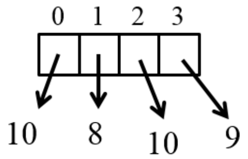
CompSci 101
Introduction to Computer Science



September 18, 2014

Prof. Rodger

Announcements

- Reading for next time on calendar page
  - RQ 7 due Tuesday
- Assignment 3 in
  - Assignment 4 out today
  - APT 3 due Tuesday
- Tutors – Peer Tutoring Center is hiring tutors now, will be available soon
- Consulting hours/office hours are free, go on non busy nights

Includes Notes from previous lectures

Lists

- A list is a collection of objects

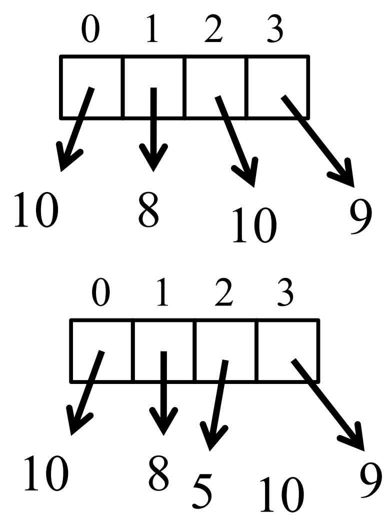
```
scores = [99, 78, 91, 84]
allAboutMe = [“Mo”,25, “934-1234”]
club=[‘Mo’,‘Jo’,‘Po’, ‘Flo’, ‘Bo’]
```

- Lists are *mutable* – use [num] to change a value
- Lists are indexed starting at 0, or -1 from the end
- Functions: max, min, len, sum
- Slice lists [:]

List Examples

```
scores = [10, 8, 10, 9]
print scores
scores[2] = 5
print scores
print max(scores)
print len(scores)
print sum(scores)
print scores[1:]
print scores[1]
```

List before/after modification



score = [10,8,10,9]

score [2] = 5

Processing List Items

- Process all the items in a list, one item at a time
- Format: `for` variable `in` list:

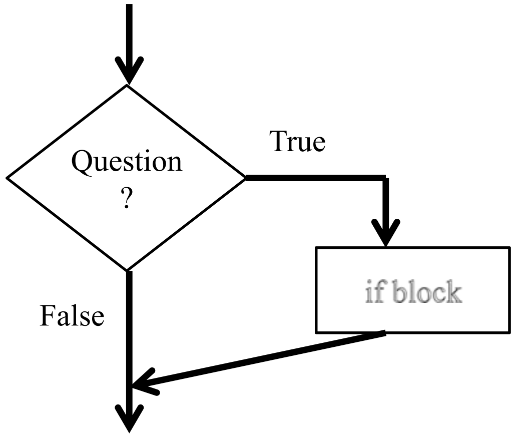
  block
- Example:

```
sum = 0
nums = [6, 7, 3, 1, 2]
for value in nums:
    sum = sum + value
print sum
```

Copying vs aliasing

```
names = [‘jo’, ‘mo’, ‘bo’]
club = names
team = names[:]
names[1] = ‘flo’
print names
print club
print team
```

Making Decisions



Making Decisions in Python

if *condition1*:

Block of code to do if condition is true

elif *condition2*:

Block of code to do if condition1 false, condition2 is true

else:

Block of code to do if other conditions false

- Can have many elifs, leave out elif, leave out else

Making Decisions tools

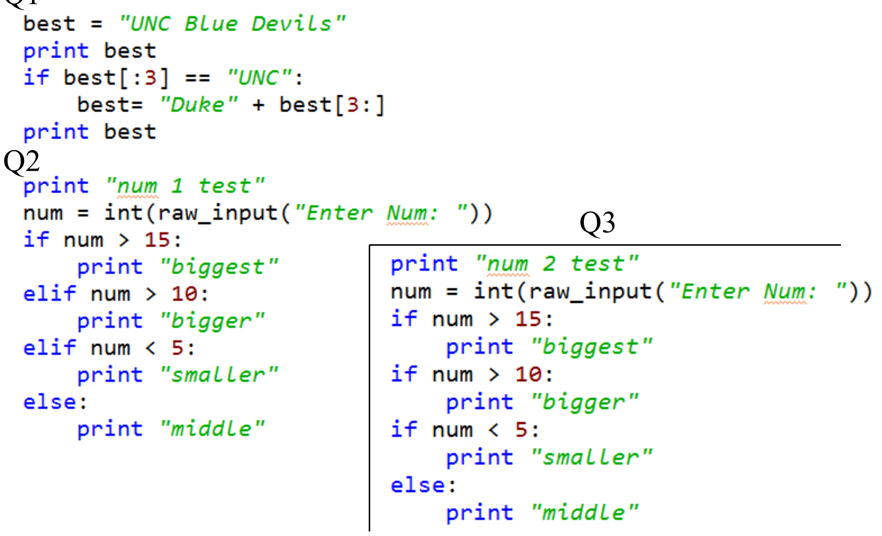
- Boolean values: True, False
- Boolean operators: and, or, not

