Introduction to Software Design

C05. String Functions, Text Games in C, Const

Tae Hyun Kim

Topics Covered

- 문자열 관련 함수들
- The “Guess the Number” Game in C
- The “Dragon Realm” Game in C
- 상수(Constant)

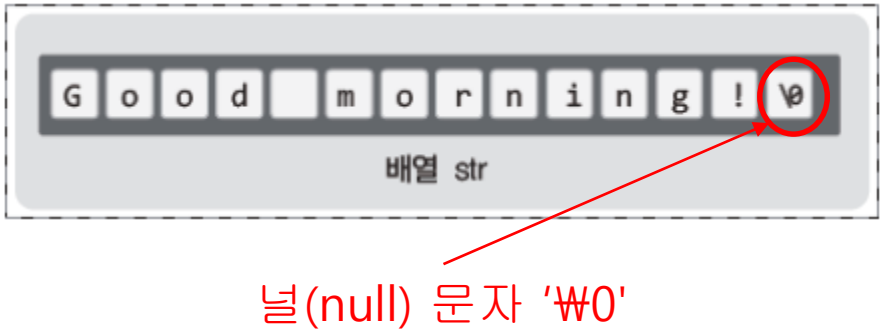
문자열 관련 함수

- strlen() : 문자열의 길이를 반환
- strcpy() : 문자열을 복사
- strcat() :두 문자열을 이어 붙임
- strcmp() : 두 문자열을 비교
- atoi(), atof() : 문자열을 숫자로 변환

문자열의 길이를 알려면?

- char str[10] = “Hello”; // 문자열의 길이는?

```
char str[14]="Good morning!";
```



- 문자열의 시작 위치에서부터 널 문자가 나올 때까지 글자수를 세면 된다.

- 이 일을 하는 함수가 strlen()

문자열의 길이를 반환하는 함수: strlen

function

strlen

```
size_t strlen ( const char * str );
```

Get string length

Returns the length of the C string *str*.

http://www.cplusplus.com/reference/cstring/strlen/

- #include <string.h> 추가해야 함.
- Parameter
  - str : 길이를 잴 문자열의 시작 주소
- Return value
  - 문자열의 길이를 반환 (size_t는 unsigned int 혹은 unsigned long을 의미)

C & Python Examples

- C

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

int main()
{
    char str[] = "abcde";
    printf("%ld\n", strlen(str));
    printf("%ld\n", strlen("asdf"));

    return 0;
}
```

- Python

```python
str = 'abcde'
print(len(str))
print(len('asdf'))
```

두 문자열이 같은지 다른지 판단은?

- str1의 내용과 str2의 내용이 같은지?

```
char str1[20] = "0123456789";
char str2[20] = "0123456789";

if(str1==str2)  // X  - 두 배열의 시작 주소를 비교하고 있음
    printf("same₩n");
else
    printf("not same₩n");
```

- → for loop로 index 증가시키며 str1과 str2 각각의 요소의 값이 같은지 확인해야 함

- 이 일을 하는 함수가 strcmp()

문자열을 비교하는 함수: strcmp

function

strcmp

```
int strcmp ( const char * str1, const char * str2 );
```

Compare two strings

Compares the C string *str1* to the C string *str2*.

http://www.cplusplus.com/reference/cstring/strcmp/

- #include <string.h> 추가해야 함.

- Parameter
  - str1, str2 : 비교할 두 문자열

- Return value
  - **0이면: 두 문자열이 같다.**
  - 0이 아니면: 두 문자열이 같지 않다.

참고:
- str1이 str2보다 크면 0보다 큰 값 반환
- str1이 str2보다 작으면 0보다 작은 값 반환

크고 작음은 아스키코드 값으로 판단
(사전편찬 순서로 뒤에 위치하면 더 큰 문자열)

C & Python Examples

- C

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

int main()
{
    char str1[20] = "0123456789";
    char str2[20] = "0123456789";
    printf("%d\n", !strcmp(str1, str2));
    return 0;
}
```

- Python

```python
str1 = '0123456789'
str2 = '0123456789'
print(str1==str2)
```

- C의 string은 사실 char형 배열이기 때문에 ==연산자를 적용하면 배열의 주소값을 비교하게 된다.
- Python의 string은 별도의 type (class)이며, string class의 경우 ==연산자는 두 string의 내용을 비교하도록 정의되어 있기 때문에 직관적인 방법으로 사용할 수 있다.

문자열을 다른 char형 배열에 복사하려면?

- str1과 같은 내용을 가지는 str2를 만들려면?

```
char str1[5] = "abcd";
char str2[5];
str2 = str1;     // X  - 배열의 이름에 다른 값 대입 불가능
```

- 그럼 어떻게 해야 할까?
- → str1의 하나 하나의 요소의 값을 str2의 각 요소로 복사해야 함

- 이 일을 하는 함수가 strcpy()

문자열을 복사하는 함수: strcpy

function

strcpy

http://www.cplusplus.com/reference/cstring/strcpy/

```
char * strcpy ( char * destination, const char * source );
```

Copy string

Copies the C string pointed by *source* into the array pointed by *destination*, including the terminating null character (and stopping at that point).

- #include <string.h> 추가해야 함.
- Parameter
  - destination : 복사가 될 목적지 char형 배열의 시작 주소
  - source : 복사할 소스 문자열의 시작 주소
- Return value
  - 목적지 char형 배열의 시작 주소 (destination)
- destination 배열의 길이가 (source 문자열의 길이 + 1)보다 작지 않도록 주의해야 함.

C & Python Examples

- C

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

int main()
{
    char str1[20] = "0123456789";
    char str2[20];

    // destination 배열 길이에 주의
    // str2[1]으로 한다면?
    strcpy(str2, str1);

    printf("str1: %s\n", str1);
    printf("str2: %s\n", str2);

    return 0;
}
```

- Python

```python
str1 = '0123456789'
str2 = str1
print('str1: %s'%str1)
print('str2: %s'%str2)
```

- Python의 string은 변경불가능(immutable)하기 때문에 어떤 string 객체(A)의 문자열 데이터를 다른 string 객체(B)에 복사할 방법이 없다.
- 대신 B 객체가 A 객체의 문자열 데이터를 '가리켜서' A와 B가 같은 문자열을 의미하게 할 수 있다.
- 위 예제에서 str2와 str1은 같은 문자열 데이터를 '가리키는' 일종의 '참조' (포인터) 라고 생각하면 된다.

두 문자열을 이어 붙이고 싶다면?

- str1 뒤에 str2를 덧붙이려면?

