Due Nov 27 at 1pm **Points** 5 **Questions** 5 **Available** until Dec 4 at 11:59pm **Time Limit** None **Allowed Attempts** Unlimited

Instructions

Prior to completing this quiz, be sure to read:

• Section 11.1: The World Wide Web (p. 372-379)

Note that this quiz will not be Python-based. You will practice identifying basic HTML elements, which will help with the later sections as we work towards web crawling.

This quiz was created for learning purposes. You may attempt this quiz as many times as you would like. The highest score *prior to the deadline* will count towards the final course grade. No late submissions will be accepted.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	3 minutes	5 out of 5
LATEST	Attempt 2	3 minutes	5 out of 5
	Attempt 1	4 minutes	3.42 out of 5

Score for this attempt: **5** out of 5 Submitted Nov 26 at 12:17pm This attempt took 3 minutes.

Question 1 1 / 1 pts

Resources on the web must each have a unique identifier. The identifier is called a Uniform Resource Locator (URL). In general, there are three parts of a URL:

- scheme how to access the resource
- host name of the server hosting the document
- · path relative pathname relative to the server's root directory

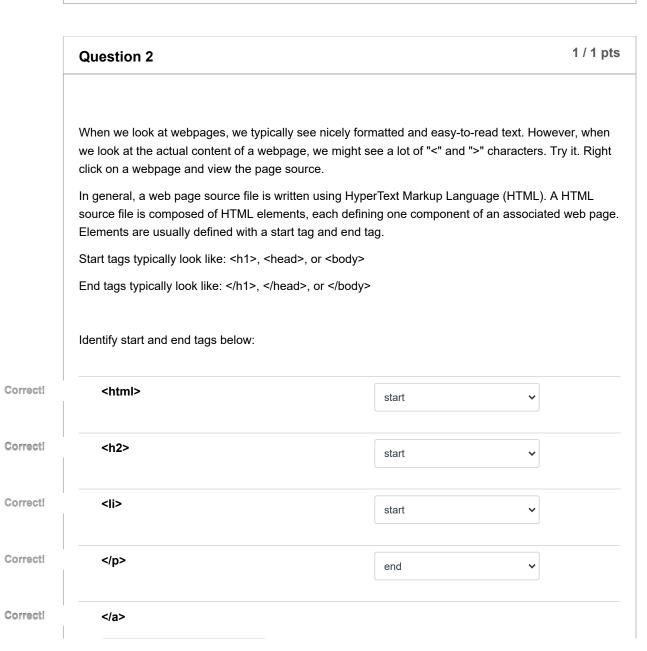
Consider the following URL:

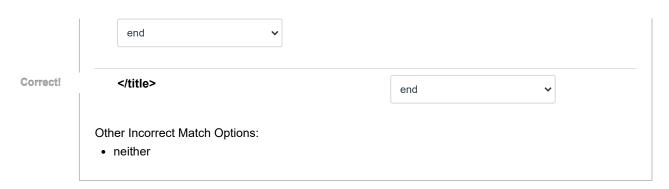
https://my.uclaextension.edu/courses/24904

https is the scheme. Other possible schemes are http, ftp, mailto, and file

my.uclaextension.edu is the host. Other hosts include www.google.com, <u>(http://www.google.com,)</u> www.python.org, and www.jetbrains.com

courses/24904 is the path. This is us	sually the extra text you see after the host.			
Consider the following link to the cou	rse e-book:			
https://www.oreilly.com/library/view/introduction-to-computing/9781118213568/				
Identify the scheme, host, and path.				
https	scheme			
www.oreilly.com	host			
library/view/introduction-to- computing/9781118213568/	path 🕶			





