

# 배열 초기화 연습

for문을 이용하여 배열의 모든 원소를 반복 : `for (i = 0; i < 배열의크기; i++)`

[배열 aa에 0, 2, 4, ... 를 차례로 대입]

```
int main()
{
    int aa[100], bb[100];
    int i;

    for (i = 0; i < 100; i++)
        aa[i] = i * 2;

    for (i = 0; i < 100; i++)
        bb[i] = aa[99-i];
}
```

[배열 bb에 배열 aa에 저장된 값을 역순으로 대입]

```
int main()
{
    int aa[10];

    for (i = 0; i < 10; i++)
    {
        printf("%d번째 숫자 : ", i+1);
        scanf("%d", &aa[i]);
    }
}
```

[배열의 값을 키보드로 입력]

# 무작위 숫자로 배열 초기화

- rand 함수를 사용해서 무작위 숫자 추출하기
  - rand() 를 쓰면 무작위 0~n 사이의 정수가 나온다.
  - rand() 를 쓰기 위해서는 **#include <stdlib.h>** 구문 필요

[rand() 함수를 사용하기 위해 헤더파일 <stdlib.h>를 추가]

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

int main()
{
    int i;
    int grade[5];

    for(i = 0; i < 5; i++)
        grade[i] = rand() % 100;

    for(i = 0; i < 5; i++)
        printf("grade[%d]=%d\n", i, grade[i]);
}
```

[rand() 함수는 0부터 n까지의 정수를 무작위로 선택해 준다. n은 시스템에 따라 다름]

[100으로 나눈 나머지를 계산하여 0에서 100 사이의 random number 생성]

[random number를 생성하여 배열에 대입]

# [실습] 배열 써 보기

- ▶ **int**형 배열에 1에서 100까지 **random number**를 10개 생성하여 저장한 후, 평균을 계산하여 출력해 보세요.

```
aa[0] : 7  
aa[1] : 49  
aa[2] : 73  
aa[3] : 58  
aa[4] : 30  
aa[5] : 72  
aa[6] : 44  
aa[7] : 78  
aa[8] : 23  
aa[9] : 9  
평균 : 44.30
```

[출력 결과]

# 배열 복사- 모든 데이터 형식

```
int main()
{
    int grade[] = {1, 2, 3, 4, 5};
    int score[5];

    score = grade;    //컴파일 오류
}
```

- ▶ 배열에 다른 배열을 대입하는 것은 불가능

```
int main()
{
    int grade[] = {1, 2, 3, 4, 5};
    int score[5];
    int i;

    for (i = 0; i < 5; i++)
        score[i] = grade[i];
}
```

- ▶ 반복문을 활용하여 원소 하나씩 복사해야 함

# 배열 크기 계산

- ▶ 배열의 크기를 명시적으로 알기 힘든 경우 또는 배열의 크기를 정확하게 계산하고자 하는 경우 사용

```
int main()
{
    int grade[] = {1, 2, 3, 4, 5, 6};
    int i, size;

    count = sizeof(grade) / sizeof(int);

    for(i = 0; i < count; i++)
        printf("%d ", grade[i]);
}
```

[배열 **grade**의 크기 계산]

[반복문 내에서 배열의 크기 활용]

# 마지막 데이터 형식 - 문자열 (string)

- 문자열은 char형 배열을 사용해서 표현함

- ▶ 문자열(string): 연속된 문자들의 집합으로 큰 따옴표로 표시함

예) "Hello World", "안녕하세요.:", "A"

- Q) 문자열은 어떠한 형태로 변수에 저장할 수 있을까?

- ▶ 프로그래밍 언어에 따라 문자열을 다루는 방식이 다름.
- ▶ 대부분의 객체지향언어(Java, C# 등)는 String이라는 객체를 제공함 `String str = "Hello world";`
- ▶ C언어에서는 문자열을 위한 데이터형식을 제공하지 않고 배열(array)을 이용함

!!배열: 동일한 형식의 데이터들을 동시에 저장하기 위한 자료구조



하나의 문자는 char형 변수로 저장



문자열은 char형 배열로 저장

# 문자열과 배열

- ▶ 문자열의 선언

```
char str[7];
```

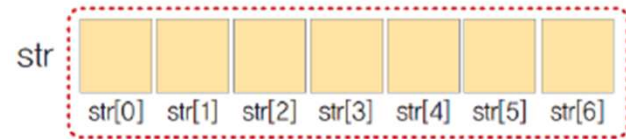


그림 3-25 배열로 표현한 문자열

- ▶ 문자열의 대입

