

2021년도 2학기
대기수치모델링 개론 및 실습
실습 02주차

조교 : 김시윤 siyunk@snu.ac.kr

- 01~05주차 (09월) Programming Warm-up
- 06주차 (10월07일) Time Differencing
- 07주차 (10월14일) Spatial Differencing
- 08주차 (10월21일) Physics Parameterization (1) PBL
- 09주차 (10월21일) Physics Parameterization (2) Convection
- 10주차 (10월21일) Physics Parameterization (3) Cloud Macrophysics
- 11주차 (10월21일) Physics Parameterization (4) Cloud Microphysics
- 12주차 (10월21일) Physics Parameterization (5) Aerosol
- 13주차 (10월21일) Physics Parameterization (6) Radiation

- Linux 기본
- Fortran 기본 문법
- NCL 자료처리 및 생산

1) Xshell 설치 (실습 컴퓨터가 Mac / Linux OS 가 아닐 경우)

<https://www.netsarang.com/ko/free-for-home-school/>

2) 리눅스 서버 원격 접속

```
$ ssh -Y student_00@147.47.215.139 ↵
```

student_04 : 김정현

student_05 : 김준하

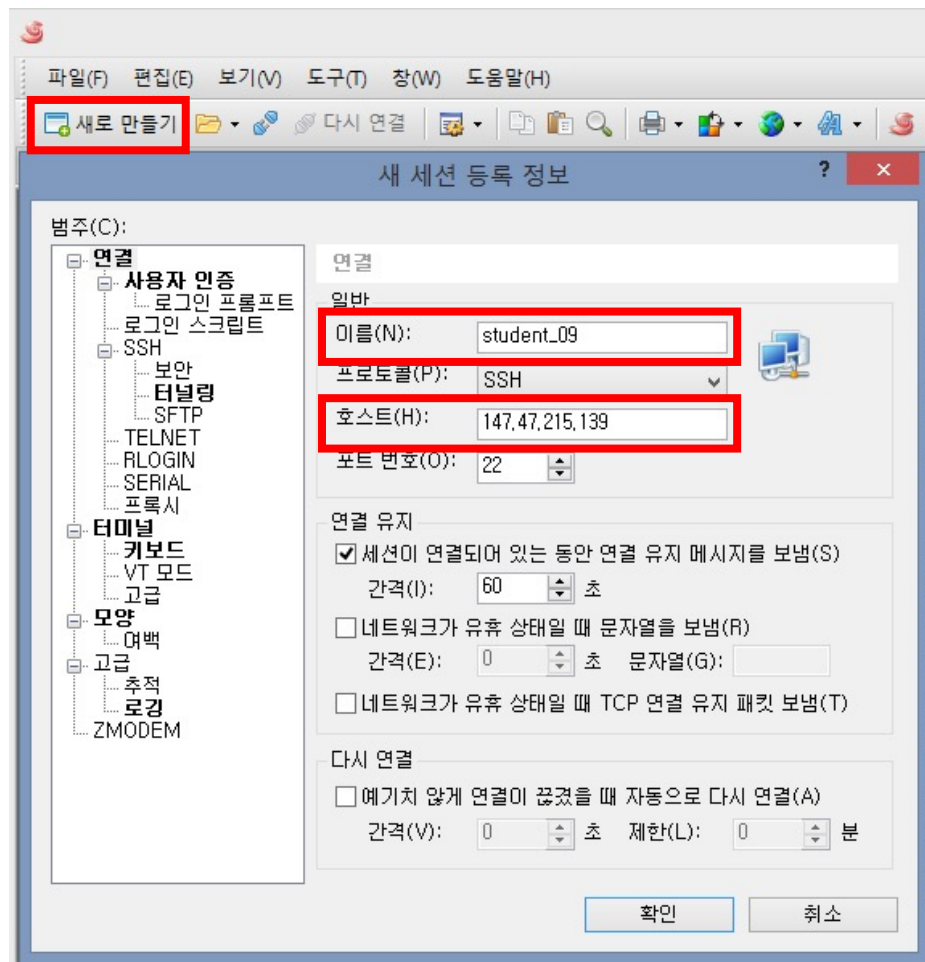
student_06 : 김준희

student_07 : 박경주

student_08 : 신상훈

3) 비밀번호 변경 (초기 비밀번호: student1234)

```
$ passwd ↵
```



- 디렉토리 / 파일 / 경로

- 리눅스 셸 기본 명령어

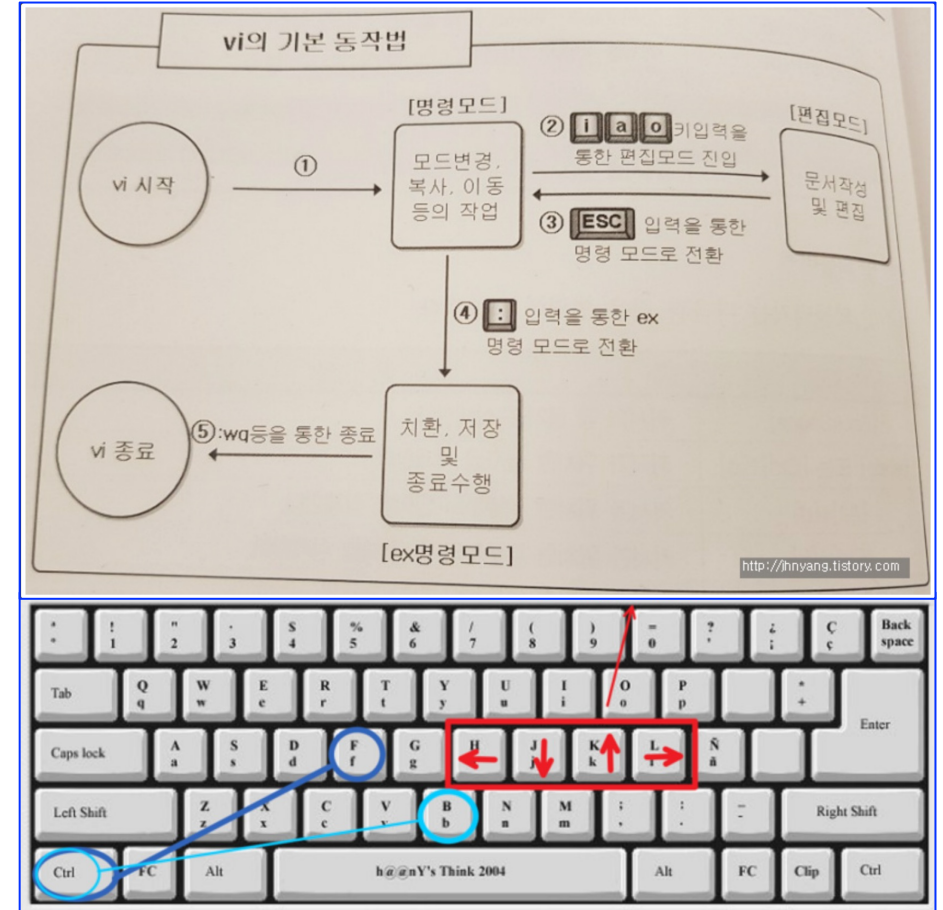
: ls / cd / pwd / mkdir / cp / mv / rm / rmdir
-option

- vi 편집기 (소스코드 작성)

: 기본 동작 (명령모드 / 입력모드 / EX모드)

- 작성한 프로그램 실행

: 컴파일러(전처리-컴파일-어셈블-링크)로 실행파일 생성
(i.e., \$ gfortran -o 실행파일이름 소스코드이름)
실행파일(exe) 실행(run)
(i.e., \$./실행파일이름)



```
$ vi week02_numo_practice.F90
```

```
program week02_numo_practice
```

```
implicit none
```

```
print *, "Hello World!"
```

```
end program week02_numo_practice
```

```
$ gfortran week02_numo_practice.F90
```

```
$ ./a.out
```

