

Chapter 5. Functions and Control Flow

April, 2016
Seungjae Baek

Dept. of software
Dankook University

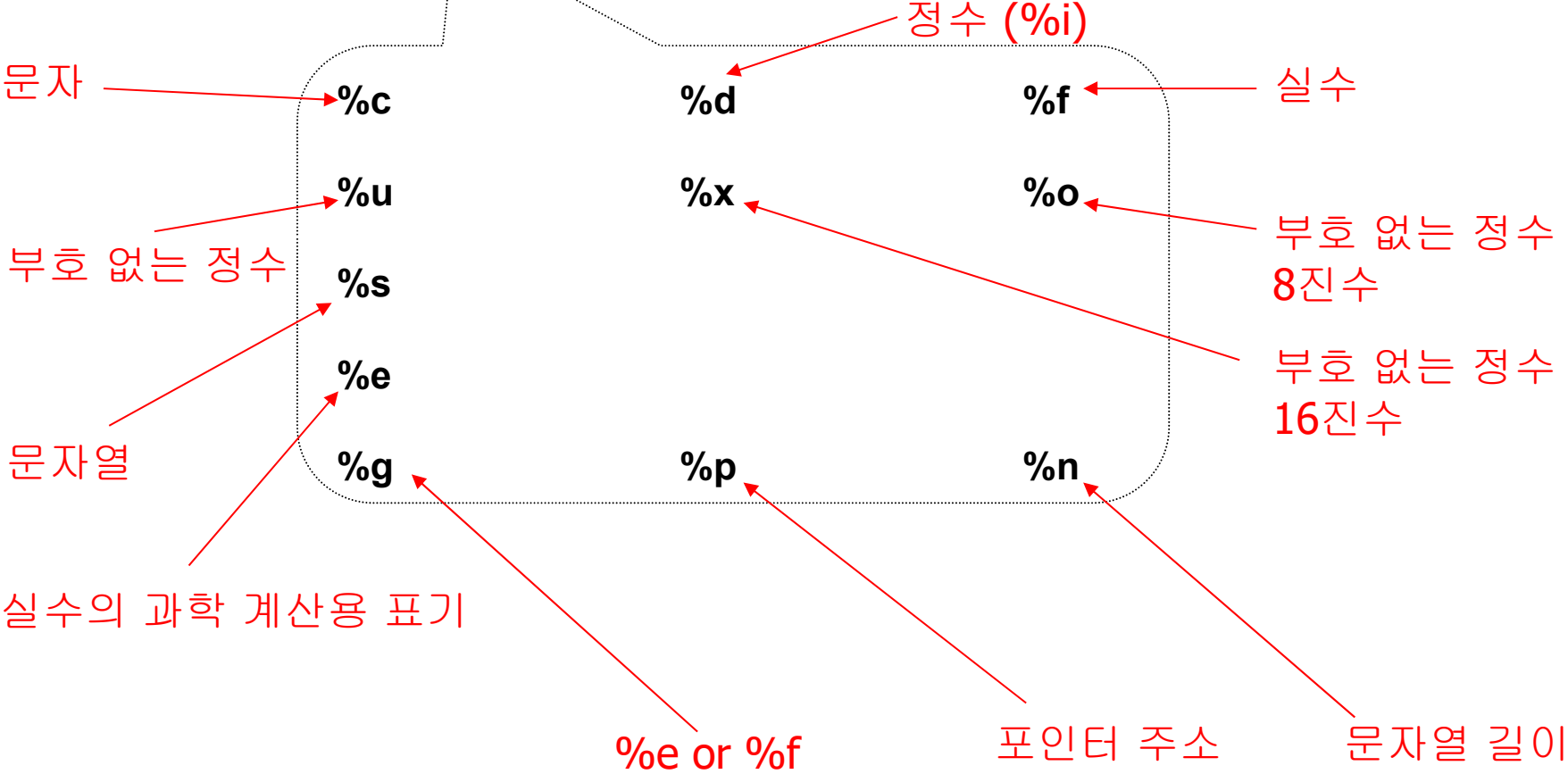
<http://embedded.dankook.ac.kr/~baeksj>

- `printf` 함수와 `scanf` 함수의 이해
- 함수의 이해
- 대표적 제어문인 `if`와 `for` 문을 이해한다.
- 코드 블록(`code block`)의 의미와 필요성을 이해한다.
- `format specifier` 와 `escape sequence`를 이해한다

printf()의 확장된 기능 (1/7)

