Application of GIS with Python

Chapter 1:Introduction to python programming

https://www.python.org/
http://www.tutorialspoint.com/python/

Table of Content

- Variables
- Literals
- keywords
- Expression
- Statements
- Operators and operands
- Types and precedence

Python Programming

Program:

- Set of instruction tell computer how to perform a particular task.
- Sequence of instructions that specifies how to perform a computation
- A program may contain multiple algorithms as the overall problem may need solutions to various sub-problems

Features:

➢Ability to formulate the problem

➢Ability to think about problem solutions

➢Ability to express a solution clearly and correctly

➢Ability to compare alternative solutions

Example of computational problems

- Given a list of cities C, what is the shortest trip that visits all cities and returns to where it started?
- Given a polygon P, and the coordinates of a point p, does point p fall inside polygon P?

Algorithm

- A program may contain multiple algorithms as the overall problem may need solutions to various sub-problems.
- A language with which the steps of an algorithm can be clearly defined(by a human being), so that they can be carried out (by a computer)

Languages:

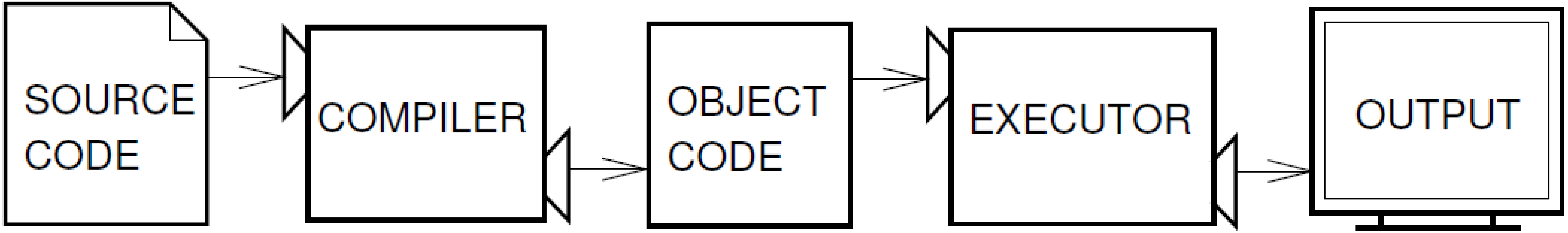
- ✓High level languages
- ✓Low level/Machine level/Assembly languages

- ➢An algorithm can be defined as “a instructions of computational steps should be follow to obtained desired solution”
- ➢An algorithm proposes to solve one isolated problem.

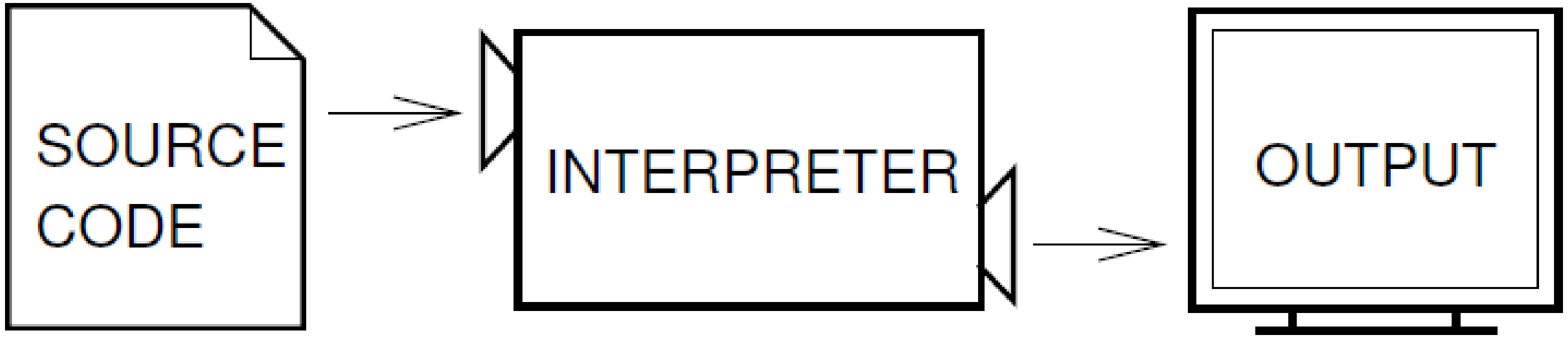
Python programming language

- High-level language take less time to write, they are shorter and easier to read, and they are more likely to be correct
- High-level languages are **portable**, meaning that they can run on different kinds of computers with few or no modifications. So most of the programs are written in high level language
- Two kinds of programs process high-level languages into low-level languages: **compilers** and **interpreters**

- A compiler reads the high-level program (**source code)** and translates it completely before the program starts running. The translated program is called the **object code** or the **executable**. Once a program is compiled, can execute it repeatedly without further translation.



