

Lesson 2

Input and Variables

Input is the term used to describe the transfer of information from the keyboard (or a disk) to the computer.

We also use the word **read** for input e.g read information from the keyboard.

The question arises – where do we store the information that we read in. This introduces the concept of **variables**. This is one of the fundamental concepts in programming

A variable is a container for information.

This means that we can store information in a variable. It is called a variable because at any time we can change (**vary**) the information it stores.

We **give each variable a unique name**, which we use to identify it. The following are examples of variable names we could use in a Python program:

```
colour
my_age
pension_age
name
taxcode
tax_rate
temperature
name
hourly_pay
```

You should choose **meaningful** variable names because it makes your programs easier to understand.

The statement **input** is used to read information from the keyboard.

- It displays a message to prompt the user for input
- It then waits for the user to type something, followed by the Return/Enter key
- The input is treated as a **string** and is stored in the variable we specify:

```
colour = input("Enter a colour: ")
```

- The message `Enter a colour:` is displayed and the text entered by the user on the keyboard is stored in the variable `colour`.

The program below reads a colour from the keyboard and then displays what you entered

```
colour = input("Enter a colour: ")  
print( colour )
```

If we execute the program the following appears on the screen where the **bolded** text is what the user entered on the keyboard. We will use this convention of showing input from the user in **bold** in the slides.

```
Enter a colour: blue  
blue
```

Note that a single `print` can display a string and the value of a variable:

```
colour = input("Enter a colour: ")  
print("You entered:", colour )
```

If we execute the program the following appears on the screen:

```
Enter a colour: blue
```

```
You entered: blue
```

Make sure you understand the difference between:

```
print( "colour" )
```

and

```
print( colour )
```

In the first case, the 6 character string *colour* is displayed on the screen.

In the second case, the **value** of the variable called *colour* is displayed for example the word *blue* or whatever value the user has given the variable like *red*, *pink* and *orange*. You can store **many words** in a string variable.

Rules for Variable Names

Python has rules on how you name variables:

- A variable name can only contain the following:
 - **letters** (lowercase and uppercase, ie a–z and A–Z)
 - **digits** (0–9)
 - the underscore “_” character
- A variable name **cannot start** with a digit
- Variable names in Python are **case-sensitive**

`colour` and `Colour` are **different variables**

- There are a number of **reserved** words or **keywords** that have built-in meanings in Python and **cannot** be used as variable names (e.g. `if`, `return`, `def`, `del`, `break`, `for`, `in`, `else`, `while`, `import`)

The following are legal or valid variable names in Python:

```
Colour, name,  
firstname, surname,  
class1, class_602,  
first_name, second_name,  
address_line1, address_line2
```

The use of the underscore character “_” is very useful in creating meaningful names made up of 2 or more words.

Do not confuse “_” with “-” (minus sign).

The following are **not** valid variable names

```
address-line1  
second-name
```

Another example of input

```
# Read the users name and age and display them

name = input("Enter your name: ")
print("Hello", name )
age = input("Enter your age: ")
print("Wow are you", age, "years old!" )
```

Running this program:

```
Enter your name: Joe
Hello Joe
Enter your age: 55
Wow are you 55 years old!
```

Time to practice !

- Copy all of the examples from the slides above and get them to run in your Python environment.
- Then complete the exercises from the Handbook and get them to run.
- Finally carry out the assignments from the Handbook and get them to run.