Practice with web scraping

# Practice with CSS Selectors

|  |  |
| --- | --- |
| HTML File | Questions |
| **<html>**  **<head>**  **<title>this is the title</title>**  **</head>**  **<body>**  **<h1>one</h1>**  **<div class="a">**  **<h2>three</h2>**  **<p class="c">four</p>**  **<p class="d">five</p>**  **</div>**  **<div class="b">**  **<h2>six</h2>**  **<p class="d">seven</p>**  **<p class="c">eight</p>**  **</div>**  **<h3>nine</h3>**  **<div class="c">**  **<p>ten</p>**  **<p>eleven</p>**  **<p>twelve</p>**  **<p>thirteen</p>**  **<p>fourteen</p>**  **<p>fifteen</p>**  **<p>sixteen</p>**  **<p>seventeen</p>**  **</div>**  **<p id="x">eighteen</p>**  **</body>**  **</html>** | 1. **The text in the HTML page to the left consists of numbers in words (ie. one, two, three, etc). For each question below, write a CSS selector that will target the specific words mentioned in the question. Try to keep the selector as short as possible. The question will be marked wrong if the selector is unnecessarily lengthy.**     1. **What CSS selector targets just the following words:  eighteen**    2. **What CSS selector targets just the following words:  three six ten**    3. **What CSS selector targets just the following words:  four seven**    4. **What CSS selector targets just the following words: thirteen fifteen seventeen**    5. **What CSS selector targets just the following words:  six seven eight** 2. **Save the HTML code in a file and display it in your browser. Modify the code so that the odd numbered words appear in green text and the even numbered words appear in red text. You may add any HTML or css code that you think is appropriate. However, do not modify the overall layout of the page (e.g. all headings should remain headings, all paragraphs should remain paragraphs, etc.)** 3. **Does WikiPedia officially disallow you from scraping the page** [**https://en.wikipedia.org/wiki/United\_States**](https://en.wikipedia.org/wiki/United_States) **from your R program? How would you know?** |

# Practice with web scraping

The following questions deal with scraping data from the website <https://www.mishnahyomit.com/> that contains audio lectures. The lectures are arranged by date according to the "mishnah yomit" schedule (see here for more info: <https://en.wikipedia.org/wiki/Mishnah_Yomis>). Here is a picture of a sample page from that website:

|  |
| --- |
| https://www.mishnahyomit.com/shiurim/Demai |
| A screenshot of a computer  Description automatically generated |

1. Write the function: getLectureData = function( url)   
   The url argument should be passed a url similar to the following:  
     
    <https://www.mishnahyomit.com/shiurim/Berachot>  
    <https://www.mishnahyomit.com/shiurim/Peah>  
    <https://www.mishnahyomit.com/shiurim/Demai>  
    etc.  
     
   The function should scrape the lecture data from the specified page. The function should continue to work even if additional lectures are added to the same page.  
     
   For example:   
     
   > lectures = getLectureData("<https://www.mishnahyomit.com/shiurim/Demai>")  
   > head(lectures, 3)  
    **lectureName length secularDate hebrewDate**1 Demai 1:1-2 8:02 Tuesday, November 23rd 2027 23 Cheshvan 5788  
   2 Demai 1:3-4 10:53 Wednesday, November 24th 2027 24 Cheshvan 5788  
   3 Demai 2:1-2 4:14 Thursday, November 25th 2027 25 Cheshvan 5788
2. In the above mentioned webpages the lectureName is a link that brings you to a different page on the website that contains the actual audio lectures. The URLs to the pages with the actual lectures are not directly visible - however when you click on the lectureName your browser is redirected to the appropriate page. Write a function -   
     
   getLectureURLs = function(url)  
     
   The url argument should be passed the same types of URLs as in the previous question.  
   The function getLectureURLs should return a dataframe with two columns. Column 1 contains the lectureName and column 2 contains the URL that the browser is redirected to when the user clicks on that lecture name. For example:  
     
   > df = getLectureURLs("<https://www.mishnahyomit.com/shiurim/Demai>")  
   > head(df, 3)  
    **lectureName lectureURL**1 Demai 1:1-2 https://www.mishnahyomit.com/shiurim/listen/Demai%201%3A1-2  
   2 Demai 1:3-4 https://www.mishnahyomit.com/shiurim/listen/Demai%201%3A3-4  
   3 Demai 2:1-2 <https://www.mishnahyomit.com/shiurim/listen/Demai%202%3A1-2>  
     
   HINT - use the rvest::html\_attr() or rvest::html\_attrs() functions as part of your answer.
3. The secular date from the data on these pages is all in one column. Write the following function that splits the data for the secular date into different columns.   
     
    splitSecularData = function(df)  
     
   The argument df should be a dataframe that was returned from function, getLectureData, described in question #1 above. For example, assuming that the getLectureData function from question #1 exists:  
     
   > lectures = getLectureData("<https://www.mishnahyomit.com/shiurim/Demai>")  
   > splitDates = splitSecularDates( lectures )  
   > head(splitDates, 3)  
    **lectureName length weekday month day year hebrewDate**1 Demai 1:1-2 8:02 Tuesday November 23 2027 23 Cheshvan 5788  
   2 Demai 1:3-4 10:53 Wednesday November 24 2027 24 Cheshvan 5788  
   3 Demai 2:1-2 4:14 Thursday November 25 2027 25 Cheshvan 5788
4. Question 1 above asked you to write a function to scrape the data from the following types of pages:  
     
    <https://www.mishnahyomit.com/shiurim/Berachot>  
    <https://www.mishnahyomit.com/shiurim/Peah>  
    <https://www.mishnahyomit.com/shiurim/Demai>  
    etc.  
     
   The end of the URLs above, i.e. "Berachot" , "Peah", "Demai", etc. are names of "tractates". Each of the above pages highlight different lectures for one specific tractate. The names of the tractates are also listed on the above webpages. Write a function - getTractateURLs. The function should return a dataframe that contains the URLs for each tractate. Theoreticaly, output of this function could be used to scrape each of the tractate pages in a loop (you don't have to do that part).   
     
   For example:  
     
   > getTractateURLs = function("<https://www.mishnahyomit.com/shiurim/Berachot>")  
     
    **tractate URL**1 Berachot <https://www.mishnahyomit.com/shiurim/Berachot>  
   2 Peah <https://www.mishnahyomit.com/shiurim/Peah>  
   3 Demai <https://www.mishnahyomit.com/shiurim/Demai>   
   4 Kilayim <https://www.mishnahyomit.com/shiurim/Kilayim>  
   etc ....  
   62 Yadayim <https://www.mishnahyomit.com/shiurim/Yadayim>

63 Uktzin <https://www.mishnahyomit.com/shiurim/Uktzin>