```
char str1[20] = "hello";
char str2[20] = "world";
str1 = str1 + str2; // X
```

- 배열의 이름에 대입 불가능 & 두 배열의 시작 주소만 더하고 있음

- 그럼 어떻게 해야 할까?
- → str1 배열의 [str1 문자열길이] index부터 index를 증가시키며 str2 배열의 각 요소를 복사하면 됨

- 이 일을 하는 함수가 strcat()

두 문자열을 이어 붙이는 함수: strcat

function

strcat

http://www.cplusplus.com/reference/cstring/strcat/

```
char * strcat ( char * destination, const char * source );
```

Concatenate strings

Appends a copy of the *source* string to the *destination* string. The terminating null character in *destination* is overwritten by the first character of *source*, and a null-character is included at the end of the new string formed by the concatenation of both in *destination*.

- #include <string.h> 추가해야 함.
- Parameter
  - destination : 덧붙여질 목적지 char형 배열의 시작 주소
  - source : 덧붙일 소스 문자열의 시작 주소
- Return value
  - 목적지 char형 배열의 시작 주소 (destination)
- destination 배열의 길이가 (destination 문자열의 길이 + source 문자열의 길이 + 1)보다 작지 않도록 주의해야 함.

C & Python Examples

- C

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

int main()
{
    char str1[20] = "hello";
    char str2[20] = "world";

    strcat(str1, str2);

    printf("%s\n", str1);

    return 0;
}
```

- Python

```python
str1 = 'hello'
str2 = 'world'

print(str1 + str2)
```

- Python의 string class에서 +연산자는 두 string의 내용을 이어 붙이도록 정의되어 있기 때문에 위와 같이 직관적으로 이어붙이기를 할 수 있다.

숫자 문자열을 숫자로 변환하고 싶다면?

- **“19”**를 **숫자 19**로 바꾸고 싶다면?
- 아스키코드 값과는 다른 이야기임
  - ‘1‘의 아스키코드 값: 49, ‘9’의 아스키코드 값: 57

- 문자열을 int형으로:

atoi

```
int atoi (const char * str);
```

Convert string to integer

- 문자형을 double형으로:

atof

```
double atof (const char* str);
```

Convert string to double

- #include <stdlib.h> 추가해야 함.

C & Python Examples

- C

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

int main()
{
    char str1[20] = "120";
    char str2[20] = "3.14";

    int i = atoi(str1);
    double d = atof(str2);

    printf("%d\n", i);
    printf("%f\n", d);

    return 0;
}
```

- Python

```python
str1 = '120'
str2 = '3.14'

print(int(str1))
print(float(str2))
```

The “Guess the Number” Game

- Python program “Guess the Number”

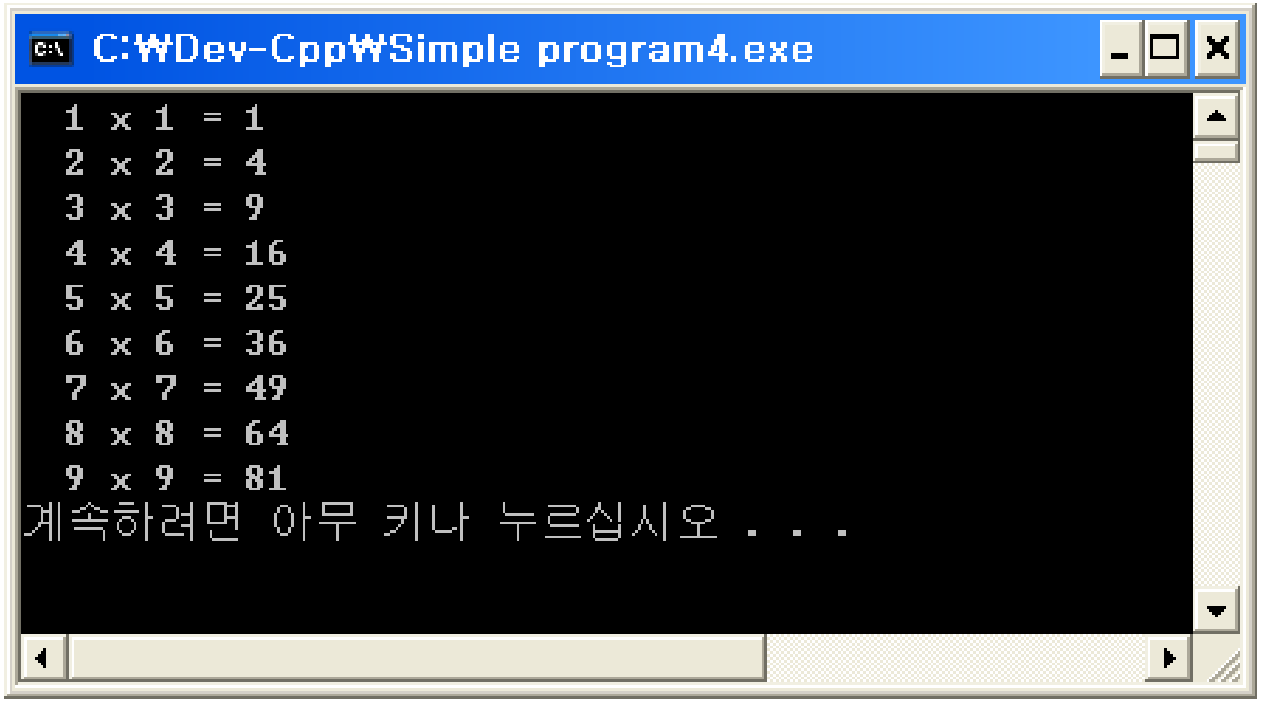
```
Hello! What is your name?
Albert
Well, Albert, I am thinking of a number between 1 and 20.
Take a guess.
10
Your guess is too high.
Take a guess.
2
Your guess is too low.
Take a guess.
4
Good job, Albert! You guessed my number in 3 guesses!
```

The “Guess the Number” Game in C

- **The “Guess the Number” Game**
  - Simple program 1
  - `while` statements
  - Simple program 2
  - `if` statements
  - Simple program 3
  - `rand()` function
- Python & C Code

A Simple Example of C programming

- **Simple program 1**



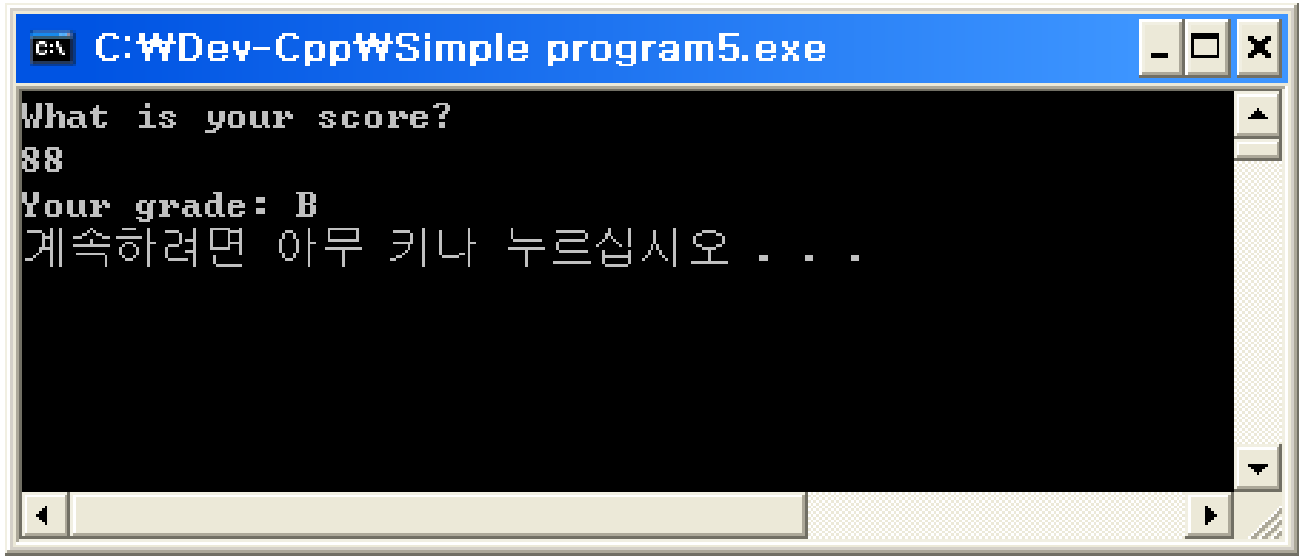
A Simple Example of C programming

- **Simple program 1 - `while` loop**