■ 형식 지정자 (format specifier)

```
printf("...%*...", ...);
```



■ 형식 지정자의 예

```
#include <stdio.h>
int main()
{
    int num;
    printf("%c      %d      %f\n", 'A', -123, 456.789);
    printf("%d      %u\n", -2, -2);
    printf("%d      %x      %o\n", 26, 26, 26);
    printf("%s\n", "Hello world");
    printf("%f      %e      %g\n", 123.45, 123.45, 123.45);
    printf("%s      %p\n", "Hi", "Hi");
    printf("1234\n%n", &num);
    printf("num = %d\n", num);
    return 0;
}
```

Q: 정상 실행이 되는가?

printf()의 확장된 기능 (3/7)

■ 형식 지정자의 확장

- ✓ 형식 지정자 앞에 **출력 양식 변환 수정자**가 사용될 수 있음

```
printf("...%[-][m][.][n][h|L]*...", ...);
```

```
#include <stdio.h>
// 형식 지정자의 확장 예 1
int main()
{
    int var_int;
    short var_short;
    long var_long;
    float var_float;
    double var_double;
    var_int = 4; var_short = 2; var_long = 4;
    printf("%d %hd %ld\n",
           var_int, var_short, var_long);
    printf("%hu %u\n\n", 65536 + 1, 65536 + 1);
    var_float = 123.456; var_double = 123.456;
    printf("%f %lf\n\n", var_float, var_double);
}
```

```
#include <stdio.h>
// 형식 지정자의 확장 예 2
int main()
{
    printf("%d\n", 1234);
    printf("%d\n", 1234);
    printf("%-10d\n\n", 1234);
    printf("%f\n", 234.5678);
    printf("%10.2f\n", 234.5678);
    printf("%10.6f\n", 234.5678);
    printf("%-10.2f\n\n", 234.5678);
}
```

Q: printf("%dWn", 1234) 에서 %2d로 변경하면?

printf()의 확장된 기능 (4/7)

- “The first alphabet is A”를 출력하고 싶다. 어떻게 할까?

```
#include <stdio.h>
int main()
{
    char ch;
    ch = 'A';
    printf("The first alphabet is %c", ch);
}
```

- O.K. 대소문자, 숫자 출력 가능.

그럼 “는 어떻게 출력할 수 있을까?

또한 Enter, Back Space, Tab 등의 특수 문자는 어떻게 프로그래밍에서 수행시킬 수 있을까?

➔ **Escape Sequence (역슬래시 문자 상수)**

printf()의 확장된 기능 (5/7)

■ Escape sequence (역슬래시 문자 상수)

\b	backspace	\f	form feed
\n	new line	\r	carriage return
\t	horizontal tab	\v	vertical tab
\0	null character	\a	bell
\"	double quotation	'	single quotation
\\	backslash	\?	question mark
\N	8진수 상수	\xN	16진수 상수

```
#include <stdio.h>
#include <string.h> // strlen()을 위해 포함
int main()
{
    printf("1234\n567\n");
    printf("1234\b567\n");
    printf("1234\f567\n");
    printf("1234\r567\n");
    printf("1234\t567\n");
    printf("1234\v567\n");
    printf("1234\a567\n");
    printf("\"Why\?\"");
    printf("length = %d\n", strlen("abc\0d")); // \0 테스트를 위한 인위적인 코드
}
```

printf()의 확장된 기능 (6/7)

- 역슬래시 문자들은 ASCII 문자 코드에 대응

0~32 : 제어문자 (bell은 7, BS는 8, HT은 9, CR은 13)
 33~47 : 특수문자 ('!'는 33, '"'는 34, '/'는 47)
 48~57 : 숫자 ('0'은 48, '1'은 49, '9'는 57)
 65~90 : 알파벳 대문자 ('A'는 65, 'B'는 66, 'Z'는 90)
 97~122 : 알파벳 소문자 ('a'는 97, 'b'는 98, 'z'는 122)
 그 외 : 특수 문자 ('\:'은 58, '['는 91, '{'는 123)

```
#include <stdio.h>
int main()
{
    printf("1234\t567\n");
    printf("1234%c567\n", '\t');
    printf("%d\n", '\t');
    printf("1234%c567\n", 9);
    return 0;
}
```

