

Outputs

To output to the screen we use the **print** procedure. Languages might differ in how you print to the screen, but in general, you do not have a space between *print* and the bracket. Inside the brackets you can place any string.

```
OUTPUT 'hello'
```

```
hello
```

It is also possible to print the value stored inside a variable. When we print variables we do not use quotes around the variable name as the variable already knows it is a string.

```
name ← 'Amy'
OUTPUT name
```

```
Amy
```

It is possible to combine two or more strings together. This process is called **concatenation**. In the example below we concatenate the string "Name: " with the variable *name* and the variable *surname*.

```
firstName ← 'Amy'
surname ← 'Smith'
OUTPUT 'Name: ' + name + surname
```

```
Name: Amy Smith
```

If we want to make a new line there isn't a symbol which we can type in. This means we have to use an **escape character**. The escape character for a new line is `\n`

```
OUTPUT 'some text\n on a new line'
```

```
some text
on a new line
```

Inputs

To get input from the user we use the **input** function. This can have a message which is printed to the user if you wish. The function will return a string containing whatever the user typed in on the keyboard. This can then be stored in a variable. In the example below the string "Jack" will be stored in the variable called *name*.

```
OUTPUT 'Type in your name: '
name ← USERINPUT
```

```
Type in your name: Jack
```

Type conversion

Variables have specific types of data which they can store. Sometimes, we need to convert between one data type, such as a string, and another data type, such as an integer. Converting one data type to another is known as **type conversion**.

```
STRING_TO_INT('7')
STRING_TO_REAL('7.3')
INT_TO_STRING(7)
REAL_TO_STRING(7.3)
```

```
7
7.3
'7'
'7.3'
```

STRING_TO_INT and STRING_TO_REAL will convert a string to an integer or real number.
INT_TO_STRING and REAL_TO_STRING will convert an integer or real number to a string.

Q 13

1. What instruction will output the text *Goodbye* onto the screen?
_____ [1]

2. The following instructions have a syntax error in. Rewrite them. [1]

OUTPUT Hi everyone → _____

OUTPU 'Hello' → _____

OUTPUT 'Good morning' → _____ [3]

3. What instruction will output the text *I Love CS* onto the screen?
_____ [1]

4. What instruction will allow the user to enter some text, and then store the answer in a variable called school?
_____ [1]

5. What is the output of this program?

```
name ← 'Sam'
age ← 15
OUTPUT 'Details: '
OUTPUT name
OUTPUT age
```


_____ [3]

6. What does \n do in programming? _____ [1]

Inputs, Outputs & Type Conversion - Questions

7. Match the functions on the left with what they return on the right.

STRING_TO_INT('19')	19.9	
INT_TO_STRING(19)	19	
STRING_TO_REAL('19.9')	'19'	[3]

8. Match the instruction on the left to the result output to the screen on the right.

OUTPUT 'help'	Syntax error	
OUTPUT 'he\nlp'	help	
OUTPUT 'help'	he lp	[3]

9. Changing one data type to another is known as what?
_____ [1]

10. Look at the following code:

```
name ← USERINPUT
age ← USERINPUT
age ← age + 1
OUTPUT 'Your age is ' + age
```

- a) What does the second + operator do? _____ [1]
- b) Which line of code is there an error on? _____ [1]
- c) What is causing this error? Fill in **one** circle. [1]
 - Incorrect data type
 - Missing bracket
 - Missing ' ' [1]