Question 3	1 / 1 pt
Start and end tags in HTML work similarly to Python. The end tag shoo ag first.	uld close the inner-most start
Notice that in the following pseudo-HTM, we close the inner tag2 befor	re closing the outer tag1:
<tag1> <tag2> This text is read on the webpage </tag2> </tag1>	
This next code is also okay:	
<tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat not tag2</tag3> </tag1>	s properties of tag1 and tag3 bu
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3</tag2></tag1></pre>	
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text ha t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3</tag2></tag1></pre> The nesting could go as far and deep as needed, but the inner-most tags	
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text ha t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3</tag2></tag1></pre> The nesting could go as far and deep as needed, but the inner-most tags	
<tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3 The nesting could go as far and deep as needed, but the inner-most tage outer ones are closed: <tag1> <tag2> <tag3> <tag4> TEXT</tag4> </tag3> </tag2> </tag1> Mark all of the tags that will work for HTML given that inner-most tags</tag2></tag1>	ag must be closed first before
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3</tag2></tag1></pre> The nesting could go as far and deep as needed, but the inner-most tag buter ones are closed: <tag1><tag2><tag3><tag4>TEXT</tag4></tag3></tag2></tag1> Mark all of the tags that will work for HTML given that inner-most tags	ag must be closed first before
<tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3 The nesting could go as far and deep as needed, but the inner-most tags outer ones are closed: <tag1> <tag2> <tag3> <tag4> TEXT</tag4> </tag3> </tag2> </tag1> Wark all of the tags that will work for HTML given that inner-most tags ones are closed.</tag2></tag1>	ag must be closed first before
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3 The nesting could go as far and deep as needed, but the inner-most tag text ones are closed: <tag1> <tag2> <tag3> <tag4> TEXT</tag4> </tag3> </tag2> </tag1> Mark all of the tags that will work for HTML given that inner-most tags ones are closed. <tag1> <tag1> <tag2> <tag3> </tag3></tag2> </tag1> </tag1></tag2></tag1></pre>	ag must be closed first before
<pre><tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text hat t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3 The nesting could go as far and deep as needed, but the inner-most tag outer ones are closed: <tag1><tag2><tag3><tag4>TEXT</tag4></tag3></tag2></tag1> Mark all of the tags that will work for HTML given that inner-most tags ones are closed. <tag1><tag1><tag2><tag3></tag3></tag2></tag1></tag1></tag2><td>ag must be closed first before</td></tag1></pre>	ag must be closed first before
The nesting could go as far and deep as needed, but the inner-most tabuter ones are closed: <tag1><tag2><tag3><tag4>TEXT</tag4></tag3></tag2></tag1> Wark all of the tags that will work for HTML given that inner-most tags ones are closed. <tag1><tag1><tag2><tag3></tag3></tag2></tag1> <tag1><tag1><tag2><tag3></tag3></tag2></tag1> <tag1><tag1><tag2><tag3></tag3></tag2> <tag1><tag1><tag2><tag3></tag3></tag2>><td>ag must be closed first before</td></tag1></tag1></tag1></tag1></tag1></tag1>	ag must be closed first before

Correct!

Correct!

Correct!

The HTML anchor element (a) callook like:	n be used to create hyperlinked text. An example of this usage
<pre><a href="https://my.uclaextension.e</pre></td><td>du/">UCLA Extension Canvas</pre>	
In the above example, we made of example, href is an <i>attribute</i> of the	ur text "UCLA Extension Canvas" a hyperlinked text. In this e anchor element.
Suppose we would like to create a appropriate HTML:	a hyperlinked text to google.com. Fill in the blanks to get the
<a href="https://w</td><td>ww.google.com/">Google Homepage< /a> >	
Answer 1:	
href	
Answer 2:	
Answer 2:	
	1 /
/a	1 /
/a Question 5	1 / g of a webpage, you might come across an anchor element wi e href attribute. For example:
/a Question 5 When looking at the source coding	g of a webpage, you might come across an anchor element wi e href attribute. For example:
Vhen looking at the source coding incomplete URL as the value to the called a rescheme and host with the contain	g of a webpage, you might come across an anchor element with e href attribute. For example: Inse Homepage Plative URL. This link will still work because we can fill in the ming document's scheme and host. So, if you found this relative te, then the scheme will automatically be https and the host wi

relative

relative

Correct!

Correct!

/library/view/introduction-to-computing/9781118213568/

/courses/24904/modules

Correct!	https://my.uclaextension.edu/	absolute	•	
Correct!	https://en.wikipedia.org/wiki/Python_(pi	absolute	•	

Quiz Score: 5 out of 5