```c
/* Simple program(4) */

#include <stdio.h>


int main(void){
    int num = 1;

    while(num < 10){
            printf("  %d x %d = %d\n", num, num, num*num);
            num = num + 1;
    }

    return 0;
}
```

A Simple Example of C programming

- **Simple program 2**



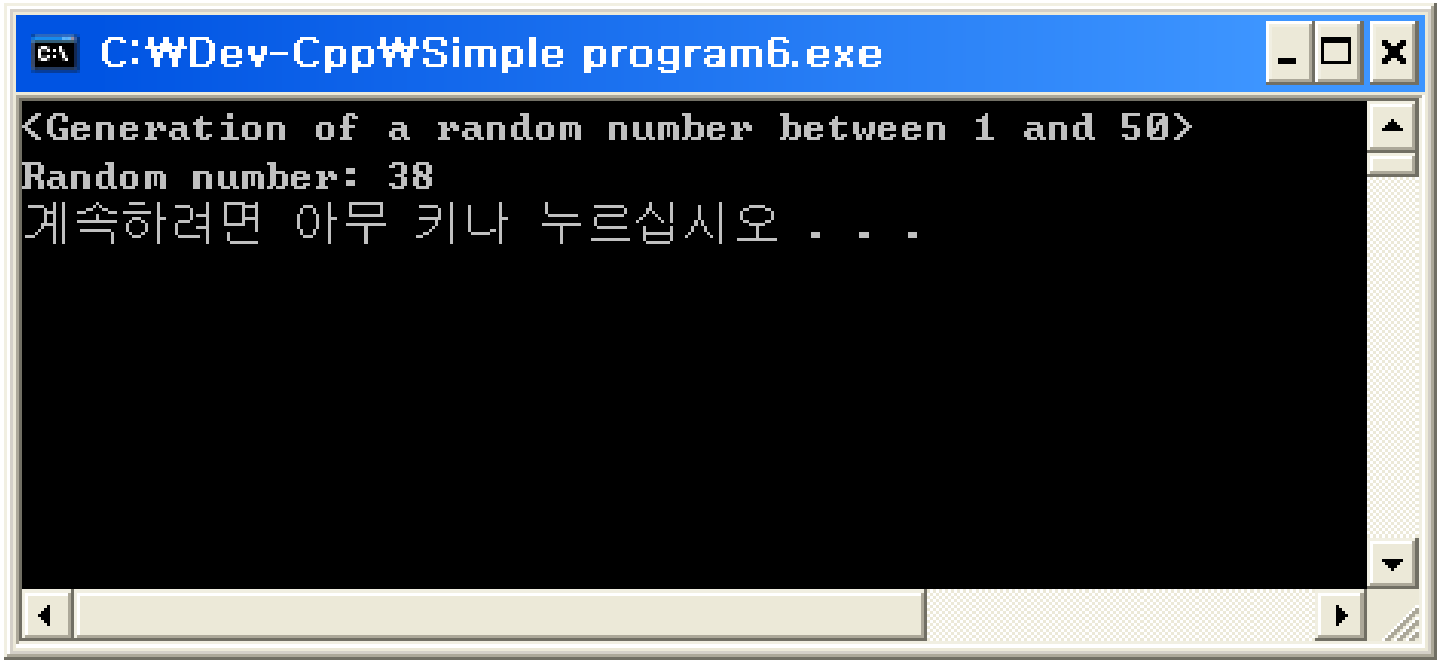
A Simple Example of C programming

- **Simple program 2 - `if` statement**

