



# 시스템프로그래밍기초 실습

---

## Ch8. The Preprocessor

# 단순 매크로 정의

```
#define SECONDS_PER_DAY (60 * 60 * 24)
#define PI      3.14159
#define C      299792.458      /* speed of light in km/sec */

#define EOF      (-1)          /* typical end-of-file value */
#define MAXINT   2147483647    /* largest 4-byte integer */

#define ITERS    50             /* number of iterations */
#define SIZE     250           /* array size */
```

- 코드의 가독성과 명확성 및 신뢰성을 높임
- 코드의 이식성을 높임

# 전처리 결과 확인

```
#cc -E filename.c
```

✓ 모든 예제코드에 대해 위의 명령어 사용해보기

# 실습 예제 1) macro\_func.c

```
#include <stdio.h>

#define SQUARE(x) ((x) * (x))
#define ADD(x, y, z) ((x) + (y) + (z))
#define MUL(x, y, z) ((x) * (y) * (z))
#define MIN(x, y) ((x) < (y)) ? (x) : (y)
#define MIN4(x, y, z, w) MIN(MIN(x, y), MIN(z, w))

#define WRONG_SQUARE(x) x * x

#define PI 3.14159
#define AREA_CIRCLE(x) ((PI) * (x) * (x))

int main(void)
{
    int a = 3, b = 4, c = 5, d = 6;
    printf("SQUARE(a) : %d\n", SQUARE(a));
    printf("ADD(a, b, c) : %d\n", ADD(a, b, c));
    printf("MUL(a, b, c) : %d\n", MUL(a, b, c));
    printf("MIN(a, b) : %d\n", MIN(a, b));
    printf("MIN4(a, b, c, d) : %d\n", MIN4(a, b, c, d));

    printf("WRONG_SQUARE(a + b) : %d?\n", WRONG_SQUARE(a + b));

    printf("AREA_CIRCLE(1) : %f\n", AREA_CIRCLE(1));
    printf("AREA_CIRCLE(5) : %f\n", AREA_CIRCLE(5));

    return 0;
}
```

## 실습 예제 2) conditon.c

```
#include <stdio.h>

#define DEBUG

int main(void)
{
#ifdef DEBUG
    printf("Debug: %s %s %s %d\n",
           __DATE__, __TIME__, __FILE__, __LINE__);
#endif

#if (defined DEBUG || defined TEST) && !defined (VERSION)
    printf("Debug\n");
#endif

#if (VERSION == 123)
    printf("Version : 1.2.3\n");
#elif (VERSION == 127)
    printf("Version : 1.2.7\n");
#endif

    return 0;
}
```

# 컴파일 시에 매크로 설정하기

```
#gcc filename.c -D[매크로]
```

## ■ 사용 예제

```
#gcc condition.c -DVERSION=123
```

- `#define VERSION 123`
- 실행파일의 이름은 `a.out`

```
#gcc -o condition condition.c -DVERSION=127
```

- `#define VERSION 127`
- 실행파일의 이름은 `condition`

## 실습 예제 3) concat.c

```
#include <stdio.h>

#define GREETINGS(a, b, c) \
    printf("#a ", " #b ", and " #c ": Hello!\n")
#define WRONG_GREETINGS(a, b, c) \
    printf("a, b, c : Hello!\n")

#define EXEC(x) func_##x()

void func_A()
{
    printf("This is func_A()\n");
}

void func_B()
{
    printf("This is func_B()\n");
}

int main(void)
{
    GREETINGS(Alice, Bob, Carole);
    WRONG_GREETINGS(Alice, Bob, Carole);

    EXEC(A);
    EXEC(B);

    return 0;
}
```