1) i/a/o 로 **입력모드** 진입

2) 소스코드 작성

3) ESC 로 편집모드 탈출 → **명령모드** 진입

4) 명령모드에서 콜론(:) 입력하여 **EX모드** 진입

5) :wq ← 저장(w)하고 vi editor 종료(q)

```
$ gfortran -o week2 week02_numo_practice.F90
```

```
$ ./week2
```

program 프로그램명

선언부

실행부

부프로그램부

end program 프로그램명

program program_name

```
implicit none  
real :: a, b, c
```

```
! comment  
a = 1.0; b = 2.0  
c = a + b  
print *, a, b, c &  
          a*b, b*c
```

contains

...

end program program_name

program_name :

- 의미 있는 이름으로 지정 (test, ex 등의 이름은 지양)
- 파일명(program_name.F90)과 동일하게 사용

F90 소스코드 작성 형식

- Fortran 코드는 1~132칼럼 중 임의의 위치에 작성 가능
- 영문 대소문자 구분하지 않음
- 특수문자 **!** 이후 작성 문장은 모두 주석
- 특수문자 **&** 는 다음 문장과 연속됨 의미
- 특수문자 **;** 는 한 문장의 끝 의미

```
program program_name
```

```
implicit none  
real :: a, b, c
```

```
! comment  
a = 1.0; b = 2.0  
c = a + b  
print *, a, b, c &  
          a*b, b*c
```

```
contains
```

```
...
```

```
end program program_name
```

변수는 자료형(data type)선언 후 사용해야 함

- 정수형 **integer**
- 실수형 **real**
- 복소수형 **complex**
- 논리형 **logical**
- 문자형 **character**

implicit none :

암묵적으로 **i, j, k, l, m, n** 으로 시작하는 모든 변수는 **integer** 로 취급됨

이를 해제하기 위해 프로그램 시작부에 **implicit none** 선언

변수의 형 선언문

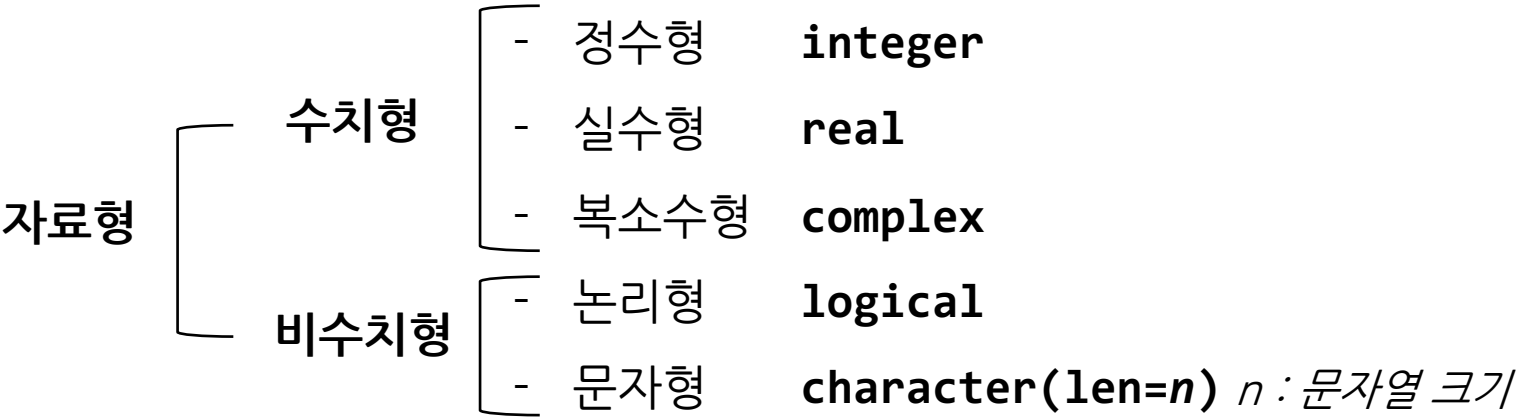
자료형 :: 변수명
자료형 :: 변수명1, 변수명2, ...