```c
/* Simple program(5) */

#include <stdio.h>


int main(void){
    int score;
    printf("What is your score?\n");
    scanf("%d", &score);
    if(score >= 90)
        printf("Your grade: A\n");
    if(score < 90 && score >= 80)
        printf("Your grade: B\n");
    if(score < 80 && score >= 70)
        printf("Your grade: C\n");
    if(score < 70 && score >= 60)
        printf("Your grade: D\n");
    if(score < 60)
        printf("Your grade: F\n");


    return 0;
  }
```

A Simple Example of C programming

- **Simple program 3**



A Simple Example of C programming

- **Simple program 3 - `rand()` function**

```c
/* Simple program(6) */

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

int main(void){
	int num;

	srand(time(NULL));
	num=(rand()%50)+1;

	printf("<Generation of a random number between 1 and 50\n");
	printf("Random number: %d\n",num);


	return 0;
}
```

A Simple Example of C programming

function
srand

http://www.cplusplus.com/reference/cstdlib/srand/?kw=srand

```
void srand (unsigned int seed);
```

Initialize random number generator

The pseudo-random number generator is initialized using the argument passed as *seed*.

function
rand

http://www.cplusplus.com/reference/cstdlib/rand/?kw=rand

```
int rand (void);
```

Generate random number

Returns a pseudo-random integral number in the range between 0 and RAND_MAX.

function
time

http://www.cplusplus.com/reference/ctime/time/?kw=time

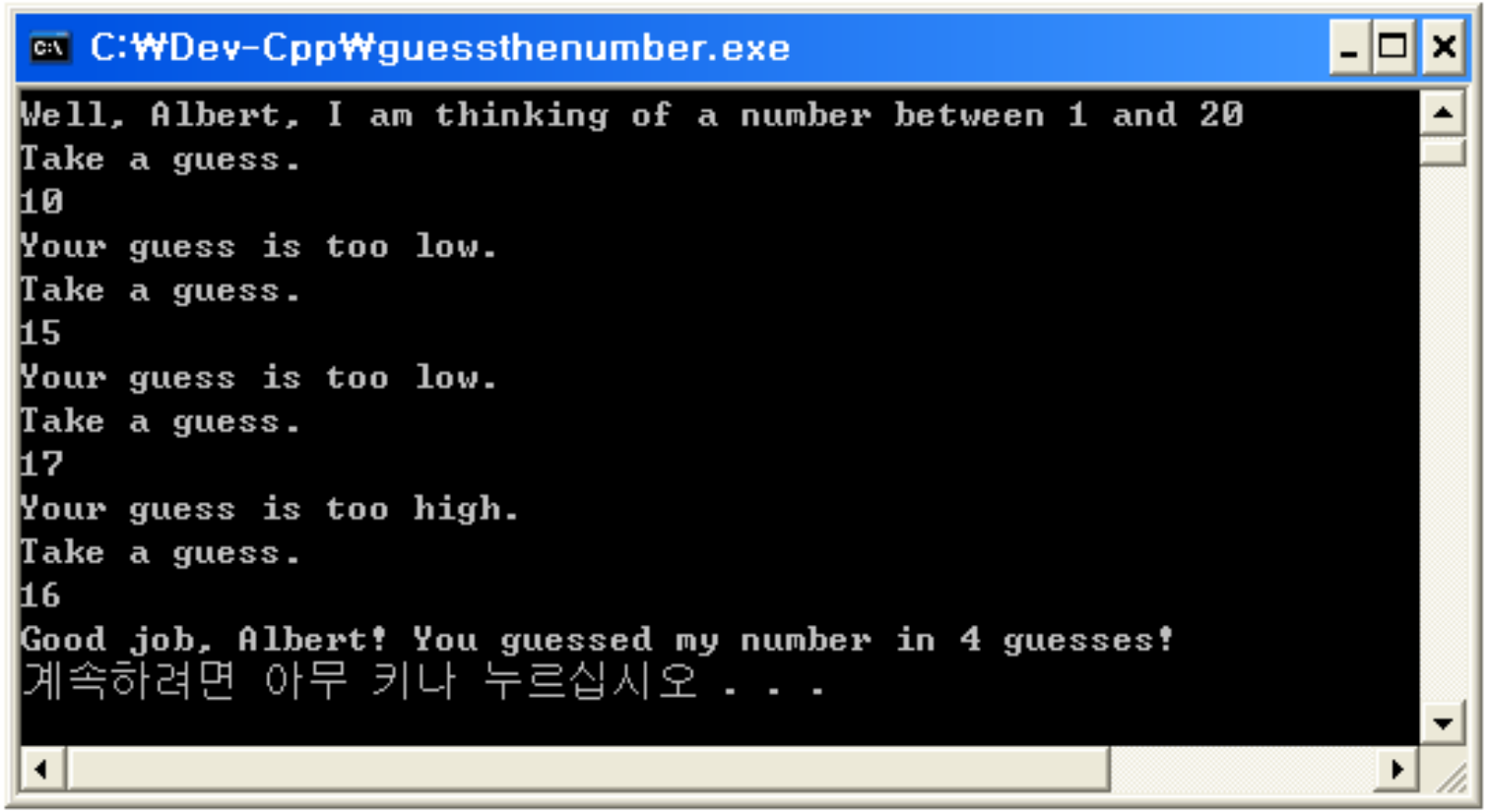
```
time_t time (time_t* timer);
```

Get current time

Get the current calendar time as a value of type time_t.

The “Guess the Number” Game with C

- C program “Guess the Number”



Python & C Code

- **Python source code**

```
# This is a guess the number game.
import random

guessesTaken = 0
```

- **C source code**

```
/* Guess the Number */

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

int main(void){
    char name[10];
    int guessesTaken, guess, number;

    guessesTaken = 0;
```

Python & C Code

- **Python source code**

```
print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

- **C source code**

```
srand(time(NULL));
number=(rand()%20)+1;


printf("Hello! What is your name?\n");
scanf("%s", name);

printf("Well, %s, I am thinking of a number between 1 and 20\n", name);
```

Python & C Code

- **Python source code**

```
while guessesTaken < 6:
    #print 'Take a guess.'
    guess = raw_input('Take a guess.')
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

- **C source code**

