

Recursion and Comprehensions

Info 206

Niall Keleher

19 September 2017

Today's Outline

1. Feedback
2. Revisiting Newton-Raphson
3. Recursion
4. Comprehensions
5. Exercises
 - Recursion
 - Comprehensions

Additional Python Exercises

- <https://www.w3resource.com/python-exercises/>
- <https://snakify.org>
- <https://developers.google.com/edu/python/exercises/basic>
- <https://learnpythonthehardway.org>

Newton-Raphson

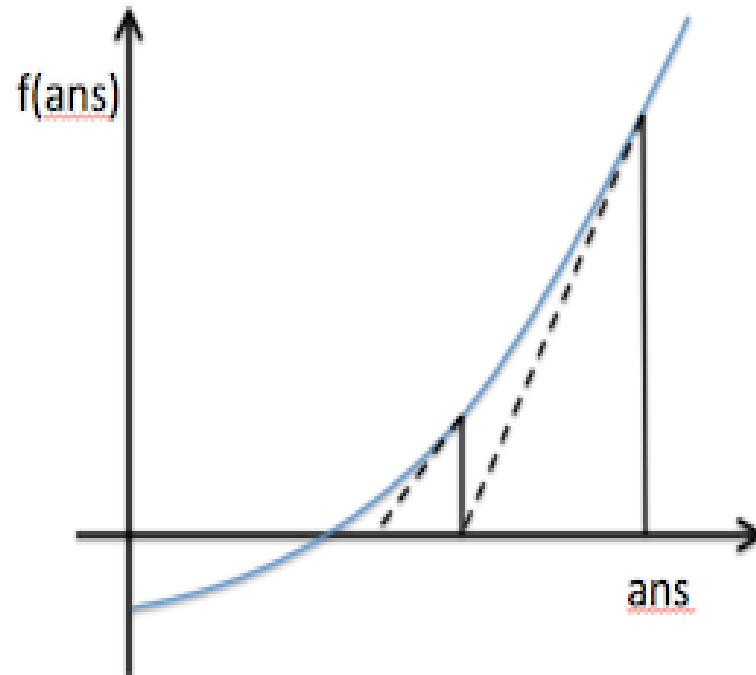
- Common method for approximating a solution
- Uses successive approximation
- Used to find the root of a wide range of functions (e.g. polynomials)

Newton-Raphson

- Supply ans (some guess)
- If ans is an approximation of the root of a polynomial, $p()$, then

is a better solution.

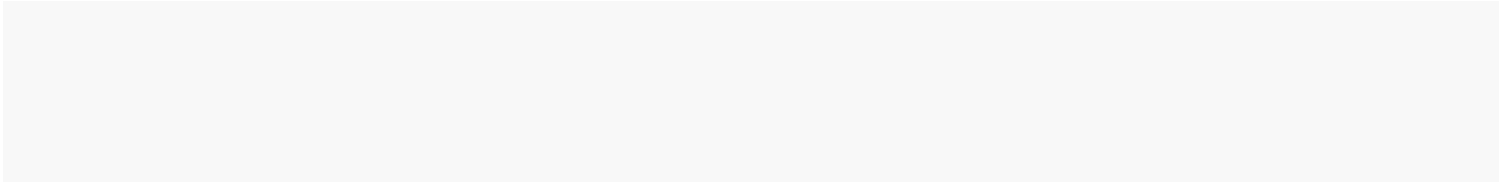
- Repeat until sufficiently close to the solution.



Newton-Raphson

Recursion

Recursion



Recursion

A recursive algorithm needs two elements.

1. A base case - a simple version of the problem that we can solve immediately
2. A recursive rule - a mathematical way to break the problem down into easier problems

Comprehensions

Comprehensions

Create a list with the square of the numbers in a range

for Loop:

List Comprehension:

Dictionary Comprehension



Team Meetings

Teams E, F, G, H, I

Exercises

Meeting 8: Recursion and Comprehensions Exercises

Exercises

- Instructions in the Github course-exercise repository
- Meeting 8 - Due at the end of the day on Friday (Sept 22)

End of Meeting #8

For next meeting

- Videos:

1. Modules and Packages (4 mins)
2. Modules and the Import Statement (10 mins)
3. Packages (13 mins)
4. [optional] The Python Standard Library (25 mins)
5. Arrays (14 mins)

- Readings:

- Lutz Chapter 22: Modules: The Big Picture
- Lutz Chapter 23: Module Coding Basics