

PFTTheWeek 9/21-9/25

- **Potpourri, Hodgepodge of Review Concepts**
 - File reading and writing
 - Boolean expressions
 - Arithmetic operators
 - Helpful methods and functions

- **Lab, Midterm, Assignment, APT Quiz**
 - Review concepts for lab, lab helps with midterm
 - How do you tackle an assignment that has so many parts?

Software in the News



```
If car.isDriving() or car.onRoad():
    gofaster()
    emit_more()
    get_performance()
elif car.connectedToMonitor():
    beclean()
    register_as_wonderful()
else:
    act_randomly()
    engine_light_on()
```

Reading from Files, Writing to Files

- **Programs generate data, store for access**
 - Notes we take, notebooks we keep
 - Files we make our programs create and add to
- **File concepts for reading and writing**
 - Call open with a path to file, how to open?
 - Choice of reading, writing, appending
 - Read or Write (depending on "r", "a", "w")
 - Close the file when done

Reading from files: see Pigify.py

- **Open file for reading**

- Read lines: `for line in f:`
- Read file: `f.read()`
- Both get strings, convert as needed

- **If fname not found?**

- **Type of f?**

- **Type of st?**

```
def readFile(fname):  
    f = open(fname)  
    st = f.read()  
    f.close()  
    return st.split()
```

Code in Pigify.py

```
def writeFile(words, fname):
    LINE_SIZE = 80
    f = open(fname, "w")
    wcount = 0
    for word in words:
        f.write(word)
        wcount += len(word)
        if wcount + 1 > LINE_SIZE:
            f.write('\n')
            wcount = 0
        else:
            f.write(' ')
    f.close()
```

Questions: File writing and Transform

<http://bit.ly/101fall15-0922-1>

Compound Boolean Expressions

- Is the second character a vowel?

```
if is_vowel(s[1]):
```

- What if word has only one character?

```
if len(s) > 1 and is_vowel(s[1]):
```

```
if is_vowel(s[1]) and len(s) > 1:
```

- Short-circuit evaluation: A false/true

```
if A and B:
```

```
if A or B:
```

Boolean expression review

- How do we figure out what this means?
 - DeMorgan's Laws help
 - $\text{NOT}(A \text{ and } B) \equiv \text{NOT}(A) \text{ or } \text{NOT}(B)$
 - $\text{NOT}(A \text{ or } B) \equiv \text{NOT}(A) \text{ and } \text{NOT}(B)$

```
if not(len(s) < 5 or is_vowel(s[1])):
```

```
if not len(s) < 5 and not is_vowel(s[1]):
```

```
if len(s) >= 5 and not is_vowel(s[1]):
```

Brian Fox

- GNU Bash Shell (developer)
 - Buddycast (co-developer)
- “each person has a sweet spot – a place where they are incredibly productive and at their happiest while doing so – okorians spend their lives living there – the okori sweet spot is the realization of the concept, the delivery of the impossible, from the germ of the idea to the instantiation of it”

<http://www.theokorigroup.com/why>



YALALA

- **Yet Another Look At Loop and Accumulate**
 - Initialize before loop: list, int, string
 - Accumulate in loop: `.append`, `+=`, `+=`
 - Return value after the loop: change as needed

- **What can we loop over?**
 - List: get elements one at a time
 - Strings: get characters one at a time
 - Range: get numbers one at a time
 - Use range elements to index lists or strings

APT and Transform Review

```
def pigall(st):  
    all = []  
    for word in st.split():  
        all.append(pigword(word))  
    return ' '.join(all)
```

```
def encrypt(str, shift):  
    all = []  
    for word in str.split():  
        all.append(shiftword(word, shift))  
    return ' '.join(all)
```

APT and Transform Review

```
def lovely(ingre, inedible) :  
    c = 0  
    for xx in yy:  
        if not x-expression:  
            c = c + 1  
    return c
```

```
def getMessage(original) :  
    trans = []  
    for word in original.split() :  
        trans.append(transform(word))  
    return ' '.join(trans)
```

From Loops to Turtles to Numbers

- **Computers are used for simulations**
 - **Weather forecasting**
 - We have models that work sometimes, use them
 - Test our simulations against past data
 - **Stock-market High Frequency Trading**
 - How do you know your new algorithm works?
 - Test it with small \$\$, Test against history, ...
- **Sometimes visualizing helps**
 - **Code visualization**
 - **Simulation visualization in lab this week**
 - Run Walker.py, look at visualization to check

What is the S in HTTPS?

- **How do we make secure transactions?**
 - Encrypt so that decryption "impossible"
 - What does impossible mean?
- **Depends on deep results from math, we can appreciate these at a high (shallow) level**
 - Multiply two prime numbers: $53 \times 47 = 2491$
 - Can we factor 2491 into two primes?
 - Primality is "easy", Factoring is "hard"

isPrime the easy way, takes "forever"

- Divisors of 28 are: 2, 4, 7, 14
- Divisors of 289 are: 17
- Divisors of N come in pairs, X and N/X
 - Use actual numbers for N and X : 64 and 16, ...
- If there's a factor ...
 - There's one less than or equal to \sqrt{N}
- Plan for testing divisors
 - Keep trying, if we find one ... otherwise ...

Writing and Testing code

- **Conceptually: when can we return True?**
 - When can we return False --- not prime
- **What are special cases?**

```
def isprime(num):  
    limit = int(math.sqrt(num)) + 1  
    # not showing special cases  
    for div in range(3, limit, 2):  
        if num % div == 0:  
            return False  
    return True
```

Counting Questions

<http://bit.ly/101fall15-0922-2>