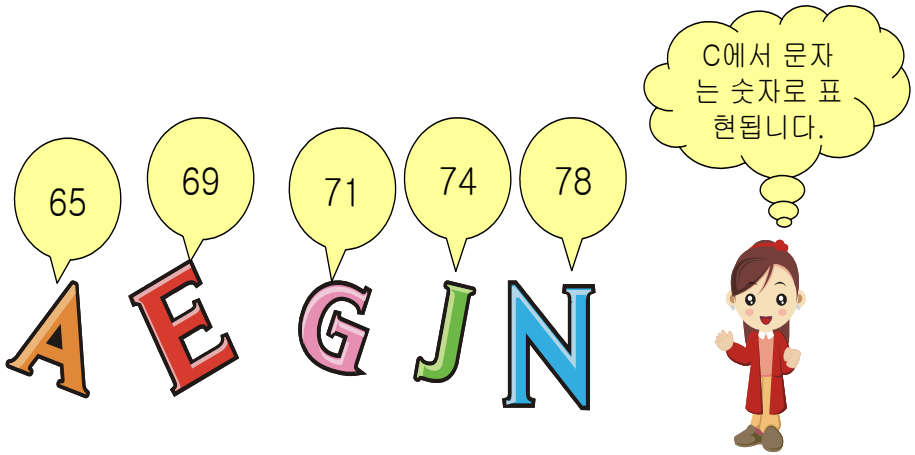
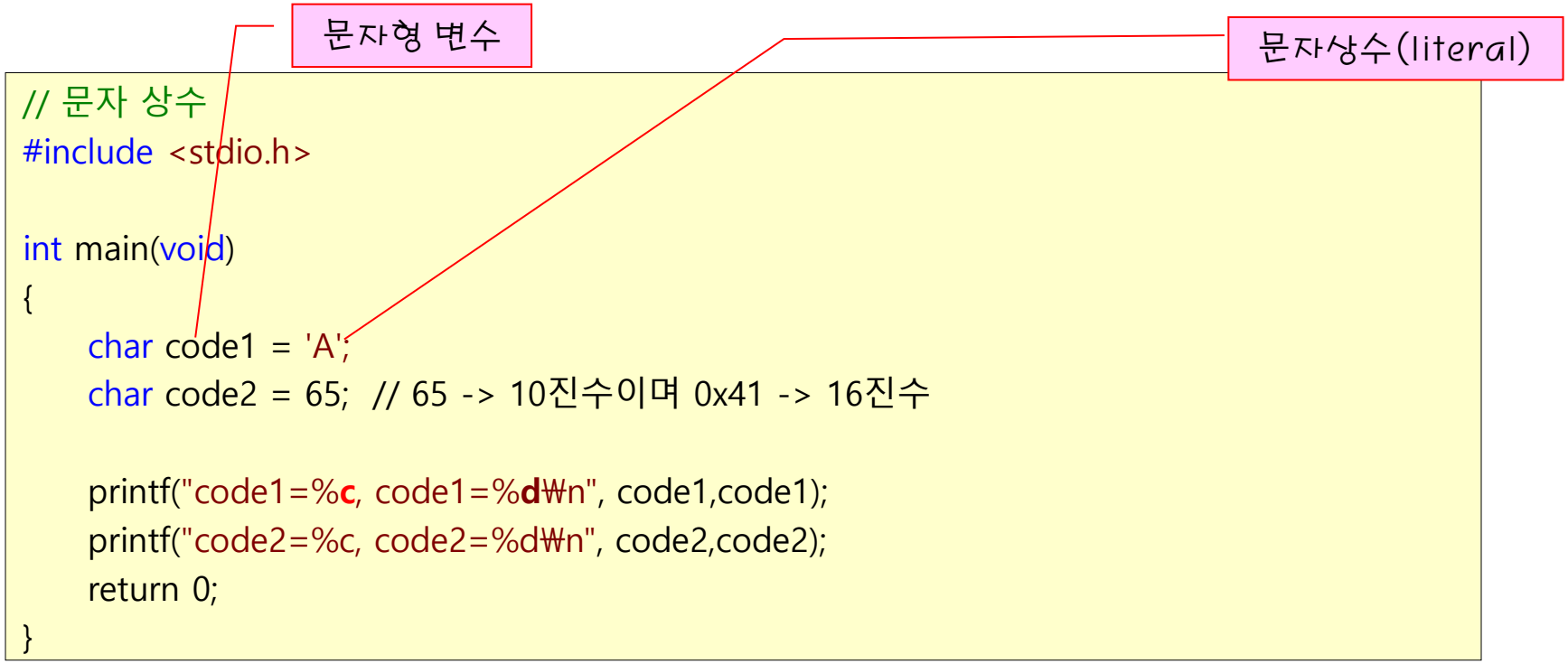
문자와 문자열 Part 1