```
while (guessesTaken < 6){
        printf("Take a guess.\n");
        scanf("%d", &guess);
        guessesTaken = guessesTaken + 1;

        if (guess < number)
                printf("Your guess is too low.\n");
        if (guess > number)
                printf("Your guess is too high.\n");
        if (guess == number)
                break;
}
```

Python & C Code

- **Python source code**

```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

- **C source code**

```
    if (guess == number)
            printf("Good job, %s! You guessed my number in %d guesses!\n", name, guessesTaken);
    if (guess != number)
            printf("Nope. The number I was thinking of was %d\n", number);


return 0;
}
```

C Code

- **C source code (1/2)**

```c
/* Guess the Number */

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

int main(void){
    char name[10];
    int guessesTaken, guess, number;

    guessesTaken = 0;

    srand(time(NULL));
    number=(rand()%20)+1;


    printf("Hello! What is your name?\n");
    scanf("%s", name);

    printf("Well, %s, I am thinking of a number between 1 and 20\n", name);
```

C Code

- **C source code (2/2)**

```
while (guessesTaken < 6){
      printf("Take a guess.\n");
      scanf("%d", &guess);
      guessesTaken = guessesTaken + 1;

      if (guess < number)
               printf("Your guess is too low.\n");
      if (guess > number)
               printf("Your guess is too high.\n");
      if (guess == number)
               break;
}

      if (guess == number)
               printf("Good job, %s! You guessed my number in %d guesses!\n", name, guessesTaken);
      if (guess != number)
               printf("Nope. The number I was thinking of was %d\n", number);


return 0;
}
```

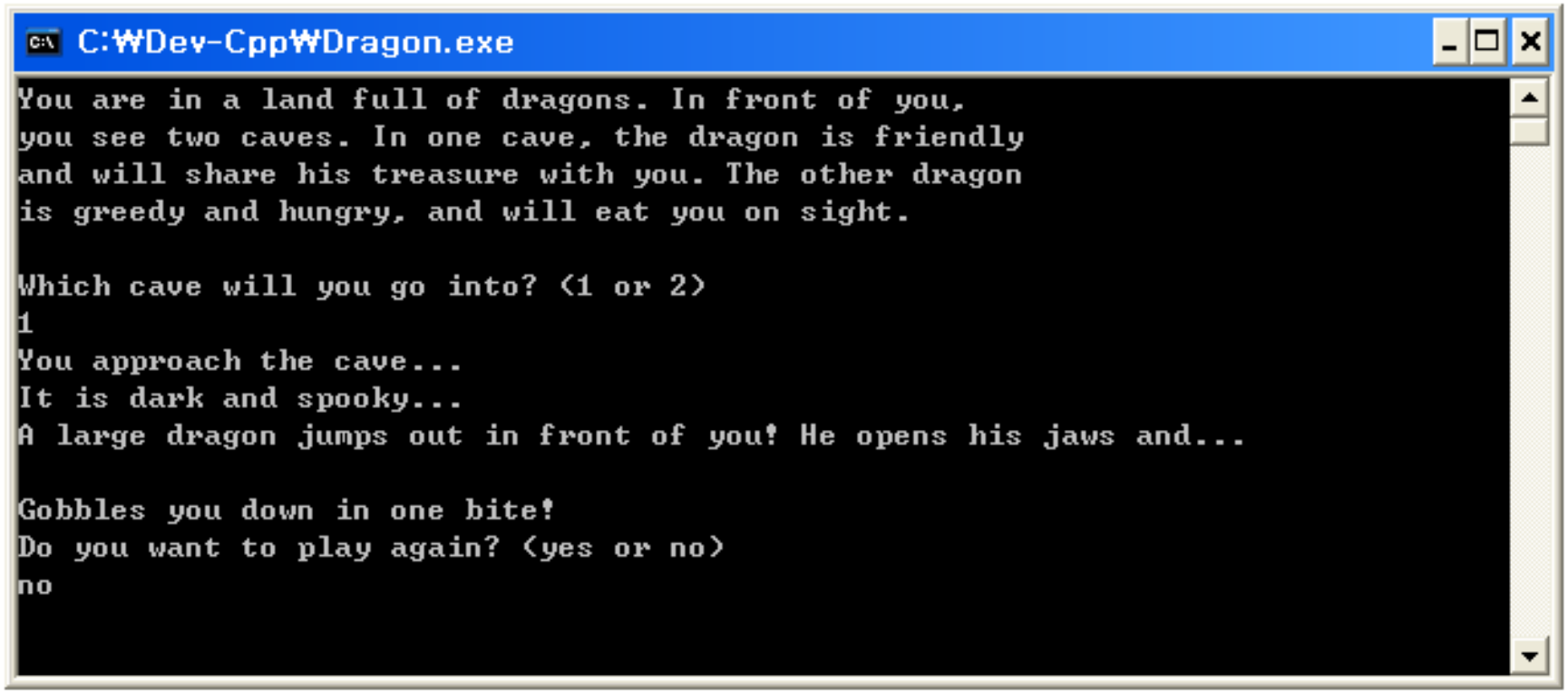
C Code

- **C source code – if else**

```
while (guessesTaken < 6){
      printf("Take a guess.\n");
      scanf("%d", &guess);
      guessesTaken = guessesTaken + 1;

      if (guess < number)
               printf("Your guess is too low.\n");
      else if (guess > number)
               printf("Your guess is too high.\n");
      else
           break;
}

      if (guess == number)
               printf("Good job, %s! You guessed my number in %d guesses!\n", name, guessesTaken);
      else
           printf("Nope. The number I was thinking of was %d\n", number);


return 0;
}
```

The “Dragon Realm” Game

- Python program “Dragon Realm”