- An interpreter reads a high-level program and executes it.
- It processes the program a little at a time, meaning that it does reading lines and performing computations.
- Python (**interpreted language**) programs are executed by an interpreter.



- Two ways to use the interpreter: **interactive mode** and **script mode**
  - **interactive mode:** type Python programs and the interpreter prints the result

    ```
    >>> 1 + 1
        2
    ```

  - **script mode:** can store code in a file and use the interpreter to execute the contents of the file. Python script file has '.py' extension

Python

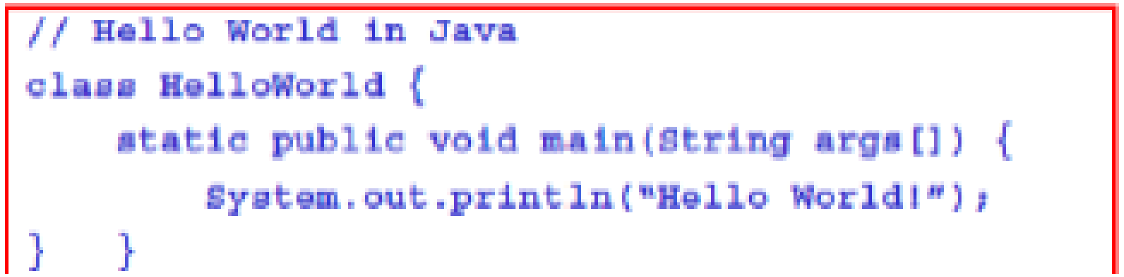
- Python is an open source, cross-platform, objected oriented scripting language (High Level Language).
- Developed by Guido van Rossum in the early 1990s
- Named after “Monty Python’s Flying Circus”
- Intended to be a scripting language on Amoeba OS
- First public release was in 1991

Why Python?

- Concise
  - Python allows writing very compact and readable programs.
  - Programs written in Python are typically much shorter than equivalent C or C++ programs, for several reasons.
- Dynamically typed: variables/objects do not have a predefined type
- Rich, built-in collection types:
  - Strings, Lists, Tuples, Dictionaries

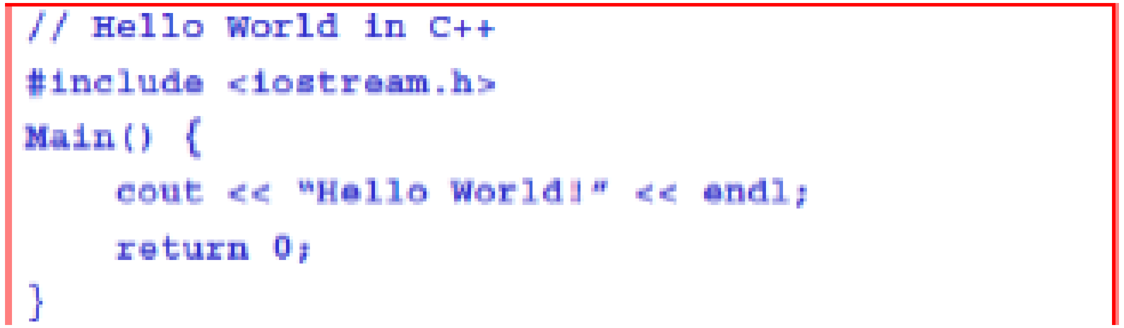
Hello World

Java



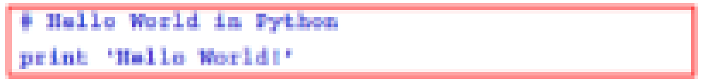
```java
// Hello World in Java
class HelloWorld {
    static public void main(String args[]) {
        System.out.println("Hello World!");
}   }
```