문자표현방법

- 컴퓨터에서는 각 문자에 특정 숫자를 할당
- 아스키코드(ASCII code): 표준적인 8비트 문자코드
  - 0에서 127까지의 숫자를 이용하여 문자표현
- 유니코드(unicode): 표준적인 16비트 문자코드
  - 전세계의 모든 문자를 일관되게 표현하고 다룰 수 있도록 설계



문자 변수와 문자 상수(literal)




```c
// 문자 상수
#include <stdio.h>

int main(void)
{
	char code1 = 'A';
	char code2 = 65;  // 65 -> 10진수이며 0x41 -> 16진수

	printf("code1=%c, code1=%d₩n", code1,code1);
	printf("code2=%c, code2=%d₩n", code2,code2);
	return 0;
}
```

```
code1=A, code1=65
code2=A, code2=65
```

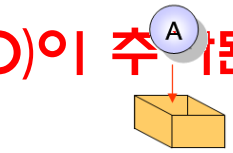
아스키 코드 출력

```c
// 아스키 코드 출력
#include <stdio.h>
int main(void)
{
	unsigned char code;

	for(code = 32; code < 128; code++)
	{
		printf("아스키 코드 %d은 %c입니다.₩n", code, code);
	}
	return 0;
}
```

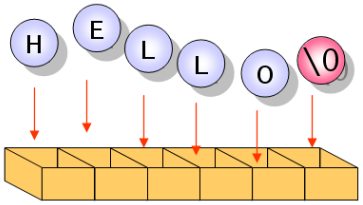
```
아스키 코드 32은 입니다.
아스키 코드 33은 !입니다.
...
아스키 코드 65은 A입니다.
아스키 코드 66은 B입니다.
...
아스키 코드 97은 a입니다.
아스키 코드 98은 b입니다.
...
아스키 코드 126은 ~입니다.
아스키 코드 127은 □입니다.
```

문자열(string) 표현 방법

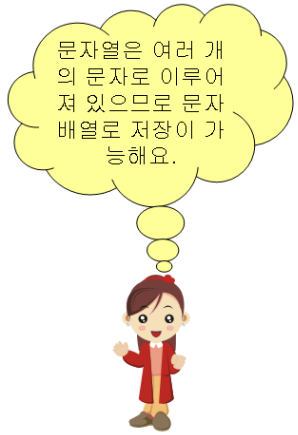
- 문자열(string): 1개 이상의 문자가 순차적으로 나열된것
  - 인용기호 " " 를 사용하여 표현한다
  - "A"
  - "Hello World!"
  - "변수 score의 값은 %d입니다“
- 문자열 상수(string constant)
  - 배열의 이름처럼 포인터로 취급된다
  - 자동으로 맨 마지막에 NULL(즉 \0)이 추가된다
  - "Hello World"
  - "Hong"
  - "guest123"
- 문자열 변수
  - Char형 데이터의 1차원 배열



