CompSci 101
Fall 2021

Lecture 11
Reminders

• **Assignments**
  • Assign 2 due 10/7
  • APT 3 due 10/7
  • Assign 3 live

• **APT Quiz 1**
  • 10/8 (8am)-10/11 (11pm)
  • **TWO PARTS**
    • 1.5 hours each
    • Notes, lecture notes, **YOUR** old code, book, and web (for info on Python)
    • **MAY NOT** search for solutions online.
  • PM via Ed only.

• **Small-group tutoring**
Key instructions

• Input ✔
• Output ✔
• Assignments* ✔
• Math/Logic ✔
• Conditionals ✔
• Repetition ✔

*not listed in book
Python Data Types

• int, float, bool ✓

• Collections
  • Strings ✓
  • Lists ✓
  • Tuples
  • Sets
  • Dictionaries
PFTD

- List comprehensions
- Transform Assignment
  - Global variable

“The mere imparting of information is not education.”
  - Dr. Carter G. Woodson
KISS Principle

• Think of the non-computing context for any word/terms
• KISS model
  • Work smarter, not harder!!
• “Good programmers are simply good designers.”
  • -Dr. Washington
• Design first and always!
• Importance of reusability
• *USE PyCharm/PythonTutor IF YOU HAVE QUESTIONS!*
People to Know: Dr. Manuel A. Pérez Quiñones

- DSc (CS)-The George Washington University
- BS/MS(CS)-Ball State University
- Professor, Software and Information Systems
  - UNC-Charlotte
- Personal information management, HCI, CS education, diversity in computing
- Co-founder-Hispanics in Computing
  - http://hispanicsincomputing.org/
def onlyPos(nums):
    ret = []
    for n in nums:
        if n > 0:
            ret.append(n)
    return ret

print(onlyPos([1,2,3,-1,-2,-3]))

return [n for n in nums if n > 0]

• **List Comprehension**
  • We will use a complete, but minimal version of list comprehensions, much more is possible
List Comprehension Syntax

- \( V \) is any variable: all list elements in order
- \( V_{\text{EXP}} \) is any expression, often use \( V \)

\[
\text{ret} = []
\text{for } V \text{ in LIST:}
\quad \text{ret.append}(V_{\text{EXP}})
\]

\[
\text{ret} = [V_{\text{EXP}} \text{ for } V \text{ in LIST}]
\]

\[
\text{ret} = []
\text{for } V \text{ in LIST:}
\quad \text{if } \text{BOOL}_{\text{EXP}}:\n\quad \quad \text{ret.append}(V_{\text{EXP}})
\]

\[
\text{ret} = [V_{\text{EXP}} \text{ for } V \text{ in LIST if } \text{BOOL}_{\text{EXP}}]
\]
List Comprehension Syntax

- if part optional - BOOL_EXP is a Boolean expression usually using V

```python
ret = []
for V in LIST:
    ret.append(V_EXP)
```

```python
ret = []
for V in LIST:
    if BOOL_EXP:
        ret.append(V_EXP)
```

```python
ret = [V_EXP for V in LIST if BOOL_EXP]
```
TPS: List Comprehension Examples

[w for w in words if w.count('e') == 0]
• What does this represent? When is the if true/false?

[v*2 for v in range(6) if v % 2 == 1]
• What does this represent? What is the length?

sum([1 for x in words if len(x) > 4])
• What does this represent?
Activity 1:
https://bit.ly/101f21-10-7-1
Assignment 3: Transform

- Reading and writing files
  - We've seen how to read, writing is similar
  - Open, read, and close
  - Open, write, and close - `.write(...)`

- Apply a function to every word in a file
  - Encrypt and decrypt
  - Respect lines, so resulting file has same structure
Encrypting and Decrypting

• We give you:
  • Transform.py
  • Vowelizer.py - Removes vowels

• You implement
  • Pig Latin
  • Caesar cipher

• Challenge: Shuffleizer
Concepts in Starter Code

- **Global variables**
  - Generally avoided, but very useful
  - Accessible in all module functions

- **FileDialog and tkinter**
  - API and libraries for building UI and UX

- **Docstrings for understanding!**
Global Variables (Best Practice)

• Best practice = help other humans read the code

• All variables that will be global are created with an initial assignment at the top of the file

• When used in a function, variable is declared global at the beginning of the function
What is global?

