Python Data Scrapping MCQs

Q-1: What protocol can be used to retrieve web pages using python?

Top of Form

A. urllib  
B. bs4  
C. HTTP  
D. GET

Bottom of Form

Q-2: What provides two way communication between two different programs in a network.

Top of Form

A. socket  
B. port  
C. http  
D. protocol

Bottom of Form

Q-3: What is a python library that can be used to send and receive data over HTTP?

Top of Form

A. http  
B. urllib  
C. port  
D. header

Bottom of Form

Q-4: What is the process by which search engines retrieve webpages and build a search index called?

Top of Form

A. scrape  
B. parse  
C. BeautifulSoup  
D. spider

Bottom of Form

Q-5: What does the following block of code do?

**import** **urllib.request**

fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')

**for** line **in** fhand:

print(line.decode().strip())

Top of Form

A. It creates a file named 'romeo.txt' in 'data.pr4e.org'  
B. It finds the urls linked to 'data.pr4e.org' and prints it.  
C. It opens a file named 'http://data.pr4e.org/romeo.txt' in local storage  
D. It prints the contents of 'romeo.txt' after retrieving it from 'data.pr4e.org'

Bottom of Form

Q-6: What does the following block of code do?

**import** **urllib.request**, **urllib.parse**, **urllib.error**

img = urllib.request.urlopen('http://data.pr4e.org/cover3.jpg').read()

fhand = open('cover3.jpg', 'wb')

fhand.write(img)

fhand.close()

Top of Form

A. It retrieves 'cover3.jpg' and saves it to your computer.  
B. It displays the image 'cover3.jpg'.  
C. It retrieves the url to download 'cover3.jpg'

Bottom of Form

Bottom of Form

Q-7: What does the following block of code do?

url = "https://www.nytimes.com"

html = urllib.request.urlopen(url, context=ctx).read()

soup = BeautifulSoup(html, 'html.parser')

Top of Form

A. retrieves and displays the webpage  
B. parses the html content of the "https://www.nytimes.com" webpage.  
C. downloads the webpage

Bottom of Form

Q-8: What does the following block of code print?

url = "https://www.nytimes.com/"

html = urllib.request.urlopen(url).read()

soup = BeautifulSoup(html, 'html.parser')

tags = soup('img')

**for** tag **in** tags:

print(tag.get('src', **None**))

Top of Form

A. retrieves and displays the webpage  
B. downloads the webpage  
C. prints the images from 'www.nytimes.com'  
D. prints all the 'img' sources under 'src' from 'www.nytimes.com'  
Check MeCompare me

Bottom of Form

Q-9: Which tag is the parent of the a tag in the above structure?

Top of Form

A. h1

B. p

C. body

D. html

Bottom of Form

Q-10: Which tag is the parent of the title tag in the above structure?

Top of Form

A. h1

B. body

C. head

D. html

Bottom of Form

Q-11: \_\_\_\_\_\_\_\_\_\_\_\_\_ is when we write a program that pretends to be a web browser and retrieves pages, then examines the data in those pages looking for patterns.

Top of Form

A. HTML parsing  
B. Crawling  
C. Search engine  
D. Web scraping

Bottom of Form

Q-12: Given the below html, how would this tag type be described in web scraping code?

<h1 class='sports'>Sports News</h1>

Top of Form

A. h1  
B. h1, class='sports'  
C. h1, class\_='sports'  
D. 'h1', class\_='sports'

Bottom of Form

Q-13: Which line of code correctly gets the first item in items and makes the most sense following the below code snippet?

soup = BeautifulSoup(response.content, 'html.parser')

items = soup.find\_all(class\_='items')

Top of Form

A. first\_item = items[0]  
B. first\_item = items.find(0)  
C. first\_item = items.get(0)  
D. first\_item = items.find[0]  
E. first\_item = soup.items[0]

Bottom of Form

Q-14: How does one parse the HTML into a BeautifulSoup object given a response object?

Top of Form

A. soup = BeautifulSoup(response.text, 'html.parser')  
B. soup = BeautifulSoup(response.content, 'html.parser')  
C. soup = BeautifulSoup(response.string, 'html.parser')

Bottom of Form

Q-15: Which of the following is the best way to get the value for the id in the first p tag?

Top of Form

A. soup.p.get('id')  
B. soup.p.get('id', None)  
C. soup.p[id]  
D. soup.p['id']

Bottom of Form

Q-16: How does one get the first header 1 tag after creating a soup object?

