FORTRAN - VARIABLES

http://www.tutorialspoint.com/fortran/fortran_variables.htm

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable should have a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

The name of a variable can be composed of letters, digits, and the underscore character. A name in Fortran must follow the following rules:

- It cannot be longer than 31 characters.
- It must be composed of alphanumeric characters *allthelettersofthealphabet*, *andthedigits*0*to*9 and underscores _.
- First character of a name must be a letter.
- Names are case-insensitive.

Based on the basic types explained in previous chapter, following are the variable types:

Type	Description
Integer	It can hold only integer values.
Real	It stores the floating point numbers.
Complex	It is used for storing complex numbers.
Logical	It stores logical Boolean values.
Character	It stores characters or strings.

Variable Declaration

Variables are declared at the beginning of a program *orsubprogram* in a type declaration statement.

Syntax for variable declaration is as follows:

```
type-specifier :: variable_name
```

For example,

```
integer :: total
real :: average
complex :: cx
logical :: done
character(len=80) :: message ! a string of 80 characters
```

Later you can assign values to these variables, like,

```
total = 20000
average = 1666.67
done = .true.
message = “A big Hello from Tutorials Point”
cx = (3.0, 5.0) ! cx = 3.0 + 5.0i
```

You can also use the intrinsic function **cmplx,** to assign values to a complex variable:

```
cx = cmplx (1.0/2.0, -7.0) ! cx = 0.5 – 7.0i
cx = cmplx (x, y) ! cx = x + yi
```

Example

The following example demonstrates variable declaration, assignment and display on screen:

```
program variableTesting
implicit none

   ! declaring variables
   integer :: total
   real :: average
   complex :: cx
   logical :: done
   character(len=80) :: message ! a string of 80 characters

   !assigning values
   total = 20000
   average = 1666.67
   done = .true.
   message = "A big Hello from Tutorials Point"
   cx = (3.0, 5.0) ! cx = 3.0 + 5.0i

   Print *, total
   Print *, average
   Print *, cx
   Print *, done
   Print *, message

end program variableTesting
```

When the above code is compiled and executed, it produces the following result:

```
20000
1666.67004
(3.00000000, 5.00000000 )
T
A big Hello from Tutorials Point
```