하나의 문자는 char형 변수로 저장

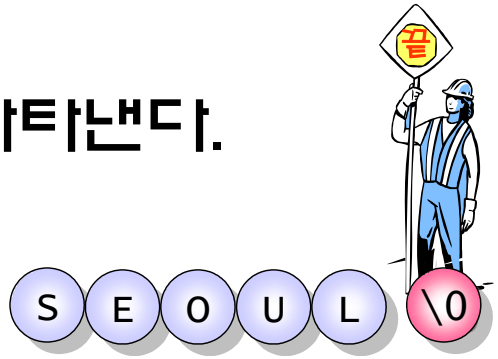


문자열은 char형 배열로 저장

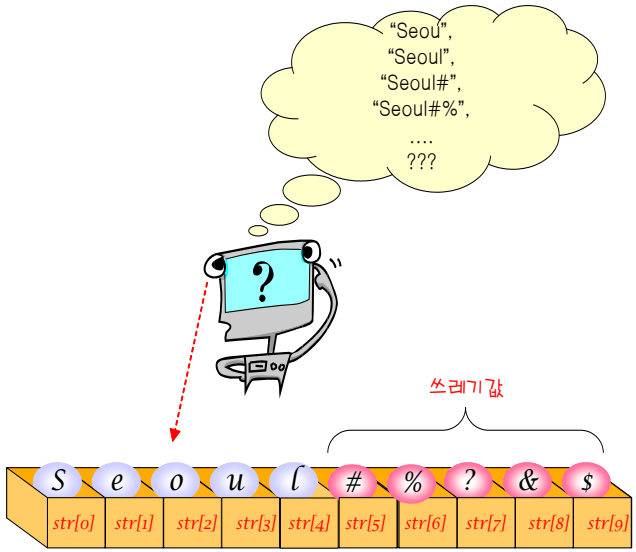


NULL 문자

- NULL 문자: 문자열의 끝을 나타낸다.



- 문자열은 어디서 종료되는지 알수가 없으므로 표시를 해주어야 한다.



문자 배열의 초기화

- 문자 배열 원소들을 중괄호를 사용하여 표기
  - char str[6] = { 'H', 'e', 'l', 'l', 'o', '\0' };
- 문자열 상수를 사용하여 초기화
  - char str[6] = "Hello";·
- 배열을 크기를 지정하지 않으면 컴파일러가 자동으로 배열의 크기를 초기화 값에 맞추어 설정
  - char str[] = "C Bible"; // 배열의 크기는 7이 된다.

문자 배열에 문자를 저장

- 각각의 문자 배열 원소에 원하는 문자를 개별적으로 대입하는 방법
  - str[0] = 'W';
  - str[1] = 'o';
  - str[2] = 'r';
  - str[3] = 'l';
  - str[4] = 'd';
  - str[5] = '\0';

- stdio.h에 기술되어 있는 문자열 처리 라이브러리인 strcpy()를 사용하여 문자열을 문자 배열에 복사
  - strcpy(str, "World");

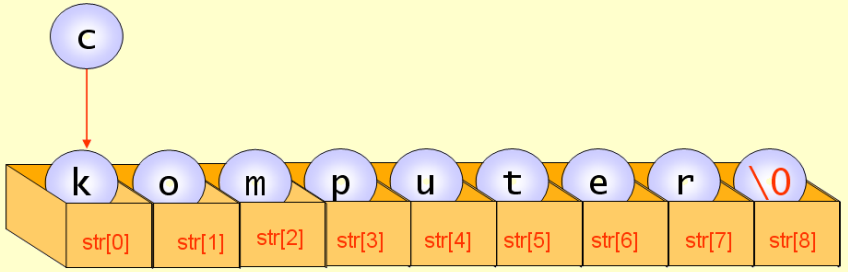
예제 #1

```c
#include <stdio.h>

int main(void)
{
	char str1[6] = "Seoul"
	char str2[3] = { 'i', 's' };
	char str3[] = "the capital city of Korea."

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

```
Seoul is the capital city of Korea.
```

예제 #2

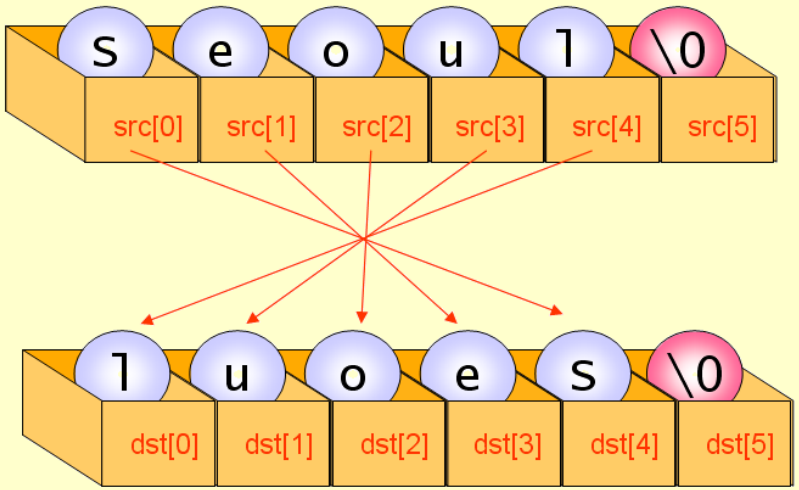
```
#include <stdio.h>

int main(void)
{
	char str[] = "komputer";
	int i;

	for(i=0;i<8;i++)
		printf("%c ", str[i]);

