DistAlgo is a high-level language for writing distributed algorithms, such that they serve as both clear specifications and runnable implementations of those algorithms. This document provides a brief, practical introduction to the major features of the language using several iterations of a familiar example.

DistAlgo is currently implemented as an extension of the Python language, and requires Python 3.4 or higher. This tutorial assumes some familiarity with Object-Oriented programming in general, and with Python, in particular.

Let's begin with the simplest possible version of "Hello World" written in DistAlgo.

```
1     def main():
2         print('Hello World.')
```

Listing 1: Hello World 01 - Main Function Definition

Every DistAlgo program must have a main function. In this case, the body of `main` merely contains a call to Python's print function, to output "Hello World" to standard output. As we will see in the next example, the inteded use of the main function in a DistAlgo program is to create, prepare, and then begin the execution of the distinct processes that participate in the distributed algorithm.

In the second version of "Hello World" we can see how process definition and creation works in a DistAlgo program.

```
1   class P (process):
2       def setup():
3           pass
4
5       def run():
6           output('Hello World from: ', self.id)
7
8   def main():
9       p = new(P)
10      setup(p, ())
11      start(p)
```

Listing 2: Hello World 02 - Process Definition