- Accessible everywhere in the file (or “module”)
- Variable is in the global frame
  - First frame in Python Tutor
- If declared global in a function:
  - The variable in the global frame can also be reassigned in that function’s
  - Despite Python being in a different frame!
- Eliminates the need to pass this value to all the functions that need it
- Demo-Pycharm
When reading code with globals

- When checking the value of a variable, ask:
  - Is this variable local to the function or in the global frame?
- When in a function and assigning a value to a variable, ask:
  - Has this variable been declared global?
    - If yes, reassign the variable in the global frame
    - If no, create/reassign the variable in the function’s local frame
- Demo-global vs. variable
TPS: What will this print?

```python
s = 'top of the file'

def change():
    global s
    s = 'I am'
    t = 'changing'

    print('Inside change:', s, t)

if __name__ == '__main__':
    print('Beginning of main:', s)
    s = 'inside main'
    t = 'text'

    print('Before change:', s, t)
    change()
    print('After change:', s, t)
```

Remember: When considering a variable ask is this variable global?
- If yes: look in global (first) frame for value AND when reassigning do it in the global frame
s = 'top of the file'

def change():
global s
    s = 'I am'
t = 'changing'

    print('Inside change:', s, t)

if __name__ == '__main__':
    print('Beginning of main:', s)
s = 'inside main'
t = 'text'

    print('Before change:', s, t)
change()
    print('After change:', s, t)
## What, where, read, write? (in 101)

<table>
<thead>
<tr>
<th>What is it?</th>
<th>Where first created?</th>
<th>Where accessible? (read)</th>
<th>Where reassign-able? (write)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular variable</td>
<td>In main</td>
<td>In main only (technically anywhere, but don’t do that)</td>
<td>In main only</td>
</tr>
<tr>
<td>in main</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regular local</td>
<td>In function</td>
<td>In function only</td>
<td>In function only</td>
</tr>
<tr>
<td>function variable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global variable</td>
<td>Top of file</td>
<td>If not reassigning the value, in main and all functions</td>
<td>In main or in any function that first declares it global</td>
</tr>
</tbody>
</table>

Python will have an error if it is not declared global and it is used and then there is a variable with the same name being assigned.

Can avoid this by ALWAYS declaring the variable global in the function (best practice) if that is the variable you are using.
Activity 2:
Tkinter and FileDialog

• This library and API is useful for creating GUIs
  • Difficult and not always about the big picture
  • Debugging can be frustrating
  • Tedium of making things right versus exultation in creating wonderful programs!

• If you don't see the rocket-ship? Oops
  • What happens when you run Transform?
Caesar Cipher

- A to Z -> 0 to 25
  - 23\textsuperscript{rd} letter is X

<table>
<thead>
<tr>
<th>Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain text</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>J</td>
<td>K</td>
<td>L</td>
</tr>
<tr>
<td>Cipher text (shift key = 23)</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
</tr>
</tbody>
</table>
findShift(…)

- **Input:** cipher text (a.k.a. encrypted)
- **Output:** shift used for encryption
- **How:** Brute force
  - Have a list of real words
  - Check all possible shift values for cipher text
  - Shift value resulting in the most real words is the decryption key (a.k.a. shift number)
  - Encryption_shift_key = 26 - decryption_key

TPS: Why is this 26 and not 25?
Eyeball Decryption Example

st = "Bxvncrvnb rc'b njbh cx lxdwc oaxv 1-10, kdc wxc jufjhb"

with this code

```python
for sh in range(26):
    setShift(sh)
    print(sh, encrypt(st))
```

results in this output:

