Week 4, Lecture 6: Lists
News: “Social Media Shaming Your College”


• Students are using apps to shame their schools into better coronavirus plans.

• We’ve all seen social media used to shame people we disagree with. Those milliseconds of tsk-tskying might feel good, but I doubt they’re helpful.

• College students are using TikTok, Twitter and other apps to embarrass their universities when they fail to care for people who have been isolated in special Covid-19 dorms or are in quarantine units because of a possible exposure.
**Announcements**

- This week:
  - Lists today, Lists and HOFs on Friday
  - Practice Problems: Optional for lab 2, required for lab 3.
  - Remember, they’re required but based on effort.
- Midterm Proctoring Guidelines are out today.
  - They’re a few pages, but hopefully not too much effort.
    - Can use: Terminal, PythonTutor, Docs, but no friends.
  - Requests from an exemption are due by Feb 22.
    - We will probably schedule a short oral exam.
Lists
Learning Objectives

- Lists are a new data type in Python.
- Lists can store any kind of data and be any length.
- We start counting items of lists at 0.
- Lists are *mutable*. We can change their data!
Lists

• A structure in Python that can hold many elements
  • Also referred to as an “array” in other programming languages.
• Lists are used to group similar items together.
  • A “contact list”, a “list of courses”, a “to do list”
• Python lists are really flexible!
  • Can contain any type of data
  • Can mix and match types!
  • Can add and delete items
Types We’ve Learned So Far

- Each *type* of data has a specific set of functions (methods) you can apply to them, and certain properties you can access.
- *ints*/Integers
  - 1, -1, 0, ...
- *floats* (“decimal numbers”)
  - 1.0, 3.14159, 20.0
- *strings*
  - ”Hello”, “CS88”
- *list*/Arrays
  - ['CS88', 'DATA8', 'POLSCI2', 'PHILR1B']
- *functions*
  - max(), min(), print(), your own functions!
List Operations

- [] ”square brackets”: Used to access items in a list. We start at 0!
- len(): The number of items in a list
- +: We can add lists together
- min(), max(): Functions that take in a list and return some info.
- Converting between types: Strings and Lists:
  - <string>.split(<separator>) → List of string
  - “I am taking CS88.”.split(‘ ‘)
  - <string>.join(<list>) → String, with the items of a list joined together.
    - ” “.join([“I”, “am”, “taking”, “CS88.”])
- Lots more interesting tools!
  - https://docs.python.org/3.7/tutorial/datastructures.html
Lists, For Loops, and Sequences
Learning Objectives: Using Lists in Practice

• **for** Loops are a "generic" way to iterate over data.
• **range** is a Python function that generates a sequence of numbers.
for statement – iteration control

• Repeat a block of statements for a structured sequence of variable bindings

<initialization statements>

for <variables> in <sequence expression>:
    <body statements>

<rest of the program>
while statement – iteration control

• Repeat a block of statements until a predicate expression is satisfied

<initialization statements>

while <predicate expression>:

<body statements>

<rest of the program>

# Equivalent to a for loop:
index = 0
while index < len(my_list)
    item = my_list[index]
    ...
    index += 1
range

- range() is a built in Python tool that generates a sequence of numbers.
  - It does not return a list unless we explicitly ask for one.
- It has many options: start, stop, and step.
- Range is lazy! It can be iterated over, but doesn’t compute all its values at once.
  - We’ll revisit this later.
- **GOTCHA:** Range is exclusive in the last value!
  - `range(10)` is a sequence on 10 numbers from 0 to 9.
  - [https://docs.python.org/3.7/library/stdtypes.html?highlight=range#range](https://docs.python.org/3.7/library/stdtypes.html?highlight=range#range)
List Comprehensions
Learning Objectives

• List comprehensions let us build lists "inline".
• List comprehensions are an expression that returns a list.
• We can easily “filter” the list using a conditional expression, i.e. if
Data-driven iteration

• describe an expression to perform on each item in a sequence
• let the data dictate the control
• In some ways, nothing more than a concise for loop.

[ <expr with loop var> for <loop var> in <sequence expr> ]

[ <expr with loop var> for <loop var> in <sequence expr> if <conditional expression with loop var> ]
Control Structures Review

• The result of list(range(0,10)) is…
  • A) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
  • B) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  • C) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  • D) [1, 2, 3, 4, 5, 6, 7, 8, 9]
  • E) an error

• http://bit.ly/88Lec3Q1

Solution:
A) list(range(m,n)) creates a list with elements from m to n-1.
iClicker Question

What is the value of `thing` after running:

- `thing = [ print(‘I like ’+ course) for course in courses ]`

- Nothing
- `[ “I like CS88”, “I like DATA8”, ... ]`
- `[]`
- `[ None, None, None, None ]`
- Error
Control Structures Review

The result of \texttt{len([i for i in range(1,10) if i \% 2 == 0])}

is…

A) 5
B) 4
C) 3
D) 2
E) 1

Solution:
B) \texttt{len([2, 4, 6, 8]) == 4}
>>> uni = 'The University of California at Berkeley'
>>> words = uni.split(' ')  
>>> thing = [ w[0] for w in words ]

A) []
B) ['The', 'University', 'of', 'California', 'at', 'Berkeley']
C) 'TUoCaB'
D) ['T', 'U', 'o', 'C', 'a', 'B']
E) Error

Solution:
D)
Control Structures Review

• The result of `[i for i in range(3,9) if i % 2 == 1]` is...

A) [3, 4, 5, 6, 7, 8, 9]  
B) [3, 4, 5, 6, 7, 8]  
C) [1, 3, 5, 7, 9]  
D) [3, 5, 7, 9]  
E) [3, 5, 7]


Solution:  
E) [3, 5, 7]