아스키 테이블

Segoe UI

문자	십진법	Hex #	이름	Ctrl 문자
	00	00	NUL	^@
r	01	01	SOH	^A
Г	02	02	STX	^B
L	03	03	ETX	^C
J	04	04	EOT	^D
	05	05	ENQ	^E
-	06	06	ACK	^F
•	07	07	BEL	^G
█	08	08	BS	^H
	09	09	HT	^I
	10	0A	LF	^J
⌘	11	0B	VT	^K
♀	12	0C	FF	^L
	13	0D	CR	^M
⌘	14	0E	SO	^N
⌘	15	0F	SI	^O
+	16	10	DLE	^P
◀	17	11	DC1	^Q

서체 선택 문자삽입
 서체 기억

printf()의 확장된 기능 (7/7)

■ ASCII 코드 사용 예

```

#include <stdio.h>
int main()
{
    int i;
    i = 49;
    printf("%d %c\n", i, i);
    printf("%c %d\n", 'A', 'A');
    printf("%c %d\n", 65, 65);
    printf("%c %d\n", 0x41, 0x41);
    printf("%c %d\n", '\x41', '\x41');
    printf("1234%c567\n", '\t');
    printf("1234%c567\n", 9);
    printf("1234%c567\n", '9');

    if ('\b' == '7')
        printf("Same\n");
    else
        printf("Different\n");

    for (i = 0; i <= 127; i++) // 모든 ASCII 문자 출력
        printf("%d-th ASCII code = %c\n", i, i);
}

```

A == 0x41

ASCII 코드에서 9번은
Horizontal Tab에 매핑.
ASCII 코드에서 57번은
숫자 '9'에 매핑

scanf()의 확장된 기능

■ 함수의 형식

```
scanf (“...%*...”, ...);
```

정수 : %d, %o, %x

실수 : %f, %Lf (double), %Lf (Long double)

```
#include <stdio.h>
int main()
{
    int num;
    float f;
    double d;
    printf("Enter an Integer: ");
    scanf("%d", &num);
    printf("Enter a floating point number: ");
    scanf("%f", &f);
    printf("Enter a floating point number again: ");
    scanf("%Lf", &d);
    printf("%d ", num);
    printf("%f ", f);
    printf("%Lf", d);
    return 0;
}
```

■ 함수의 정의

- ✓ 작업을 수행하기 위한 문장(statement)들의 그룹
- ✓ 모든 C 프로그램은 하나 이상의 함수가 존재 : main()
- ✓ 모든 프로그램은 추가 함수 정의가 가능

■ 함수가 필요한 예 - 1

```
#include <stdio.h>
int main()
{
    int x, y;
    x = 4; y = 2;

    printf("=====\n");
    printf("%d + %d = %d\n", x, y, x + y);
    printf("=====\n");
    printf("%d - %d = %d\n", x, y, x - y);
    printf("=====\n");
    printf("%d * %d = %d\n", x, y, x * y);
    printf("=====\n");
    printf("%d / %d = %d\n", x, y, x / y);
    return 0;
}
```

■ 함수가 필요한 예 - 2

```
#include <stdio.h>
int main()
{
    int x = 2;
    printf("%d x %d = %d\n", x, 1, x * 1);
    printf("%d x %d = %d\n", x, 2, x * 2);
    printf("%d x %d = %d\n", x, 3, x * 3);
    printf("%d x %d = %d\n", x, 4, x * 4);
    printf("%d x %d = %d\n", x, 5, x * 5);
    printf("%d x %d = %d\n", x, 6, x * 6);
    printf("%d x %d = %d\n", x, 7, x * 7);
    printf("%d x %d = %d\n", x, 8, x * 8);
    printf("%d x %d = %d\n", x, 9, x * 9);
}
```

