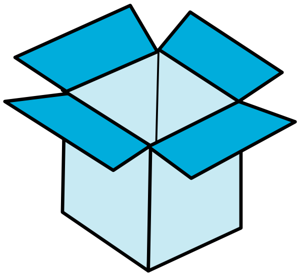
Employing variables



In programming, a "variable" is a container in which a data value can be stored within the computer's memory. The stored value can then be referenced using the variable's name. The programmer can choose any name for a variable, except the Python keywords, and it is good practice to choose meaningful names that reflect the variable's content.

Data to be stored in a variable is assigned in a Python program declaration statement with the = assignment operator. For example, to store the numeric value eight in a variable named "a":

```
a = 8
```

The stored value can then be referenced using the variable's name, so that the statement **print(a)** will output the stored value **8**. That variable can subsequently be assigned a different value, so its value can vary as the program proceeds – hence the term "variable".

In Python programming, a variable must be assigned an initial value ("initialized") in the statement that declares it in a program – otherwise the interpreter will report a "not defined" error.

Multiple variables can be initialized with a common value in a single statement using a sequence of = assignments. For example, to initialize variables named "a", "b" and "c", each with a numeric value of eight, like this:

```
a = b = c = 8
```

Alternatively, multiple variables can be initialized with differing values in a single statement using comma separators. For example, to initialize variables named "a", "b" and "c" with numeric values of one, two and three respectively, like this:

```
a , b , c = 1 , 2 , 3
```

Some programming languages, such as Java, demand you specify what type of data a variable may contain in its declaration. This reserves a specific amount of memory space and is known as "static typing". Python variables, on the other hand, have no such limitation and adjust the memory allocation to suit the various data values assigned to their variables ("dynamic typing"). This means they can store integer whole numbers, floating-point numbers, text strings, or Boolean values of **True** or **False** as required.

Don't forget

String data must be enclosed within quote marks to denote the start and end of the string.

Hot tip

Programming languages that require variable types to be specified are alternatively known as "strongly typed" whereas those that do not are alternatively known as "loosely typed".

Optionally, comments can be added to your Python scripts to describe the purpose of statements or sections of code if preceded by a **#** hash character. Everything following the **#** hash character up to the end of the line is ignored by the Python interpreter. It is useful to comment your code to make its purpose clear to others or when revisiting the code yourself later.

var.py

1. Launch a plain text editor, then declare and initialize a variable – then display its stored value

```
# Initialize a variable with an integer value.
var = 8
print( var )
```

2. Next, assign a new value and display that stored value

```
# Assign a float value to the variable.
var = 3.142
print( var )
```

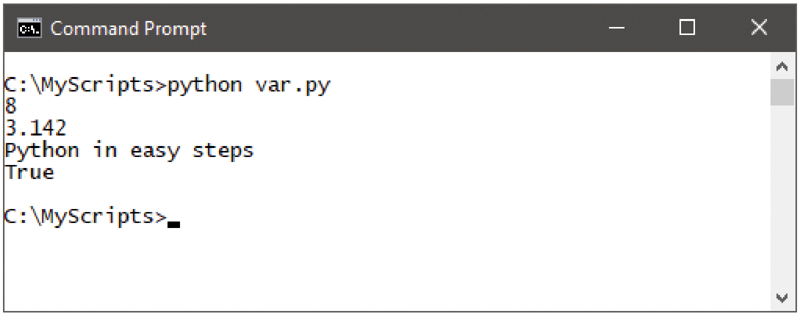
3. Now, assign a different value and display the stored value

```
# Assign a string value to the variable.
var = 'Python in easy steps'
print( var )
```

4. Finally, assign another value and display the stored value

```
# Assign a boolean value to the variable.
var = True
print( var )
```

5. Save the file in your scripts directory, then open a Command Prompt window there and run the program – to see the stored values output as the program proceeds



```
C:\MyScripts>python var.py
8
3.142
Python in easy steps
True

C:\MyScripts>_
```

Hot tip

Multi-line comments can be added to a script if enclosed between triple quote marks “““...””” .