컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 11강 – 문자열

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved

학습 목표

문자열

1) 전통적인 C언어에서 문자열을 다루는 방법에 대해서 학습합니다.
2) 다양한 문자열 관련 함수를 익히고 활용합니다.

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열의 개념

1) 문자열을 말 그대로 문자들의 배열입니다.
2) 문자열은 컴퓨터 메모리 구조상에서 마지막에 널(NULL) 값을 포함합니다.

0	1	2	3	4	5	6	7	8	9	10	11
H	E	L	L	O		W	O	R	L	D	₩0

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved

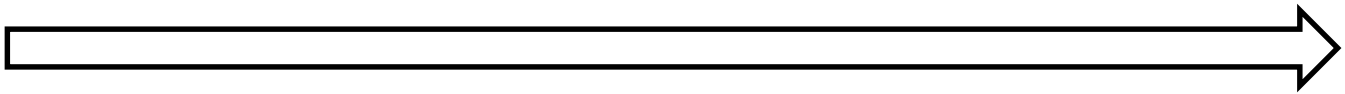
Fast campus

문자열

문자열의 개념

1) 널(NULL) 값은 문자열의 끝을 알리는 목적으로 사용됩니다.
2) printf() 함수를 실행하면 컴퓨터는 내부적으로 NULL을 만날 때까지 출력합니다.

0	1	2	3	4	5	6	7	8	9	10	11
H	E	L	L	O		W	O	R	L	D	₩0



출력: HELLO WORLD

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열과 포인터

1) 문자열 형태로 포인터를 사용하면 포인터에 특정한 문자열의 주소를 넣게 됩니다.
2) 다음 코드는 "Hello World" 문자열을 읽기 전용으로 메모리 공간에 넣은 뒤에 그 위치를 처리합니다.
3) 이러한 문자열을 '문자열 리터럴'이라고 말합니다. 이는 컴파일러가 알아서 메모리 주소를 결정합

```
#include <stdio.h>

int main(void) {
  char *a = "Hello World";
  printf("%s₩n", a);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열과 포인터

1) 포인터로 문자열을 선언했다고 하더라도 기존의 배열처럼 처리할 수 있습니다.

```
#include <stdio.h>

int main(void) {
  char *a = "Hello World";
  printf("%c₩n", a[1]);
  printf("%c₩n", a[4]);
  printf("%c₩n", a[8]);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 입출력 함수

1) 문자열 입출력을 수행합니다.
2) scanf() 함수는 공백을 만날 때까지 입력 받지만 gets() 함수는 공백까지 포함하여 한 줄을 입력 받습니다.

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

int main(void) {
  char a[100];
  gets(a);
  printf("%s₩n", a);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 입출력 함수

1) gets() 함수는 버퍼의 크기를 벗어나도 입력을 받아버립니다.
2) C11 표준부터는 버퍼의 크기를 철저히 지키는 gets_s() 함수가 추가되었습니다.

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void) {
  char a[100];
  gets_s(a, sizeof(a));
  printf("%s₩n", a);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 입출력 함수

1) gets_s()를 이용하는 경우 범위를 넘으면 그 즉시 런타임(Runtime) 오류가 발생하게 됩니다.

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

1) C언어의 문자열처리와 관련해서는 기본적인 문자열 함수를 알고 있는 것이 좋습니다.
2) 나중에 C++을 이용하면 더욱 간편하고 다양한 함수를 사용할 수 있습니다.
3) C언어에서의 문자열 함수는 <string.h> 라이브러리에 포함되어 있습니다.

Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

strlen()	문자열의 길이를 반환합니다.
strcmp()	문자열 1이 문자열 2보다 사전적으로 앞에 있으면 -1, 뒤에 있으면 1을 반환
strcpy()	문자열을 복사합니다.
strcat()	문자열 1에 문자열 2를 더합니다.
strstr()	문자열 1에 문자열 2가 어떻게 포함되어 있는지를 반환합니다.

Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

1) strlen()는 문자열의 길이를 반환합니다.

```
#include <stdio.h>

int main(void) {
  char a[20] = "Dongbin Na";
  printf("문자열의 길이: %d₩n", strlen(a));
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

Fast campus

문자열

문자열 처리를 위한 다양한 함수

1) strcmp()는 문자열 1이 문자열 2보다 사전적으로 앞에 있으면 -1, 뒤에 있으면 1을 반환합니다.

```
#include <stdio.h>

int main(void) {
  char a[20] = "Dongbin Na";
  char b[20] = "Hojoon Seok";
  printf("두 배열의 사전 순 비교: %d₩n", strcmp(a, b));
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

1) strcpy()는 문자열을 복사합니다.
2) C언어에서는 기본적으로 'a = b'와 같은 간단한 방식으로는 문자열 복사가 안 됩니다.

```
#include <stdio.h>

int main(void) {
  char a[20] = "My Name";
  char b[20] = "Dongbin Na";
  strcpy(a, b);
  printf("복사된 문자열: %s₩n", a);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

1) strcat()은 뒤에 있는 문자열을 앞에 있는 문자열에 합칩니다.

```
#include <stdio.h>

int main(void) {
  char a[20] = "My Name is ";
  char b[20] = "Dongbin Na";
  strcat(a, b);
  printf("합쳐진 결과 문자열: %s₩n", a);
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

문자열

문자열 처리를 위한 다양한 함수

1) strstr()은 긴 문자열에서 짧은 문자열을 찾아 그 위치를 반환합니다.
2) 짧은 문자열을 찾은 주소 값 자체를 반환하므로 단순히 출력하도록 하면, 찾은 이후 모든 문자열이 반환됩니다

```
#include <stdio.h>

int main(void) {
  char a[20] = "I like you";
  char b[20] = "like";
  printf("찾은 문자열: %s₩n", strstr(a, b));
  system("pause");
  return 0;
}
```


Copyright FASTCAMPUS Corp. All Rights Reserved

배운 내용 정리하기

문자열

1) C언어에서 문자열은 배열이므로 포인터 형태로 사용할 수 있습니다.
2) C언어에서 문자열 비교, 연산, 탐색 등의 알고리즘의 사용 방법은 각각 함수 형태로 제공됩니다.

FASTCAMPUS
Copyright FASTCAMPUS Corp. All Rights Reserved