Data and Variables

A variable is a named memory location that is used to store data.

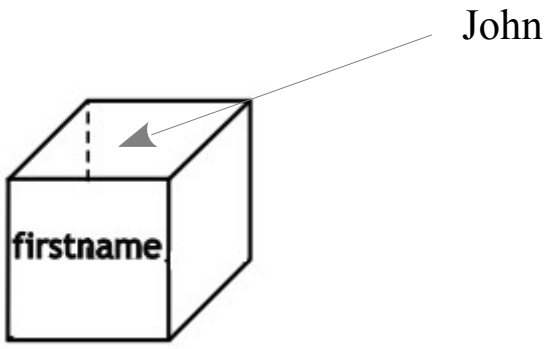
Data is any value or information that you wish to store.

Variables must be declared before they are used.

VARIABLE NAME= EXPRESSION

firstname = “John”

It may help to imaging memory as a warehouse full of boxes. Each time we wish to store information we take a box and put the information into the box. The variable is the name on the box and the expression is the value which goes into it.



When you assign an expression to a variable (memory location) using the equals sign it is called an ASSIGNMENT STATEMENT. The expression on the right hand side is evaluated and stored in the variable on the left hand side of the equals (=)

To create a new variable just assign it a value.

Last_name = “Doe” # creates the variable Last_name and assigns it the value “Doe”

In python **variables are untyped**, **values have associated types** (or classes – but types is fine)

number1 = 5 # creates the variable number1 and assigns it an integer value 5

number1 = 5.0 # reassigns the variable number1 a float value 5.0

number1 = “5” # reassigns the variable number1 a string value “5”

The values have the datatype (string, boolean, int, float, ect...) associated with them. The variable names can be assigned to anything.

Naming Variables

Variables must begin with a letter or an underscore. After this they can have any combination of letters, underscores or digits.

Variables are case sensitive.

Number1
number1
nUmber1

all three of the above would be different variables. They would be referencing different locations in memory.

Reserved words

Reserved words are words that have a special meaning to the python interpreter and cannot be used as variable names. You will get a syntax error. Examples include : if, else, for ,and

Function names are not reserved words but using them as variable names is a bad idea as it reassigns them.

Example:

```
print("hello") # prints hello
print = 5 # this is legal
print("hello again") # We have a problem now because print is a variable with 5 not a function
```

Programmer and End-user

Programmers write and test the code that makes up a computer system. They usually work as part of a team.

An end-user is the person (or organisation) for whom a software system is developed. End-users are the customers and, very often, do not know how to program.

Language Syntax

The syntax of a language is that part of the grammar which defines how sentences are constructed – syntax is mostly concerned with legitimate words, symbols and the order in which they are used.

A syntax error is when a programmer writes code that breaks the grammar rules.

Comments

Comments are used by programmers to improve the readability of their code for the benefit of other programmers. Comments in Python start with the hash character, # , and extend to the end of the

physical line. When Python comes across the hash character it ignores the rest of the text on that line.

Case Sensitivity

Python is case sensitive. This means that Python treats upper and lower case letters differently.

Strings and Numbers

In programming text is referred to as a string. A string is any text enclosed inside quotation marks. It can also be referred to as an array or list of characters.

Numbers in quotation marks are treated as strings, and outside of quotation marks they are treated as numbers.

Quotation Marks

Quotation marks can be single (’) or double (") – it does not matter as long as they match.

print()

The python print() command allows strings and numbers to be separated by commas.

Each of the following are syntactically correct, (they don't have a syntax error).

```
print("What is your age? ")
print(21)
print("My age is 21")
print("My age is", 21)
print("I am", 18, "and my friend is", 21)
print("The print command", "can handle", "more than 1 string.")
```

Revision Exercise 2

Type the above commands into Thonny what do you notice about the output formatting?

For example, the last line has no spaces in the different strings. The comma adds a space between each input.

Indentation

In python, indentation refers to the empty spaces at the beginning of a line of code.

Every line of python code must be properly indented – logical related lines of code (code blocks) must all start the same number of spaces in from the start of the line

An indentation error is the error which appears when one or more lines of code are not correctly spaced. An indentation error is a form of syntax error.

The following will give rise to an indentation error as the second line has a different indent – is a

different number of spaces from the beginning.

```
print(“Hello”)
 print(“there”)
print(“Bob”)
```

Flow of control

Flow of control refers to the order in which lines of a computer program are run by the computer. **Sequential flow** is when the lines are executed in the sequence they appear one after another.

```
print("As I was going out one day")
print("My head fell off and rolled away,")
print("But when I saw that it was gone,")
print("I picked it up and put it on.")
```

The above code will print as follows to the screen:

```
print("As I was going out one day")
```

```
print("As I was going out one day")
print("My head fell off and rolled away,")
```

```
print("As I was going out one day")
print("My head fell off and rolled away,")
print("But when I saw that it was gone,")
```

```
print("As I was going out one day")
print("My head fell off and rolled away,")
print("But when I saw that it was gone,")
print("I picked it up and put it on.")
```

When you run it all four lines appear at the same time because the computer is so fast, but in reality they appear one after another in sequence.

Escape Sequences

Strings are text enclosed in quotation marks. Sometimes we may wish to actually print quotation marks to indicate a quote from a person or some text. So how do we tell the computer that we wish to write our text in actual quotation marks?

For example, if we wished to print “Change is the law of life” - including the quotation marks

We can not do as follows:

```
print(""Change is the law of life"")
```

The second quote closes the first and the rest of the line is not understood by Python

To fix this syntax error we place a backslash before the quote we wish to print.

print("\"Change is the law of life\"")

The combination of \" tells python to include the double quotes as part of the string. The combination is called an escape sequence. It changes how Python interprets the character after the backslash.

Other common escape sequences are:

\n for a Newline
\t for a Tab space
\' for a single quote
\\ for a backslash – for example when putting in a path to a file.

The input() command

Allows a user to enter a value into a running program and have the value stored in a variable.

The input() and print() commands are examples of built-in functions. They are functions that come pre-defined with the programming language.

personName = input("Enter your name:")

Run the above line of code in Thonny. The text in brackets is a prompt for the user to tell them the information required. The input provided by the user is stored in the variable personName.

Hard coding and soft coding

When you assign a variable a value in your code it is said to be hard coded

personName = "Alex" - The variable personName is hard coded

When you allow the variable to be changed by the user or some external agent it is said to be soft coded

personName = input("Enter your name:") - The variable personName is soft coded

Datatypes

Programs use different types of data. For example, strings and numbers are two different types of data. The data a program can use is called a datatype.

Python has many different datatypes.

The datatypes we care about at the minute are:

The number datatypes of integer and float. There is also complex for complex numbers but we

won't use it.

The boolean datatype – is something True or False

The sequence datatype string – a combination of characters enclosed in quotes.

type() command

If you wish to see what datatype a variable is, you can use the type()