C



```cpp
// Hello World in C++
#include <iostream.h>
Main() {
    cout << “Hello World!” << endl;
    return 0;
}
```

Python



```python
# Hello World in Python
print 'Hello World!'
```

Language features

- ➢Newline separates statements
- ➢Indentation instead of braces
- ➢Several sequence types
  - ➢Strings '…': made of characters, immutable
  - ➢Lists […]: made of anything, mutable
  - ➢Tuples (…) : made of anything, immutable
- ➢Powerful subscripting (slicing)
- ➢Functions are independent entities (not all functions are methods)

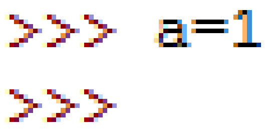
- Case sensitivity
- File paths (/, \\ )
- Quotation marks (", ')
- Commenting (#)

Variables

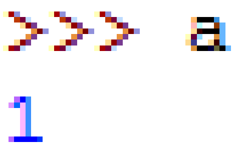
- A variable is a name that refers to a value. (store a value)
- An algorithm may use variables during: (box as variable name)
  - **Assignment** - give the variable a value: box = 1.60159
  - **Reading** - use the stored value for computations: 2 * box + 15
  - **Testing** - see whether the stored value meets a condition: box >= 3.15
  - **Output** - look up the value and show it to the user: print box

Variables and types

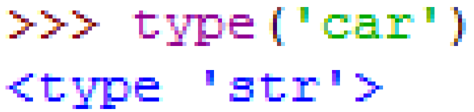
- Variables need no declaration
- As a variable assignment is a statement, there is no printed result
- Variable name alone is an expression, so the result is printed
- Variables must be created before they can be used
- The type of a variable is the type of the value it refers



```
>>> a=1
>>>
```



```
>>> a
1
```



```
>>> type('car')
<type 'str'>
```

Variable names

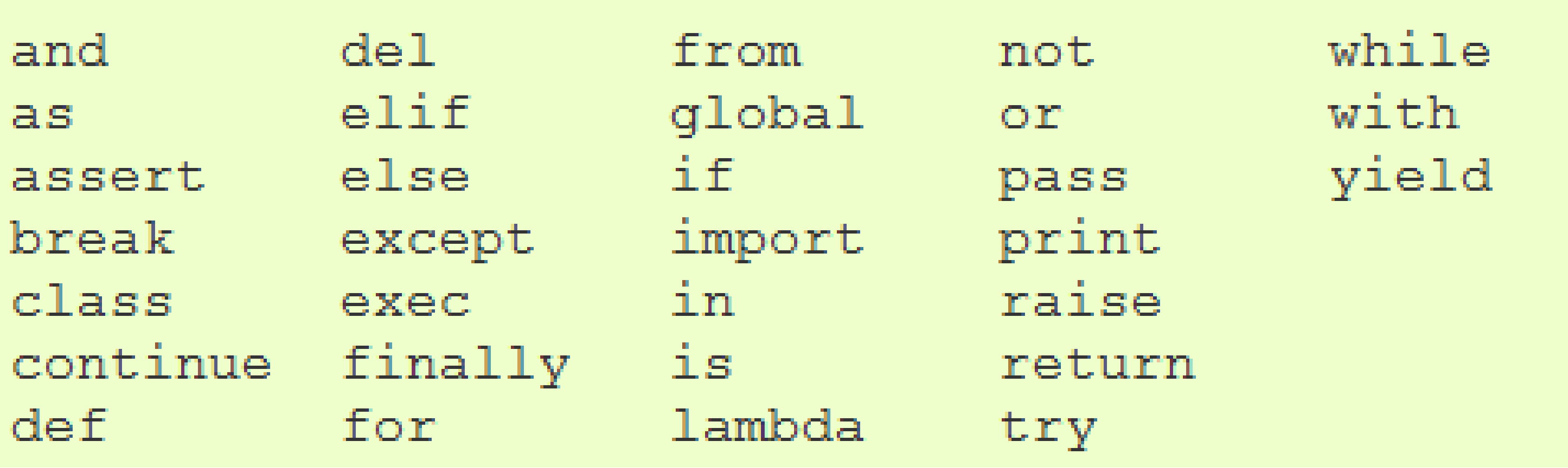
- Can contain both letters and numbers.
- But they have to begin with letter.or with underscore like _a = 3 # Valid
- Legal to use uppercase, but by convention not used.
- Case matters: Bruce and bruce are different variables.
- Underscore character(_) can appear in a name.

