separator: Capital
double  tax;
float   complex voltage;
char    code, kind;         // Poor style-see text
int     a, b;               // Poor style-see text
```

변수 초기화하기

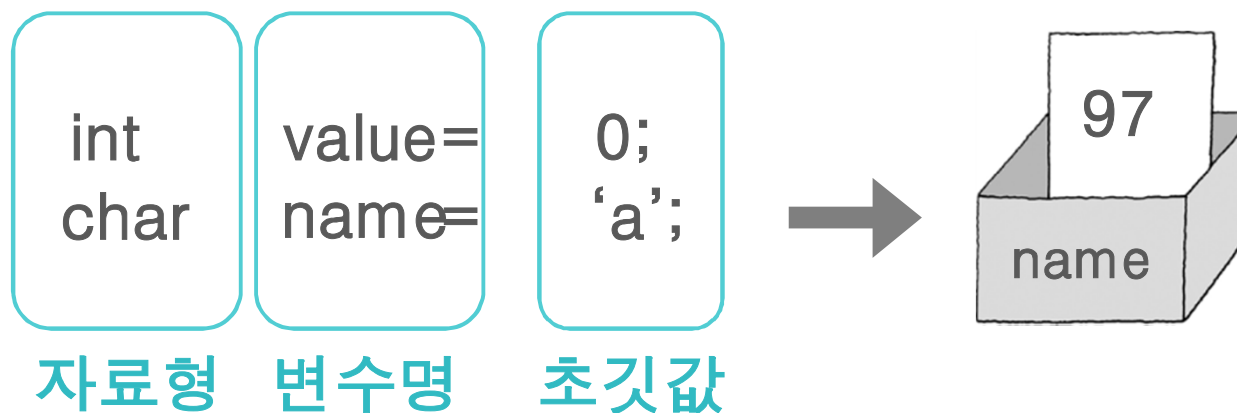


변수를 사용하기 전 초깃값을 저장해서
깨끗하게 만드는 것을 초기화라고 합니다.

변수의 초기화

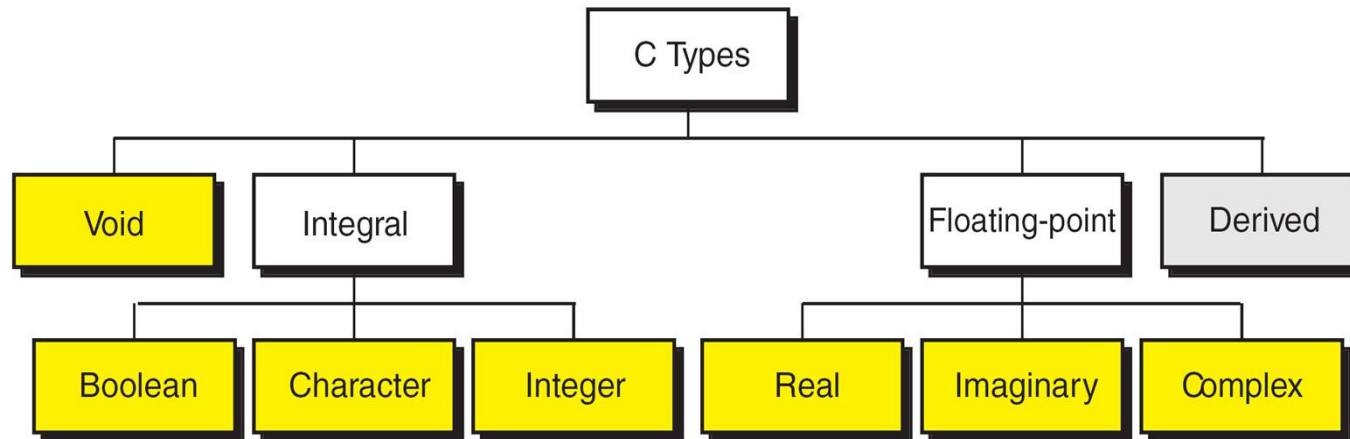
- variable initialization(변수의 초기화)
 - 변수를 초기화하지 않으면 어떤 값이 들어있는지 알 수 없음
초기화 이전에 저장되어 있는 값을 쓰레기 값(garbage)이라 함

