Compsci 101
List and String Operations, For loop

Susan Rodger
January 27, 2022

G is for …

- Google
  - How to find the answer to everything
- Global Variable
  - Accessible everywhere, typically do not do
- GIGO
  - Garbage In, Garbage Out
- Git
  - Working Together or Solo

Sir Tim Berners-Lee

- Invented World Wide Web
- Turing award 2016
- HTTP vs. TCP/IP
  - Just protocols?

“The Web as I envisaged it, we have not seen it yet. The future is still so much bigger than the past.”

“We need diversity of thought in the world to face the new challenges.”

Did you sign up for compsci@duke.edu mailing list?

- Mailing list to get the CompSci weekly newsletter
  - Events, research and job opportunities
- To add yourself:
  - Go to lists.duke.edu
  - Authenticate and then add compsci@duke.edu
- Sample item:
  - Duke Women in Tech looking for new members and to get our mailing lists. Fill out this form: https://tinyurl.com/witspring22
Announcements

• Assignment 1 Faces due today 11:30pm
  • Also REFLECT Form due same time
  • Get one grace day, but no consulting hours on Friday

• Exam on Tuesday!, Feb 1

PFTD

• Exam 1
• Lists continued
• String methods and more
• For Loops

Exam 1
Read all rules posted in Announcement in Sakai

• This is your own work, no collaboration
• No book, No notes, only Exam 1 Python Ref Sheet

• Do not search for answers on the internet
• Do not type in code where it can be compiled and run
  • Do not use Pycharm, textbook code boxes, Python tutor or any other place the code can be run
• Do not talk to anyone about the exam during the exam, and until it is handed back!

Exam 1 Logistics

• Take on Tues. Feb 1 between 8am and 11pm
• You pick the start time
  • Must start by 9:30pm
• You get 1 hour 30 min
  • Longer if you have accommodations
• Once you start, your timer starts and you must finish in 1 hour, 30 minutes
• You cannot pause the timer
Exam 1 Logistics (2)

- Go to Gradescope to start
- Click on Exam 1 to start
- Gradescope saves answers as you type them in
  - Type 4 spaces to indent code
- Disconnected? Just log back in to Gradescope
- Question? Post a private post on Ed Discussion

- We do not have lecture on Feb. 1, Just take exam

Don’t go to Gradescope site until you are ready to start!

You click it, you have started!

We do not restart it!

Compare assign with integers, strings and lists – 1

```python
Python 3.6
(known limitations)

Frames | Objects
--- | ---
1 | x = 6
2 | y = x
3 | x = 3
4 | m = "pink"
5 | n = m
6 | m = "red"
7 | a = ["pig", "cow", "dog"]
8 | b = a
9 | a[-1] = "ant"

Edit this code

- line that just executed
- next line to execute
```

Compare assign with integers, strings and lists – 2

```python
Python 3.6
(known limitations)

Frames | Objects
--- | ---
1 | x = 6
2 | y = x
3 | x = 3
4 | m = "pink"
5 | n = m
6 | m = "red"
7 | a = ["pig", "cow", "dog"]
8 | b = a
9 | a[-1] = "ant"

Edit this code

- line that just executed
- next line to execute
```
List Cloning (or copying)

```python
lst1 = ['a', 'b', 1, 2]
lst2 = lst1
lst3 = lst1[:]
```

List Cloning (or copying)

```python
lst1 = ['a', 'b', 1, 2]
lst2 = lst1
lst3 = lst1[:]
```

WOTO-1 Cloning

List Concatenation Steps

1. Calculate the **length** of the new list
2. **Create** list of that length
3. **Copy** values from first list
4. **Copy** values from second list
5. **Assign the variable to the new list**

```python
lst0 = [1, 2]
lst1 = [3, 4, 5]
lst2 = lst0 + lst1
```
Concatenation: length, create, copy, copy, assign

1. `lst0 = [1,2]`
2. `lst1 = [3,4,5]`
3. `lst2 = lst0 + lst1`

Concatenation: Makes new List

1. `lst0 = [1,2]`
2. `tmp = lst0`
3. `lst0 = lst0 + [4]`

What will Python Tutor Display? How many lists will there be?

Concatenation: Makes new List

1. `lst0 = [1,2]`
2. `tmp = lst0`
3. `lst0 = lst0 + [4]`

What will Python Tutor Display? How many copies of ['b', 3.0] will be present?

Concatenation: Makes new List

1. `lst0 = [1,2]`
2. `tmp = lst0`
3. `lst0 = lst0 + [4]`

• How is the inner list copied?

1. `lst0 = [1, ['b', 3.0]]`
2. `lst1 = [4]`
3. `lst2 = lst0 + lst1`

What will Python Tutor Display? How many copies of ['b', 3.0] will be present?
List Mutation: `.append(…)`

- `.append()` – list function that adds element to end of list
  - Mutates list to left of “.”
  - “.” – call function to the right of the dot on the thing to the left of the dot (LEFT.RIGHT)

```python
x = [6, 2, 4]
x.append(3)
x.append([5,2])
```

What will Python Tutor Display? One or two lists?

WOTO-2 – Mutable and Append

Anatomy of a **for** loop

```python
for VARIABLE in SEQUENCE:
    CODE_BLOCK
```

- Think of as:
  - “For each element in the SEQUENCE put it in the VARIABLE and execute the CODE_BLOCK.”
- Also called: *iterate* over the sequence
- What type(s) are sequences?
  - Strings, Lists
- Will VARIABLE likely be in CODE_BLOCK?

---

Example for loop with a list

```python
1 lst = [5, 3, 2]
2 sum = 0
3 for num in lst:
4    sum = sum + num
5 print(sum)
```

- What is first value of `num`?
- What is final value of `num`?

---

Trace through for loop – 1

```python
1 lst = [5, 3, 2]
2 sum = 0
3 for num in lst:
4    sum = sum + num
5 print(sum)
```
Example for loop with a string

- What does this for loop do?

```
word = 'cat'
for ch in word:
    word = word + ch
print(word)
```

- What is first value of ch?

- What is final value of ch?
String's split(...) 

- Strings have functions too!
- TYPE_STRING.FUNCTION(PARAMETERS)
  - "." means apply function to what is on the left
  
  'one fish two fish'.split() returns a list

- What did it divide the string by?
  - When no parameter, default whitespace
  
  'one fish, two fish'.split(',')

String's join(...) 

- TYPE_STRING.join(SEQ_OF_STRINGS)
  - Opposite of .split()
  - Creates string from sequence's items separated by the string to the left of join
  
  ' '.join(['one','fish','two','fish'])

  '+' .join(['one','fish','two','fish'])

  'ish'.join(['f','w','d','end'])

More Methods

<table>
<thead>
<tr>
<th>String</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>.find(s)</td>
<td>sum(lst)</td>
</tr>
<tr>
<td>.rfind(s)</td>
<td>sum of the elements</td>
</tr>
<tr>
<td>.upper()/</td>
<td>in lst</td>
</tr>
<tr>
<td>.lower()</td>
<td>max(lst)</td>
</tr>
<tr>
<td>.strip()</td>
<td>min(lst)</td>
</tr>
<tr>
<td>.count(s)</td>
<td>.append(elm)</td>
</tr>
<tr>
<td>.startswith(s)</td>
<td>Mutates the list by</td>
</tr>
<tr>
<td>.endswith(s)</td>
<td>adding elm to the</td>
</tr>
<tr>
<td></td>
<td>end of the list</td>
</tr>
</tbody>
</table>

WOTO-3 – Split and Join