Variable names

- >>>1car=‘big parade’
- >>>more$=100000
- >>>Class=‘geomatics’
- Get Syntax errors?

Keywords

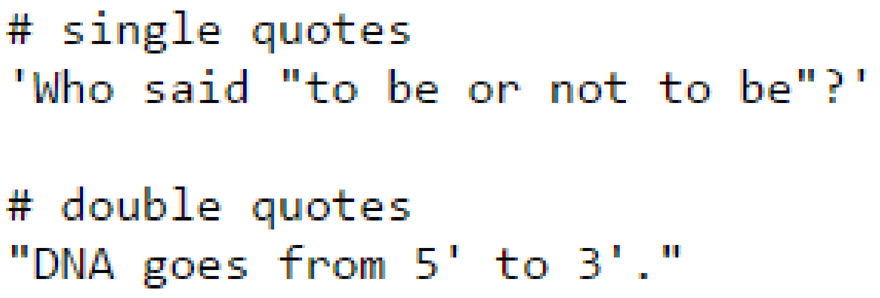
- Predefined /used/reserved words or names by python to recognize the structure of the program
- They must be spelled exactly as written
- Cannot be used as variable name



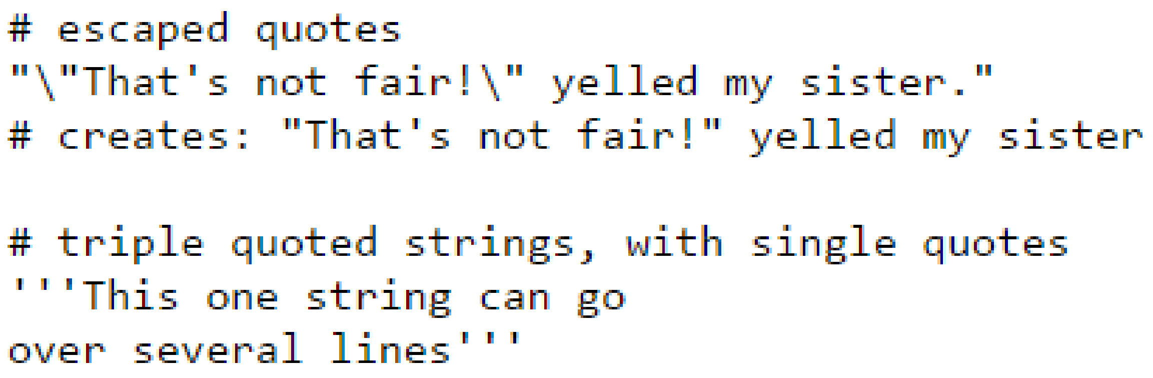
and
as
assert
break
class
continue
def
del
elif
else
except
exec
finally
for
from
global
if
import
in
is
lambda
not
or
pass
print
raise
return
try
while
with
yield

Literals

- Python comes with some built-in data structures or objects (string, unicode string, integer, float, long, list, tuple and dictionary), the way of representing values to such objects (creating objects), called **literals.**
- **String Literals**



```
# single quotes
'Who said "to be or not to be"?'

# double quotes
"DNA goes from 5' to 3'."
```

