

Formatting Output

Formatting – Location

By default, output occurs one line at a time, and is always aligned with the left side of the screen.

It is possible to manipulate the output in a crude way using extra spaces and new lines.

Turing also provides a built-in method (procedure) to accomplish the same task in a much more controlled and direct manner.

```
locate (row, column : int)
```

The `locate()` procedure accepts two parameters (row and column) and moves the cursor to that location on the screen. The next put command will start at that location.

Useful parameters that can be used with `locate()` are `maxrow` and `maxcol`.

See “locate” in the Turing Help File for more information and a coding example.

```
colour ( setColour : int )
```

It is possible to change the colour of any text output to the screen using the `colour()` procedure (also spelled `color` in Turing).

After calling `colour()`, all new text will be displayed in the newly selected colour.

`colorback(setColour : int)` will change the background colour of text.

Colour values range from 0 to `maxcolor` (which is usually 15, 255, or possibly higher)

Example – Locate & Colour

```
var row, column, setColour : int
loop
  randint (row, 1, maxrow - 1)
  randint (column, 1, maxcol)
  randint (setColour, 0, maxcolor)
  color (setColour)
  locate (row, column)
  put "*" ..
  % Use .. to avoid clearing end of line
  delay (100)
end loop
```

Formatting – Numbers

When outputting multiple integer values (e.g., line numbers), they won't naturally line up. This can be forced with spacing, but there is a more elegant option available.

```
8 : Line 8  
9 : Line 9  
10 : Line 10
```

or

```
8 : Line 8  
9 : Line 9  
10 : Line 10
```

Formatting Numbers – Width

It is possible to specify the total width of a number using the `put` statement. If we specify a width less than the width of the number, the whole number is still output

`put number : width`

```
put 123 : 1      123
put 123 : 2      123
put 123 : 3      123
put 123 : 4       123
put 123 : 5       123
```

Numbers will
right-justify to fit
the width

Formatting Strings – Width

The same formatting can be used with a string. As with numbers, it is not possible to output less than the whole string this way.

put string : width

```
put "hi" : 1, 9      hi9
put "hi" : 2, 9      hi9
put "hi" : 3, 9      hi 9
put "hi" : 4, 9      hi  9
put "hi" : 5, 9      hi   9
```


Formatting Reals – Decimal Places

By default, in Turing, decimal values will round to 6 decimal places. Take this into account when selecting the width.

put number : width : decimalPlaces

```
put sqrt(2) : 1 : 1.414214
put sqrt(2) : 1 : 1
put sqrt(2) : 2 : 1
put sqrt(2) : 3 : 1
put sqrt(2) : 4 : 1
put sqrt(2) : 5 : 1
```