Indexing, Slicing

Announcements

- Lab 1 is due tomorrow night
  - Ninja session tonight 7-9pm
- First quiz is next Friday
  - Covers weeks 1 and 2
  - 25 minutes
- Extra practice questions on website
- Lab 2 available Sunday

Today's plan

- Review string formatting and accumulator pattern
- `+`, `*`, `len`, `type` for sequences
- Indexing
  - Indexing within a `for` loop
- Slicing

String formatting

- <format string> % (value1, value2, …)
- Each value corresponds to a placeholder in the format string
- Values are converted to strings, then replace placeholders in format string **based on order**
- %s, %d, and %f are placeholders for strings, ints, and floats respectively

String formatting

- We can also specify the **padding** and in the case of floats, the **precision**.
- `%<padding>s, %<padding>d`
- `%<padding>.<precision>f`

Accumulator pattern

- Use this pattern when you want to *combine the values in a sequence into one value*

Accumulator Pattern

- Initialize accumulator, a variable
  - *What should the initial value be?*
- Inside of a loop, update accumulator to new value, using old value
  - *What operation do I use to do update?*
- Use final value after the loop finishes
  - *What do I do with it?*

Revisiting average program

```
"""
This program averages a list of numbers.

David Mauskop, CS 21
"""

def main():
  total = 0
  n = int(raw_input("How many numbers? "))

  for i in range(1, n+1):
    prompt = "%2d) " % i
    next_number = float(raw_input(prompt))
    total = total + next_number

  # why don't I need to convert total to a float?
  average = total/n
  print("The average is: %.1f" % average)

main()
```

Sequence operators/functions

- `+` does concatenation
- `*` does replication
- `len` function finds the length
- `type` function would return one of:
  - `<type 'str'>`
  - `<type 'list'>`

Indexing a sequence

- Isolate one of the values in a sequence
- Syntax: `<sequence>[<index>]`

```
>>> L = [3, 5, 10]
>>> L[0]
3
>>> L[1]
5
>>> L[2]
10
>>> L[3]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>> "swat"[3]
't'
```

Indexing in a `for` loop

```
for i in range(len(seq)):
    print("Index: %d, Value: %s" % (i, seq[i]))
```

Slicing a sequence

- Get a sub-sequence of a sequence
- Syntax: `<sequence>[<start>:<stop>]`

```
>>> s = "superb"
>>> s[1:len(s)-1]
'uper'
>>> s[:2]
'su'
>>> s[2:]
'perb'
>>> s[:]
'superb'
>>> L = [3, 5, 10]
>>> L[1:]
[5, 10]
```

More on slicing

- `<sequence>[<start>:<stop>:<step>]`
- If `<start>` is omitted, it's assumed to be 0
- If `<stop>` is omitted, it's assumed to be `len(<sequence>)`
- If `<step>` is omitted, it's assumed to be 1

Practice

Enjoy the weekend!