12장 문자열

한빛미디어 Hanbit Media, Inc.

12-1 문자열과 포인터

❖ 문자열 상수 구현 방법

문자열 상수가 주소란 증거 소스 코드 예제12-1.c

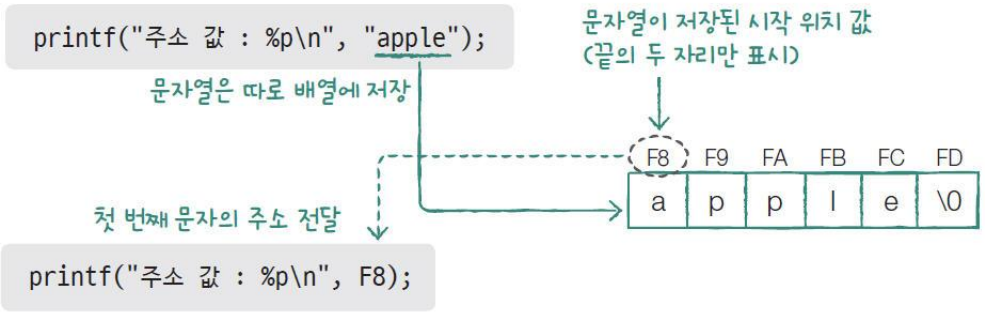
```c
#include <stdio.h>

int main(void)
{
    printf("apple이 저장된 시작 주소 값 : %p\n", "apple");
    printf("두 번째 문자의 주소 값 : %p\n", "apple" + 1);      // 주소 값 출력
    printf("첫 번째 문자 : %c\n", *"apple");                   // 간접 참조 연산
    printf("두 번째 문자 : %c\n", *("apple" + 1));             // 포인터 연산식
    printf("배열로 표현한 세 번째 문자 : %c\n", "apple"[2]);   // 배열 표현식

    return 0;
}
```

실행결과

```
apple이 저장된 시작 주소 값 : 00EA58F8
두 번째 문자의 주소 값 : 00EA58F9
첫 번째 문자 : a
두 번째 문자 : p
배열로 표현한 세 번째 문자 : p
```

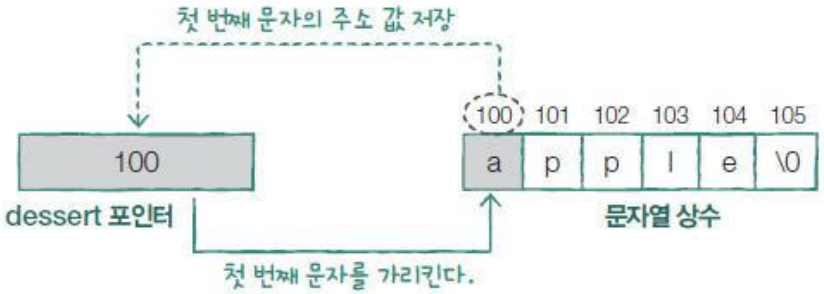


12-1 문자열과 포인터

❖ char 포인터로 문자열 사용

포인터로 문자열을 사용하는 방법 소스 코드 예제12-2.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char *dessert = "apple";
06
07      printf("오늘 후식은 %s입니다.\n", dessert);    // 문자열 출력
08      dessert = "banana";                            // 새로운 문자열 대입
09      printf("내일 후식은 %s입니다.\n", dessert);    // 바뀐 문자열 출력
10
11      return 0;
12  }
```



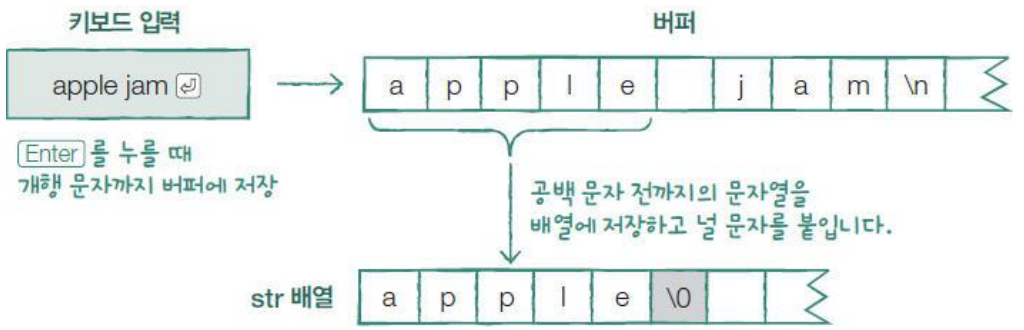
실행결과

