

- `char *변수이름 = "문자열";`

다음 내용을 소스 코드 편집 창에 입력한 뒤 실행해보세요.

string\_pointer.c

```
#include <stdio.h>

int main()
{
    char c1 = 'a';          // 변수에 문자 'a' 저장
    char *s1 = "Hello";     // 포인터에 문자열 "Hello"의 주소 저장

    printf("%c\n", c1);     // a: %c로 문자 출력
    printf("%s\n", s1);     // Hello: %s로 문자열 출력

    return 0;
}
```

↑ \*기호 없이 해당 변수를 넣는다.

실행 결과

a  
Hello

문자(char)는 'a' 처럼 글자가 하나만 있는 상태를 뜻하고 문자열(char \*)은 "Hello" 처럼 글자 여러 개가 계속 이어진 상태를 뜻합니다. 즉, 문자는 1바이트 크기의 char에 저장할 수 있지만 문자열은 크기가 1바이트를 넘어서므로 char에 저장할 수 없습니다. 따라서 문자열은 변수에 직접 저장하지 않고 포인터를 이용해서 저장합니다.

다음은 문자가 변수 c1에 저장된 상태와 문자열이 변수 s1에 저장된 상태를 나타냅니다.

#### 참고 | 문자열의 메모리 주소 출력

printf로 문자열 대신 문자열이 저장된 메모리 주소를 출력하고 싶다면 서식 지정자로 %p를 사용하면 됩니다.

```
char *s1 = "Hello";

printf("%p\n", s1);    // 00A5585: %p로 문자열이 저장된 메모리 주소 출력
```

← 핵심!

문자열을 사용할 때는 char 포인터에 " "(큰따옴표)로 묶은 문자열만 할당하면 된다는 점만 기억하세요. 그리고 printf로 문자열을 출력할 때는 서식 지정자로 %s를 사용하면 됩니다.

```
printf("%s\n", s1);    // Hello: %s로 문자열 출력
```