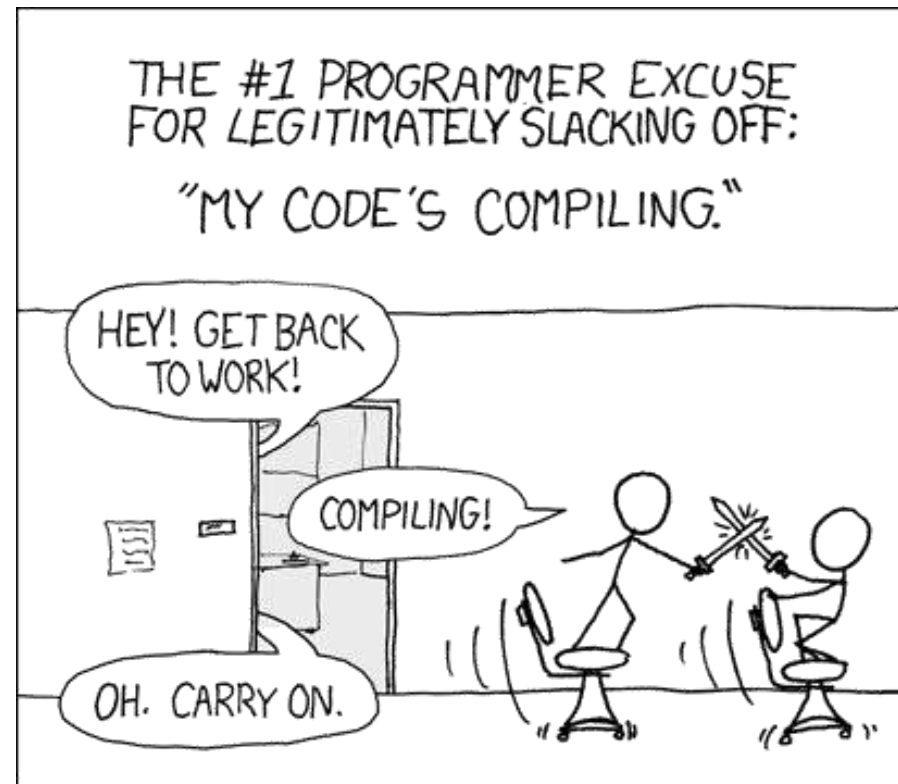


CSc 110, Spring 2017

Lecture 20: Lists for Tallying; Text Processing

Adapted from slides by Marty Stepp and Stuart Reges

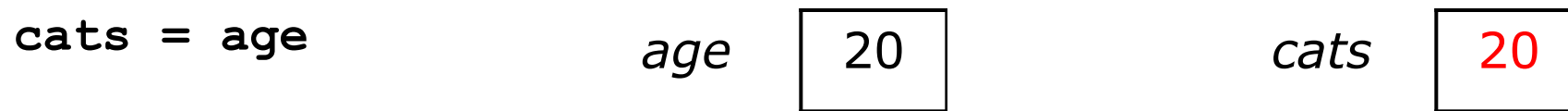


Value/Reference Semantics - Review

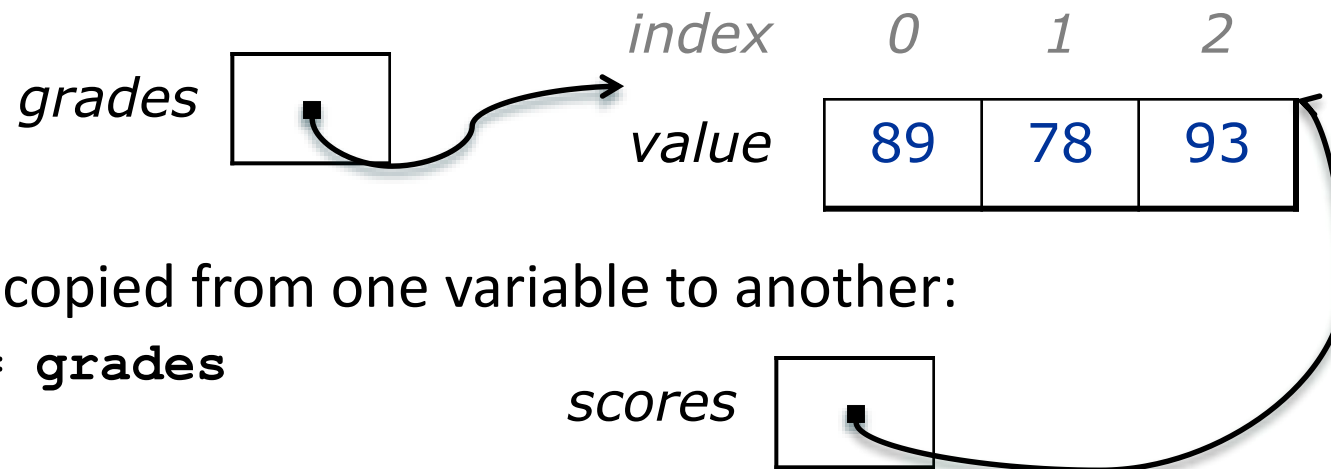
- Variables of type int, float, boolean, store values directly:



- Values are copied from one variable to another:



- Variables of other types (like lists) store references to memory:



- References are copied from one variable to another:

scores = grades

Lists for Tallying

Extracting digits

- Given a number, how do we extract the digits one at a time? Ex: 590823
- Hint: use % and //

```
>>> n = 590823
>>> n % 10
3
>>> n = n // 10
>>> n
59082
>>> n % 10
2
>>> n = n // 10
>>> n
5908
>>> n % 10
8
>>>
```

A tallying problem

- Problem: Write a function `most_frequent_digit(n)` that returns the digit of a number `n` that occurs most frequently.
 - Example: the number 669260267 contains:
one 0, two 2s, four 6s, one 7, and one 9.
`most_frequent_digit(669260267)` returns 6.
 - If there is a tie, return the digit with the lower value.
`most_frequent_digit(57135203)` returns 3.