	str[0] = 'c';
	printf("₩n");

	for(i=0;i<8;i++)
		printf("%c ", str[i]);
	return 0
```




```
k o m p u t e r
c o m p u t e r
```

문자열 역순 예제

```
#include <stdio.h>

int main(void)
{
	char src[] = "Seoul";
	char dst[6];
	int i;

	printf("원래 문자열=%s₩n", src);

	i = 0;
	while(src[i] != '₩0')
	{
		dst[i] = src[4 - i];
		i++;
	}
	dst[i] = '₩0';

	printf("역순 문자열=%s₩n", dst);
	return 0;
}
```




```
원래 문자열=Seoul
역순 문자열=luoeS
```

문자열 길이 계산 예제

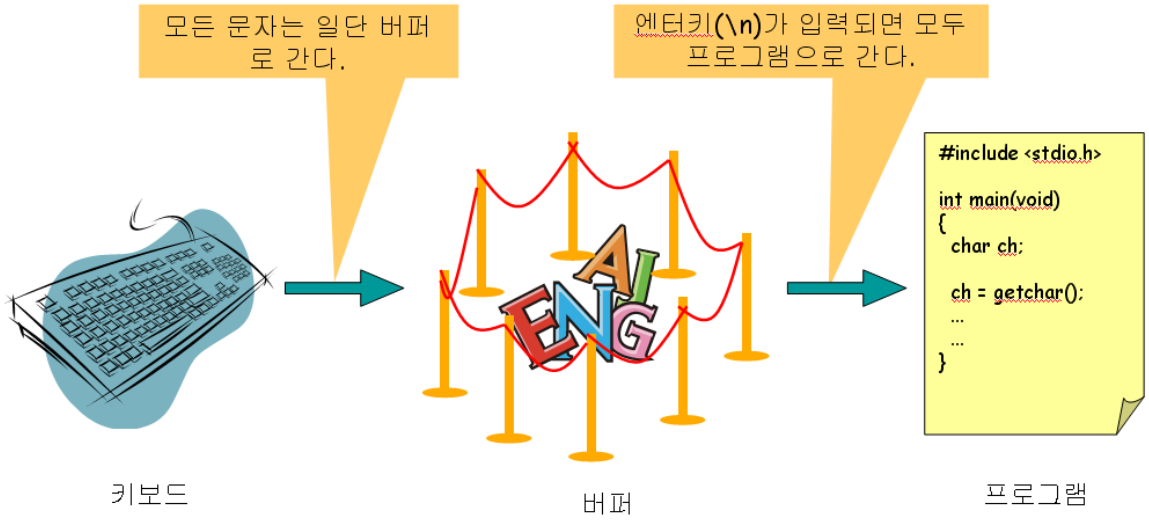
```c
// 문자열의 길이를 구하는 프로그램
#include <stdio.h>

int main(void)
{
	char str[30] = "C language is easy";
	int i = 0;

	while(str[i] != 0)
		i++;

	printf("문자열₩"%s₩"의 길이는 %d입니다.₩n", str, i);

	return 0;
}
```

```
문자열 "C language is easy"의 길이는 18입니다.
```

- 스트링의 길이를 구하는 라이브러리 함수 strlen(cont char *s)

문자 입출력 라이브러리

입출력 함수	설명
int getchar(void)	하나의 문자를 읽어서 반환한다.
void putchar(int c)	변수 c에 저장된 문자를 출력한다.
int getch(void)	하나의 문자를 읽어서 반환한다(버퍼를 사용하지 않음).
void putch(int c)	변수 c에 저장된 문자를 출력한다(버퍼를 사용하지 않음).
scanf("%c", &c)	하나의 문자를 읽어서 변수 c에 저장한다.
printf("%c", c);	변수 c에 저장된 문자를 출력한다.



getchar(), putchar()

```c
// getchar()의 사용
#include <stdio.h>

int main(void)
{
	int ch;				// 정수형에 주의

	while(1)
	{
		ch = getchar();		// 문자를 입력받는다.
		if( ch == 'q' ) break;
		putchar(ch);
	}
	return 0;
}
```

```
A   → getchar
A   → putchar
B
B
q
```

- 입력되는 키는 버퍼에 저장되며, Enter 키를 누르면 전달
- 입력된 문자는 화면에 출력

getch(), putch()