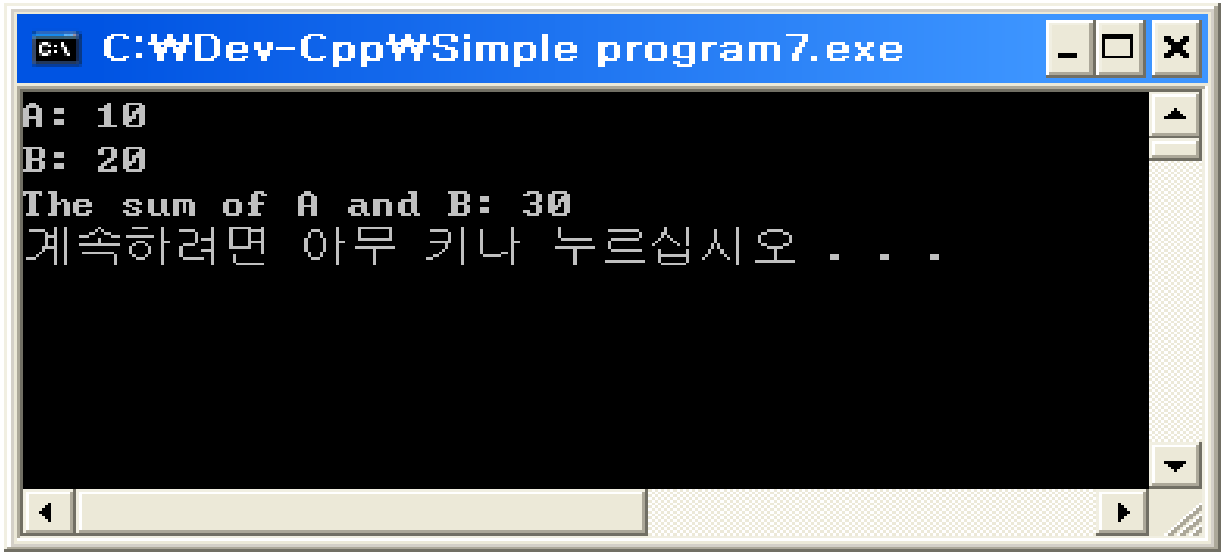


The “Dragon Realm” Game in C

- **The “Dragon Realm” Game**
  - Simple program 1
  - User defined function
  - Simple program 2
  - `strcmp()` function
  - Simple program 3
  - `Sleep()` function
- **Python & C Code**

A Simple Example of C

- **Simple program1**
  - **User defined function**



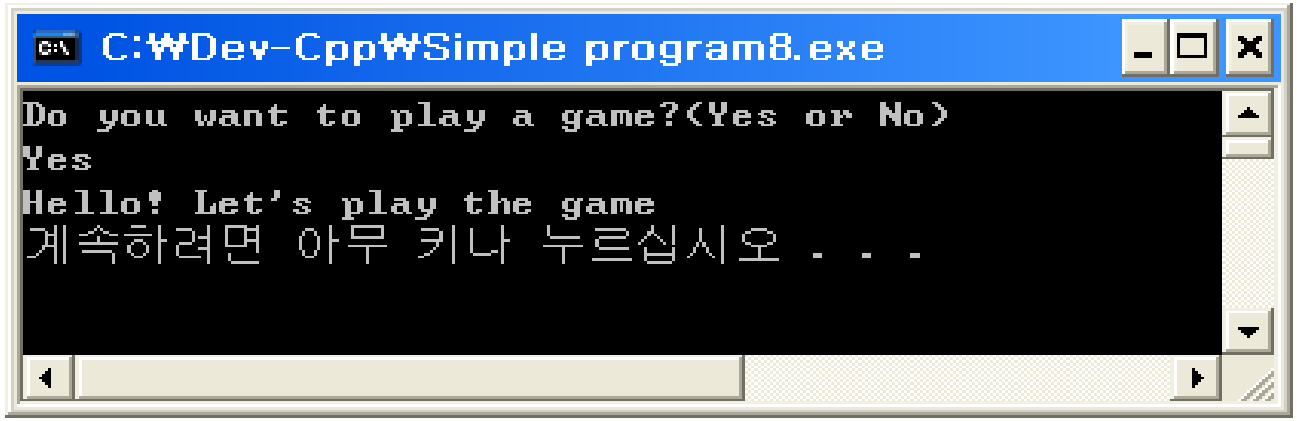
A Simple Example of C

- **Simple program1**
  - **Source code**

```c
/* Simple program(7) */

#include <stdio.h>


int sum(int a, int b);

int main(void){
    int a = 10, b = 20;
    sum(a, b);


    return 0;
}

int sum(int a, int b){
     printf("A: %d\n", a);
     printf("B: %d\n", b);
     printf("The sum of A and B: %d\n", a+b);
}
```

A Simple Example of C

- **Simple program2**
  - **`strcmp()` function**



A Simple Example of C

- **Simple program2**
  - **Source code**

```c
/* Simple program(8) */

#include <stdio.h>

#include <string.h>

int main(void){

    char str[5];

    printf("Do you want to play a game?(Yes or No) \n");
    scanf("%s", str);

    if(strcmp(str, "Yes") == 0)
        printf("Hello! Let's play the game\n");

    else if(strcmp(str, "No") == 0)
        printf("Good bye!\n");


    return 0;
}
```

A Simple Example of C

- **Simple program2**
  - **`strcmp()` function**
  - Compare characters of two strings .
  - Return Value
    - if Return value if **< 0** then it indicates **S1 is less than S2**
    - if Return value if **> 0** then it indicates **S1 is greater than S2**
    - if Return value if **= 0** then it indicates **S1 is equal to S2**