자료형 :: 변수명 = 값
자료형 :: 변수명1=값1, 변수명2=값2, ...

자료형, **parameter** :: 변수명 = 값

← 변수의 초기화 (실행문에서 변수 값 수정 가능)

← 상수변수 지정 (실행문에서 변수 값 수정 불가능)



표준출력문

- 사용법

```
print*, 출력 목록  
write(*,*) 출력 목록
```

** : 서식지정자 (다음시간에)*

- 예

```
distance=500  
print*,distance  
print*, "distance is", distance  
print*, "distance is", distance, "meters."
```

표준입력문

- 사용법

```
read*, 입력 목록  
read(*,*) 입력 목록
```

- 화면으로 입력 받음
- 입력될 자료는 read 문의 입력 목록에 있는 변수들과 개수 및 자료형 일치
- 각 입력되는 자료는 적어도 하나 이상의 공백이나 comma(,)로 분리
- 문자열을 입력할 때는 " " 가 필요 없음

IF 구문

```
if( 논리식 ) 실행문
```

```
if( 논리식 ) then  
    실행문블럭1  
else  
    실행문블럭2  
endif
```

```
if( 논리식1 ) then  
    실행문블럭1  
elseif( 논리식2 ) then  
    실행문블럭2  
else  
    실행문블럭3  
endif
```

CASE 구문

```
select case( 선택자 )  
    case( 값1 )  
        실행문블럭1  
    case( 값2 )  
        실행문블럭2  
    case default  
        실행문블럭3  
end select
```

DO 구문

횟수 제어형

```
do 루프변수 = 초기값, 최종값, 증감값  
    실행문블럭  
enddo
```

- 루프변수는 정수형이어야 함

```
do  
    실행문블럭1  
    if( 논리식 ) exit  
    실행문블럭2  
enddo
```

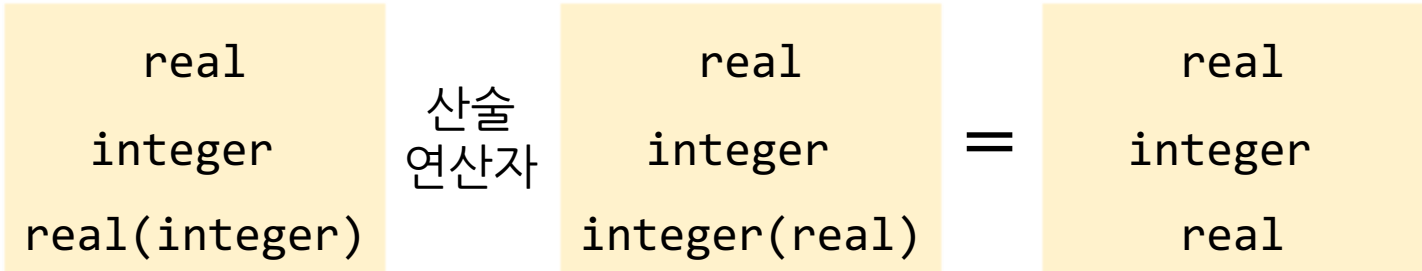
exit : 전체 loop 종료

```
do  
    실행문블럭1  
    if( 논리식 ) cycle  
    실행문블럭2  
enddo
```

cycle : 현재 loop 종료 후
다음 loop 실행

연산자의 종류 및 계산 우선순위

- 1) 산술연산자 : +, -, *, /, **
문자연산자 : //
- 2) 관계연산자 : >, <, >=, <=, ==(eq.), .ne.(/=)
- 3) 논리연산자 : .and. .or. .not.



