CS 111: Program Design I

Lecture 21: Midterm postmortem, HTML Law, Start Networks

Robert H. Sloan & Richard Warner
University of Illinois at Chicago
November 12, 2019

UIC

MIDTERM POSTMORTEM

Based on grading to date

- There will be an extremely wide range of grades
- Significant minority of class had difficulty writing the functions

- And Crawler deadline extension from Friday night to Monday night

docstrings

- Unique to Python among popular languages (and all languages that Prof. Sloan knows)
- Distinct from Python comments; (so far) CS 111 has *never* asked you to write comments (have asked for docstrings!)
- Docstrings *not* very similar to comments in other languages
  - So if you know Java or something, don't try to use docstring like comments in that language
- Exactly one per function, must be immediately *after* def line

```
def an_example_function(a_formal_parameter):
    " " "This is only place for docstring!" " "
```

Functions: Most important analysis

- First question to ask when you have to design function:
  - What is *number of input parameters and type of each,* and
  - *Is there a return value?* And if so,
  - *What type is it?*

For loop vs. while loop

- When you need to count and/or use a sequence of integers
  - *Almost always easier to use for loop rather than while loop*
  - Easier =
    - Less likely to make mistakes
    - Easier for humans to read
    - Easier for *you* to read

What is a for loop over (most of the time)

- for item in a_list
- for ch in a_string
- for i in range(n)
- for i in range(start, end)
- for i in range(start, end, skip)
- Much less common: for i in range(len(list_or_string))

- WRONG: for i in len(anything) # No, no no, please no!

Accumulator pattern

- Occurs again and again

```
<sometimes need set-up work here first>
answer_holder = <appropriate start/nil value>
for i in <something, often range statement>:
    answer_holder = adjustment for i case
return answer_holder
```

Midterm programming solutions live in Spyder

- Starting with factorial, as example of accumulator pattern

Remark: Can *always* convert numerical for to while

- for is there to make life better, not because we have to have it

Generic conversion

```
answer_h = <start/nil value>
for i in range(a, b, s):
    answer_h = update for i
return answer_h
```

```
answer_h = <start/nil value>
i = a
while i < b:
    answer_h = update for i
    i += s
return answer_h
```

And factorial conversion at Spyder

WEB CRAWLER

def crawl(start, limit):

```
    to_visit = [start]
    visited = [ ]
    while to_visit:
        address = to_visit.pop()
        if address not in visited:
            content = text @ URL address
    #(earlier)
            do_the_visit(page, emails, etc.)
            visited.append(address)
            if len(visited) >= limit:
                break
```

Crawl and *Scrape!*

- Crawlers crawl for a purpose
- Our assignment: Grab email addresses
- Could just as easily grab all .jpg or all .mp3 or all … files
- Or, a search engine:
  - Build a dictionary showing which words/phrases show up on which web pages
- Or ...

A LITTLE ABOUT WHAT'S ON A WEBPAGE: HTML

HTML is a markup language

- That has evolved
- Was simple c. 1996, and pretty simple c. 2005
  - Now, people want to control look-and-feel of page down to pixels and fonts.
  - Plus, we want to grab information more easily out of Web pages.
    - Leading to XML, the eXtensible Markup Language.
    - XML allows for new kinds of markup languages (that, say, explicitly identify prices or stock ticker codes) for business purposes.

Four kinds of HTML languages

1. Original HTML: Simple, what the earliest browsers understood.
2. CSS, Cascading Style Sheets
   - Ways of defining more of the formatting instructions than HTML allowed.
- (3a) XHTML: HTML re-defined in terms of XML.
  - A little more complicated to use, but more standardized, more flexible, more powerful.
  - Never 100% caught on
- (3b) HTML 5: Today's standard; subsumes XHTML
  - Many new syntactic elements for multimedia and graphical content

