Test_5_Resuelto

May 13, 2025

Creado por:

Isabel Maniega

1 Test 5

1.1 Question 1

Which of the following methods is used to find elements in the *xml.etree.ElementTree* module?

(Select all that apply)

a) findall

b) iter

c) find

d) findone

1.2 Solution 1

a, b, c

1.3 Question 2

You are preprocessing a dataset and need to standardize the "Height" and "Weight" columns. Which Python library and method would you use to standardize these columns using z-score normalization?

a) sklearn.preprocessing.StandardScaler()

b) pandas.DataFrame.standarize()

c) numpy.zscore()

d) scipy.stats.standarize()

1.4 Solution 2

a

The sklearn library provides the StandardScaler() class for standardizing data using z-score normalization. Using scaler.fit_transform(df[['Height', 'Weight']]) will standardize the "Height" and "Weight" columns.

1.5 Question 3

How can you retrieve all records from a SQL database table named Orders using Python's sqlite3 library?

a) `cursor.execute("SELECT * FROM orders").fetchall()`

b) `cursor.query("GET * FROM orders").all()`

c) `cursor.get("SELECT * IN Orders").collect()`

d) `cursor.run("FECTH ALL IN Orders")`

1.6 Solution 3

a

This choice is correct because it uses the execute method of the cursor object to run a SQL query that selects all records from the Orders table in the database. The fetchall method is then used to retrieve all the selected records as a list of tuples, effectively retrieving all records from the table.

1.7 Question 4

Your Python script is giving you unexpected results in data analysis. Which of the following is generally not advisable for debugging this issue?

a) Suppress all exceptions using a general `except` block.

b) Review and validate data inputs and transformations.

c) Examine the Python traceback to identify where the error originates.

d) Use conditional breakpoints to isolate the problem

1.8 Solution 4

a

This is generally not advisable because you may overlook the root cause of the problem, making it difficult to debug effectively.

1.9 Question 5

An XML document includes:

(Select all that apply)

a) atributes

b) a prolog

c) a root element

d) properties

1.10 Solution 5

a, b, c

1.11 Question 6

An object of any class:

a) is not JSON-serializable by default

b) is either-serializable by default, or can be turned into such an object.

c) is always JSON-serializable.

1.12 Solution 6

b

1.13 Question 7

An XML comment is a text surrounded by:

a) `//` and `\\`

b) `<!--` and `-->`

c) `<*` and `*>`

1.14 Solution 7

b

1.15 Question 8

A value, described by Python as `True`, is reflected by the JSON element called:

a) `true`

b) `True`

c) `TRUE`

d) `1`

1.16 Solution 8

a

1.17 Question 9

The `json.dumps()` method: (Select all that apply)

a) returns a JSON string

b) takes Python data as its argument

c) takes a JSON string as its argument.

d) returns Python data

1.18 Solution 9

a, b

1.19 Question 10

The `json.loads()` method: (select all that apply)

a) takes Python data as its argument

b) returns a JSON object.

c) takes a Json string as its argument.

d) returns a string

1.20 Solution 10

b, c

1.21 Question 11

In the JSON processing context, the term *deserialization*:

a) names a process in which a JSON string is remodeled and transformed into a new JSON string.

b) names a process in which the Python data is turned into a JSON string

c) refers to nothing as there is no such string as JSON deserialization.

d) names a process in which a JSON string is turned into Python data.

1.22 Solution 11

d

1.23 Question 12

```
You have analyzed website traffic data and created a visualization showing the user engagement

a) Use technical terms and metrics without explanation.
b) Create a narrative around user behavior patterns and trends.
c) Provide raw data and detailed charts for audience exploration.
d) Focus on the aesthetics of the visualization rather than the insights.
```

1.24 Solution 12

b

Create a narrative around user behavior patterns and trends. For a non-technical audience, creating a narrative around user behavior patterns and trends helps contextualize the insights and make them more relatable. This approach allows the audience to understand the implications of the data in terms of user experience and engagement.