```
int score = 100;
```



변수 타입

- A set of values and a set of operations that can be applied on those values.



- void : void 타입은 값을 갖고 있지 않으며, 연산 없음
 - Derived : 다른 타입들로 구성된 복잡한 구조
- 예) pointer, enumerated type, union, array, and structure

변수명(식별자, Identifier)

■ 식별자(변수명)의 작명 규칙

- 변수명의 첫 번째 문자는 영문자, _, \$ 만 된다.
- 변수명의 두 번째 부터는 영문자, _, \$, 숫자 등이 가능
- 대문자와 소문자는 구별된다.
- 최대 63글자만 가능하다.
- 변수명에 예약어를 사용할 수 없다.

첫번째는 영문자만 된다.
두번째부터는 영문자와 숫자도 된다.
기호는 _, \$만 된다.



식별자(변수 이름) 선언의 예

```
int speed;
```

```
int earthPopulation;
```

```
int _count;
```

```
int $value;
```

```
int 반복횟수;
```

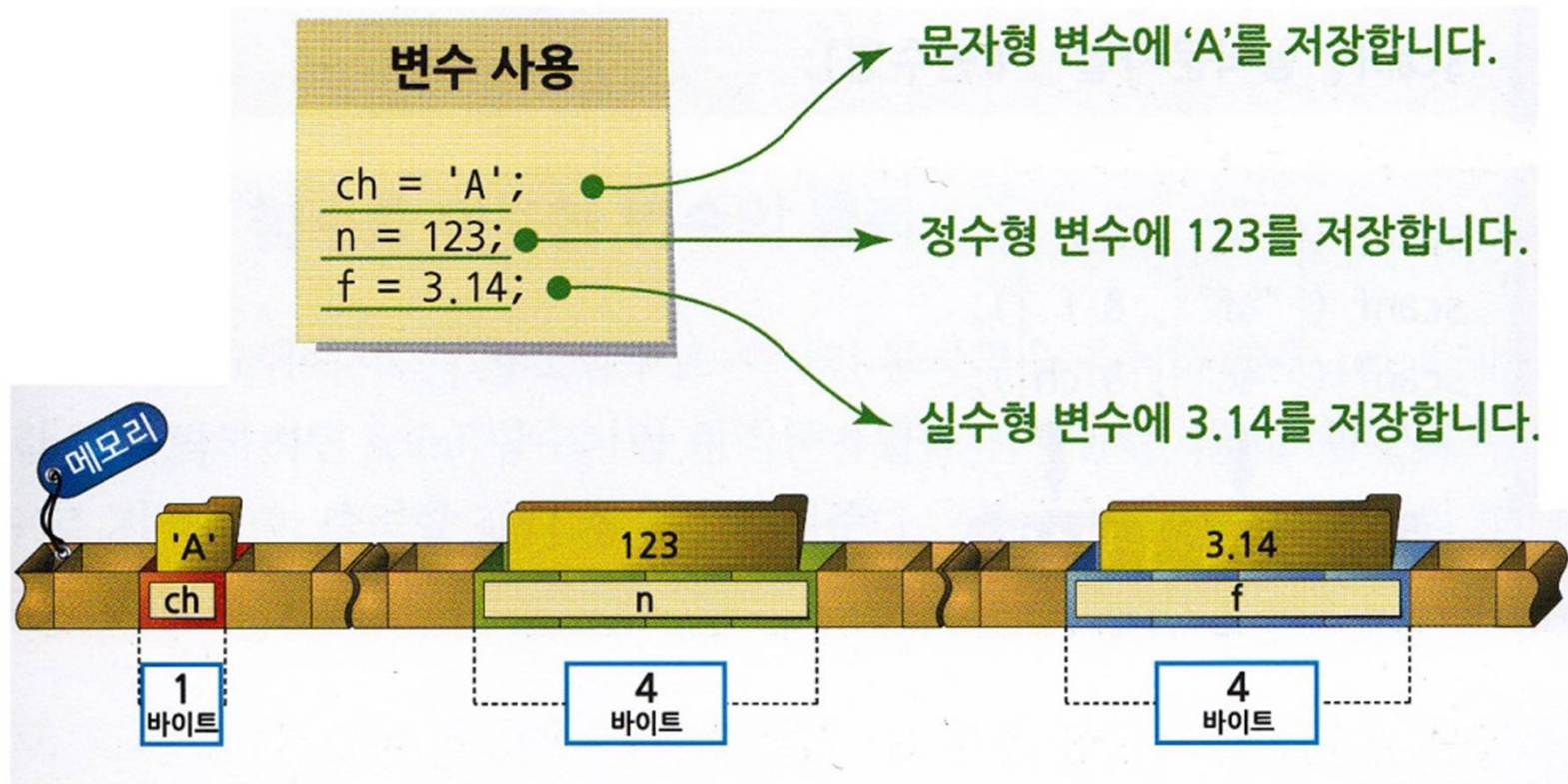
```
int Henry8;
```

```
int 1stPrizeMoney;
```

```
int break;
```

```
int #ofComputer;
```

Types



실습 예제 ex04-01.c

```
#include <stdio.h>
#include <stdlib.h>
void main(void)
{
    // 변수 초기화
    char c = 'A';
    int n = 123;
    double d = 3.14;

    // 출력 : printf
    printf(" [%c] [%d] [%lf]", c, n, d);

    // debug시 결과화면 스톱시키기 위해.
    getchar();
}
```

상수(Constants)

■ 상수(Constants) :

- 변하는 양의 값

■ 상수(Constants) :

- define 상수
 - const 상수
 - 리터럴 상수
-

ASCII Table

