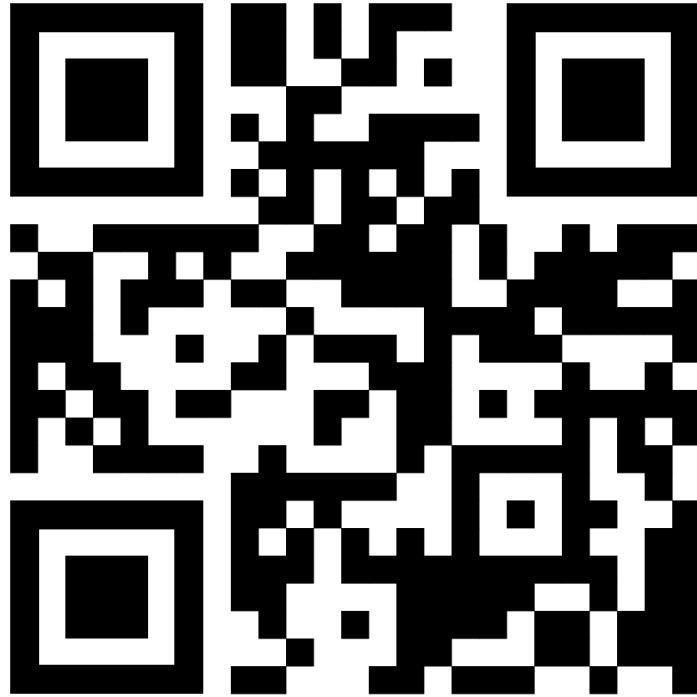


Statements and Syntax, Part 2

Info 206

Niall Keleher

12 September 2017



Today's Quiz: <http://bit.ly/zwTDOjF>

Today's Outline

1. Iteration

2. Comprehensions

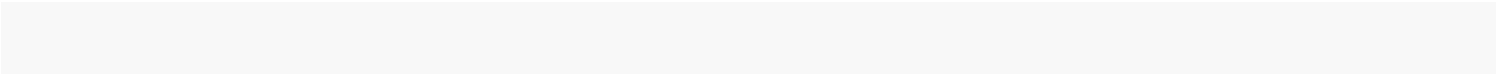
3. Exercises

- File Input & Output
- Exhaustive Search
- Bisection Search

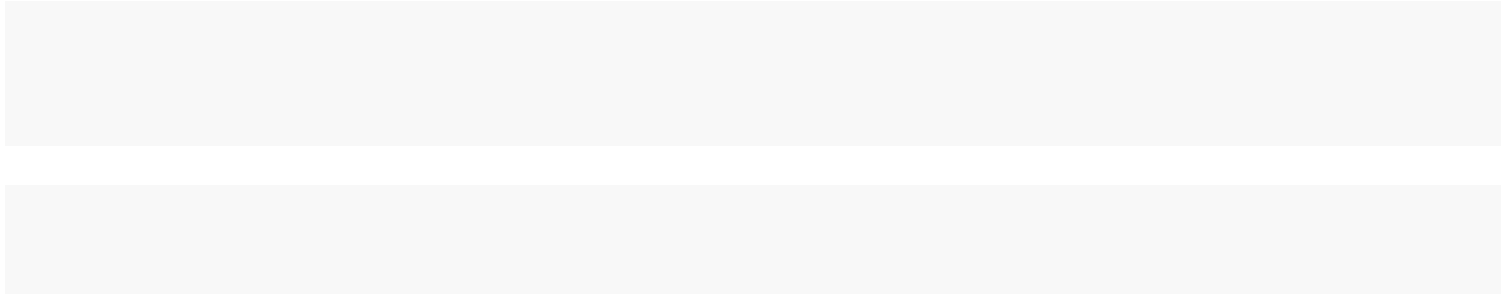
Iteration

Comprehensions

Comprehensions

- A concise way to apply an operation to the values of a sequence.
 - Requires less coding relative to for Loops
 - Tends to be faster than for Loops - iterations are performed in C
 - List comprehensions:
- 

List comprehensions



Exercises

File Input and Output

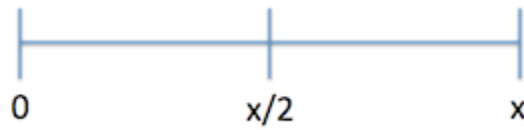
- using `open()`, `write()`, `read()`, `close()`
- printing lines through iteration
 - for Loops
 - list comprehension

Exhaustive Search

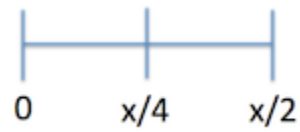
- an example of a
- Searches through all possible solutions until the "right" one is found.
- Can take a rather long time.

Bisection Search

- Reduce the search space for each iteration of the algorithm
- Improves the run time of the search algorithm
- Requires the decision of how close you want to get to the correct solution



Bisection Search - Step 1



Bisection Search - Step 2

Exercise - Meeting 6

- Instructions in the Github course-exercise repository
- Due at the end of the day on Friday

End of Meeting #6

For next meeting

- Videos:
 1. Functions (10 mins)
 2. Calling Functions (3 mins)
 3. Nesting Functions (6 mins)
 4. Functions and the Call Stack (2 mins)
 5. The Stack Trace (3 mins)
 6. Advantages of Functions (1 min)
 7. Namespaces (6 mins)
 8. Accessing Global Variables (12 mins)
 9. Using Parameters (13 mins)
 10. Functions are Objects (6 mins)
- Readings:
 - Lutz Chapter 16: Function Basics
 - Lutz Chapter 17: Scope
 - Lutz Chapter 18: Arguments