1.25 Question 13

```
You have analyzed a dataset containing customer purchase behavior and created a visualization s

a) Present detailed statistical analysis with technical terms.
b) Use storytelling and simple language to explain the trends and patterns.
c) Provide raw* data and complex charts for audience interpretation.
d) Focus only on the technical aspects without context or storytelling.

*raw: sin procesar
```

1.26 Solution 13

b

Use storytelling and simple language to explain the trends and patterns. For a non-technical audience, it's important to use storytelling techniques and simple language to convey the insights from visualizations effectively. This approach helps the audience understand the trends and patterns without overwhelming them with technical details.

1.27 Question 14

```
You are analyzing the performance of an email marketing campaign. Which of the following metri
```

a) Email Conversion Rate (Email Conversion Rate measures the perentage of recipients who take a
b) Time Spent on website.
c) Email Open Rate.
d) Bounce Rate (Bounce rate is a metric that reprensents the percentage of visitors who enter a

1.28 Solution 14

b

While it can indicate engagement, it doesn't directly reflect the effectiveness of the email marketing campaign in generating sales.

1.29 Question 15

When using Matplotlib, what does the figsize parameter control in the plt.figure() function?

a) The number of subplots in the figure.

b) The color scheme of the figure.

c) The resolution of the saved figure.

d) The type of chart to be rendered (e.g., bar, pie, line)

e) The aspect ratio and size of the figure.

1.30 Solution 15

e

The figsize parameter in the plt.figure() function controls the aspect ratio and size of the figure that will be displayed when using Matplotlib. It allows you to adjust the width and height of the figure to customize the visual representation of your data.

1.31 Question 16

Which of the following presents the proper form of putting a quote inside a JSON string?

a) ' " '

b) ' ' " ' '

c) " "" "

d) " \" "

1.32 Solution 16

d

1.33 Question 17

```
In the context of an ETL (Extract, Transform, Load) process, what is the primary purpose of the

a) Creating visualizations and reports for end-users.
b) Retrieving and reading data from multiple heterogeneous data sources.
c) Loading data into a data warehouse or database.
d) Performing data cleaning and preparation for analysis.
```

1.34 Solution 17

b

The 'Extract' phase is all about extracting data from various sources, which can include databases, APIs, flat files, etc. The key task is to efficiently retrieve data in its original format.

1.35 Question 18

Which of the following techniques should be avoided to ensure that your Python data scripting code is easily understandable and follows best practices?

a) Documenting functions with docstrings.

b) Using descriptive variable names.

c) Refactoring repetitive code into reusable functions.

d) Using single-letter variable names for main variables.

1.36 Solution 18

d

This should be avoided, as it can make the code hard to understand and maintain.

1.37 Question 19

While running an ETL job, you encounter issues with data types mismatch during the transformation stage. The source data contains strings that are supposed to be cast to integers. What would be the most robust way to handle this issue?

a) Manually inspect and clean the source data before running the ETL job.

b) Automatically convert any non-convertible string to the integer 0.

c) Use Python's try-except block to cast each value and catch exceltions, logging the errors.

d) Skip any records that cannot be cast to integers.

1.38 Solution 19

c

Using a try-except block allows you to attempt the cast and handle errors gracefully, which is generally the most robust solution.

1.39 Question 20

When working on a data manipulation script, what is the best approach for debugging and ensuring the quality of the code?

a) Use `print()` statements throughout the code for debugging.

b) Write test and perform code reviews.

c) Use `assert`statements for checking intermediate results.

d) Use comments to describe what each section of the code does.

1.40 Solution 20

b

This is considered a best practice in software development, including data scripting. Unit tests help you catch errors early, and code reviews provide an extra layer of quality assurance.

1.41 Question 21

When working on a Python script that processes a large dataset, which of the following techniques is considered a best practice for handling exceptions and errors?

a) Ignoring errors to improve script performance.

b) Using generic except blocks to catch all exceptions.

c) Using global variables to store error states

d) Using specific exception handling with try-except blocks for known issues.