A Simple Example of C

- **Simple program2**
  - **Logical operators in c**

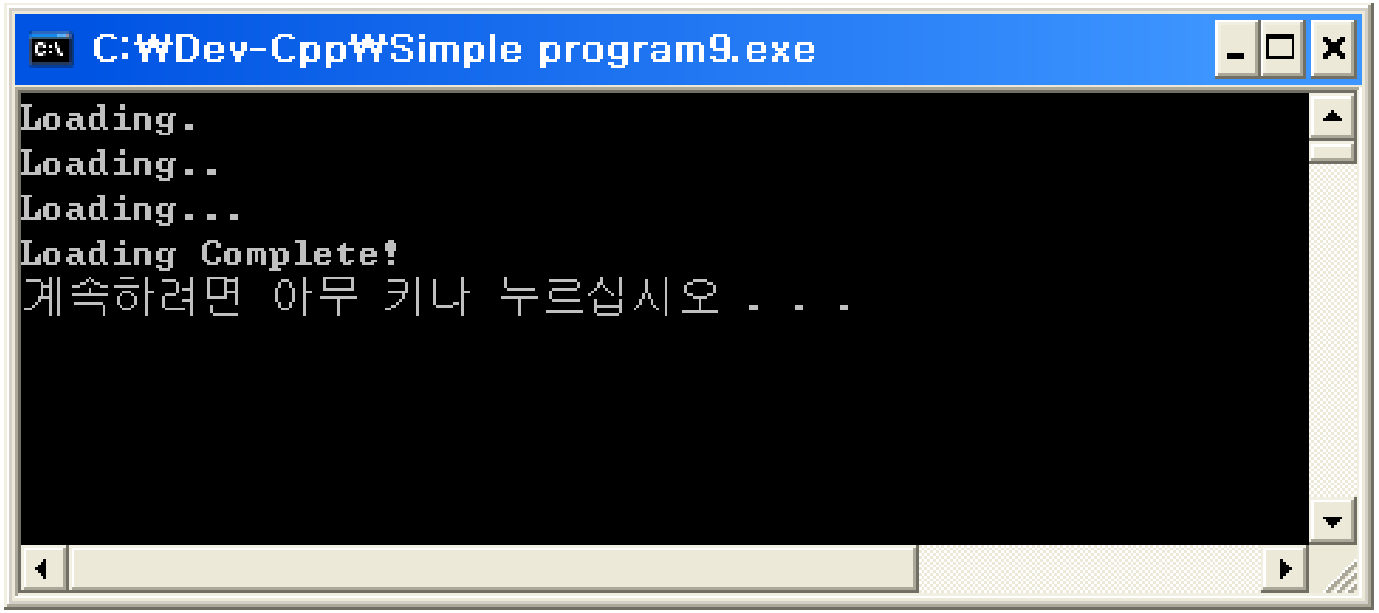
Operator name	Python syntax	C syntax
Logical AND	`a and b`	`a && b`
Logical OR	`a or b`	`a || b`

  - **example**

```
if(strcmp(str, "Yes") == 0 || strcmp(str, "Y") == 0)
    printf("Hello! Let's play the game\n");
```

A Simple Example of C

- **Simple program3**
  - **`Sleep()` function**



A Simple Example of C

Simple program3

- **`Sleep()` function – only in Windows!**
- Suspend execution for an interval of time.
- The time interval for which execution is to be suspended, in milliseconds.

- **example**

```
#include <windows.h>

int main(void){
    Sleep(2000);
……
```

  - number is in milliseconds 2Sec = 2000 Milliseconds.

A Simple Example of C

- **Cross-platform version of Sleep()**

  - Use Sleep() declared in <windows.h> on Windows
  - Use usleep() declared in <unistd.h> on Linux suspended, in milliseconds.

```
#ifdef _WINDOWS
#include <windows.h>

#else
#include <unistd.h>
#define Sleep(x)
usleep((x)*1000)

#endif
```

A Simple Example of C

- **Simple program3**
  - **Source code**

```c
/* Simple program(9) */

#include <stdio.h>

int main(void){

    printf("Loading.\n");
    Sleep(3000);
    printf("Loading..\n");
    Sleep(2000);
    printf("Loading...\n");
    Sleep(1000);
    printf("Loading Complete!\n");


    return 0;
}
```

```c
#ifdef _WINDOWS
#include <windows.h>

#else
#include <unistd.h>
#define Sleep(x) usleep((x)*1000)

#endif
```

Python & C Code

- **Python source code**

```
import random
import time
```

- **C source code**

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

#include <string.h>
#include <time.h>

int displayIntro(void);
int chooseCave(void);
int checkCave(int chooseCave);
```

```
#ifdef _WINDOWS
#include <windows.h>

#else
#include <unistd.h>
#define Sleep(x) usleep((x)*1000)

#endif
```

Python & C Code

- **Python source code**

```
def displayIntro():
    print 'You are in a land full of dragons. In front of you,'
    print 'you see two caves. In one cave, the dragon is friendly'
    print 'and will share his treasure with you. The other dragon'
    print 'is greedy and hungry, and will eat you on sight.'
    print
```

- **C source code**

```
int displayIntro(void){
    printf("You are in a land full of dragons. In front of you,\n");
    printf("you see two caves. In one cave, the dragon is friendly\n");
    printf("and will share his treasure with you. The other dragon\n");
    printf("is greedy and hungry, and will eat you on sight.\n\n");
}
```

Python & C Code

- **Python source code**

```
def chooseCave():
    cave = ''
    while cave != '1' and cave != '2':
        print 'Which cave will you go into? (1 or 2)'
        cave = raw_input()

    return cave
```

- **C source code**

```
int chooseCave(void){
    int cave;
    cave = 0;

    while (cave != 1 && cave != 2){
          printf("Which cave will you go into? (1 or 2)\n");
          scanf("%d", &cave);

    return cave;
    }
}
```

Python & C Code

- **Python source code**

```
def checkCave(chosenCave):
    print 'You approach the cave...'
    time.sleep(2)
    print 'It is dark and spooky...'
    time.sleep(2)
    print 'A large dragon jumps out in front of you! He opens his jaws and...'
    print
    time.sleep(2)
```

- **C source code**

```
int checkCave(int chooseCave){
    int friendlyCave;

    printf("You approach the cave...\n");
    Sleep(2000);
    printf("It is dark and spooky...\n");
    Sleep(2000);
    printf("A large dragon jumps out in front of you! He opens his jaws and...\n\n");
    Sleep(2000);
```

Python & C Code

- **Python source code**

```
friendlyCave = random.randint(1, 2)

if chosenCave == str(friendlyCave):
     print 'Gives you his treasure!'
else:
     print 'Gobbles you down in one bite!'
```

- **C source code**

```
    srand(time(NULL));
    friendlyCave=(rand()%2)+1;

    if(chooseCave == friendlyCave)
         printf("Gives you his treasure!\n");

    else
         printf("Gobbles you down in one bite!\n");
}
```

Python & C Code

- **Python source code**

```
playAgain = 'yes'
while playAgain == 'yes' or playAgain == 'y':

    displayIntro()
    caveNumber = chooseCave()
    checkCave(caveNumber)

    print 'Do you want to play again? (yes or no)'
    playAgain = raw_input()
```

- **C source code**

