Args kata

KATA Most of us have had to parse command-line arguments from time to time. If we don't have a convenient utility, then we simply walk the array of strings that is passed into the main function. There are several good utilities available from various sources, but they probably don't do exactly what we want. So let's write another one! The arguments passed to the program consist of flags and values. Flags should be one character, preceded by a minus sign. Each flag should have zero, or one value associated with it.

You should write a parser for this kind of arguments. This parser takes **a schema** detailing what arguments the program expects. The schema specifies the number and types of flags and values the program expects.

Once the schema has been specified, the program should pass the actual argument list to the args parser. It will verify that the arguments match the schema.

The program can then **ask the args parser for each of the values**, using the names of the flags. The values are returned with the correct types, as specified in the schema.

For example if the program is to be called with these arguments:

```
-l -p 8080 -d /usr/logs
```

this indicates a schema with 3 flags: l, p, d.

The "l" (logging) flag has no values associated with it, it is a boolean flag, True if present, False if not. The "p" (port) flag has an integer value, and the "d" (directory) flag has a string value.

If a flag mentioned in the schema is missing in the arguments, a suitable default value should be returned. For example "False" for a boolean, 0 for a number, and "" for a string.

If the arguments given do not match the schema, it is important that a good error message is given, explaining exactly what is wrong. If you are feeling ambitious, extend your code to support lists eg

```
-g this,is,a,list -d 1,2,-3,5
```

So the "g" flag indicates a list of strings, ["this", "is", "a", "list"] and the "d" flag indicates a list of integers, [1, 2, -3, 5].

Make sure **your code is extensible**, in that it is **straightforward and obvious** how to add new types of values.