```
오늘 후식은 apple입니다.
내일 후식은 banana입니다.
```

❖ scanf 함수를 사용한 문자열 입력

scanf 함수를 사용한 문자열 입력 소스 코드 예제12-3.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char str[80];
06
07      printf("문자열 입력 : ");
08      scanf("%s", str);                    // %s를 사용하고 배열명을 준다.
09      printf("첫 번째 단어 : %s\n", str);      // 배열에 입력된 문자열 출력
10      scanf("%s", str);
11      printf("버퍼에 남아 있는 두 번째 단어 : %s\n", str);
12
13      return 0;
14  }
```



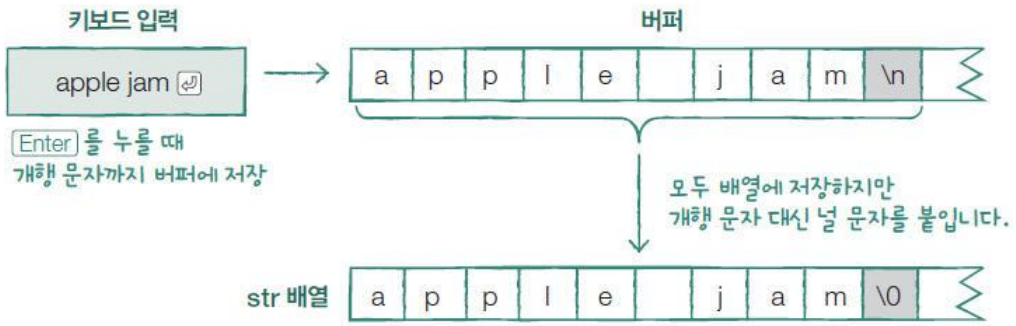
실행결과

```
문자열 입력 : apple jam
첫 번째 단어 : apple
버퍼에 남아 있는 두 번째 단어 : jam
```

❖ gets 함수를 사용한 문자열 입력

gets 함수로 한 줄의 문자열 입력 소스 코드 예제12-4.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char str[80];
06
07      printf("공백이 포함된 문자열 입력 : ");
08      gets(str);              // 배열명으로 주고 함수 호출
09      printf("입력한 문자열은 %s입니다.", str);
10
11      return 0;
12  }
```



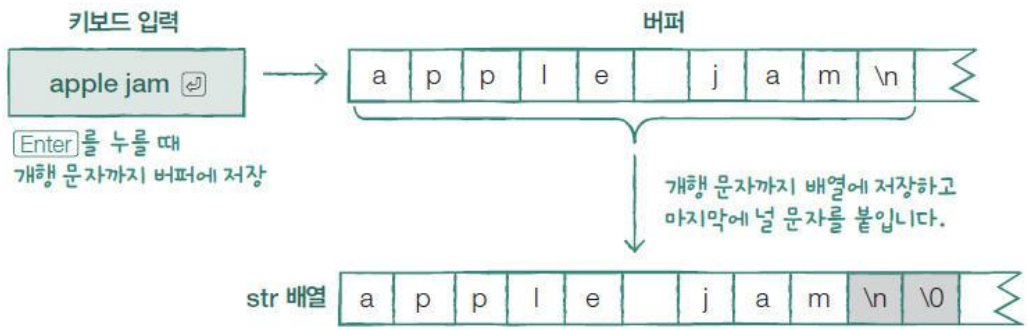
실행결과