■ 함수 형식

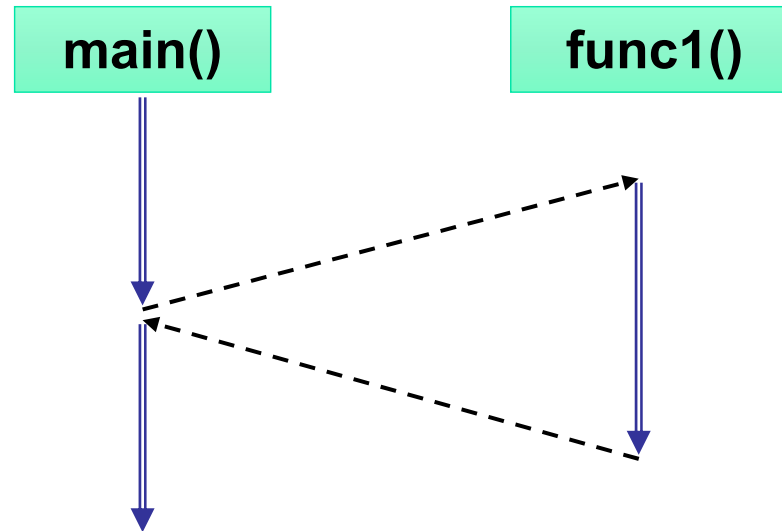
```
return_type function_name(parameter_lists)
{
    statements;
    statements;
}
```

```
#include <stdio.h>
void func1(void);

int main()
{
    printf("I ");
    func1();
    printf("C.\n");
}

void func1(void)
{
    printf("like ");
    return;
}
```

- 함수 제어 흐름 (control flow)



- ☞ 함수 내부에서 함수를 호출하면?
- ☞ 응용이 두 개 이상 수행되는 경우 흐름도는? (multitask, multithread)

■ 함수가 필요한 예 - 1 (update)

Q: PrintLine() 함수를 main() 함수 뒤에 작성하려면 ?

```
#include <stdio.h>

void PrintLine()
{
    printf("===== \n");
}

int main()
{
    int x, y;
    x = 4; y = 2;
    PrintLine();
    printf("%d + %d = %d \n", x, y, x + y);
    PrintLine();
    printf("%d - %d = %d \n", x, y, x - y);
    PrintLine();
    printf("%d * %d = %d \n", x, y, x * y);
    PrintLine();
    printf("%d / %d = %d \n", x, y, x / y);
    return 0;
}
```


- 함수가 필요한 예 - 2 (update)

```
#include <stdio.h>

void Multi_Table(int x);

int main()
{
    Multi_Table(2);
    Multi_Table(5);
}

void Multi_Table(int x)
{
    printf("%d x %d = %d\n", x, 1, x * 1);
    printf("%d x %d = %d\n", x, 2, x * 2);
    printf("%d x %d = %d\n", x, 3, x * 3);
    printf("%d x %d = %d\n", x, 4, x * 4);
    printf("%d x %d = %d\n", x, 5, x * 5);
    ...
}
```

- 인자와 리턴 값이 있는 경우

```
#include <stdio.h>

int Add(int num1, int num2)
{
    return num1 + num2;
}

int main(void)
{
    int result;

    result = Add(3, 4);
    printf("Result1: %d \n", result);

    result = Add(5, 8);
    printf("Result2: %d \n", result);

    return 0;
}
```

Q: 함수로 가능한가?

- 인생은 선택의 연속임!!
- if 조건문 형식

```
if (expression)
    statement;
```

- expression이 0이 아니면 참, 0이면 거짓

```
#include <stdio.h>

int main(void)
{
    int num;

    printf("enter a number: ");
    scanf("%d", &num);

    if (num < 0)
        printf("number is negative\n");
    if (num > -1)
        printf("number is non-negative\n");

    return 0;
}
```

관계 연산자 (>, ==, <, !=, ...)
산술 연산자보다 우선 순위가 낮음

- 예제 프로그램
 - ✓ 다음 프로그램의 기능은?

```
#include <stdio.h>

int main(void)
{
    int answer;

    printf("What is 10 + 14 ? ");
    scanf("%d", &answer);

    if (answer == 10 + 14)
        printf("Right!");
    if (answer != 10 + 14)
        printf("Wrong!");

    return 0;
}
```

■ 연습문제:

- ✓ 정수를 입력 받아 홀수인지 짝수인지 알려주는 프로그램 작성

```
#include <stdio.h>

int main(void)
{
    int num;

    printf("enter a number: ");
    scanf("%d", &num);

    if ((num % 2) == 0)
        printf("number is even\n");
    if ((num % 2) == 1)
        printf("number is odd\n");

    return 0;
}
```