```
strcpy(str, "Basic");
```



그림 3-26 str 배열에 문자열 대입하기

- ▶ 문자열의 선언과 동시에 대입

```
char str[7] = "Basic";
```

- ▶ C에서는 문자열의 끝에 항상 null문자 (\0)를 저장함.

- ▶ 문자열의 끝을 알려주는 역할을 하지만 출력되지는 않는다

- ▶ 문자열의 출력

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

- ▶ 주의) scanf()함수로 문자열을 입력받을 때는 %s 서식을 사용하고, 아래 예와 같이 변수이름 앞에 &를 생략

```
char name[20];  
scanf("%s", name);
```

```
int a;  
scanf("%d", &a);
```

# 문자열 사용 예시

- 문자열 복사 함수 **strcpy** 사용

```
#include <stdio.h>

int main()
{
    char str1[10];
    char str2[10];
    char str3[10] = "CookBook";

    strcpy(str1, "Basic-C");
    strcpy(str2, str3);

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

실행결과

---

```
str1 ==> Basic-C
str2 ==> CookBook
str3 ==> CookBook
```



# 배열 비교

```
int main()
{
    int grade[] = {1, 2, 3, 4, 5};
    int score[] = {1, 2, 3, 4, 6};

    if (grade == score) //배열끼리 비교는 불가능
        ...
    else
        ...
}
```

```
int main()
{
    int grade[] = {1, 2, 3, 4, 5};
    int score[] = {1, 2, 3, 4, 6};
    int i;

    for (i = 0; i < 5; i++)
    {
        if (grade[i] != score[i])
        {
            printf("같지 않음.\n");
            break;
        }
    }
}
```

- ▶ 배열끼리의 비교는 불가능하고, 반복문을 이용하여 배열의 원소를 하나씩 비교해야 함

배열을 사용할 때 반복문(for문)의 활용은 필수!!

# 문자열 관련 함수들

- ▶ 헤더파일 **<string.h>**는 문자열을 쉽게 사용할 수 있도록 도와주는 함수들을 제공
- ▶ **strlen()**, **strcpy\_s()**, **strcat\_s()**, **strcmp()** : 교과서 251-256페이지를 참고하기 바람

```
#include <stdio.h>
#include <string.h>
int main()
{
    char ss[] = "XYZ";
    int len;

    len = strlen(ss);
    printf("문자열 ss의 길이 : %d \n", len);

    char tt[4];
    strcpy_s(tt, 4, ss);
    printf("문자열 tt : %s \n", tt);
}
```

← 헤더파일 **<string.h>** 추가

문자열의 길이를 계산해 준다

tt에 문자열 ss를 복사한다

문자열 ss의 길이 : 3
문자열 tt : XYZ

# 문자열 입출력 함수

- ▶ 문자열의 입력은 `scanf()` 대신에 `gets_s()`, 출력은 `printf()` 대신에 `puts()`를 쓰는 것이 유용함

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

```
int main()
{
```

```
    char ss[20];
```

```
    puts("문자열을 입력하세요.");
```

```
    gets_s(ss, 15);
```

```
    puts(ss);
```

```
}
```

← `puts()`는 `\n`이 없이도 자동으로 줄을 넘김

← 최대 14개의 문자를 입력받음 (1개는 null문자)

## [실습1]

- ▶ `int`형 배열에 1에서 1000까지 `random number`를 10개 생성하여 저장한 후, 최소값을 계산하는 프로그램을 작성해 보세요.

```
0: 807
1: 249
2: 73
3: 658
4: 930
5: 272
6: 544
7: 878
8: 923
9: 709
최소값은 73입니다.
```

[실행 예]

## [실습2]

- ▶ 문자열 배열을 이용하여 입력된 문자열을 거꾸로 출력하는 프로그램을 작성해 보세요.
  - ▶ **Hint 1.** `puts()`와 `gets_s()`를 이용하여 문자열을 출력하고 입력받음
  - ▶ **Hint 2.** `char`형 배열 `ss`를 선언하여 문자열을 입력받고, `strlen()`을 이용하여 문자열의 크기를 계산한다. 계산한 문자열의 크기만큼 `for`문을 반복하면서 `ss`의 마지막 문자부터 처음 문자까지 차례로 새로운 `char`형 배열 `tt`에 저장한다. 최종적으로 `tt`에 저장된 문자열을 출력한다.

문자열을 입력하세요 :  
programming  
gnimmargorp

[실행 예]