Announcements

• Reading and RQ 10 due next time
• APT 4 is due on Tuesday

• Today:
  – A different way to process elements
    • List comprehension
  – Coming – more ways to process data
  – Exam 1 back

Creating a list

• Given a list of numbers, create a second list of every number squared.

```python
tenums = [8, 3, 5, 4, 1]
sqnums = []
for v in nums:
    sqnums.append(v*v)
print sqnums
```

[64, 9, 25, 16, 1]

More on List operations

• Previous page
  – `nameOfList “dot” function (parameter)`
    `sqnums.append(v*v)`
  – See list operations on next page
  – Mutator vs hybrid vs return
    – Mutator changes the list (no return value)
    – Hybrid changes list and returns value
    – Return – returns value, no change to list
### Problem

- Remove all negative numbers from list 
  \[ [4, -2, 5, 6, -3] \rightarrow [4, 5, 6] \]

- Two ways
  1) return a new list with all negative numbers removed
  2) Modify a list to remove negative numbers

---

```python
def removeNegatives(numberlist):
    answer = []
    for num in numberlist:
        if num >= 0:
            answer.append(num)
    return answer

somenums = [3, -1, 8, -5, -2, 6, 7]
nonegs = removeNegatives(somenums)
```

```python
def removeNegatives2(numberlist):
    for x in range(len(numberlist)):
        value = numberlist[x]
        if value < 0:
            numberlist.pop(x)

somenums = [3, -1, 8, -5, -2, 6, 7]
removeNegatives2(somenums)
```
```python
def removeNegatives3(numberlist):
    pos = 0;
    while (True):
        if pos >= len(numberlist):
            break
        value = numberlist[pos]
        if value < 0:
            numberlist.pop(pos)
        pos = pos + 1
```

somenums = [3, -1, 8, -5, -2, 6, 7]
removeNegatives3(somenums)

---

**List Comprehension**

- Take advantage of patterns, make a new list based on per element calculations of another list
- Format:
  
  `<expression with variable> for <variable> in <old list>]`
- Example:
  
  `nums = [8, 3, 5, 4, 1]`
  `sqnums = [v*v for v in nums]`

---

**Richard Stallman**

- MacArthur Fellowship (Genious grant)
- ACM Grace Murray Hopper award
- Started GNU – Free Software Foundation (1983)
  
  - GNU Compiler Collection
  - GNU Emacs

---

These result in the same list!

```
nums = [8, 3, 5, 4, 1]
```

1) `sqnums = []`
   `for v in nums:`
   `sqnums.append(v*v)`

2) `sqnums = [v*v for v in nums]`
Examples of List Comprehensions

bit.ly/101sp16-0225-4

nums = [4, 3, 8]
[v for v in nums]
[2 for v in nums]
sum([v*2 for v in nums])
[v+5 for v in nums][1]

Creating a list with just the even numbers

nums = [8, 3, 5, 4, 1]
evennums = []
for v in nums:
    if v % 2 == 0:
        evennums.append(v)
print evennums

[8, 4]

List Comprehension with Filtering

• Create list and use “if” to filter out elements to the list
• Format:
• [<expression with variable> for <variable> in <old list> if <filter with variable> ]
• Example: nums = [8, 3, 5, 4, 1]
evennums =
    [v for v in nums if v%2==0]

More on List Comprehensions


names = [“Bo”, “Moe”, “Mary”, “Aaron”, “Joe”]
• What is the list for the following:
  1) [w for w in names if w.endswith(“e”)]
  2) [w for w in names if w.lower()[0] > ‘c’]
  3) [j+1 for j in range(20) if (j%3) == 0]
  4) [i*2 for i in [j+1 for j in range(20)
      if (j%3) == 0] if i*i > 19]
Giving Back Exam 1...

- Will post solutions
- Try working problem you missed first
  - Then look at solution
- Once you think you understand
  - Get blank sheet of paper – try again
- Understand all solutions

Will provide extra help
Group Tutoring