Python 101 Quiz 2

1. Email:
2. First Name:
3. Last Name:

Section 1: Python Basics

1. Which of the following is an example of an escape characters?
2. \c
3. \n
4. \u
5. None of the above
6. Which of the following is the correct syntax of a Lambda function?
7. lambda(*arguments*, *expression*)
8. lambda *expression*: *arguments*
9. lambda *arguments* : *expression*
10. lambda(*expression*, *arguments*)
11. What datatype is this example: True
12. bool
13. str
14. int
15. char
16. What does this result in: re.split("\_", "Memorial\_Hermann\_Data\_Science\_Team")?
17. ["Memorial\_", "Hermann\_", "Data\_", "Science\_", "Team"]
18. ["memorial", "hermann", "data", "science", "team"]
19. ["Memorial", "\_Hermann", "\_Data", "\_Science", "\_Team"]
20. ["Memorial", "Hermann", "Data", "Science", "Team"]
21. Which the following is required for a Python conditional statement?
22. if
23. else
24. elif
25. All of them are required for a Python conditional statement

Section 2: Data Ingestion

1. What does the following do? pd.read\_excel("file.xlsx")
2. Reads a flat csv file named "file"
3. Reads a flat json file named "file"
4. Reads a flat excel file named "file"
5. Reads a flat text file named "file"
6. Which of the following is a Pandas reader method?
7. pd.read\_text()
8. pd.read\_json()
9. pd.read\_pdf()
10. pd.read\_file()
11. Which of the following is a Pandas writer method?
12. pd.to\_text
13. pd.to\_sas
14. pd.to\_csv
15. pd.to\_file
16. How do you write to a csv file called "result" with a pipe ("|") delimiter?
17. pd.to\_csv("result.csv")
18. pd.to\_csv("result.xlsx")
19. pd.to\_csv("result.csv", sep= "|")
20. pd.to\_csv("result.xlsx", sep= "|")
21. Which of the following is NOT a flat file?
22. filename.csv
23. filename.txt
24. filename.xlsx
25. filename.sql

Section 3: Data Exploration

1. How do you remove empty columns from DataFrame df?
2. df.dropna(axis=0, how='all')
3. df.dropna(axis=1, how='all')
4. df.dropcol(axis=0)
5. df.drops(axis=1)
6. What does the df.describe() method produce?
7. Generates a descriptive analysis of the columns with int64 or float64 datatype by default.
8. Generates a descriptive analysis of all columns in the dataframe.
9. Generates a descriptive analysis of the rows with int64 or float64 values by default.
10. Generates a descriptive analysis of all text columns in the dataframe.
11. What does df['colA'].unique() return?
12. A series object that shows the count of unique values in the "colA" column of Dataframe df.
13. A list of unique values from the "colA" column of Dataframe df.
14. A count of unique values from the "colA" column of Dataframe df.
15. A series object of unique values from the "colA" column of Dataframe df.
16. How do you get the count of unique values from column "colB" from Dataframe df?
17. df["colB"].count\_unique()
18. df["colB"].num\_unique()
19. df["colB"].nunique()
20. df["colB"].unique()
21. What does df["colC"].value\_counts() return?
22. A series containing counts of unique values within column "colC" of Dataframe df
23. A list containing counts of unique values within column "colC" of Dataframe df
24. A dataframe containing counts of unique values within column "colC" of Dataframe df
25. A dict containing counts of unique values within column "colC" of Dataframe df

Section 4: Data Wrangling

1. How to convert a Pandas Series object x to a number or numeric type?
2. x.to\_number()
3. x.to\_numeric()
4. pd.to\_number(x)
5. pd.to\_numeric(x)
6. Which of the following is the INCORRECT way to use the loc method?
7. df.loc[:, ['column\_one', 'column\_two']]
8. df.loc[0, 5]
9. df.loc[0, 'column\_one']
10. df.loc[df['column\_one'] == 0]
11. What does df.iloc[0:2, 2:4] do?
12. Selected from a dataframe (df) row indices from 0 to 1 and column indices from 2 to 3.
13. Selected from a dataframe (df) row indices from 0 to 2 and column indices from 2 to 4.
14. Selected from a dataframe (df) column indices from 0 to 1 and row indices from 2 to 3.
15. Selected from a dataframe (df) column indices from 0 to 2 and row indices from 2 to 4.
16. Which of the following concat dataframe df2 to df1?
17. pd.concat(df1, df2)
18. df1.concat(df2)
19. df1 = pd.concat(df2)
20. pd.concat([df1, df2])
21. Which of the following left joins df1 and df2 with df1 as the left dataframe using 'column\_one' as joining key?
22. pd.merge(df2, df1, how='left', on=['column\_one'])
23. pd.merge(df2, df1, on=['column\_one'])
24. df1.merge(df2, how='left', on=['column\_one'])
25. df1.merge(df2, on=['column\_one'])

Section 5: Data Analysis & Reporting

1. Which of the following fills missing values in Dataframe df with forward fill method?
2. df.fillna(method='ffill')
3. df.ffill()
4. df.pad()
5. All the above
6. Which method drops duplicate rows from Dataframe df and returns a dataframe without these duplicated rows?
7. df.drop\_duplicate()
8. df.drop\_duplicated()
9. df.drop\_duplicates()
10. df.drop\_duplicate\_rows()
11. What does df.groupby() do?
12. Groups a dataframe by a mapper or dataframe of columns
13. Groups a dataframe by a mapper or Series of columns
14. Groups a Series by a mapper or dataframe of columns
15. Groups a Series by a mapper or Series of columns
16. Which of the following is NOT a data visualization supported by Pandas?
17. Bar Graph
18. Pie Graph
19. Line Graph
20. All of them are supported by Pandas