```
공백이 포함된 문자열 입력 : apple jam
입력한 문자열은 apple jam입니다.
```

❖ fgets 함수를 사용한 문자열 입력

fgets 함수의 문자열 입력 방법 소스 코드 예제12-5.c

```
01  #include <stdio.h>
02  // 나중에 입력할 공간입니다.
03
04  int main(void)
05  {
06      char str[80];
07
08      printf("공백이 포함된 문자열 입력 : ");
09      fgets(str, sizeof(str), stdin);
10      // 나중에 입력할 공간입니다.
11      printf("입력된 문자열은 %s입니다\n", str);
12
13      return 0;
14  }
```




```
str[strlen(str) - 1] = '\0'
```

널 문자 이전까지 문자 수 10개

9는 배열에서 개행 문자가 저장된 곳의 위치

개행 문자를 널 문자로 바꿈

실행결과

```
공백이 포함된 문자열 입력 : apple jam
입력된 문자열은 apple jam
입니다
```

❖ 표준 입력 함수의 버퍼 공유 문제

개행 문자로 인해 gets 함수가 입력을 못하는 경우 소스 코드 예제12-6.c

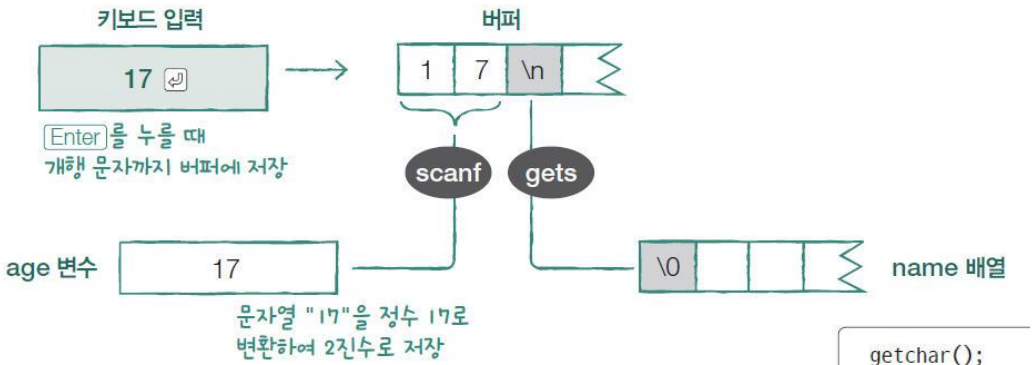
```c
#include <stdio.h>

int main(void)
{
    int age;
    char name[20];

    printf("나이 입력 : ");
    scanf("%d", &age);                  // scanf 함수로 나이 입력

    printf("이름 입력 : ");
    gets(name);                         // gets 함수로 이름 입력
    printf("나이 : %d, 이름 : %s\n", age, name);

    return 0;
}
```



실행결과

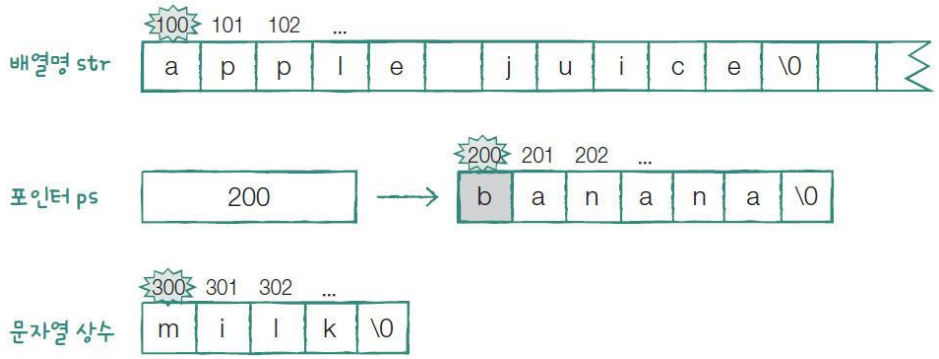
```
나이 입력 : 17
이름 입력 : 나이 : 17, 이름 :
```

12-1 문자열과 포인터

❖ 문자열을 출력하는 puts, fputs 함수

문자열을 출력하는 puts와 fputs 함수 소스 코드 예제12-7.c

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      char str[80] = "apple juice";
06      char *ps = "banana";
07
08      puts(str);          // apple juice 출력하고 줄 바꿈
09      fputs(ps, stdout);// banana만 출력
10      puts("milk");       // banana에 이어 milk 출력
11
12      return 0;
13  }
```



