Introduction to Python

# Knowledge

* What does a computer do?
  + Perform **calculations**
  + **Remembers** results
* Kinds of calculations
  + Built in language
  + Ones defined by programmer
* Simple calculations aren’t enough to solve problems
* Good algorithm design is needed to accomplish some tasks
* Computers have limits
  + Problems too complex
    - Accurate weather prediction at a local scale
    - Some encryption schemes
  + Fundamentally impossible to compute
    - Predicting whether a piece of code will always halt with an answer for any input (check on this [here](https://en.wikipedia.org/wiki/Halting_problem))
* Types of knowledge
  + Computers know what you tell them
  + Declarative knowledge([wiki](https://en.wikipedia.org/wiki/Descriptive_knowledge))
    - Statement of fact
    - Example: There is candy taped to the underside of one chair
    - It declares or states something
  + Imperative knowledge
    - Is a recipe or “how-to”
      * Sequence of simple steps
      * Flow of control that specifies when each step is executed
      * A mean of determining when to stop
    - Example: A cooking recipe, instructions manual, etc.
    - AKA: An algorithm

# Machines

* Fixed Program

Machine does only one thing

* Stored Program

Machine stores and executes instructions

* Basic Machine Architecture
  + Memory
    - Where data is stored
  + Control Unit([wiki](https://en.wikipedia.org/wiki/Control_unit))
    - Has a program counter (controls what operation comes next)
  + Arithmetic Logic Unit
    - Does primitive operations

# Languages

# Types

# Variables

# Operators and Branching