Lab-6 (part1) - Hongyu (Ray)

July 30, 2019

0.1 Generator

Create fibonacci numbers that are less than 100 using Generator

0.2 Iterable Class

Create an iterable class Range that has the exact same functionality as the built-in range() function, which takes three arguments: start, stop, and step. This class Range can be iterated by using the next() method and will raises an error when out of range occurs. You should also prevent possible infinite loop by raising RuntimeError when necessary.

0.3 Decorator (first class function && closure)

1. Create a decorator function that can check zero division error.
2. Create a logging function that logs every decorated function into a file named by this function with the following format:

```
function_name.log:
Ran with arguments: arg1, arg2, arg3... and keyworded arguments: kwarg1, kwarg2, kwarg3...
```

In this assignment, you could use the `logging` model (`import logging`)

3. Create a timing decorator that prints out the running time of decorated functions.
4. Slightly modify the logging and timing decorators created in question 2 and 3 so that they can be chained together properly without losing information. (You may need to use the wraps decorator from the functools model.)
5. Assume you want to create a simple HTML web page. However, you only want to write the content do not want to repeatedly type all the tags yourself. Greate a reusable decorator to generate HTML tags automatically for you and using it write the following html. (Ignore the indentation)

```
<h1>Learning Python</h1>
    <h2>Chapter 1</h2>
    <h2>Chapter 2</h2>
        <p>
          <h3>Section2.1</h3>
        </p>
    <h2>Chapter 3</h2>
```