실행결과

```
apple juice
bananamilk
```

마무리

키워드로 끝내는 핵심 포인트

- **문자열**은 첫 번째 문자가 저장된 메모리의 주소로 바뀐다.
- **scanf** 함수는 중간에 공백이 포함된 문자열을 입력할 수 없다.
- **gets** 함수는 한 줄의 데이터를 char 배열에 저장한다.
- **fgets** 함수는 배열의 크기를 검사하는 문자열 입력 함수다.

마무리

표로 정리하는 핵심 포인트

표 12-1 문자열을 저장하는 배열과 포인터의 차이

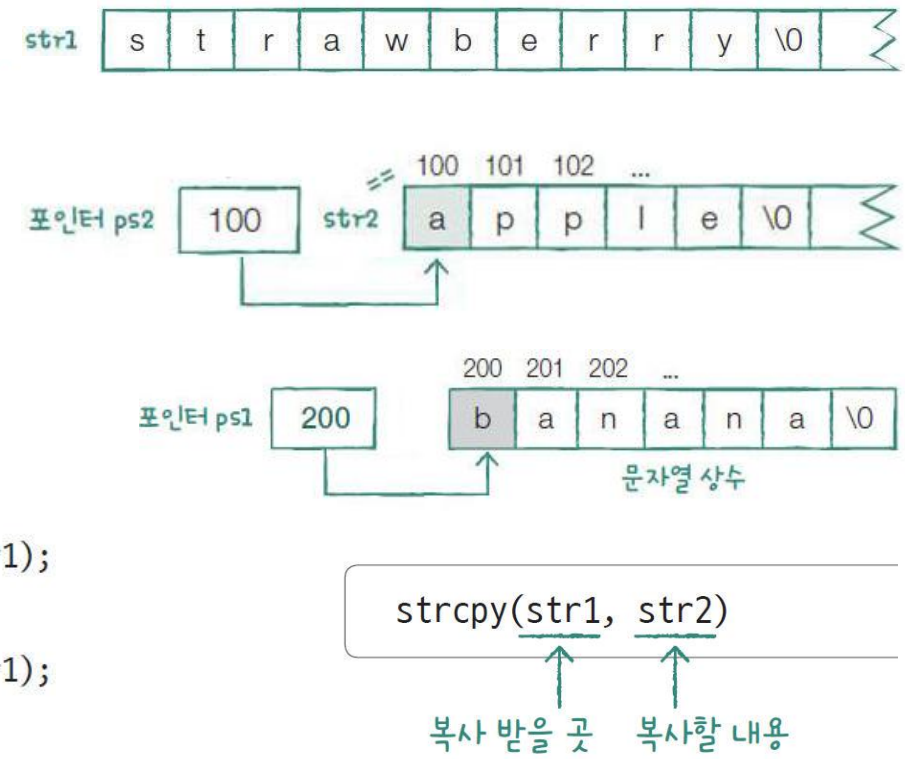
구분	char 포인터	char 배열
초기화	`char *pc = "mango";`	`char str[80] = "mango";`
대입	`pc = "banana";`	`strcpy(str, "banana");`
크기	`sizeof(pc)` ➡ 4바이트	`sizeof(str)` ➡ 80바이트
수정	`pc[0] = 't';` (X)	`str[0] = 't';` (O)
입력	`scanf("%s", pc);` (X)	`scanf("%s", str);` (O)