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

ASCII Table

128	Ç	144	É	160	á	176	☐	193	⌞	209	ƒ	225	ß	241	±
129	ü	145	æ	161	í	177	☐	194	⌞	210	π	226	Γ	242	≥
130	é	146	Æ	162	ó	178	☐	195	⌞	211	ℓ	227	π	243	≤
131	â	147	ô	163	ú	179		196	—	212	ℓ	228	Σ	244	∫
132	ä	148	ö	164	ñ	180	†	197	†	213	ƒ	229	σ	245	∫
133	à	149	ò	165	Ñ	181	†	198	†	214	ƒ	230	μ	246	+
134	â	150	û	166	²	182	‡	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	167	°	183	‡	200	ℓ	216	‡	232	Φ	248	°
136	ê	152	—	168	¿	184	‡	201	ƒ	217	∫	233	Θ	249	.
137	ë	153	Ö	169	—	185	‡	202	‡	218	∫	234	Ω	250	.
138	è	154	Û	170	¬	186	‡	203	ƒ	219	■	235	δ	251	√
139	ï	156	£	171	½	187	‡	204	‡	220	■	236	∞	252	—
140	î	157	¥	172	¼	188	‡	205	=	221	■	237	φ	253	²
141	ï	158	—	173	¡	189	‡	206	‡	222	■	238	ε	254	■
142	Ä	159	ƒ	174	«	190	‡	207	⌞	223	■	239	∩	255	
143	Å	192	Ł	175	»	191	‡	208	⌞	224	α	240	≡		

Source: www.asciitable.com

그러나 ASCII 테이블만 가지고는 일본어, 한글, 중국어와 같은 2byte 문자는 표현할 수 없다. 2byte 문자는 char를 2개 이상 사용해서 저장해야 한다.

이제 마지막으로 문자열이 남았다. C는 문자열을 위한 데이터 타입을 가지고 있지 않다. C에서 문자열을 처리하기 위해서는 배열을 사용해야만 한다. 이것은 원시 데이터 타입을 여러개를 포함하고 있는 데이터 구조다. 예를 들어 문자열은 char를 여러개 포함할 수 있는 데이터 구조를 이용하면 표현할 수 있을 것이다. 배열은 다음장에서 자세히 다루도록 하겠다.