

## More Fortran: Keyword and optional arguments

If the interface of the procedure is explicit (e.g., in a module)

- one does not have to use all arguments in a procedure call
  - omissions at the end of the argument list ok
  - any omission ok if keyword (dummy variable name) is used
- one can use any order of the arguments if keywords are used

Example: keyword.f90

```
module test
```

```
contains
```

```
  subroutine keywordsub(a,b)
```

```
    integer, optional :: a
```

```
    integer, optional :: b
```

```
    if (present(a)) write(*,*) 'a = ',a
```

```
    if (present(b)) write(*,*) 'b = ',b
```

```
  end subroutine keywordsub
```

```
end module test
```

```
program testkeyword
use test
integer :: arg1,arg2

read(*,*)arg1,arg2

write(*,*)
call keywordsub(arg1)
write(*,*)
call keywordsub(b=arg2)
write(*,*)
call keywordsub(a=arg1,b=arg2)

end program testkeyword
```

If arg1=1 and arg2=2 are read in, this is the output:

a=1

b=2

a=1

b=2

# Fortran 90 intrinsic random number generator

`random_number(r)` initialized with `random_seed()`

```
integer :: i,size  
integer, allocatable :: seed(:)  
real :: r
```

```
call random_seed(size)  
allocate (seed(size))  
write(*,*)'give ',size,' random seeds '  
read(*,*)seed
```

```
call random_seed(put=seed)  
do i=1,10  
    call random_number(r)  
    write(*,*)r  
end do
```

```
call random_seed(get=seed)  
write(*,*)seed
```