

# 11장 1차원 배열

1. 배역의 이해와 배역의 서어 및 초기화 방법
2. 배역을 이용한 문자열 변수의 표현



# 문자표현방법

3

- 컴퓨터에서는 문자를 숫자코드를 이용하여 저장함
- 아스키코드(ASCII code): 미국 표준 문자코드
  - ▣ 컴퓨터에서 문자를 저장할 때 문자를 숫자로 변환하는 규칙
  - ▣ 영어 알파벳 문자, 숫자, 기호들에 0에서 127까지의 숫자를 할당





# 아스키 코드표

4

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS, BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCLE]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

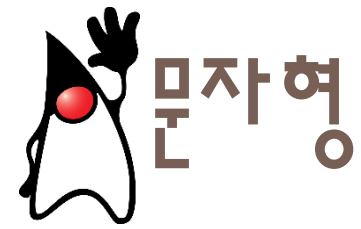


# 문자형 상수

5

- 아스키코드를 외우기 힘들므로 작은 따옴표(' ')를 사용하여 문자형 상수를 나타냄
- 문자형 상수는 아스키코드(숫자)로 변환되어 저장됨

'A'	// 65으로 저장됨
'b'	// 98으로 저장됨
'8'	// 56으로 저장됨
'!'	// 33으로 저장됨



- 문자형 : 문자를 저장하기 위하여 사용하는 자료형

자료형	종류	메모리 크기	저장범위
문자형	char	1바이트(8비트)	-128 ~ 127

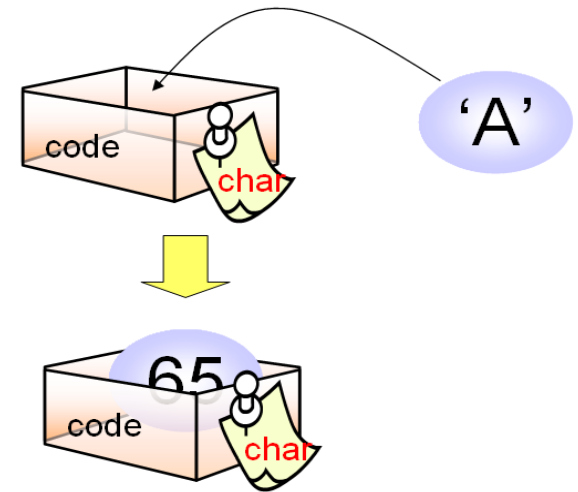
- char형의 변수를 이용하여 문자 저장, 메모리 크기는 8비트

```
char code;
```

- char형의 변수에 문자를 저장하는 방법
  - ▣ 문자 상수를 만나면 아스키코드로 변환하여 저장함

```
code = 'A';    // 'A' 저장
```

```
code = 65;     // 아스키코드를 외워야한다.
```





# 문자 입출력

8

- 문자입력 : 형식지정자 %c 이용

```
char ch;  
scanf("%c", &ch);
```

- 문자출력 : 형식지정자 %c 이용

```
char ch = 'a';  
printf("%c", ch);           // a
```





# 문자 변수와 문자 상수

9

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char code1 = 'A';    // 65가 저장됨
```