■ 형식

```
if (expression)
    statement;
else
    statement;
```

✓ else를 사용해 다시 작성하면...

```
#include <stdio.h>

int main(void)
{
    int num;
    printf("enter a number: ");
    scanf("%d", &num);

    if ((num % 2) == 0)
        printf("number is even\n");
    else
        printf("number is odd\n");

    return 0;
}
```

■ Example of Exception Handling (예외 처리 예제)

```
#include <stdio.h>

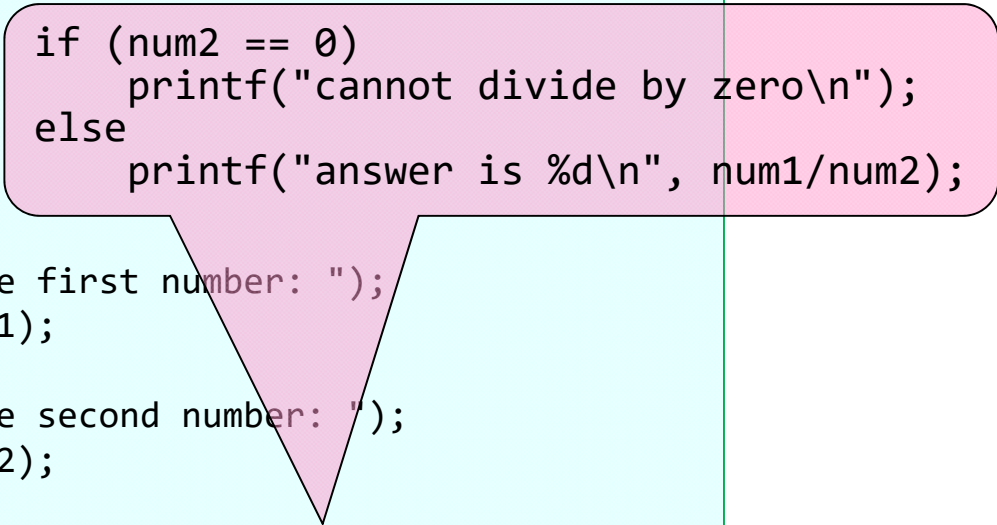
int main(void)
{
    int num1, num2;

    printf("enter the first number: ");
    scanf("%d", &num1);

    printf("enter the second number: ");
    scanf("%d", &num2);

    printf("answer is %d\n", num1/num2);

    return 0;
}
```



```
if (num2 == 0)
    printf("cannot divide by zero\n");
else
    printf("answer is %d\n", num1/num2);
```

☞ **What's wrong?** 실습 시간에 반드시 해 보세요. 재미있는 결과가..

☞ 소수점까지 계산하려면?

코드 블록 생성 (1/2)

- 연관된 문장들을 **block**으로 만들 수 있음
 - ✓ 논리적인 하나의 단위가 됨
 - ✓ {} 사용, code block은 독립된 스택 사용

```
#include <stdio.h>
int main()
{
    float num;
    int choice;

    printf("1:feet to meters, 2: meters to feet\n");
    printf("enter choice: ");
    scanf("%d", &choice);

    if (choice == 1) {
        printf("enter number of feet: ");
        scanf("%f", &num);
        printf("meters: %f\n", num / 3.28);
    }
    else {
        printf("enter number of meters: ");
        scanf("%f", &num);
        printf("feet: %f\n", num * 3.28);
    }
}
```

code block: logically single unit!!

☞ 이 프로그램의 문제는 ?
 항상 예외 처리 고려

■ 예제 수정

```
#include <stdio.h>
int main()
{
    int choice;

    printf("1:feet to meters, 2: meters to feet\n");
    printf("enter choice: ");
    scanf("%d", &choice);

    if (choice == 1) {
        float num;
        printf("enter number of feet: ");
        scanf("%f", &num);
        printf("meters: %f\n", num / 3.28);
    }
    else {
        float num;
        printf("enter number of meters: ");
        scanf("%f", &num);
        printf("feet: %f\n", num * 3.28);
    }
}
```

■ if 문

- ✓ 두 정수를 입력 받아 두 수의 차를 출력
- ✓ 두 수의 차는 함수로 작성 (차를 리턴 값으로 받음)
- ✓ 두 수의 차는 항상 큰 값에서 작은 값을 뺀 값임
- ✓ 예) 12와 7 입력시 값은 5, 4와 16 입력시 값은 12

```
Enter 2 integer : 4 16
```

```
Differnce : 12
```

