Optional and Keyword Arguments

Fortran

- Subroutines and functions may take optional arguments. Such arguments need not be passed. If they are passed, they take on the passed value; otherwise they do not exist.
- They are declared with the OPTIONAL attribute.

Example

```
SUBROUTINE mysub(a,b,c,d)

REAL, INTENT(IN)                :: a

REAL, INTENT(OUT)               :: b

REAL, OPTIONAL, INTENT(IN)      :: c, d
```

Using Optional Arguments

- The call to the previously-defined subroutine could be

  ```
  call mysub(a,b)
  ```

  in which case c and d would have no values and the subroutine would need to handle that situation appropriately. The call could also be

  ```
  call mysub(a,b,c)
  ```

  or

  ```
  call mysub(a,b,c,d)
  ```

  depending on how many of the optional arguments needed to be passed.

Keyword Arguments

- Suppose it were desired to change d but not c in the preceding subroutine. The c parameter can be skipped by using a keyword argument; the optional argument is called as

  ```
  dummy=actual
  ```

  where dummy is its name in the program unit where it is defined, and the actual argument is its name in the calling program unit.

  Example:

  ```
  call mysub(aa, bb, d=dd)
  ```

The PRESENT Intrinsic

- The PRESENT() function tests whether a particular optional argument is present in the argument list of the caller. If it is not present, defaults can be set or other action taken.

Example

```
IF (PRESENT(d)) then

   dd=d

ELSE

   dd=3.14

ENDIF
```