

# Mid-course feedback, Course Projects, Functions

Info 206

Niall Keleher

14 September 2017

# Today's Outline

1. Mid-course feedback

2. Group Projects

- Status and Expectations
- Team meetings - today

3. Functions

4. Exercises

- Continue with meeting 6 exercises
- Functions

# Taking Stock

- Using git

# Taking Stock

- Using git
- Python object types

# Taking Stock

- Using git
- Python object types
- Python Statements & Syntax

# Taking Stock

- Using git
- Python object types
- Python Statements & Syntax
- Iteration

# Taking Stock

- Using git
- Python object types
- Python Statements & Syntax
- Iteration
- Functions

# The road ahead

- Recursion
- Modules
- Classes & OOP
- Complexity
- Exceptions & Tests



# Group Projects

# Course Objectives

- work collaboratively using source control
- understand and operationalize core Python objects
- create user-written functions in Python
- work with classes in Python
- clearly explain and execute good practices in software prototyping

# Project Objectives

Teams are responsible for:

- identifying the problem that they aim to address
- working collaboratively and sharing code through Github
- building and testing software
- display final project results on the last day of the course (17 October)

# MVP & Design Document

# Team meetings

- Today and Tuesday
- Review the problem statement
- Discuss ways of decomposing and modularizing the proposed solution
- How is the team dividing up work?
- Achievable and stretch goals
- What is your MVP?

# Functions

# Functions

## Decomposition

## Modularity

## Abstraction

# Scope & Namespace

- What is scope?



# Scope & Namespace

- What is scope?
- LEGB

# Scope & Namespace

- What is scope?
- LEGB
  - Local - Names assigned within a function
  - Enclosing - Names in the local scope of a function
  - Global - Names assigned at the top-level of a module or declared global in a function
  - Built-in - Preassigned names

# Newton-Raphson (Heron's Method)

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

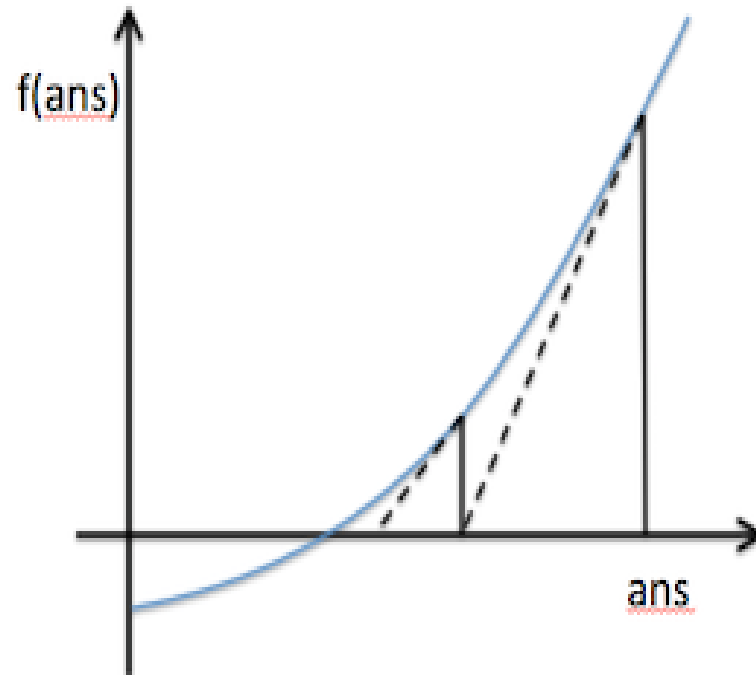
# Newton-Raphson

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

$$ans - p(ans)/p'(ans)$$

is a better solution.

- Repeat until sufficiently close to the solution.



Newton-Raphson

# Exercises

# Questions about Meeting 6 Exercises?

# Meeting 7: Function Exercises



# Exercises

- Instructions in the Github course-exercise repository
- Meeting 6 - Due at the end of the day on Friday (Sept 15)
- Meeting 7 - Due at the end of the day on Tuesday (Sept 19)

# Exercises

- Instructions in the Github course-exercise repository
- Meeting 6 - Due at the end of the day on Friday (Sept 15)
- Meeting 7 - Due at the end of the day on Tuesday (Sept 19)

**Get ready for more exercises between now and early October!**

End of Meeting #7

# For next meeting

- Videos:
  1. Recursions Basics (10 mins)
  2. Traversing Nested Dictionaries with Recursion (13 mins)
  3. Comprehensions (10 mins)
- Readings:
  - Lutz Chapter 20: Comprehensions and Generations