PFTW: Functions, Control, Python/Tools

- How do functions work and why do we use them?
  - Functions we call (APIs), Functions we write
  - Parameters, names, return values
  - Modules, functions, abstractions
- How do we alter control of execution
  - Conditionally execute code in certain situations
  - Repeatedly execute code for set # times or until condition
  - Blocks of code are indented, following :
    - Functions, if/else, loops, other …
- Tools used in Compsci101
  - Submit, APT running, webpages, help, books, LINK, …

Functions: abstractions over code

- Naming something gives you power
  - How do you read a file into a string?
  - What is length of a string? Of a list?
- We can write and call functions
  - Re-use and/or modify
  - Store in module, import and re-use functions
  - Import standard modules and use functions from them
- Functions can (should?) return a value
  - We've seen len return an int, what about file.read()?
  - Other functions return Strings, floats, or other types

Functions: BMI (Body Mass Index)

- What is formula? How to use it?
  - For one person can simply print the BMI
    - `W/(H*H) * 703, W in pounds, H in inches`
  - What if we want to validate data?
  - What if we want to notify folks who might need guidance?

```
def bmi(weight, height):
    return 703 * weight/(height*height)

if bmi(170,72) < 18.5:
  print "underweight"
```

call replaced by return value, why use function?

What does return statement do?

- Programs execute one line at a time
  - After one statement finishes, the next executes
  - Calling a function causes its code to execute
    - What happens in the code that calls the function?
- The value returned *replaces* the function call
  - `print math.sqrt(25.0)`
  - `if bmi(170,72) < 18.5: print "underweight"`
- What if nothing returned?
  - `None` by default in Python

Re-use: Counting words in file

```
def word_count(filename):
    f = open(filename)
    all = f.read()
    words = all.split()
    return len(words)
if __name__ == "__main__":
  name = "/data/romeo.txt"
  print "# words in",name,
  print "=",wordCount(filename)
```

Running Python Program/Module

- Python is an interpreter, platform specific
  - So is Java, so is Android, ... contrast compilers
  - Python can execute a .py file, need a "launch point"
- Convention in Python and other languages
  - Start with section labeled __main__, that's run

```
if __name__ == "__main__":
    statements here
    statements here
```

- Boilerplate, don't memorize, let Eclipse do work!

Nancy Leveson: Software Safety

- Mathematical and engineering aspects, invented the discipline
  - Air traffic control
  - Microsoft word

  "There will always be another software bug; never trust human life solely on software" huffington post?

- Therac 25: Radiation machine
  - http://en.wikipedia.org/wiki/Therac-25
  - http://bit.ly/5qOjoH
- Software and steam engines

Anatomy of a Python function

```
def name(params):
    body
```

- Define a function, provide a name, provide parameters, provide a function body
  - How to decide on name?
  - Do we need parameters?
  - What does body of function do
- Functions provide a named abstraction over code
  - Huh? What is `math.factorial(15)` `"hello".upper()`

Revisiting functions

- Python Heron's formula or BMI (Body Mass Index)
  - What's the name of the function
  - What are parameters that enable function use/call
- How do we write and test APTs
  - What's the name of the function
  - What are parameters that enable function use/call
  - Who writes the function? Who calls the function?
- How will you decide on these things in writing your own code?

Design, Implementation, Testing

- Designing Python code and functions
  - What do you want the code to do?
  - Other aspects of code, e.g., portability, efficiency, size, …
  - Understand how to solve a problem without computer
- Implementing design in code
  - Translation of ideas into vocabulary of Python
  - We don't have a large vocabulary, but it will grow!
- Testing code, functions
  - How do you know when function is right?
  - What confidence can testing provide?
  - APT testing is similar to Unit Testing (well known)

Running Python

- Can run in Eclipse Console Window
  - How to start? What to type?
  - Also run on command-line, e.g. simple Mac/Linux
- Can import code into another module/.py file
  - See APT examples and how to run them
  - Understand how your Python code is executed
  - Understand where Python code is and how it got there
- How do we test your code to grade it, evaluate it?
  - APTs: auto-test, other assignments, human-in-the-loop

Eclipse Particulars

- Supports many languages: we care about Python
  - PyDev perspective: Windows>Open Perspective>Other>…
  - Also use console: Windows>Show View>Console
  - Use PyDev console (right click console icon)
- Preferences are per project or per 'concept'
  - Interpreter for Python? Color for code? Indentation?
  - See Preferences (Windows/Windows, Mac/Eclipse)
- Creating projects, Python Module
- Submitting and check via Ambient

Language and Problems in Context

- Convert Spanish Wikipedia page to English
  - How do we convert HTML to text?
- How do you determine if 2040 is a leap year?
  - Any specified year is a leap year?
- How do we make an image larger, more red, …
  - What is an image? How do read it? Convert it? Access it?
- How do we find the BMI for everyone
  - What's the right tool for this? Why use Python? Why not?

What years are leap years?

- 2000, 2004, 2008, …
  - But not 1900, not 2100, yes 2400!
  - Yes if divisible by 4, but not if divisible by 100 unless divisible by 400! (what?)

```
def is_leap_year(year):
    if year % 400 == 0:
        return True
    if year % 100 == 0:
        return False
    if year % 4 == 0:
        return True
    return False
```

- There is more than one way to skin a cat, but we need at least one way

Python if statements and Booleans

- In python we have if: else: elif:
  - Used to guard or select block of code
  - If guard is True then, else other
- What type of expression used in if/elif tests?
  - ==, <=, <, >, >=, !=, and, or, not, in
  - Value of expression must be either True or False
  - Type == bool, George Boole, Boolean,
- Examples with if
  - String starts with vowel (useful for Piglatin, e.g.,)
  - BMI out of range

Alan Kay

- Turing award 2003
  - OO programming, Dynabook
- "The best way to predict the future is to invent it"
- "Americans have no past and no future, they live in an extended present."

I think the main thing about doing …any kind of programming work, is that there has to be some exquisite blend between beauty and practicality. There's no reason to sacrifice either one of those, and people who are willing to sacrifice either one of those, I don't think really get what computing is all about.