```
    char code2 = 65;
```

```
    printf("code1=%c, code2=%c\n", code1, code2);
```

```
    return 0;
```

```
}
```

code1=A, code2=A



# 아스키 코드 출력

10

```
#include <stdio.h>
int main(void)
{
    char code;
    for(code = 32; code =< 127; code++)
    {
        printf("아스키 코드 %d은 %c입니다.\n", code, code);
    }
    return 0;
}
```

아스키코드 32은 입니다.  
아스키코드 33은 !입니다.  
...  
아스키코드 97은 a입니다.  
아스키코드 98은 b입니다.  
아스키코드 127은 입니다.



# 문자열 표현 방법

11

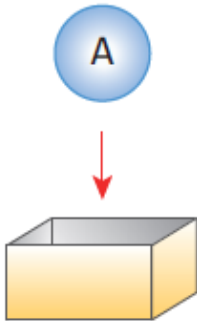
- 문자열(string): 문자들을 여러 개 모아 놓은 것(큰 따옴표 이용 표현)
  - ▣ "Hello World!"
  - ▣ "GilDong Hong"
- 예: 이름, 주소, 등 문자보다 많이 사용됨
- 문자형은 문자 1개 만을 나타내는 자료형



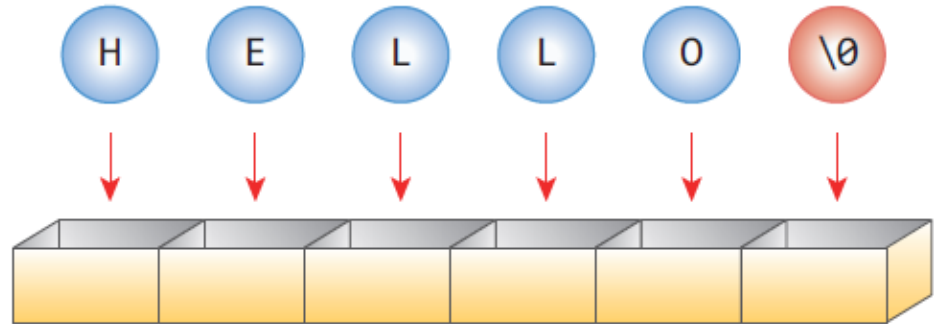
# 문자열 상수와 저장

12

- 문자열 상수: 고정된 값을 갖는 문자열, 큰 따옴표 사용하여 표현
  - ▣ "Hello", "World"
- 문자열을 저장하기 위한 별도의 자료형은 없음
  - ▣ char형 배열을 이용하여 저장
  - ▣ `char str[6] = "HELLO" ;`



하나의 문자는 char형 변수로 저장



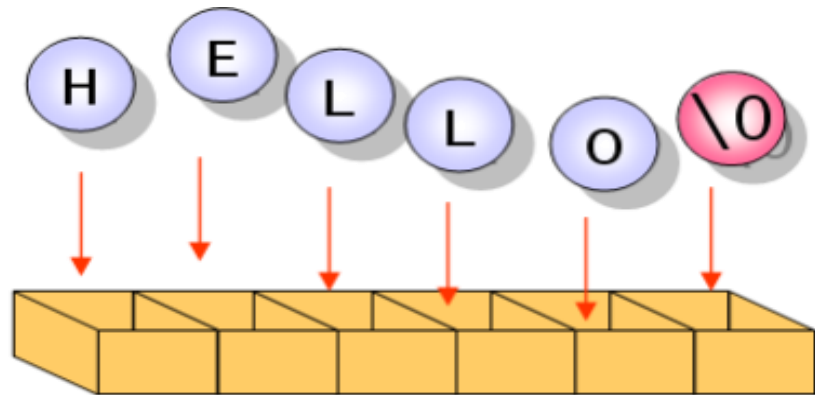
문자열은 char형 배열로 저장

# NULL 문자

13

- NULL 문자: 문자열의 끝을 나타내는 특수문자
  - ▣ 기호 : `'\0'`
  - ▣ 아스키 코드 : 0

```
char str[6] = "Hello";
```



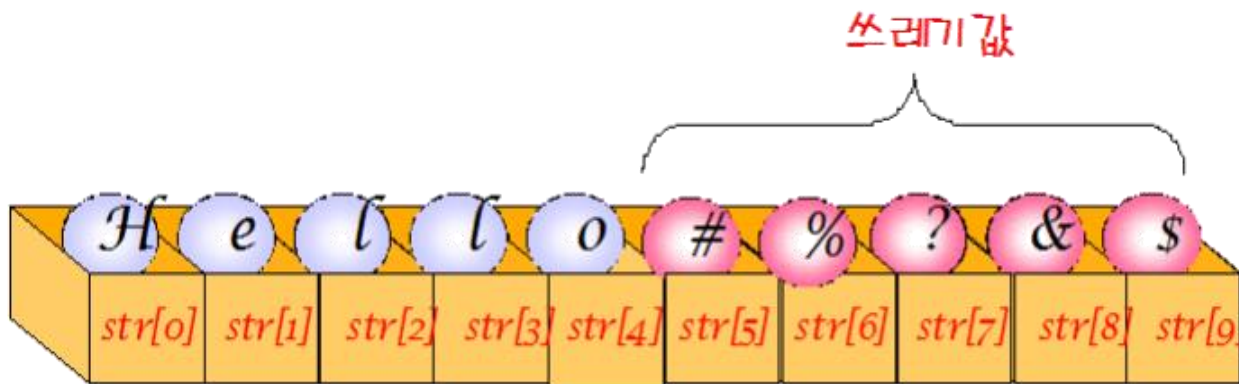
- 주의사항
  - ▣ `'A'` : 문자 상수, 1바이트
  - ▣ `"A"` : 문자열 상수, 2바이트(`'A'`, `'\0'`)



# 왜 NULL 문자가 필요한가?

14

- 문자열은 영문자, 기호, 숫자 모두 포함가능
  - ▣ "Hello", "Hello#", "Hello#%", "Hello88"
- 널문자가 없다면 컴퓨터가 문자열에 포함된 값으로 문자열의 끝이 어디인지 알 수가 없다
- 널문자를 이용하여 문자열의 끝을 표시해줘야 함.



NULL 문자의 필요성: 정상적인 데이터와 쓰레기 값을 분리하기 위해서이다.



# 문자형 배열의 초기화

15

- 문자형 배열 원소들을 중괄호 안에 넣어주는 방법
  - ▣ `char str[6] = { 'H', 'e', 'l', 'l', 'o', '\0' }; // 널문자 포함`
- 문자열 상수를 사용하여 초기화하는 방법
  - ▣ `char str[6] = "Hello"; // 널문자를 자동으로 저장해줌`
- 만약 배열을 크기를 지정하지 않으면 컴파일러가 자동으로 배열의 크기를 초기화 값에 맞추어 설정
  - ▣ `char str[] = "C Bible"; // 배열의 크기는 8이 된다.`
- 문자 배열선언시 널문자를 포함하여 배열 크기를 지정해야 함



# 문자열 입력

16

- 문자열 입력 : 형식지정자 %s 이용, scanf함수가 널문자를 자동으로 저장해줌, 배열명 앞에 &를 붙이지 않음
- 문자열 출력 : 형식지정자 %s 이용, 문자 배열의 시작주소를 전달해야함, 널문자가 나올때까지 출력, 배열이름 str의 앞에는 & 연산자를 붙이지 않음