1.42 Solution 21

d

This is a best practice, as it allows you to handle known errors gracefully without affecting other parts of the script.

1.43 Question 22

You've written a Python data script and find that it's throwing exceptions intermittently. What is the least recommended way to handle exceptions in this scenario?

a) Use a broad `except:`block to catch all exceptions and continue execution.

b) Investigate the exceptions and fix the root cause.

c) Replace the code block with a try-except block specific to the excepted exceptions.

d) Log the exceptions for further analysis.

1.44 Solution 22

a

Using a broad except: block to catch all exceptions is not recommended as it can mask specific errors and make it difficult to identify and address the root cause of the intermittent exceptions. It is better to have specific exception handling to ensure that only the expected errors are caught and handled appropriately.

1.45 Question 23

Which of the following Seaborn plots is used to visualize the relationship between two numerical variables?

a) Bar plot

b) Scatter plot

c) Box plot

d) Heatmap

1.46 Solution 23

b

A scatter plot is used to show relationships between two numerical variables by plotting points based on their values.

1.47 Question 24

What is the best way to organize a dataset with multiple variables and observations in Python?

a) Use a dictionary with variable names as key and observations as values.

b) Use a Pandas Dataframe with rows as observations and columns as variables.

c) Create multiple list for each variable and store them separately.

d) Use a single list with sub-lists containing each observation.

1.48 Solution 24

b

Using a Pandas DataFrame is the best way to organize a dataset with multiple variables and observations in Python. DataFrames are specifically designed for tabular data with rows representing observations and columns representing variables. This structure allows for easy manipulation, analysis, and visualization of the data.

1.49 Question 25

When debugging a complex Python script for data analysis, which of the following practices is least recommended?

a) Inserting print statements at different points in the script.

b) Ignoring warnings as long as the script does not crash.

c) Writing unit tests to verify each function.

d) Using a debugger to step through the code.

1.50 Solution 25

b

Ignoring warnings, even if the script does not crash, can lead to potential issues that may affect the accuracy and reliability of the data analysis. It is important to address warnings as they may indicate underlying problems that could impact the script's performance or results.

1.51 Question 26

You are cleaning a dataset and need to replace missing values in the "Income" column with the mean income. Which Python library and method would you use to perform this task?

a) pandas.DataFrame.fillna()

b) numpy.fillna()

c) sklearn.preprocessing.Imputer()

d) matplotlib.pyplot.fill_missing()

1.52 Solution 26

a

pandas.DataFrame.fillna() The pandas library provides the fillna() method for replacing missing values in a DataFrame. Using `df['Income'].fillna(df['Income'].mean())` will replace missing values in the "Income" column with the mean income.

1.53 Question 27

You have been given a list of mixed data types and your task is to validate the numeric data within the list for further analysis. Which of the following Python code snippets would successfully extract all the valid numeric data into a new list?

```
data_list = ['apple', 42, 'banana', 99.9, 0, 'cherry', 11, None]
```

a) `new_list = [int(x) for x in data_list if x.isnumeric()]`

b) `new_list = [x if type(x) in (int, float) else 'Invalid' for x in data_list]`

c) `new_list = [x for x in data_list if type(x) == 'int' or type(x) == 'float']`

d) `new_list = [x for x in data_list if isinstance(x, (int, float))]`

1.54 Solution 27

[1]:
```
data_list = ['apple', 42, 'banana', 99.9, 0, 'cherry', 11, None]
```

[2]:
```
# a
new_list = [int(x) for x in data_list if x.isnumeric()]
new_list
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[2], line 2
      1 # a
----> 2 new_list = [int(x) for x in data_list if x.isnumeric()]
      3 new_list

AttributeError: 'int' object has no attribute 'isnumeric'
```

[3]:
```
# b
new_list = [x if type(x) in (int, float) else 'Invalid' for x in data_list]
new_list
```

