 Due by midnight today

CMPSC 102
Discrete Structures
Fall 2018

Practical 7: Using Generator Functions to Output in Morse Code from Strings

Refer to your notes, slides and sample Python code from this week and other weeks. In particular, follow the python code that we created in class this week: `yay.py` *to help you see how strings are implemented in generators.*

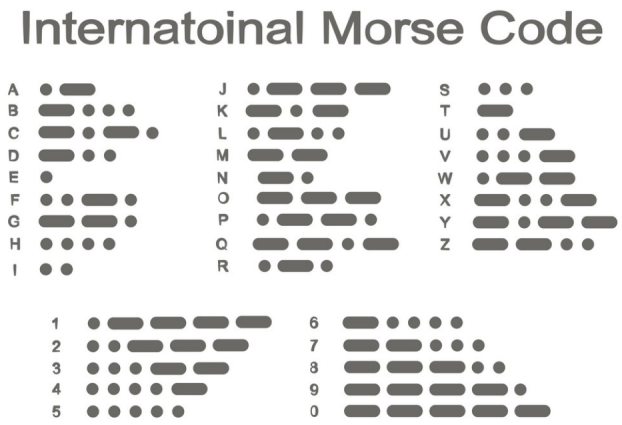

Figure 1: Morse code (written in *dots* and *dashes*) was used to send messages over telegraphs and early radio communication (before the voice could be transmitted). In the latter part of the 19th and early 20th centuries, seemingly all high-speed international communication was encoded in Morse code to be sent through telegraph lines, undersea cables and radio circuits. Take that, Internet!

Summary

A class of generator functions in Python rely on iterators to drive their calculation operations. Iterators do not compute the value of each item when instantiated, instead, they only compute values when they are requested.

Another important part of this practical is to work with Morse code which was developed to encode messages before the voice could be transmitted. Ships used to use Morse code to send and receive all their communications to each other and to the shore.

In this practical, we will be completing Python code to incorporate generator functions to work with strings to translate them into Morse code outputs, shown in Figure 1. In reality, the generator function only translates the string into Morse code when it is requested by other code. Although, translating strings into Morse code could be done in absence of using generator functions, there is no amusement in such a silly task.

Handed out: 12^{th} Oct 2018

GitHub Starter Link

https://classroom.github.com/a/eOvdUgIP

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):
  - `git commit <nameOfFile> -m ‘‘Your notes about commit here’’`
  - `git push`

  Alternatively, you can use the following commands to add multiple files from your repository:
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

What to do for this practical

Locate the incomplete Python3 code from the file `src/myMorseGenerator_starter.py`. While this code will run and produce Morse code outputs from user-entered inputs, there needs to be a generator function to output Morse code, as well. Here, you will be completing the function, `myMorseGen()` in the source code to implement a generator that will translate the user-entered strings into Morse code.

Output

You might want to consider adding a print statement to prove that your `type` is actually a generator. Your output will look like the following.

```
 Input your message : my Output
   M  :  --
   y  :  -.--
      :  /
   O  :  ---
   u  :  ..-
   t  :  -
   p  :  .--.
   u  :  ..-
   t  :  -
  Generator component
type : <class 'generator'>
-- -.-- / --- ..- - .--. ..- -
```

Deliverables

1. Your completed (and working) Python code (`src/myMorseGenerator_starter.py`)

```
--. --- / --. . - / .----. . --
```