```
char str[10];
```

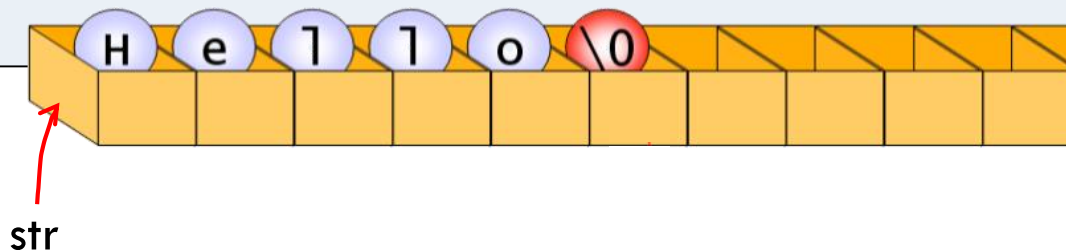
```
printf("문자열을 입력하시오:");
```

```
scanf("%s", str);
```

```
printf("%s\n", str);
```

문자열을 입력하시오: Hello <엔터/>  
Hello

// &str -> 오류  
// Hello 출력







# 예제 1 : 널문자 출력

17

```
#include <stdio.h>
int main(void)
{
    char str[] = "Good morning!";
    printf("배열 str의 크기: %d \n", sizeof(str));
    printf("널 문자 문자형 출력: %c \n", str[13]);
    printf("널 문자 정수형 출력: %d \n", str[13]);

    str[12] = '?';
    printf("문자열 출력: %s \n", str);
    return 0;
}
```

배열 str의 크기: 14  
널 문자 문자형 출력:  
널 문자 정수형 출력: 0  
문자열 출력: Good morning?



# 널(NULL) 문자와 공백 문자의 비교

18

- 널 문자를 %c를 이용해서 출력 시 아무것도 출력되지 않으나 그렇다고 해서 널 문자가 공백 문자는 아님
- 널 문자의 아스키 코드 값은 0이고, 공백 문자의 아스키 코드 값은 32임
- 널 문자는 모니터 출력에서 의미를 갖지 않고 아무것도 출력이 되지 않을 뿐임

```
int main(void)
{
    char nu = '\0';    // 널 문자 저장
    char sp = ' ';     // 공백 문자 저장
    printf("%d %d", nu, sp);    // 0과 32 출력
    return 0;
}
```



## 예제 2 : 문자열 입력

19

```
#include <stdio.h>
int main(void)
{
    char str[50];
    int idx = 0;
    printf("문자열 입력: ");
    scanf("%s", str);
    printf("입력 받은 문자열: %s\n", str);
    printf("문자 단위 출력: ");
    while (str[idx] != '\0')
    {
        printf("%c", str[idx]);
        idx++;
    }
    printf("\n");
    return 0;
}
```

문자열 입력: Simple  
입력 받은 문자열: Simple  
문자 단위 출력: Simple



## 예제 2 : 문자열 입력

20