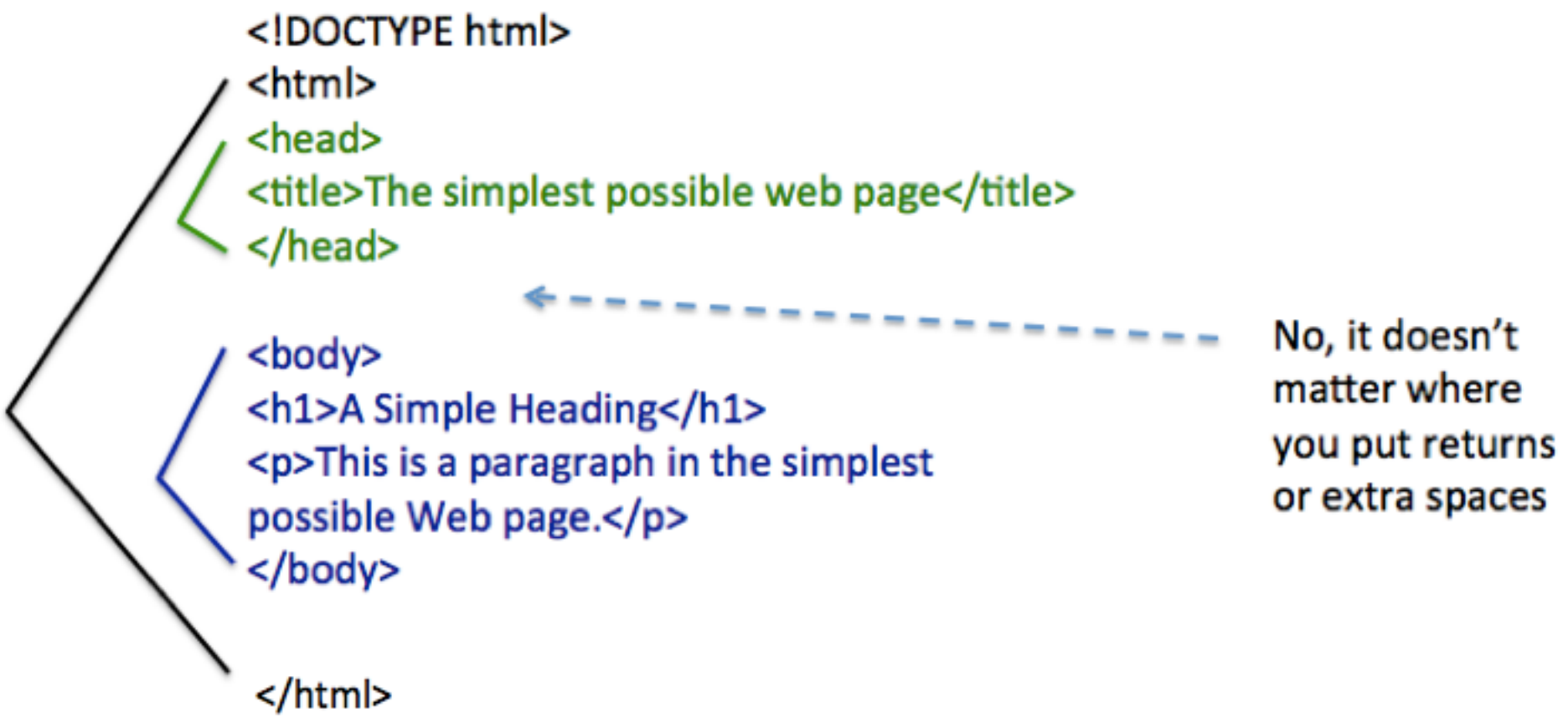
Markup means adding tags

- A *markup language* adds *tags* to regular text to identify its parts
- Tag in HTML enclosed by <angle brackets>
- Most tags have starting tag and ending tag
  - A paragraph is identified by a at its start and a at its end
  - A heading is identified by a <h1> at its start and a </h1> at its end

HTML is just text in a file

- Enter text and tags in just plain ole ordinary text file.
- Use extension “.html” (“.htm” if your computer only allows three characters) to indicate HTML.
- Any *text or code* editor (e.g., WordPad, TextEdit, VS Code) works just fine for editing and saving HTML files.

Parts of a Web Page



Parts of a Web Page

- Start with a DOCTYPE
  - It tells browsers what kind of language you're using below.
- Whole document is enclosed in <html> </html> tags.
  - The heading is enclosed with <head> </head>
    - That's where you put the <title> </title>
  - The body is enclosed with <body> </body>
    - That's where you put <h1> headings and <p> paragraphs.

First page of crawler sample site

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>     <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>crawlerstart</title>
  </head>
  <body>     <h1>Welcome to the CS 111 Webcrawler Start Page!</h1>
    <p><br>    </p>
    Here we have the email addresses of some of our favorite students:<br>
        OMMITTED TEXT TO MAKE IT FIT ON ONE SLIDE
    <p>And we have links to some other pages such as
    <a href="https://www.cs.uic.edu/%7Esloan/CS111Law/Page42.html">this one</a> that is linked from
    <a href="https://www.cs.uic.edu/%7Esloan/CS111Law/crawlerstart.html">right here</a><br>    </p>
    <p>And <a href="https://www.cs.uic.edu/%7Esloan/CS111Law/page7.html">this other one</a> too!<br>
         </p>
  </body>
</html>
```

Other things in HTML

- We're simplifying these tags a bit.
- More can go in the <head>
  - uic.edu header (checked early November 2018) had 327 lines in head, of which about 325 are a combination of:
    - Javascript (tons of it)
    - References to documents like cascading style sheets
- The <body> tag can also, e.g., set colors.
  - <body bgcolor="#ffffff" text="#000000" link="#3300cc" alink="#cc0033" vlink="#550088">

HTML is not a programming language

- Using HTML is called “coding” and it is about getting your codes right.
- But it's not about coding programs.
- HTML has no
  - Loops
  - IFs
  - Variables
  - Data types
  - Ability to read and write files
- Bottom line: HTML does not communicate process!