Learning Quiz 31: The World Wide Web

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, (http://www.google.com,) www.python.org, and www.jetbrains.com

courses/24904 is the path. This is usually the extra text you see after the host.

Consider the following link to the course e-book:

```
https://www.oreilly.com/library/view/introduction-to-computing/9781118213568/
```

Identify the scheme, host, and path.

Correct!	**https**	scheme
Correct!	**www.oreilly.com**	host
Correct!	**library/view/introduction-to-computing/9781118213568/**	path

Question 2

1 / 1 pts

When we look at webpages, we typically see nicely formatted and easy-to-read text. However, when we look at the actual content of a webpage, we might see a lot of "<" and ">" characters. Try it. Right click on a webpage and view the page source.

In general, a web page source file is written using HyperText Markup Language (HTML). A HTML source file is composed of HTML elements, each defining one component of an associated web page. Elements are usually defined with a start tag and end tag.

Start tags typically look like: <h1>, <head>, or <body>

End tags typically look like: </h1>, </head>, or </body>

Identify start and end tags below:

Correct!	**<html>**	start
Correct!	**<h2>**	start
Correct!	** **	start
Correct!	**</p>**	end
Correct!	****	

end

Correct! **</title>** end

Other Incorrect Match Options:

- neither

Question 3

1 / 1 pts

Start and end tags in HTML work similarly to Python. The end tag should close the inner-most start tag first.

Notice that in the following pseudo-HTM, we close the inner tag2 before 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 has properties of tag1 and tag3 bu
t not tag2</tag3> </tag1>
```

Here is another example:

```
<tag1> <tag2> This text is read on the webpage </tag2> <tag3> This text has properties of tag1 and tag3 bu
t not tag2</tag3> <tag2> This text has tag2 properties again, but not tag3</tag2></tag1>
```

The nesting could go as far and deep as needed, but the inner-most tag must be closed first before 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 must be closed before outer ones are closed.

- [ ] <tag1><tag2><tag3></tag2></tag3></tag1>
- [ ] <tag1><tag2><tag3></tag3></tag2></tag3>
- [ ] <tag1><tag2><tag3></tag1></tag2></tag3>
- [x] Correct! <tag1><tag2></tag2><tag3></tag3></tag1>
- [x] Correct! <tag1><tag2><tag3></tag3></tag2></tag1>
- [x] Correct! <tag1></tag1><tag2></tag2><tag3></tag3>

Question 4

1 / 1 pts

The HTML anchor element (a) can be used to create hyperlinked text. An example of this usage might look like:

```
<a href="https://my.uclaextension.edu/">UCLA Extension Canvas</a>
```

In the above example, we made our text "UCLA Extension Canvas" a hyperlinked text. In this example, **href** is an *attribute* of the anchor element.

Suppose we would like to create a hyperlinked text to google.com. Fill in the blanks to get the appropriate HTML:

<a [href] ="https://www.google.com/">Google Homepage< [/a] >

Answer 1:

Correct!

href

Answer 2:

Correct!

/a

Question 5

1 / 1 pts

When looking at the source coding of a webpage, you might come across an anchor element with an incomplete URL as the value to the href attribute. For example:

```
<a href="/courses/24904">Python Course Homepage</a>
```

This incomplete URL is called a relative URL. This link will still work because we can fill in the missing scheme and host with the containing document's scheme and host. So, if you found this *relative* URL on UCLA Extension's Canvas page, then the scheme will automatically be https and the host will automatically be my.uclaextension.edu.

Identify the relative and absolute URLs.

Correct!

/courses/24904/modules — relative

Correct!

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

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

Quiz Score: **5** out of 5