<table>
<thead>
<tr>
<th>Shift</th>
<th>Encrypted String</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Bxvncrvnb rc'b njbh cx lxdwc oaxv 1-10, kdc wxc jufjhb</td>
</tr>
<tr>
<td>1</td>
<td>Cywodswoc sd'c okci dy myexd pbyw 1-10, led xyd kvgkic</td>
</tr>
<tr>
<td>2</td>
<td>Dzxpetxp te'd pldj ez nzfye qczx 1-10, mfe yze lwhljd</td>
</tr>
<tr>
<td>3</td>
<td>Eayqfuyqe uf'e qmek fa oagzf rday 1-10, ngf zaf mximke</td>
</tr>
<tr>
<td>4</td>
<td>Fbzrgvzrf vg'f rnf1 gb phabit sebz 1-10, ohg abg nyjlnf</td>
</tr>
<tr>
<td>5</td>
<td>Gcashwasg wh'g sogm hc qcibh tfca 1-10, pin bch ozkomoq</td>
</tr>
<tr>
<td>6</td>
<td>Hdbtixbth xi'h tphn id rdjci ugdib 1-10, qji cdi palpnh</td>
</tr>
<tr>
<td>7</td>
<td>Iecujycui yj'i uqio je sekdj vhec 1-10, rkj dej qbmqoi</td>
</tr>
<tr>
<td>8</td>
<td>Jfdvkzdvj zk'j vrjp kf tflek wifd 1-10, slk efk rcrnpj</td>
</tr>
<tr>
<td>9</td>
<td>Kgewlaewk al'k wskq lg ugmfl xgje 1-10, tml fgl sdosqk</td>
</tr>
<tr>
<td>10</td>
<td>Lhfxmbxf1 bm'l xtlr mh vhngm ykhf 1-10, unm ghm teptrl</td>
</tr>
<tr>
<td>11</td>
<td>Migyncgym cn'm yums ni wiohn zlig 1-10, von hin ufqusm</td>
</tr>
<tr>
<td>12</td>
<td>Njhzodhzn do'n zvnt oj xjpio amjh 1-10, wpo ijo vgrvt</td>
</tr>
<tr>
<td>13</td>
<td>Okiapeiaeo ep'o awou pk ykqjp bnki 1-10, xgp jkp whswuo</td>
</tr>
<tr>
<td>14</td>
<td>Pljbqfjbp fq'p bpxv ql zlrkq colj 1-10, yrq klq xitzvp</td>
</tr>
<tr>
<td>15</td>
<td>Qmkmcrqkcq gr'q cyqw rm amslr dpmk 1-10, zsr lmr yjuywq</td>
</tr>
<tr>
<td>16</td>
<td>Rnlshblrh hs'r dzrx sn bntms eqnl 1-10, ats mns zkvvxr</td>
</tr>
<tr>
<td>17</td>
<td>Sometimes it's easy to count from 1-10, but not always</td>
</tr>
<tr>
<td>18</td>
<td>Tnpfnjft ju't fbtz up dpvou gspn 1-10, cvu opu bmxbzt</td>
</tr>
<tr>
<td>19</td>
<td>Uqogvkogu kv'u gcuva vq eqwpv htnq 1-10, dwv pqv cnycau</td>
</tr>
<tr>
<td>20</td>
<td>Vrphwlphv lw'v hdvb wr frxqwh iurp 1-10, exw qrw dozdbv</td>
</tr>
<tr>
<td>21</td>
<td>Wsqixmqiw mx'w iewc xs gsyrx jvsq 1-10, fyx rsx epeacw</td>
</tr>
<tr>
<td>22</td>
<td>Xtrjynrjx ny'x jfxd yt htsy kwtr 1-10, gzy sty fqbfdx</td>
</tr>
<tr>
<td>23</td>
<td>Yuskzosoz oz'y kgye zu iuatx lxus 1-10, haz tuz grcgey</td>
</tr>
<tr>
<td>24</td>
<td>Zvtlapltz pa'z lhzf av jvbua myvt 1-10, iba uva hsdhfz</td>
</tr>
<tr>
<td>25</td>
<td>Awumbquma qb'a miag bw kwcvb nzwu 1-10, jcb vwb iteiga</td>
</tr>
</tbody>
</table>

...
Reminders

• Work smarter, not harder
• Design first
• Try to identify where you are stuck
  • Identify resources to help solve problem
• Leverage your design and PythonTutor to understand program flow of control
  • http://pythontutor.com