A tallying problem

- This is well-suited for a list.
- Note that there are 10 digits. Consider a list of 10 elements.
 - The value at index i holds the number of occurrences of digit i
 - Example for 669260267:

<i>index</i>	0	1	2	3	4	5	6	7	8	9
<i>value</i>										

Creating a list of tallies

```
# assume n = 669260267
counts = [0] * 10
while (n > 0):
    # pluck off a digit and add to its counter
    digit = n % 10
    counts[digit] = counts[digit] + 1
    n = n // 10
```

index 0 1 2 3 4 5 6 7 8 9

<i>value</i>	1	0	2	0	0	0	4	1	0	1
--------------	---	---	---	---	---	---	---	---	---	---

Tally solution

Returns the digit value that occurs most frequently in n.

Breaks ties by choosing the smaller value.

```
def most_frequent_digit(n):
    counts = [0] * 10
    while (n > 0):
        digit = n % 10      # pluck off a digit and tally it
        counts[digit] = count[digit] + 1
        n = n // 10

    # find the most frequently occurring digit
    best_index = 0
    for i in range(1, len(counts)):
        if (counts[i] > counts[best_index]):
            best_index = i
    return best_index
```


Data transformations

- In many problems we transform data between forms.
 - Example: digits \rightarrow count of each digit \rightarrow most frequent digit
 - A transformation is computed/stored as a list.
- Sometimes we map between data and list indexes.
 - tally (if digit is i , store its count at index i)
- The problem structure affects the mapping

Section attendance question

- Read a file of section attendance (*see next slide for structure*):
- And produce the following output:

```
Section 1  
Student points: [20, 16, 17, 14, 11]  
Student grades: [100.0, 80.0, 85.0, 70.0, 55.0]
```

```
Section 2  
Student points: [16, 19, 14, 14, 8]  
Student grades: [80.0, 95.0, 70.0, 70.0, 40.0]
```

```
Section 3  
Student points: [16, 15, 16, 18, 14]  
Student grades: [80.0, 75.0, 80.0, 90.0, 70.0]
```

- Students earn 3 points for each section attended up to 20.

Section input file

student	1234512345123451234512345123451234512345123451234512345
week	1 2 3 4 5 6 7 8 9
section 1	yyynyynnayayynyyyaayanyyyaaynayaayyanayyyanayayna
section 2	ayyanyyyyayanaayyanayyyananaayayaynyayayynynya
section 3	yyayaynyyayyanynnyyyayyanayaynannnyyayyayayny

- Each line represents 9 weeks of attendance data for a section.
- Each week has 5 characters because there are 5 students in all sections.
- Within each week, each character represents one student's attendance:
 - a means the student was absent (+0 points)
 - n means they attended but didn't do the problems (+2 points)
 - y means they attended and did the problems (+3 points)

Section input file (fragment)

- Look at 2 weeks of one section (one line of the file):

<i>student</i>	1	2	3	4	5	1	2	3	4	5
<i>index</i>	0	1	2	3	4	5	6	7	8	9
<i>value</i>	y	y	n	y	y	y	n	a	y	a

- For index i , a student is $i \% 5$.

Student 1	Student 2	Student 3	Student 4	Student 5
$0 \% 5$	$1 \% 5$	$2 \% 5$	$3 \% 5$	$4 \% 5$
$5 \% 5$	$6 \% 5$	$7 \% 5$	$8 \% 5$	$9 \% 5$

Need a list of length 5 to calculate the cumulative points for each student

Get the points for each student of a section

#Computes points earned for each student in a particular section.

```
def count_points(line):
    points = [0] * 5
    for i in range(0, len(line)):
        student = i % 5
        earned = 0
        if (line[i] == 'y'):           # values are 'y', 'n' or 'a'
            earned = 3
        elif (line[i] == 'n'):
            earned = 2
        points[student] = points[student] + earned)
    return points
```

Note: fix the code to cap the points earned

Section Attendance - Answer

```
# This program reads a file representing which students attended  
# which discussion sections and produces output of the students'  
# section attendance and scores.
```

```
def main():  
    file = open("sections.txt")  
    lines = file.readlines()  
    section = 1  
    for line in lines:  
        # process one section  
        points = count_points(line)  
        grades = compute_grades(points)  
        results(section, points, grades)  
        section += 1  
  
# Produces all output about a particular section.
```

```
def results(section, points, grades):  
    print("Section " + str(section))  
    print("Student scores: " + str(points))  
    print("Student grades: " + str(grades))  
    print()
```

...

Section Attendance - answer

...

Computes the points earned for each student for a particular section.

```
def count_points(line):
    points = [0] * 5
    for i in range(0, len(line)):
        student = i % 5
        earned = 0
        if (line[i] == 'y'):           #values are 'y', 'n' or 'a'
            earned = 3
        elif (line[i] == 'n'):
            earned = 2
        points[student] = min(20, points[student] + earned)
    return points
```

Computes the percentage for each student for a particular section.

```
def compute_grades(points):
    grades = [0] * 5
    for i in range(0, len(points)):
        grades[i] = 100.0 * points[i] / 20
    return grades
```