



Cheat Sheet - Format Strings

This sheet is intended to provide an overview of Fortran string input and output formats.

- 01 Format strings

- String that represents how to display other variables
- Mixture of **format specifiers** that describe how to display individual variables and **fixed strings** that are printed as is when the format is used
- Used in **PRINT** and **WRITE** functions
 - Also used in **READ** to input data but is usually unnecessary and can be quite messy and complex

- 02 General format

- Each format string begins and ends with round brackets and all of the format specifiers and fixed strings in the format string are separated by commas
 - "(A, I3.3)"

- 03 Format specifiers

- A - Write a string variable
 - A{w} - where **w** is an integer describing how many characters to write
- I - Write an integer variable. Must be either
 - I{w} where **w** is an integer saying how many characters to output. If you don't specify enough characters to represent the number then the output will be replaced by *s. Note that the length of the output string **will** be **w** with spaced added to the front if needed
 - e.g. PRINT, "(I4)", 10 will output " 10"
 - I{w.m} where **w** is the number of characters to output and **m** is the number of characters that must contain numerical values. If **w** is too small to represent the number then the output will be all *s. If **m** is larger than **w** that is an error and your code will fail, typically with a runtime error.

- PRINT, "(I4.3)", 10 will output " 010"
PRINT, "(I6.6)", 10 will output "000010"
- F{w.d} - Write a real variable in decimal form
 - **w** is the number of characters to output in total and **d** is the number of digits to the right of the decimal point. **w** must be greater than **d+1** or the output will be replaced by *s (The +1 is because of the decimal point itself). **w** must be large enough to display all of the digits before the decimal point and the decimal point itself or the output will be replaced by *s
 - PRINT "(F6.2)", 1.2345 will output " 1.23"
PRINT "(F6.4)", 1.2345 will output "1.2345"
PRINT "(F6.2)", 123.4567 will output "123.45"
PRINT "(F6.3)", 123.4567 will output "*****"
- E - Write a real variable in exponential form. All numbers are written in exponential form with no non-zero digits written to the left of the decimal point. Must be either
 - E{w.d} - where **w** is the number of characters to output in total and **d** is the number of digits to the right of the decimal point. **w** must be at least large enough to contain a decimal point, **d** digits of the number, the "E" in the representation of the exponential number, the exponent sign symbol (+ or -) and the digits of the exponent. If the number is negative then an extra character must be specified in **w**. The exponent will always have at least two digits but will use more if needed. A leading zero will be added if possible but will be removed if there are insufficient digits specified in **w** to display it. If insufficient characters are selected then the output will be *s. Rounding will be applied to the number before printing. You may not use this form if the absolute value of the exponent is larger than 999
 - PRINT "(E8.3)", 1.2345 will output ".123E+01"
PRINT "(E8.2)", 1.2345 will output "0.12E+01"
PRINT "(E10.4)", 1.2345 will output "0.1235E+01" (not 0.1234 because of the rounding)
 - E{w.d}E{e} - where **w** is the number of characters to output in total, **d** is the number of digits to the right of the decimal point and **e** is the number of digits

in the exponent. Same rules as E{w.d} for number of characters required but now the number of characters needed for the exponent is known. You must use this form if the absolute value of the exponent is larger than 999

- PRINT "(E8.3E1)", 1.2345 will output "0.123E+1"
PRINT "(E10.3E3)", 1.2345 will output "0.123E+001"
- ES - Write a real variable in scientific form. All numbers are represented in exponential form with a single digit to the left of the decimal point. Must be either
 - ES{w.d} - where **w** is the number of characters to output in total and **d** is the number of digits to the right of the decimal point. **w** must be at least large enough to contain the leading digit, a decimal point, **d** digits of the number after the decimal point, the "E" in the representation of the exponential number, the exponent sign symbol (+ or -) and the digits of the exponent. If the number is negative then an extra character must be specified in **w**. The exponent will always have at least two digits but will use more if needed. If insufficient characters are selected then the output will be *s. Rounding will be applied to the number before printing. You may not use this form if the absolute value of the exponent is larger than 999
 - PRINT "(ES8.3)", 1.2345 will output "*****"
 - PRINT "(ES9.3)", 1.2345 will output "1.234E+00"
 - PRINT "(ES8.2)", 1.2345 will output "1.23E+00"
 - PRINT "(ES10.4)", 1.2345 will output "1.2345E+00"
 - PRINT "(ES10.3)", 1.2345 will output "1.235E+00" (rounding behaviour again)
 - ES{w.d}E{e} - where **w** is the number of characters to output in total, **d** is the number of digits to the right of the decimal point and **e** is the number of digits in the exponent. Same rules as ES{w.d} for number of characters required but now the number of characters needed for the exponent is known. You must use this form if the absolute value of the exponent is larger than 999
 - PRINT "(ES8.3E1)", 1.2345 will output "1.235E+0" (rounding again)
 - PRINT "(ES10.3E3)", 1.2345 will output "1.235E+000" (still more rounding)
- EN - Write a real variable in engineering form. All numbers are represented in exponential form with a 1-3 digits to the left of the decimal point and the exponent always a multiple of 3. Must be either

- EN{w.d} - where **w** is the number of characters to output in total and **d** is the number of digits to the right of the decimal point. **w** must be at least large enough to contain the leading 1-3 digits, a decimal point, **d** digits of the number after the decimal point, the "E" in the representation of the exponential number, the exponent sign symbol (+ or -) and the digits of the exponent. If the number is negative then an extra character must be specified in **w**. The exponent will always have at least two digits but will use more if needed. If insufficient characters are selected then the output will be *s. Rounding will be applied to the number before printing. You may not use this form if the absolute value of the exponent is larger than 999

- PRINT "(EN9.3)", 1.2345 will output "1.235E+00"
- PRINT "(EN9.3)", 12.345 will output "*****" (now have two leading digits)
- PRINT "(EN11.3)", 12.345 will output " 12.345E+00"
- PRINT "(EN9.3)", 123.45 will output "*****" (now have three leading digits)
- PRINT "(EN11.3)", 123.45 will output "123.45E+00"
- PRINT "(EN9.3)", 1234.5 will output "1.234E+03"

- EN{w.d}E{e} - where **w** is the number of characters to output in total, **d** is the number of digits to the right of the decimal point and **e** is the number of digits in the exponent. Same rules as EN{w.d} for number of characters required but now the number of characters needed for the exponent is known. You must use this form if the absolute value of the exponent is larger than 999

- PRINT "(EN9.3E1)", 12.345 will output "12.345E+0"
- PRINT "(EN11.3E3)", 12.345 will output "12.345E+000"

- 04 Combining elements

- Remember that both "" and '' are valid ways of marking strings in Fortran but you can have one inside the other so '' 'Test' '' is a string containing the text 'Test' (including the inner set of quotes). Similarly ' "Test" ' contains the text "Test"
- CHARACTER(LEN=3) :: name = "Bob"
- INTEGER :: age = 99
- PRINT "('Hello ', A, '. You are ', I2, ' years old')", name, age

Hello Bob. You are 99 years old

- All of the parts of the format string are separated by commas and the fixed strings are specified by putting them inside the type of quote mark that I didn't use to start the format string.