```
int main(void){
    char playAgain[5] = "yes";
    int caveNumber;

    while(strcmp(playAgain, "yes") == 0 || (strcmp(playAgain, "y") == 0)){
        displayIntro();
        caveNumber = chooseCave();
        checkCave(caveNumber);

        printf("Do you want to play again? (yes or no)\n");
        scanf("%s", playAgain);
    }

    return 0;
}
```

C Code

- **C source code (1/3)**

```c
/* Dragon Realm */

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

#include <string.h>
#include <time.h>


int displayIntro(void);
int chooseCave(void);
int checkCave(int chooseCave);

int main(void){
    char playAgain[5] = "yes";
    int caveNumber;

    while(strcmp(playAgain, "yes") == 0 || (strcmp(playAgain, "y") == 0)){
         displayIntro();
         caveNumber = chooseCave();
         checkCave(caveNumber);

         printf("Do you want to play again? (yes or no)\n");
         scanf("%s", playAgain);
    }

    return 0;
}
```

```c
#ifdef _WINDOWS
#include <windows.h>

#else
#include <unistd.h>
#define Sleep(x) usleep((x)*1000)

#endif
```

C Code

- **C source code (2/3)**

```
int displayIntro(void){
    printf("You are in a land full of dragons. In front of you,\n");
    printf("you see two caves. In one cave, the dragon is friendly\n");
    printf("and will share his treasure with you. The other dragon\n");
    printf("is greedy and hungry, and will eat you on sight.\n\n");
}

int chooseCave(void){
    int cave;
    cave = 0;

    while (cave != 1 && cave != 2){
          printf("Which cave will you go into? (1 or 2)\n");
          scanf("%d", &cave);

    return cave;
    }
}
```

C Code

- **C source code (3/3)**

```
int checkCave(int chooseCave){
    int friendlyCave;

    printf("You approach the cave...\n");
    Sleep(2000);
    printf("It is dark and spooky...\n");
    Sleep(2000);
    printf("A large dragon jumps out in front of you! He opens his jaws and...\n\n");
    Sleep(2000);

    srand(time(NULL));
    friendlyCave=(rand()%2)+1;

    if(chooseCave == friendlyCave)
         printf("Gives you his treasure!\n");

    else
         printf("Gobbles you down in one bite!\n");
}
```

“상수”?

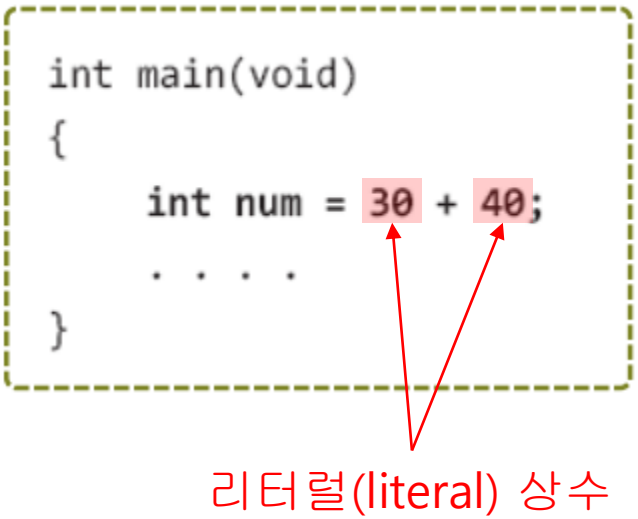
- 변수
  - 값을 저장할 수 있는 메모리 공간에 붙여진 이름
  - 값이 바뀔 수 있다

```
int a = 3;
a = 5;
```

- 상수
  - 값이 바뀔 수 없다

- 이름이 없는 상수: 리터럴(literal) 상수
- 이름이 있는 상수: 심볼릭(symbolic) 상수

이름이 없는 리터럴(Literal) 상수



- CPU에서 덧셈 연산을 하기 위해서는 30과 40도 메모리상에 존재해야 함
- 변수와 달리 할당된 메모리 공간에 이름이 없고 값도 변경할 수 없음.

리터럴 상수의 자료형

- 리터럴 상수도 자료형을 가진다 (자료형이 결정되어야 메모리에 저장될 수 있다).

```
#include <stdio.h>

int main()
{
    printf("%d₩n", sizeof(7));
    printf("%d₩n", sizeof(3.14));
    printf("%d₩n", sizeof('C'));
    return 0;
}
```

- 정수형 상수는 기본적으로 int형으로 저장
- 실수형 상수는 기본적으로 double형으로 저장

리터럴 상수의 자료형

- 아래 접미사를 이용하면 리터럴 상수의 자료형을 지정할 수 있다.

접미사	자료형	사용의 예
U	unsigned int	unsigned int n = 1025U
L	long	long n = 2467L
UL	unsigned long	unsigned long n = 3456UL
LL	long long	long long n = 5768LL
ULL	unsigned long long	unsigned long long n = 8979ULL

접미사	자료형	사용의 예
F	float	float f = 3.15F
L	long double	long double f = 5.789L

참고) 위 접미사들은 대소문자 구분 안 함. 아래 두 문장 의미 같음.

```
float f = 3.15f;
float f = 3.15F;
```

이름이 있는 심볼릭(Symbolic) 상수

```
int main(void)
{
    const int MAX=100;    // MAX는 상수! 따라서 값의 변경 불가!
    const double PI=3.1415;    // PI는 상수! 따라서 값의 변경 불가!
    . . . .
}
```

```
int main(void)
{
    const int MAX;    // 쓰레기 값으로 초기화 되어버림
    MAX=100;    // 값의 변경 불가! 따라서 컴파일 에러 발생!
    . . . .
}
```

참고) 심볼릭 상수의 이름은 모든 대문자로 표시하고, 둘 이상 단어를 연결할 때는 MY_AGE와 같은 식으로 하는 것이 관례.

C & Python Examples

- C

```c
# include <stdio.h>

int main()
{
    const int MAX=100;
    printf("%d\n", MAX);
    MAX = 200;  // build error
}
```

- Python

```python
MAX = 100
print(MAX)
MAX = 200 # no error
```

- Python에는 상수(const)의 개념이 없다.
- 상수를 쓰고 싶으면 프로그래머가 특정 변수에 '쓰기'를 하지 않고 사용하면 된다.