EdX 6.00x Notes

Lecture 3:

- Iteration:
  - A method repeated multiple times in order to reuse the computation to execute something an arbitrary number of times.
  - How it works:
    - Starts with a test.
    - If evaluates to True, then execute loop body once, and go back to reevaluate the test.
    - Repeat until test evaluates to False, after which code following iteration statement is executed.
  - Some properties of iteration loops:
    - Need to set an iteration variable outside of the loop.
    - Need to test that variable to determine when done.
    - Need to change that variable within the loop, in addition to other work.
      - *Not changing this can lead to an infinite loop!*
- while() a.k.a. While Loops
  - A keyword in python that has a Boolean test which allows for iteration.
  - Note: A colon is required after the last parenthesis.
  - Generally iterates over a sequence of choices.
- Branching structures (conditionals):
  - Let us jump to different pieces of code based on a test.
  - Programs are constant time.
- Looping structures (e.g., while):
  - Let us repeat pieces of code until a condition is satisfied.
  - Programs now take time that depends on values of variables and length of the program.
- break:
  - If break is executed within a loop, it halts evaluation of the loop at that point and passes control to the next expression
- Guess and Check method:
  - Iterate through guessing different answers to a problem one is trying to solve, and then checking to see if it is the right one.
- abs()
  - Built-in function that returns the absolute value of input.
- Loop characteristics:
  - Need a loop variable
    - Initialized outside of loop
    - Changes within loop

        - Test for termination depends on variable
- Decrementing Function:
    - Maps set of program variables into an integer
    - When loop is entered, value is non-negative
    - When value is <= 0, loop terminates, and value is decreased every time through loop
- Traceback:
    - Found in error messages, shows us where an error is at.
- Exhaustive Enumeration:
    - When you start at one end of the possible range of values and try each one in turn until a solution is found.
    - It is a good way to generate guesses in an organized manner.
- for() a.k.a. For loops:
    - Note: A colon is required after the last parenthesis.
    - Iterates over a sequence of choices.
    - How it works:
        - Identifier bound to first value in sequence.
        - Code block executed.
        - Identifier bound to next value.
        - Code block executed.
        - Continues until sequence exhausted or a break statement is executed.
- range():
    - In Python version 2.7 Built-in function that generates a sequence of integers
    - range(n) = [0,1,2,3,…,n-1]
    - range(m,n) = [m, m+1,…,n-1]
    - Note: In Python version 3.3 range(4) will return range(0, 4) because it is treated as a class of immutable iterable object
        - Further details: http://stackoverflow.com/questions/13092267/if-range-is-a-generator-in-python-3-3-why-can-i-not-call-next-on-a-range
- Float:
    - A python type that approximates real numbers. (i.e., 3.14)
    - The equality of floats is not always exact due to limitations within Python.
        - Always use abs(x-y) < 0.00 rather than x==y because the approximations may be different.
- Approximate Solutions:
    - Can't guarantee exact answer, but just look for something close enough.
    - Start with exhaustive enumeration:
        - Take small steps to generate guesses in order.
    - In general it will take x/step_size time through code to find a solution.
    - Step Size Concerns:
        - If the step size is too big it may miss the solution to the problem. However if the step size is too small it make take an unnecessarily large number of guesses.

- Bisection Search:
    - Start with a guess at the midpoint of a range and cycles through the problem cutting the size of the problem in half each round of iteration.
    - It radically reduces computation time.
    - It works well on problems where there's a sort of ordering property. Meaning, that the value of the function being solved varies monotonically with the input value.
- Newtown-Raphson Algorithm:
    - General approximation algorithm to find roots of a polynomial in one variable.
    - $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$
- Iterative algorithms (Review)
    - Guess and check methods build on reusing same code.
        - Use a looping construct to generate guesses, then check and continue.
    - Generating guesses:
        - Exhaustive enumeration
        - Bisection search
        - Newton-Raphson (for square roots)