

Introduction to Programming in Turing

More Advanced
Output of Information

Output in Turing

Joining Multiple Lines

It is possible to join the output from two different commands on a single line. Use the `put` command, but add `..` to the end of the line.

Note: The last `put` command should not have the `..` on the end.

```
put "To be or not to be, " ..  
put "that is the question."
```



Output in Turing

Joining Multiple Lines

The ".." operator can join more than two lines (although this example is a terrible waste). Don't forget that the last line ends without the dots.

```
put "To " ..  
put "be " ..  
put "or " ..  
put "not " ..  
put "to " ..  
put "be "
```

Output in Turing

Joining Multiple Elements

It is also possible to join multiple elements using a comma on a single output line. This is not useful when using only strings, but as we learn more programming, it will become more useful.

```
put "To ", "be ", "or " ..  
put "not ", "to ", "be "
```

Output in Turing

Output Quotation Marks

Quotation marks are used to represent strings in Turing. They are also used to represent an actual quotation in written text.

How can we display quotation marks as part of our Turing output?

```
put ""quote?""
```

Although a reasonable guess, this fails. You can try this yourself.

Output in Turing

Output Quotation Marks

When Turing sees a quotation mark, it interprets it as a string. What we need is a way to tell Turing to interpret it in a second way.

Turing uses another character, the slash `\`, to indicate an alternate, or secondary, meaning.

```
put " a "
```

```
put " \" "
```

```
put " \\" "
```

Output in Turing Escape Sequence

The slash `\`, indicates an alternate, or secondary, meaning. This is commonly referred to as an escape sequence (i.e., "escape" from the primary meaning). Here are some examples:

symbol	primary meaning	escape sequence	secondary meaning
<code>"</code>	string output	<code>\"</code>	output quotation mark character
<code>n</code>	output letter 'n'	<code>\n</code>	add new line to output
<code>\</code>	escape sequence	<code>\\</code>	output slash character