example

example

example

plies the `formatSpec` to all elements of arrays `A1,...An` example
. `fprintf` uses the encoding scheme specified in the

a and displays the results on the screen. example

f bytes that `fprintf` writes, using any of the input example

collapse all

the screen.

```
%8.3f mm\n';
```

he first value in each line of output is a floating-point number with a
fter the decimal point. `%8.3f` in the `formatSpec` input specifies that the
g-point number with a field width of eight digits, including three digits
er that starts a new line.

n fractions to integer values.

in the vector, `round(a)`, as a signed integer. `\n` is a control character

to a text file called `exp.txt`.

```
ɔ(x)');
```

nd `exp(x)`, and the second call prints the values from variable `A`.

epad, use `'\r\n'` instead of `'\n'` to move to a new line. For example,
g:

```
exp(x)');
);
```

ions, and Microsoft Word and WordPad recognize `'\n'` as a newline

ɯmand.

ytes written.

ıumber of bytes that `fprintf` writes.

```
d %5d\n',A)
```

e.

ımand.

e) on the screen.

```
ite,title)
```

values of the variables `site` and `title`, should be printed as strings.

collapse all

g:

ıg containing formatting operators. `formatSpec` also can include

n, %, and ends with a conversion character. The conversion character
ier, flags, field width, precision, and subtype operators between % and
between operators and are shown here only for readability).

Conversion character

Subtype

Precision

mat numeric and character data as strings.

ın	Details
	Base 10
	Base 10
	Base 8 (octal)
	Base 16 (hexadecimal), lowercase letters `a–f`
	Same as `%x`, uppercase letters `A–F`
	Fixed-point notation (Use a precision operator to specify the number of digits after the decimal point.)
	Exponential notation, such as `3.141593e+00` (Use a precision operator to specify the number of digits after the decimal point.)
	Same as `%e`, but uppercase, such as `3.141593E+00` (Use a precision operator to specify the number of digits after the decimal point.)
	The more compact of `%e` or `%f`, with no trailing zeros (Use a precision operator to specify the number of significant digits.)
	The more compact of `%E` or `%f`, with no trailing zeros (Use a precision operator to specify the number of significant digits.)
	Single character
	String of characters

sion, and subtype operators further define the format of the output

t list. Use the syntax *n*$, where *n* represents the position of the value in

s inputs `'A'`, `'B'`, `'C'` as follows: `C B A B`.

aracter (+ or –) for any value.
the value.
zeros before the value.
eric conversions: rint `0`, `0x`, or `0X` prefix. rint decimal point even when precision is 0. ot remove trailing zeros or decimal point.

ne field width operator can be a number, or an asterisk (*) to refer to

is equivalent to `('%*d', 12, intmax)`.

es before the value unless otherwise specified by flags.

nber of digits to the right of the decimal point **mple:** `'%.4f'` prints `pi` as `'3.1416'`
nber of significant digits **mple:** `'%.4g'` prints `pi` as `' 3.142'`

or an asterisk (*) to refer to an argument in the input list.

equivalent to `('%*.*f', 6, 4, pi)`.

r for floating-point values that exceeds the precision of the input
natch the input values to the precision you specified. The result
d operating system.

t a subtype. The subtype operator immediately precedes the
e conversions that can use subtypes.

type and Conversion aracter	**Output Value Type**
or %bX	Double-precision hexadecimal, octal, or decimal value **Example:** %bx prints `pi` as 400921fb54442d18

or %tX	Single-precision hexadecimal, octal, or decimal value **Example:** %tx prints pi as 40490fdb
or %li or %lX	64-bit value
or %hi or %hX	16-bit value

efore a percent sign, %, or after a conversion character. The text can

s ordinary text. This table shows how to represent special characters

CONTENTS	Close **Representation**
	''
	%%
	\\
	\a
	\b
	\f
	\n
	\r
	\t
	\v
decimal number, N	\xN
number, N	\N

atting Operators

mponent of complex numbers.

t the data, such as a string conversion for a numeric value, MATLAB ses %e.

+00.

ger values, MATLAB converts values that correspond to valid

BC.

ay

calar, vector, matrix, or multidimensional array.

int32 | int64 | uint8 | uint16 | uint32 | uint64 | logical | char

collapse all

l as a scalar. When writing to a file, nbytes is determined by the
screen, nbytes is the number of characters displayed on the screen.

collapse all

sscanf and fscanf differ from the formats for the writing functions
ons do not support a precision field. The width field specifies a
ading.

rogramming Language, Second Edition, Prentice-Hall, Inc., 1988.

ing Language C," ANSI, 1430 Broadway, New York, NY 10018.

| fseek | ftell | fwrite | sprintf