12- 2 문자열 연산 함수

❖ 문자열을 대입하는 strcpy 함수 (1/2)

strcpy 함수의 사용법 소스 코드 예제12-8.c

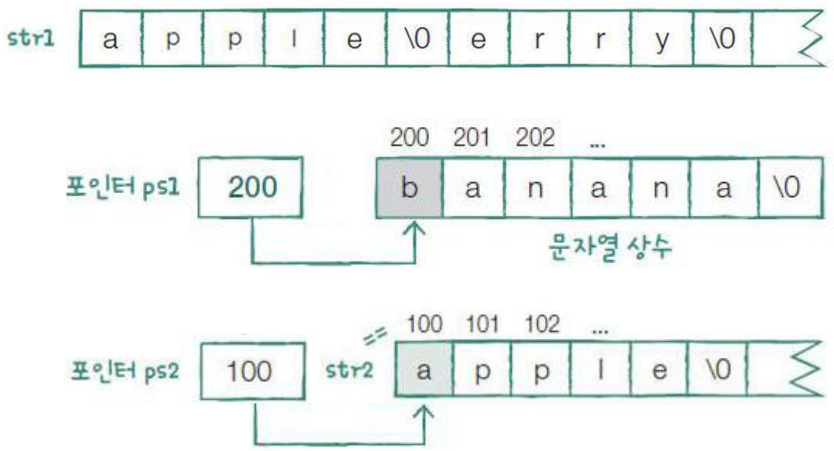
```
01  #include <stdio.h>
02  #include <string.h>
03
04  int main(void)
05  {
06      char str1[80] = "strawberry";
07      char str2[80] = "apple";
08      char *ps1 = "banana";
09      char *ps2 = str2;
10
11      printf("최초 문자열 : %s\n", str1);
12      strcpy(str1, str2);
13      printf("바뀐 문자열 : %s\n", str1);
14
```



❖ 문자열을 대입하는 strcpy 함수 (2/2)

strcpy 함수의 사용법 소스 코드 예제12-8.c

```
15      strcpy(str1, ps1);
16      printf("바뀐 문자열 : %s\n", str1);
17
18      strcpy(str1, ps2);
19      printf("바뀐 문자열 : %s\n", str1);
20
21      strcpy(str1, "banana");
22      printf("바뀐 문자열 : %s\n", str1);
23
24      return 0;
25  }
```



실행결과

```
최초 문자열 : strawberry
바뀐 문자열 : apple
바뀐 문자열 : banana
바뀐 문자열 : apple
바뀐 문자열 : banana
```

12-2 문자열 연산 함수

❖ 원하는 개수의 문자만을 복사하는 strncpy 함수

strncpy 함수를 사용한 문자열 복사 소스 코드 예제12-9.c

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

int main(void)
{
    char str[20] = "mango tree";    // 배열 초기화

    strncpy(str, "apple-pie", 5);   // "apple-pie"에서 다섯 문자만 복사

    printf("%s\n", str);            // 복사 받은 문자열 출력

    return 0;
}
```

```c
strncpy(str, "apple-pie", 5);
```

- str: 복사 받을 배열명
- "apple-pie": 복사할 문자열
- 5: 복사할 문자 수

실행결과

```
apple tree
```

12-2 문자열 연산 함수

❖ 문자열을 붙이는 strcat, strncat 함수

strcat, strncat 함수를 사용한 문자열 붙이기 소스 코드 예제12-10.c

```
01  #include <stdio.h>
02  #include <string.h>
03
04  int main(void)
05  {
06      char str[80] = "straw";          // 문자열 초기화
07
08      strcat(str, "berry");            // str 배열에 문자열 붙이기
09      printf("%s\n", str);
10      strncat(str, "piece", 3);        // str 배열에 3개의 문자 붙이기
11      printf("%s\n", str);
12
13      return 0;
14  }
```

실행결과