```
#include <stdio.h>
int main(void)
{
    char str[50];
    int idx = 0;
    printf("문자열 입력: ");
    scanf("%s", str);
    printf("입력 받은 문자열: %s \n", str);
    printf("문자 단위 출력: ");
    while (str[idx] != '\0')
    {
        printf("%c", str[idx]);
        idx++;
    }
    printf("\n");
    return 0;
}
```

문자열 입력: He is my friend  
입력 받은 문자열: He  
문자 단위 출력: He

*scanf 함수는 공백을 기준으로 데이터의 수를 구분한다. 따라서 공백을 포함하는 문자열을 한번의 scanf 함수호출을 통해서 읽어 들이지는 못한다.*



## 예제 3 : 너문자의 역할

21

```
#include <stdio.h>
int main(void)
{
    char str[50];
    int idx = 0;
    printf("문자열 입력: ");
    scanf("%s", str);
    printf("입력 받은 문자열: %s\n", str);
    printf("문자 단위 출력: ");
    while (str[idx] != '\0')
    {
        printf("%c", str[idx]);
        idx++;
    }
    printf("\n");
    return 0;
}
```

string: I like C programming  
string: I like C  
string: I like  
string: I



# 예제 1

22

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char str1[6] = "Seoul";
```

```
    char str2[3] = { 'i', 's', '\0' };           // char str2[3]="is";
```

```
    char str3[] = "the capital city of Korea.";
```

```
    printf("%s %s %s\n", str1, str2, str3);
```

```
}
```

Seoul is the capital city of Korea.



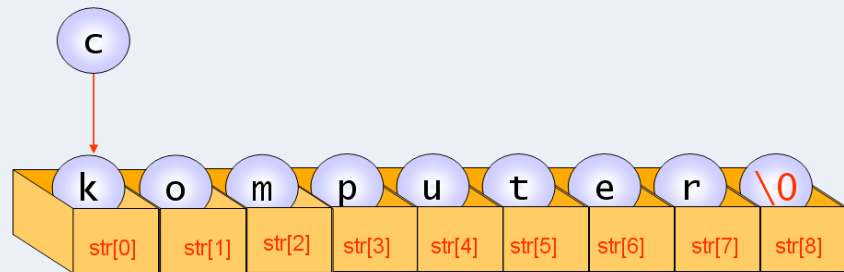
## 예제 2

23

```
#include <stdio.h>
int main(void)
{
    char str[] = "komputer";
    printf("%s\n", str);
    str[0] = 'c';
    printf("%s\n", str);

    return 0;
}
```

komputer  
computer





## 예제 3

24

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char str[30] = "A barking dog never bites";
```

```
    int i = 0;
```

```
    while( str[i] != '\0' )
```

```
        i++;
```

```
    printf("문자열 %s의 길이는 %d입니다.\n", str, i);
```

```
    return 0;
```

```
}
```

문자열 A barking dog never bites의 길이는 25입니다.





25

- B, 'B', "B"의 차이점을 설명하라.



## 실습과제 2

26

- 키보드로부터 문자를 입력 받아 화면에 출력해주는 프로그램을 작성하시오. 'q' 또는 'Q'를 입력하면 종료한다.
- 힌트: 무한 루프와 break문을 이용하라.
- 힌트: 문자를 2번 이상 반복해서 입력 받을 때는 `scanf(" %c", &ch);` 문장에서 `%c` 앞에 공백 넣을 것

문자입력 : a<엔터>

입력된 문자는 a

문자입력: 1<엔터>

입력된 문자는 1

문자입력: C<엔터>

입력된 문자는 C

문자입력: q<엔터>

- 문자열을 입력 받아 모든 문자를 대문자로 변환하여 출력 하시오.
- 힌트: 예제3번 처럼 널(NULL)문자를 만날 때까지 반복하면서 소문자를 대문자로 변환 하시오. 대문자 = 소문자 - ( 'a'-'A' )

문자열입력 : *hello*<엔터>  
대문자로 변환결과 : *HELLO*



# 11장 정리문제

28

- 1) 배열이란 무엇인가?
- 2) 배열이 왜 필요한지 구체적인 예를 들어 설명하십시오.
- 3) 정수형 변수 10개를 선언하는 것과 정수배열 10개를 선언하는 것의 차이는?
- 4) 널문자의 용도는 무엇인가?
- 5) 널문자는 누가 저장해 주는가?
- 6) 정수 5개를 입력 받아 최대값을 구하는 코드를 작성하라
- 7) 정수 5개를 입력 받아 최소값을 구하는 코드를 작성하라
- 8) `strlen`, `strcpy`, `strcat` 함수를 조사하고 같은 기능을 `main` 함수안에 구현하라.



# 과제제출방법

29

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```