

Formatting Strings

Formatting Strings Using `format`

1. Read a string of arbitrary length, and display it centred in a block 60 characters wide.
2. Repeat Q1, but add a padding character, like '\$', to the string.
3. Repeat Q1, but ask the user how they want to justify the string: left, right or centred.
4. Read a very large integer (at least 12 digits), and display it in two ways: by separating thousands using a comma, and in scientific notation. For example, the number 123456789012 would be displayed as 123,456,789,012 and 1.234568e+08 (representing 1.234568×10^8).
5. Generate 10 strings of asterisks, each with a random length between 1 and 20 characters. Display all strings in a centred column of width 20.
6. Read a positive integer value and display it in both binary and hexadecimal forms.
7. Display a random integer between 1 and 99 999 inside of a box, as shown below. The value should be centred in the box, and the total width of the box should be 9 characters.

```
+-----+
|  375  |
+-----+
```

8. Generate a table of the first 20 natural numbers and their squares. Each row of the table should contain a value and its square, with the two values each right-aligned in their columns, as shown:

```
...      ...
 3         9
 4        16
 5        25
...      ...
```

9. Read two integers from the user and calculate their sum. Display the calculation vertically, as in the example below, with all values right-aligned.

```
   421
+ 3516
-----
 3937
```