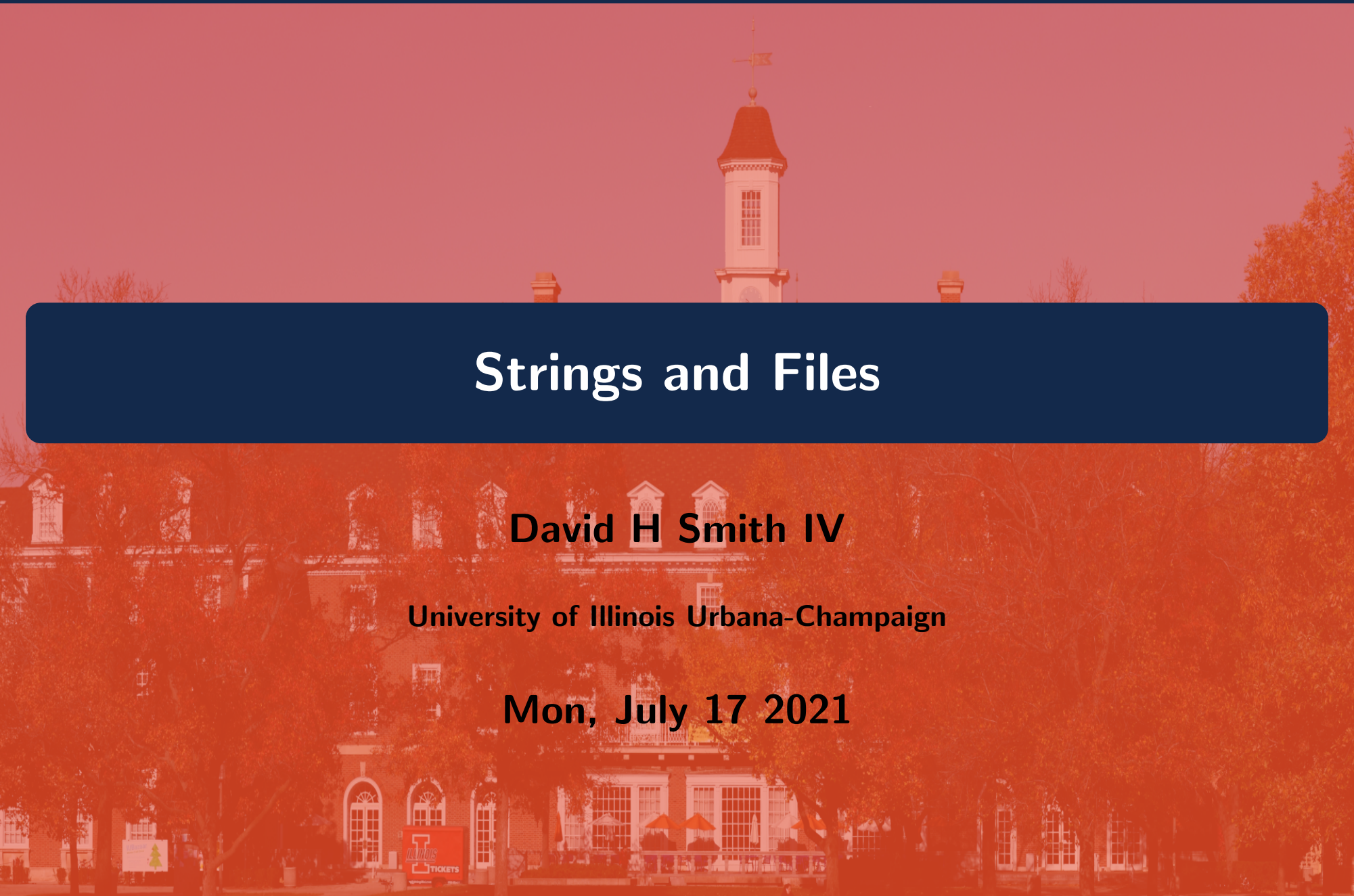

Strings and Files

David H Smith IV

University of Illinois Urbana-Champaign

Mon, July 17 2021

Course Overview

Reminders

- Signup for the quiz or submit a conflict request
- Homework 6 is up
- Practice quiz is up
- zyBooks and post reading are due tomorrow

The Internet

Poll Question: Acronyms

When you request a web page your request is part of what protocol?

