2nd and 4th Order Runge-Kutta Methods

Authors: Howard Lin, Danny Lu

For our Math 352 Project, we have decided to implement Runge-Kutta in Python.
Our module/file is rk.py, and there are 4 notable functions:

- Runge-Kutta of Order 2, single iteration
- Runge-Kutta of Order 2, n iterations
- Runge-Kutta of Order 4, single iteration
- Runge-Kutta of Order 4, n iterations

Running the program

We have set it up so that you can run our program through two ways:

- Through the Python3 shell
- Editing the main function in rk.py

Depending on how you want to run or test the program, follow ONE of the instructions below.

Using the Python3 shell

1. Enter the python shell by entering the terminal and type in

`python3` in the directory that rk.py is in.

2. Type in `>>> import rk`
3. Define your math function that only takes in the parameter `t` and `x` in that order (The order important, if your function takes in `x` and then `t` the Runge-Kutta module will fail)
4. Call a function from the library and follow the parameters in the comments of the file. See example below:

```
$ python3
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import rk
>>> def f(t,x):
...     return 2 + (x - t - 1)**2
...
>>> result = rk.fourth_order_rk(f,1,2,0.5625)
>>> # we are gonna approximate x(1.5624) with initial value x(1) = 2
...
>>> print(result)
3.192481613086853
>>>
```

Using the main function

If for whatever reason you can’t run the program in the shell, just edit rk.py

1. First change `def f(t,x)` to the function you would like, using any python3 math functions
2. In the `if __name__ == "__main__":` function, call the desired Runga-Kutta Method and the `f` function.
3. In the terminal or Python3 interpreter (e.g. IDLE), run the program using `$ python3 rk.py`

Example:

In the end of `rk.py`:

```
'''
This is the f(t,x) function
'''
def f(t,x):
    return 2 + (x - t - 1)*(x - t -1)

'''
    This is the main function
'''
if __name__ == "__main__":
    print("Hello there! This is the main file")
    result = fourth_order_rk(f,1,2,0.5625)
    # we are gonna approximate x(1.5624) with initial value x(1) = 2
    print(result)
```

And then in the terminal:

```
 ✘ $ python3 rk.py
Hello there! This is the main file
3.192481613086853
```