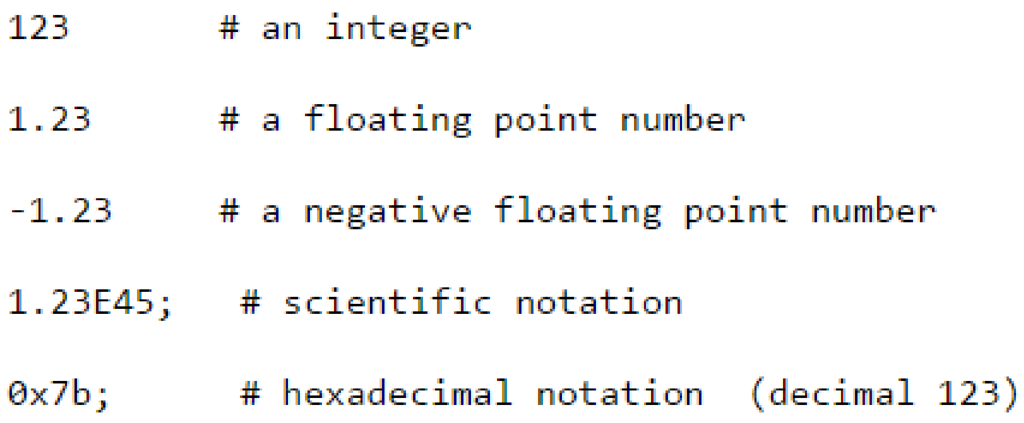


```
# escaped quotes
"\"That's not fair!\" yelled my sister."
# creates: "That's not fair!" yelled my sister

# triple quoted strings, with single quotes
'''This one string can go
over several lines'''
```

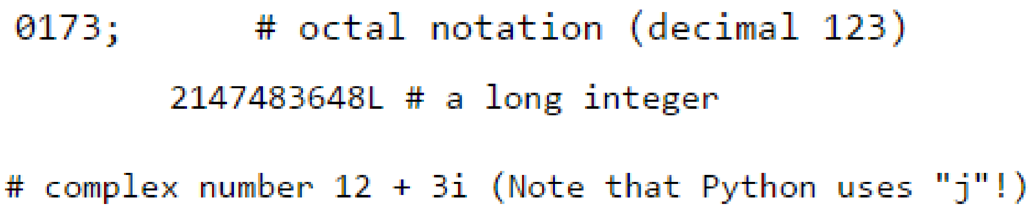
Literals

➢Numeric literals

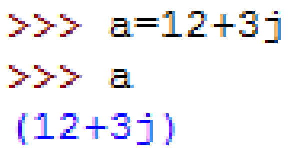
- Can refer to numeric values using integers, floating point numbers, scientific notation, hexadecimal notation, octal, and complex



```
123        # an integer

1.23       # a floating point number

-1.23      # a negative floating point number

1.23E45;    # scientific notation

0x7b;       # hexadecimal notation  (decimal 123)
```



```
0173;      # octal notation (decimal 123)

       2147483648L # a long integer

# complex number 12 + 3i (Note that Python uses "j"!)
```



```
>>> a=12+3j
>>> a
(12+3j)
```

Statement

- A statement is a unit of code that the Python interpreter can execute.
- A script usually contains a sequence of statements; the results appear one at a time as the statements execute.

For example, the script

```
print 1
x = 2
print x
```

produces the output

```
1
2
```

The assignment statement produces no output

Operators and Operands

- Operators are special symbols that represent computations like addition and multiplication.
- The values which the operator is applied to are called operands.
- Python's has many operators, like ( ),+, -, /, *, ** and %(arithmetic); and, or, not (logical); <, >, <<, >>, ==, !=(comparison)
- They are polymorphic and work differently for the different data types.

Operators (5+9)*(15-7) operands

Expression

- An expression is a combination of values, variables and operators, which when evaluated leads to a value (or possibly an error).
- Not every expression always contains all three; valid expressions are:

  17 "Hello" "Hello" + " mum"

  ‘17’ 3.2 1 + 17 pi ** 2 + n / 4
- The expression pi ** 2 + n / 4 has three operators: **, + and /.
- How the expression is evaluated depends on the rules of operator precedence.

Rules of arithmetic precedence

PEDMAS

- Parentheses: these are used to overwrite an unwanted precedence, or just for sake of clarity
- Exponentiation: your highest priority after ( )
- Multiplication and Division: they are at the same level, because (3 * 4) / 2 equals 3 * (4 / 2).
- Addition and Subtraction: same thing
- Order: left before right of two equal precedence
- Operators (the left one goes first).

Assignment

- Explain three types of errors in program.
- Python is an open source, cross-platform, objected oriented scripting language (High Level Language), Justify the statement with suitable example.
- Write Short notes on following:
  - Object oriented programming language
  - Compiler
  - Interpreter
  - Literals

Evaluation

Internal			Final	Lab/Practical
Attendance	Assignment	Assessment		
5	5	10	80	25

References:

- Books available in the internet
- Materials provided
- YouTube Tutorials