A. HTML
B. DNS
C. CSS
D. HTTP
E. JS
F. WWW

Poll Question: Acronyms

When you request a web page your request is part of what protocol?

A. **HTML:** Hyper Text Markup Language
B. **DNS:** Domain Name Service
C. **CSS:** Cascading Style Sheets
D. **HTTP:** Hyper Text Transfer Protocol
E. **JS:** Javascript
F. **WWW:** World Wide Web

Poll Question:

A. Tags in HTML documents typically come in pairs
B. HTML documents are hierarchical
C. CSS properties consists of attribute: value pairs
D. The preferred method of styling web pages is through CSS
E. Javascript can be used to modify both the content of a web page and its presentation
F. All of the statements are true

Patterns

Counting Pattern

```
1 def count(collection):
2   counter = 0
3   for item in collection:
4     if <item meets condition>:
5       counter += 1
6   return counter
```

Computing a Sum/Total

```
1 def sum(collection):
2   total = 0
3   for item in collection:
4     total += item
5   return total
```

Finding (single thing) in a Collection

```
def find_thing(collection):
  for thing in collection:
    if <thing meets condition>:
      return thing
```

```
def find_thing(collection):
  found = None
  for thing in collection:
    if <thing meets condition>:
      found = thing
      break
  return found
```

Using Loop else when nothing found

```
def find_thing(collection):
  for thing in collection:
    if <thing meets condition>:
      found = thing
      break
  else:
    found = <something>

 return found
```

Finding best in collection

```
def find_best(collection):
  currentbest = ??
  for thing in collection:
    if <thing is better than current best>:
      currentbest = thing
  return currentbest
```

- If we're searching over a list and we want to return the largest or smaller number: `currentbest = stufflist[0]`

Finding best in collection

```
def find_best(collection):
  currentbest = ??
  for thing in collection:
    if <thing is better than current best>:
      currentbest = thing
  return currentbest
```

- If we're searching over a list and we want to return the largest or smaller number: `currentbest = stufflist[0]`
- If we're searching over a list of strings and we want to return the longest string: `currentbest = stufflist[0]` or `currentbest = ""`

Finding best in collection

```
def find_best(collection):
  currentbest = ??
  for thing in collection:
    if <thing is better than current best>:
      currentbest = thing
  return currentbest
```

- If we're searching over a list and we want to return the largest or smaller number: `currentbest = stufflist[0]`
- If we're searching over a list of strings and we want to return the longest string: `currentbest = stufflist[0]` or `currentbest = ""`
- If you know the list contains only non-negative integers: `currentbest = -1`

Filtering a collection

```
1 def filter(collection):
2   new_list = []
3
4   for thing in collection:
5     if <thing meets criteria>:
6       newlist.append(thing)
7
8   return new_list
```

Poll Question: Patterns

Given a list of names, make a new list containing only those enrolled in a given course.

A. Sum
B. Counter
C. Finding "best" in collection
D. Filtering a collection
E. None of the above

```
def find_student_enrolled_in_course(student_registrations,
    course):
    students = []
    for student in student_registrations:
        if course in student_registrations[student]:
            students.append(student)
    return students

```

Poll Question: Patterns

Given a list of strings find the longest string.

A. Sum
B. Counter
C. Finding "best" in collection
D. Filtering a collection
E. None of the above

```
1  def find_longest_string(strings):
2      longest_string = strings[0]
3      for string in strings[1:]:
4          if len(string) > len(longest_string):
5              longest_string = string
6      return longest_string
7
```

Poll Question: Patterns

Given a list of strings, find the number of strings that contain the substring —"tion"—.

A. Sum
B. Counter
C. Finding "best" in collection
D. Filtering a collection
E. None of the above

```
1     def count_substring(strings):
2       counter = 0
3       for string in strings:
4         if "tion" in strings:
5           counter += 1
6       return counter
7
```