```
// getch()의 사용
#include <conio.h>

int main(void)
{
      int ch;                 // 정수형에 주의

      while(1)
      {
            ch = getch();           // 문자를 입력받는다.
            if( ch == 'q' ) break;
            putch(ch);
      }
      return 0;
}
```

버퍼를 사용하지 않는다

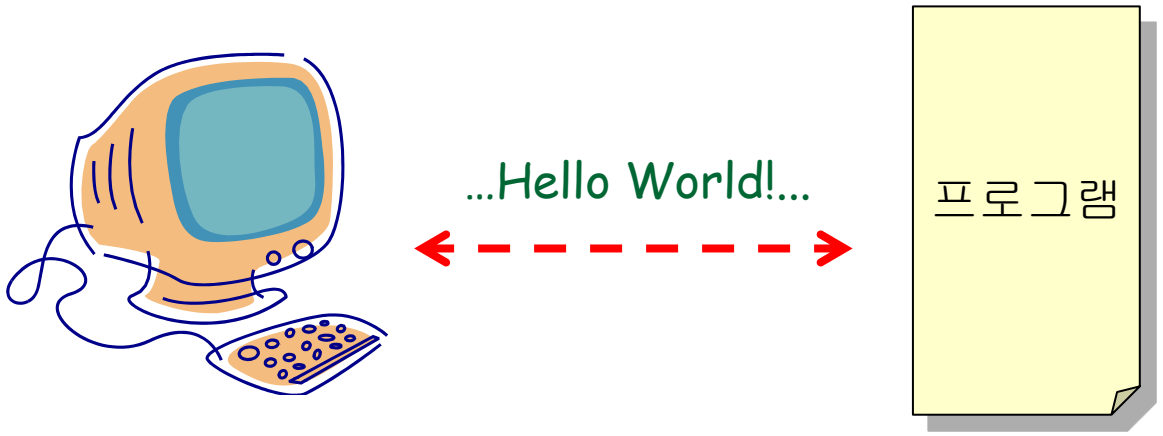
```
ABCDEFGH
```

getch(), getche(), getchar()

	헤더파일	버퍼사용여부	에코여부	응답성	문자 수정 여부
getchar()	<stdio.h>	사용함 (엔터키를 눌러입력됨)	에코	줄단위	가능
getch()	<conio.h>	사용하지 않음	에 코 하 지 않음	문자단위	불가능
getche()	<conio.h>	사용하지 않음	에코	문자단위	불가능

문자열 입출력 라이브러리 함수

입출력 함수	설명
int scanf("%s", s)	문자열을 읽어서 문자배열 s[]에 저장
int printf("%s", s)	배열 s[]에 저장되어 있는 문자열을 출력한다.
char *gets(char *s)	한 줄의 문자열을 읽어서 문자 배열 s[]에 저장한다.
int puts(const char *s)	배열 s[]에 저장되어 있는 한 줄의 문자열을 출력한다.



scanf(), printf() 문자열 입출력

- scanf()의 사용법
  - char str[10];
  - scanf("%s", str);

- scanf()는 한 번에 두 개 이상의 문자열을 입력받을 수 있다.
  - char s1[10];
  - char s2[10];
  - char s3[10];
  - scanf("%s %s %s", s1,s2,s3);
  - // 사용자가 one two three와 같이 입력하면 s1에는 one이, s2에는 two가, s3에는 three가 할당된다.

gets()와 puts() 문자열 입출력

```
char *gets(char *buffer);
int    puts(const char *str);
```

- gets()
  - 표준 입력으로부터 엔터 키가 나올 때까지 한 줄의 라인을 입력
  - 문자열에 줄바꿈 문자('\n')는 포함되지 않으며 자동으로 NULL 문자('\0')를 추가한다.
  - 입력된 문자열은 buffer가 가리키는 주소에 저장된다.
- puts()
  - str이 가리키는 문자열을 받아서 화면에 출력
  - NULL 문자('\0')는 줄바꿈 문자('\n')로 변경

```
char *menu = "파일열기: open, 파일닫기: close";
puts("메뉴에서 하나를 선택하시오.");
puts(str);
```

예제

```c
#include <stdio.h>

int main( void )
{
    char buffer[21]; // 20개의 문자와 '\0'을 저장할 수 있다.

    printf("문자열을 입력하시오.\n");
    gets( buffer );

    printf("입력된 라인은 다음과 같습니다.\n");
    puts(buffer);
    return 0;
}
```

```
문자열을 입력하시오.
Hello!
입력된 라인은 다음과 같습니다.
Hello!
```