■ for 반복문 형식

```
for (initialization; conditional test; operation)
    statement; or code block
```

■ 1에서 10까지의 합은?

```
#include <stdio.h>

int main()
{
    int i;
    int total = 0;

    for (i=1; i<=10; i=i+1)
        total = total + i;

    printf("total = %d\n", total);
}
```

initialization

conditional test

operation(s) at each iteration

for 반복문 사용 (2/6)

■ for 반복문의 다양한 예

```
// for 반복문의 예
#include <stdio.h>

int main()
{
    int i, total;

    total = 0;
    for (i=1; i<10; i=i+1)
        total = total + i;
    printf("total = %d\n", total);

    total = 0;
    for (i=1; i<10; i=i+3)
        total = total + i;
    printf("total = %d\n", total);

    total = 0;
    for (i=10; i>5; i=i-1)
        total = total + i;
    printf("total = %d\n", total);
}
```

```
// for 반복문의 예 계속

    total = 0;
    for (i=10; i<10; i=i+1)
        total = total + i;
    printf("total = %d\n", total);

    total = 0;
    for (i=1; i<10; i=i+1)
        if ((i % 2) == 0)
            total = total + i;
    printf("total = %d\n", total);

    total = 0;
    for (i=1; i<10; i=i+1);
        total = total + i;
    printf("total = %d\n", total);

    return 0;
}
```

☞ 좀 더 좋은 다른 방법 없을까?

■ 연습문제 1

- ✓ 1) 0 부터 100 까지 정수를 출력하시오.
- ✓ 2) 0 부터 100까지 정수 중에 짝수만 출력하시오.

```
#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 100; i = i + 1)
    {
        printf("%d ", i);
    }

    return 0;
}
```

■ 연습문제 2

- ✓ 0 이상의 정수 값을 입력 받아 정수 값까지 합을 구하시오.
- ✓ 조건: 최대 정수값은 50

```
input integer number : 10  
total = 55
```

■ 연습문제 3

- ✓ 5 이하의 정수 중에 소수(prime number)를 찾는 프로그램을 작성하시오.
- ✓ 소수를 찾는 부분은 함수로 작성하시오.

```
enter any integer: 15
15 in not prime number
enter any integer: 13
13 is prime number
enter any integer:
...
```

- initialization, condition test, operation은 생략 가능.
 - ✓ 무한 루프를 돌거나, for 루프를 한번도 수행하지 않게 할 수 있음
 - ✓ 콤마 연산자 사용 가능
- for 반복문은 매우 중요!!. 실제 실습을 많이 해 보시길
 - ✓ 6자리 lotto 복권 번호를 예상해 주는 프로그램 작성

```
#include <stdio.h>
#include <stdlib.h>          /* int rand(void) 함수를 위해 포함 */

main()
{
    int i, num;

    for (i=1; i<=6; i++) {
        num = rand();
        num = num % 46;
        printf("%d-th number = %d\n", i, num);
    }
}
```

☞ 위 프로그램은 3가지 정도의 문제가 있다. 수정해 보세요. ➔ 과제

■ “수 알아 맞추기”

- ✓ 컴퓨터에서 임의의 숫자 생성 (0~500 사이)
- ✓ 사용자가 숫자를 맞추기 위해서 10번 까지 숫자 입력
 - 입력한 값이 정답과 같으면 **Right** 출력
 - 입력한 값이 정답이 아니면 정답 보다 큰지 작은지 알려줌

I select a number from 0 to 500. Guess what the number is?

1-th Try.. Guess the number: 100

Sorry.. The number I've selected is larger than 100.

2-th Try.. Guess the number: 400

Sorry.. The number I've selected is larger than 400.

3-th Try.. Guess the number: 450

Sorry.. The number that I've selected is smaller than 450.

4-th Try.. Guess the number: 432

Sorry.. The number I've selected is larger than 432.

- printf 함수
 - ✓ 서식문자, Escape sequence 문자
- scanf 함수
 - ✓ printf 서식 문자와 유사
- 함수
 - ✓ 함수의 정의 및 필요성, 사용자 정의 함수
- 대표적 제어문인 if와 for 문을 이용한 프로그램 작성
- 코드 블록(code block)의 의미 이해
- for 반복문 이해