

More functions etc. examples

Ben Bolker

18:44 29 January 2017

More examples

conversion to/from Roman numerals

- Based on CSE 231 project “Shannon’s Throbac”
- The rules:
 - (I,V,X,L,C,D,M) \leftrightarrow (1,5,10,50,100,500,1000)
 - *subtraction rule*: I, X, or C preceding the next two elements correspond to subtraction (e.g. IV=4, XC=90)
- string methods are useful, especially `in`, `s.count()`, `s.replace()` are useful
- for the reverse direction we need *integer division* and *modulo* operations, `//` and `%`
- we should write tests first, e.g.

```
rom = ["I","II","III","IV","V","VI","VII","VIII","IX",
      "X","XI","XIV","XV","LX","XC","XCXLI","CCC"]
int = [1,2,3,4,5,6,7,8,9,10,11,14,15,60,90,131,300]
for i in range(len(rom)):
    print(rom,int,rom_to_int(rom[i])==int[i])
```

- we can try to write this as a gigantic `if` statement, or we can try to make the rules more compact
- which direction is easier?

bisection search for a root

- suppose we have a continuous function $f(x)$ which we can evaluate at any point x and we know that the function has opposite signs at points x_L and x_H (i.e., $f(x_L)f(x_H) > 0$). We don’t know anything else about $f(x)$, but we have been given a Python function `f` to compute it. How do we find (approximately) the *root*, i.e. the value where $f(x) = 0$? (Assume f has only one root between x_L and x_H .)

compute sine/cosine

- CSE project
- power series approximation:

$$\begin{aligned}\sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \\ &= \sum_{k=0} (-1)^k \frac{x^{2k+1}}{(2k+1)!}\end{aligned}$$

$$\begin{aligned}\cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \\ &= \sum_{k=0} (-1)^k \frac{x^{2k}}{(2k)!}\end{aligned}$$

- stop when the absolute value of the next term is $< 10^{-8}$ (`1e-8` in computer notation)
- divide x by π , take the remainder (modulo: `%`)

- write tests first (what are some easy cases, using `math.pi`?)
- use `math.factorial`? use the factorial function we already wrote? augment terms as we go along?
- after we write the two functions, can we combine them?
- extra credit/puzzle: what do we get if we interpolate between these two functions?

letter-scrambling

- CSE project
- scramble words within a text
- start with some simpler cases
 - reverse every word in a line
 - again, string methods – especially `str.split()`
 - now read from a file
- test cases? easier with just a few lines
- harder: preserve punctuation at the end of the word
- harder: shuffle
- `from numpy.random import shuffle`
- split words and join them: `list()`, `''.join()`
- test case

```
f = open("../misc/darwin.txt")
for line in f:
    print(line)
```

It is interesting to contemplate a tangled bank, clothed with many plants of many kinds, with birds s