```
strawberry
strawberrypie
```

12-2 문자열 연산 함수

❖ 문자열 길이를 계산하는 strlen 함수

두 문자열 중 길이가 긴 단어 출력 소스 코드 예제12-11.c

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

int main(void)
{
    char str1[80], str2[80];                // 두 문자열을 입력할 배열
    char *resp;                             // 문자열이 긴 배열을 선택할 포인터

    printf("2개의 과일 이름 입력 : ");
    scanf("%s%s", str1, str2);              // 2개의 문자열 입력
    if (strlen(str1) > strlen(str2))        // 배열에 입력된 문자열의 길이 비교
        resp = str1;                        // 첫 번째 배열이 긴 경우 선택
    else
        resp = str2;                        // 두 번째 배열이 긴 경우 선택
    printf("이름이 긴 과일은 : %s\n", resp);  // 선택된 배열의 문자열 출력

    return 0;
}
```

strlen(str)

크기를 확인할 배열명

실행결과

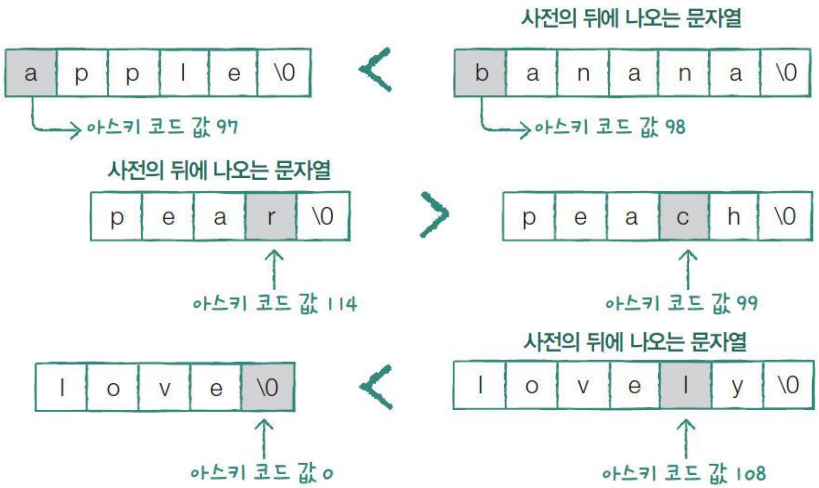
```
2개의 과일 이름 입력 : banana strawberry
이름이 긴 과일은 : strawberry
```

12-2 문자열 연산 함수

❖ 문자열을 비교하는 strcmp, strncmp 함수

strcmp, strncmp 함수를 사용한 문자열 비교 소스 코드 예제12-12.c

```
01 #include <stdio.h>
02 #include <string.h>
03
04 int main(void)
05 {
06     char str1[80] = "pear";
07     char str2[80] = "peach";
08
09     printf("사전에 나중에 나오는 과일 이름 : ");
10     if (strcmp(str1, str2) > 0)     // str1이 str2보다 사전에 나중에 나오면 1 반환
11         printf("%s\n", str1);       // str1이 str2보다 사전에 먼저 나오면 -1 반환
12     else                            // str1과 str2가 같은 문자열이면 0 반환
13         printf("%s\n", str2);
14
15     return 0;
16 }
```



실행결과

```
사전에 나중에 나오는 과일 이름 : pear
```

마무리

키워드로 끝내는 핵심 포인트

- **strcpy** 함수에서 문자열을 복사 받는 곳은 배열이어야 한다.
- **strcat** 함수로 문자열을 최초로 붙일 때는 초기화를 해야 한다.
- **strlen** 함수로 배열에 저장된 문자열의 길이를 알 수 있다.
- **strcmp** 함수로 문자열의 사전 등록 순서를 확인할 수 있다.

마무리

표로 정리하는 핵심 포인트

표 12-2 기본적인 문자열 연산 함수

연산 기능	사용 방법	실행결과
대입	`strcpy(str1, str2);`	문자열 str2를 str1에 복사
길이 계산	`strlen(str);`	문자열 str의 길이(문자 수)를 구해 반환
붙이기	`strcat(str1, str2);`	문자열 str2를 str1문자열 뒤에 이어 붙임
비교	`strcmp(str1, str2);`	문자열 str1이 str2보다 크면 1 반환 문자열 str1이 str2보다 작으면 –1 반환 str1과 str2가 같은 문자열이면 0 반환