X	Y	X and Y	X or Y
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

- Relational operators: <, <=, >, >=
- Equality operators: ==, !=
- Look at if examples: miscIf.py

Compare Ifs

Form: `www.bit.ly/101fall14-0918-01`



Q1

```
best = "UNC Blue Devils"
print best
if best[:3] == "UNC":
    best= "Duke" + best[3:]
print best
```

Q2

```
print "num 1 test"
num = int(raw_input("Enter Num: "))
if num > 15:
    print "biggest"
elif num > 10:
    print "bigger"
elif num < 5:
    print "smaller"
else:
    print "middle"
```

Q3

```
print "num 2 test"
num = int(raw_input("Enter Num: "))
if num > 15:
    print "biggest"
if num > 10:
    print "bigger"
if num < 5:
    print "smaller"
else:
    print "middle"
```

More on lists

- range (1,11)
  - Generates a list of numbers from 1 to 10

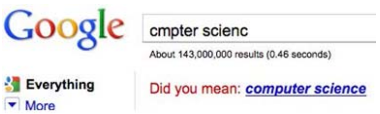
- Example:

```
for num in range(1,11):
   print num
```

Example

```
answer = 0
for num in range(1,10):
    if num % 3 == 0:
        answer = answer + num
    else:
        answer = answer + 1
```

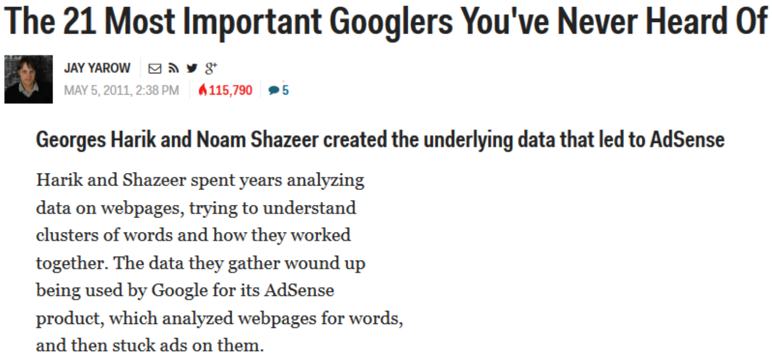
Computer Science Duke Alum



cmpter scienc

About 143,000,000 results (0.46 seconds)

Did you mean: ***computer science***



The 21 Most Important Googlers You've Never Heard Of

JAY YAROW
MAY 5, 2011, 2:38 PM 115,790 5

Georges Harik and Noam Shazeer created the underlying data that led to AdSense

Harik and Shazeer spent years analyzing data on webpages, trying to understand clusters of words and how they worked together. The data they gather wound up being used by Google for its AdSense product, which analyzed webpages for words, and then stuck ads on them.

Dissect the for loop

for VARIABLE in STRUCTURE:
    BODY

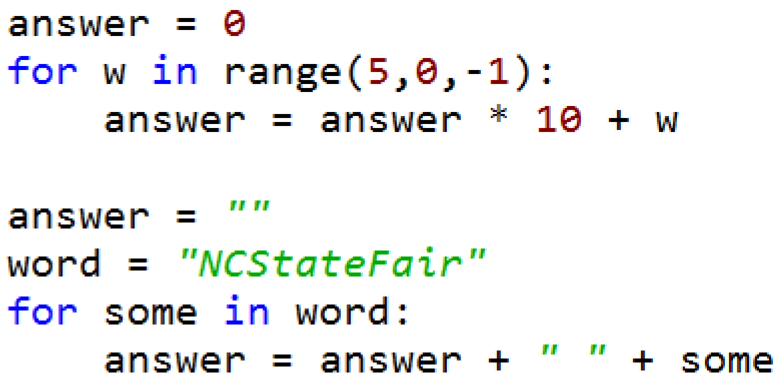
Repeat the BODY with the VARIABLE equal to each item in structure

What can the structure be? Variable be?

- STRUCTURE → Variable
- String → character
- List → item in list

- There are other types of structures we will see

Examples



```
answer = 0
for w in range(5,0,-1):
    answer = answer * 10 + w

answer = ""
word = "NCStateFair"
for some in word:
    answer = answer + " " + some
```

Reading from Files

- Must open file, close file
- Easiest way, read one line as a string and then process it

```
inputfile = open(“datafile.txt”)
for line in inputfile:
   line = line.strip()
   ‘’’  do something with line ‘’’

inputfile.close()
```

Dissect the for loop (again)

```
for  VARIABLE in STRUCTURE:
     BODY
```

```
inputFile = open(“somefile.txt”)
for str in inputFile:
     process str
```

Writing to File

- Must open file, close file
- Open file for writing

```
outfile = open(“outputfile.txt”, ‘w’)
phrases = [“hello there”, … ]
for phr in phrases:
  outfile.write(phr + “\n”)

outfile.close()
```

Note: refresh to see the file

Exercise with files

- Snarf code
- Form:

www.bit.ly/101fall14-0918-02