내장 함수

Fortran 내장함수				
내 장 함수	수 학 적 의 미	인자형	결과	
ABS(X)	x	실수	실수	실수
ACOS(X)	cos-1(x)	실수	실수	실수
AIMAG(Z)	복소수 z의 허수 부분	복소수	실수	실수
ALOG10(X)	log10(X)	실수	실수	실수
ALOG(X)	ln(X)	실수	실수	실수
ASIN(X)	sin-1(x)	실수	실수	실수
ATAN(X)	tan-1(x)	실수	실수	실수
ATAN2(X,Y)	tan-1(x/y)	실수	실수	실수
CABS(Z)	z	복소수	실수	실수
CCOS(Z)	cos(z)	복소수	복소수	복소수
CEXP(Z)	exp(z)	복소수	복소수	복소수
CHAR(I)	정수 i를 문자로 변환	정수	문자	문자
CLOG(Z)	log(z)	복소수	복소수	복소수
CMPLX(X,Y)	x + iy로 변환	실수	복소수	복소수
COS(X)	cos(x)	실수	실수	실수
CONJG(Z)	z의 complex conjugate	복소수	복소수	복소수
COSH(X)	쌍곡선 cosh(x)	실수	실수	실수
CSIN(Z)	sin(z)	복소수	복소수	복소수
CSQRT(Z)	z의 제곱근	복소수	복소수	복소수
EXP(X)	exp(x)	실수	실수	실수
FLOAT(I)	정수 i를 실수로 변환	정수	실수	실수
IABS(I)	i	정수	정수	정수
ICHAR(CH)	문자 CH를 정수로 변환	문자	정수	정수
INDEX(ST1,ST2)	문자열 ST1내의 부분열(ST2)위치	문자열	정수	정수
INT(X)	실수 x를 정수로 변환	실수	정수	정수
LEN(STR)	문자열 STR의 길이	문자열	정수	정수
MAX(I1,I2,...)	최대값 선택	정수	정수	정수
MIN(I1,I2,...)	최소값 선택	정수	정수	정수
MOD(I1,I2)	i1/i2의 나머지	정수	정수	정수
NINT(X)	x를 반올림	실수	정수	정수
REAL(I)	정수 i를 실수로 변환	정수	실수	실수
REAL(Z)	복소수 z의 정수 부분	복소수	실수	실수
SIGN(X)	x의 부호	실수	실수	실수
SIN(X)	sin(x)	실수	실수	실수
SINH(X)	쌍곡선 sinh(x)	실수	실수	실수
SQRT(X)	x의 제곱근	실수	실수	실수
TAN(X)	tan(x)	실수	실수	실수
TANH(X)	쌍곡선 tanh(x)	실수	실수	실수

*내장함수에서 모든 삼각함수의 각도 단위(x)는 라디안이다.

1. 임의의 연도를 입력 받아 윤년 여부를 출력하는 프로그램을 작성하시오.
(if 문을 이용한 코드와 case 문을 이용한 코드 2개 작성)
2. 1부터 임의의 자연수까지의 합이 100을 넘을 때, 그 때의 합과 해당 자연수를 출력하는 프로그램을 작성하시오.
3. 주어진 정수의 자릿수를 헤아리기 위한 루프를 사용하는 프로그램을 작성하시오.

- 소스코드

돌아가는 완성 코드 제출 요망 (컴파일 안되면 감점 상당...)

파일 백업은 xftp (scp / git) 이용

- 이메일 제출

제목 : **[수치실습] 과제 01 (김시윤 2018-#####)**

첨부 파일 : 하나의 zip file

제출 기한 : 다음 수요일(9/15) 23:59 까지

받는 사람 : siyunk@snu.ac.kr (김시윤)