Top of Form

A. soup.h1  
B. soup.header1  
C. soup.h1[0]  
D. soup.h1[1]

Bottom of Form

Q-17: Which of the following gets the first link tag and returns a dictionary of all attributes and values for that link tag?

Top of Form

A. soup.a.attributes  
B. soup.link.attrs  
C. soup.a.attrs  
D. soup.link.attributes

Bottom of Form

Q-18: Which of the following finds all link tags?

Top of Form

A. all\_links = soup.find('a')  
B. all\_links = soup.findall('a')  
C. all\_links = soup.findall('link')  
D. all\_links = soup.find\_all('a')  
E. all\_links = soup.find\_all('link'

Bottom of Form

Q-19: Which of the following finds all paragraph tags with class b-soup?

Top of Form

A. all\_links = soup.find\_all('p', class='b-soup')  
B. all\_links = soup.find\_all('paragraph', class='b-soup')  
C. all\_links = soup.find\_all('p', class\_='b-soup')  
D. all\_links = soup.find\_all('paragraph', class\_='b-soup')

Bottom of Form

Q-20: After creating an empty dictionary and getting a list of all link tags, how does one put the link\_tag text as keys and the link\_tag href attribute as values for the dictionary?

Top of Form

A. loop through the elements of the list and do dictionary[link\_tag.text] = a.get('href', None)  
B. loop through the elements of the list and do dictionary[link\_tag.text] = a['href']  
C. loop through the elements of the list and do dictionary[link\_tag.text] = link\_tag.get('href', None)  
D. loop through the elements of the list and do dictionary[link\_tag.text] = link\_tag[href]

Bottom of Form

Q-21: Given the below html, after importing re, what will be returned after for tag insoup.find\_all(re.compile("t")): print(tag.name) is run?

<html>

<head>

<title>Site</title>

</head>

<body>

<p>There **is** lots of content.</p>

</body>

</html>

Top of Form

A. html  
B. title  
C. Site  
D. There is lots of content.

Bottom of Form

Q-22: What does the following block of code do?

url = "https://www.nytimes.com"

html = urllib.request.urlopen(url, context=ctx).read()

soup = BeautifulSoup(html, 'html.parser')

Top of Form

A. retrieves and displays the webpage  
B. parses the html content of the "https://www.nytimes.com" webpage.  
C. downloads the webpage

Bottom of Form

Q-23: What does the following block of code print?

url = "https://www.nytimes.com/"

html = urllib.request.urlopen(url).read()

soup = BeautifulSoup(html, 'html.parser')

tags = soup('img')

**for** tag **in** tags:

print(tag.get('src', **None**))

Top of Form

A. retrieves and displays the webpage  
B. downloads the webpage  
C. prints the images from 'www.nytimes.com'  
D. prints all the 'img' sources under 'src' from 'www.nytimes.com'

Bottom of Form

Q-24: Given the below html, how would this tag type be described in web scraping code?

<h1 class='sports'>Sports News</h1>

Top of Form

A. h1  
B. h1, class='sports'  
C. h1, class\_='sports'  
D. 'h1', class\_='sports'

Bottom of Form

Q-26: Which line of code correctly gets the first item in items and makes the most sense following the below code snippet?

soup = BeautifulSoup(response.content, 'html.parser')

items = soup.find\_all(class\_='items')

Top of Form

A. first\_item = items[0]  
B. first\_item = items.find(0)  
C. first\_item = items.get(0)  
D. first\_item = items.find[0]  
E. first\_item = soup.items[0]

Bottom of Form

Q-27: How does one parse the HTML into a BeautifulSoup object given a response object?

Top of Form

A. soup = BeautifulSoup(response.text, 'html.parser')  
B. soup = BeautifulSoup(response.content, 'html.parser')  
C. soup = BeautifulSoup(response.string, 'html.parser')

Bottom of Form

Q-28: Which of the following is the best way to get the value for the id in the first p tag?

Top of Form

A. soup.p.get('id')  
B. soup.p.get('id', None)  
C. soup.p[id]  
D. soup.p['id']

Bottom of Form

Q-29: How does one get the first header 1 tag after creating a soup object?

Top of Form

A. soup.h1  
B. soup.header1  
C. soup.h1[0]  
D. soup.h1[1]

Bottom of Form

Q-30: Which of the following gets the first link tag and returns a dictionary of all attributes and values for that link tag?

Top of Form

A. soup.a.attributes  
B. soup.link.attrs  
C. soup.a.attrs  
D. soup.link.attributes

Bottom of Form