Introduction to: Computers & Programming: Exercises Using:

(the print function and Mathematical Operators and Data Types)

Adam Meyers
New York University
Exercise 1 to 3

1. What are some simple ways to print the same thing 3 times, with a space in between each instance? Let's assume one variable as input to the function `print_3_times`.

2. What is a simple way to print the same thing 3 times, each on a newline?

3. How would you convert an arbitrary temperature from Fahrenheit to Celsius?
How to Solve a Complex Problem

• Break down big problem into small problems
  – For each small problem either:
    • Solve the problem if you know how
    • Break it down further if you don't know how to solve it and:
      – Try to solve each of the smaller problems
      – Etc.
Until we find problems easy enough to solve.
Exercise 4

• Given
  – Today's date:
    • Month, day, year, and day of the week
  – An arbitrary number of days in the future (e.g., 1500)

• How should we go about identifying that day in the future?
  – Month, day, year and day of the week
How Can We Solve This Problem?

1. Some Pieces of the Problem
   a) Find the year
   b) Find the month
   c) Find the day
   d) Find the day of the week

2. Are all calculation dependent on each other or are any independent of the others, i.e., is there an order?

3. Write Out Pseudo Code for Solving This Problem
   • We can implement this late in the term
Find Day of the Week

• Interval mod 7 = offset from today in days of the week

• Number days from 0 to 6
  – Conventionalize days based on problem
  – If today is sunday, make sunday 0, ...saturday 6
  – If today is monday, make monday 0, ... sunday 6
  – Etc.
  – Then offset will tell you day of week

• Alternative set sunday to 0, ... saturday to 6 and add number to today's number
Finding year, month, day

• Find out current day of year
  – Add days in prior months to current day in this month.

• Number of years in the future = (Interval + current day of year) \( \div \) 365 (+ some adjustment for leap years)

• New day of year = (Interval + current day of year) \% 365
Find the Month

• Make ranges for months
  – start(january) = 1
  – end(month) = start(month) + length(month) – 1
  – start(month) = end(month-1) + 1

• New day of year is in month, iff
  – start(month) <= new day <= end(month)

• Day of month = (new day – start(month))+ 1

• New year = old year + difference in years
Leap Year Problem

• Changes start/end dates of all months after February as well as the end date of February
  – Calculating ranges for day of month
  – Current year and result year

• Number of days in years that we go over
  – Modulus and integer division need to be replaced with more complex functions
Summary

• A Few Simple Problems
  – Implemented in Python

• One Larger Problem
  – Broken Down into Small Problems
  – The Problems were Ordered
  – Then Solved Using Pseudo Code

• Limitations of Our Current Python Programming:
  – Variables
  – If/Then Structures
  – Repetition Structures