CSC 111:
Intro to Computer Science through Programming

Spring 2017
Prof. Sara Mathieson

Admin

- No Lab 4 due to Rally Day
- Quiz solutions are posted
- **Start Homework 3 early!**
- You should be spending roughly 10 hours per week on CSC 111 outside of class/lab

Lab reminders

- Make sure to come to lab and come on time
  - Lab attendance is required and more than one absence will affect your lab grade (unless you have a deans notice)
- Work with your partner
  - It is unfair to your partner if they never get the chance to code
- Talk to your partner
  - Part of the goal of lab is not only to program, but to learn to communicate using this new language

Outline: 2/17

- Feedback from Monday
- Continue guess my number program
- Recap string/list parallels
- Slicing strings and lists

Feedback from Monday

Monday anonymous questions

- Pace seems mostly okay, but a bit fast during live coding (too fast switching between code and slides)
- Forming study groups ("Search for Teammates" on Piazza)
- Coding techniques and how to approach problems
- More ungraded practice problems (try the book exercises)
- TA hours are busy (try Wed/Thurs, try to find others working on the same part of the homework)
- Go over homework and labs

Homework 3 Preview: ELIZA

http://www.masswerk.at/elizabot/

Guess my number (live coding)

Guess my number extensions

- Use a Boolean variable to "flip" the state from not having guessed the answer to having guessed the answer
- Compute the number of tries it took the computer to guess

```
Enter a integer between 0 and 9 inclusive: 6
8 is wrong, guess again.
1 is wrong, guess again.
8 is wrong, guess again.
0 is wrong, guess again.
You guessed my number!
You took 4 tries to guess my number.

Enter a integer between 0 and 9 inclusive: 7
5 is wrong, guess again.
4 is wrong, guess again.
3 is wrong, guess again.
8 is wrong, guess again.
6 is wrong, guess again.
8 is wrong, guess again.
9 is wrong, guess again.
0 is wrong, guess again.
2 is wrong, guess again.
1 is wrong, guess again.
3 is wrong, guess again.
1 is wrong, guess again.
3 is wrong, guess again.
6 is wrong, guess again.
You guessed my number!
You took 14 tries to guess my number.

Enter a integer between 0 and 9 inclusive: 8
9 is wrong, guess again.
6 is wrong, guess again.
5 is wrong, guess again.
3 is wrong, guess again.
9 is wrong, guess again.
You guessed my number!
You took 5 tries to guess my number.
```

Recap parallels between strings and lists

Similarities between list and str

`lst = ["zero","one","two","three", 'four","five"]`	Assign a value	`string = "Smith College"`
`len(lst)` `6`	length	`len(string)` `13`

`lst2 = ["six","seven","eight"]`
`lst + lst2`
`['zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']`
	concatenate + sign	`string2 = " is in Northampton"`
		`string + string2`
		`'Smith College is in Northampton'`

`lst[3]` `'three'` `lst[0]` `'zero'`	indexing [square brackets]	`string[3]` `'t'` `string[0]` `'S'`
`>>> type(lst)` `<class 'list'>`	get type	`>>> type(string)` `<class 'str'>`
`>>> lst.count("two")` `1`	count	`>>> string.count("e")` `2`

Strings are special though…

```
>>> lst.split()
Traceback (most recent call last):
  File "<pyshell#39>", line 1, in <module>
    lst.split()
AttributeError: 'list' object has no attribute 'split'
```

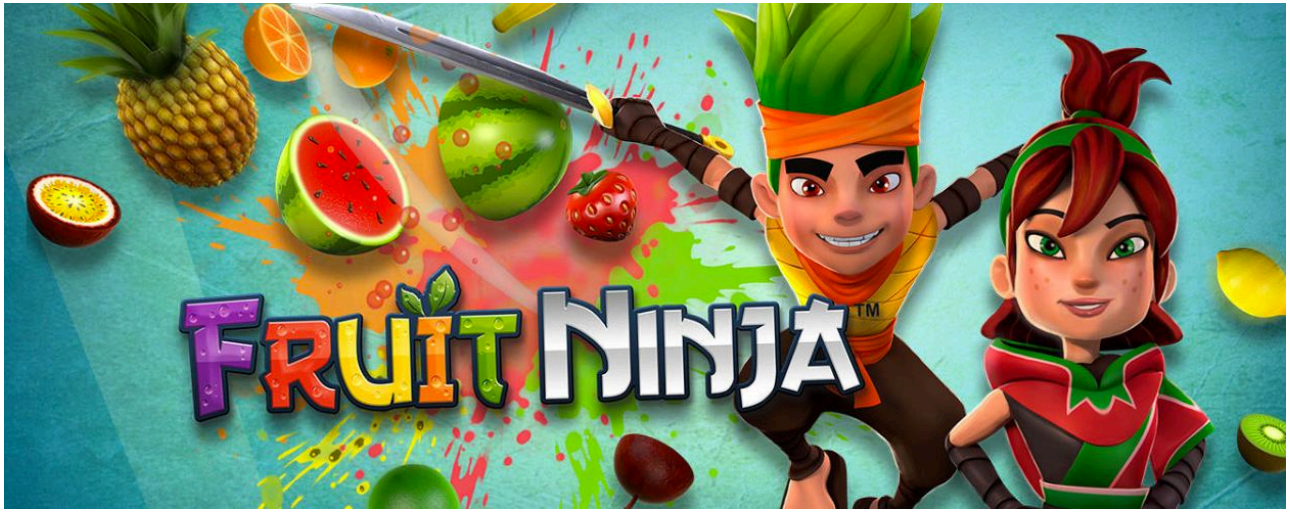
```
>>> string.split()
['Smith', 'College']
```

```
>>> string.lower()
'smith college'
```

```
>>> string.upper()
'SMITH COLLEGE'
```

```
>>> string.replace(" ","")
'SmithCollege'
```

“Slicing and dicing”



Idea: create a substring or sublist

- Still use square brackets, but with start and stop elements separated by a colon:

```
>>> college = "Smith College"
>>> college[0]
'S'
>>> college[4]
'h'
>>> college[0:4]
'Smit'
>>> # slicing is like range: include start but exclude end
```

- If start is omitted, start from the beginning (zeroth element), and if end is omitted, go until the very end

```
>>> college[:4]
'Smit'
>>>
>>> college[:5]
'Smith'
```

```
>>> college[5:8]
' Co'
>>> college[6:len(college)]
'College'
>>> college[6:]
'College'
```

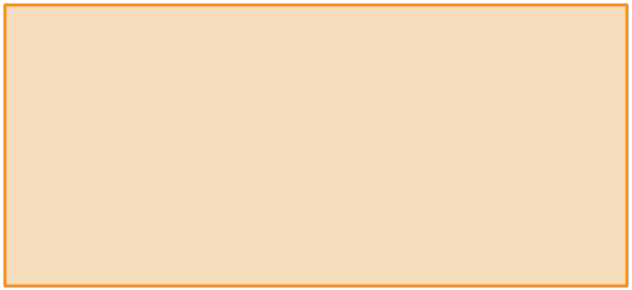
Exercise: months of the year

- Write a function to print the month abbreviation, given the month number

```
months = "JanFebMarAprMayJunJulAugSepOctNovDec"
```

```
>>> month_name(4)
The number 4 month is Apr
>>>
>>> month_name(1)
The number 1 month is Jan
>>>
>>> month_name(9)
The number 9 month is Sep
>>>
>>> month_name(2)
The number 2 month is Feb
```

```
# given a month number n (1,2,...,12), print the
# 3 letter month abbreviation
def month_name(n):
```



Why start numbering from zero?

Argument from influential computer scientist Dijkstra:

http://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD831.html