[3]:
```
['Invalid', 42, 'Invalid', 99.9, 0, 'Invalid', 11, 'Invalid']
```

[4]:
```
# c
new_list = [x for x in data_list if type(x) == 'int' or type(x) == 'float']
new_list
```

[4]:
```
[]
```

[5]:
```
# d
new_list = [x for x in data_list if isinstance(x, (int, float))]
new_list
```

[5]:
```
[42, 99.9, 0, 11]
```

d

This code snippet uses list comprehension to iterate over each element in the data_list and checks if the element is an instance of either int or float data types. If the element is numeric, it is added to the new_list. This approach successfully extracts all valid numeric data from the mixed data types list.

1.55 Question 28

Consider the Python code snippet below:

```
try:
    with open('data.txt', 'r') as file:
        lines = file.readlines()
        value = int(lines[0].strip())
        print(value)
except (FileNotFoundError, ValueError, IndexError) as e:
    value = 0
```

If the file data.txt contains the following lines:

```
10
20
30
```

What will be the value of value after executing the code?

a) 10

b) 20

c) 0

d) 30

e) 102030

1.56 Solution 28

[6]:
```
try:
    with open('data.txt', 'r') as file:
        lines = file.readlines()
        value = int(lines[0].strip())
        print(value)
except (FileNotFoundError, ValueError, IndexError) as e:
    value = 0
```

```
10
```

a

The code snippet opens the file 'data.txt', reads the lines, and extracts the integer value from the first line. Since the first line in the file is '10', the value variable will be assigned the integer value of 10 after executing the code.

1.57 Question 29

When connecting to a SQL database from Python to collect data, what limitation should you be most concerned about when dealing with very large datasets?

a) SQL databases usually enforce strong data typing, which may cause type mismatch errors in Python.

b) Large data pulls can exhaust Python's memory, causing the program to crash.

c) SQL databases typically limit the number of rows that can be returned in a single query.

d) SQL databases often require complex join operations that Python cannot handle.

1.58 Solution 29

b

Large data pulls can indeed exhaust Python's memory, especially when dealing with very large datasets. This can lead to the program crashing due to memory limitations, making it a significant concern when collecting data from a SQL database.

1.59 Question 30

Select the true statements about XML:

(Select all that apply)

a) Each XML document must have a root element.

b) Each element contains at least one attribute

c) Each open XML tag must have a corresponding closing tag.

d) Each XML document must have a prolog.

1.60 Solution 30

a, c

1.61 Question 31

Which of the following best practices ensures that a data script can be easily understood, modified, and debugged by other data analysts in your team?

a) Making your code run as fast as possible.

b) Minimizing the number of functions used.

c) Employing meaningful variable names and commenting your code.

d) Keeping lines of code as short as possible.

1.62 Solution 31

c

Employing meaningful variable names and commenting your code is essential for ensuring that other data analysts can easily understand, modify, and debug the script. Clear and descriptive variable names help convey the purpose of each variable, while comments provide additional context and explanations for the code logic, making it easier for team members to follow along and make changes if needed.

1.63 Question 32

You are preprocessing a dataset and need to standardize the "Height" and "Weight" columns. Which Python library and method would you use to standardize these columns using z-score normalization?

a) sklearn.preprocessing.StandardScaler()

b) pandas.DataFrame.standarize()

c) numpy.zscore()

d) scipy.stats.standarize()

1.64 Solution 32

a

sklearn.preprocessing.StandardScaler() The sklearn library provides the StandardScaler() class for standardizing data using z-score normalization. Using scaler.fit_transform(df[['Height', 'Weight']]) will standardize the "Height" and "Weight" columns.

1.65 Question 33

John is tasked with ensuring the reliability and accuracy of collected data from various sources. Which data validation technique should he focus on to achieve this goal?

a) Data type validation

b) Range validation

c) Consistency validation

d) Completeness validation

1.66 Solution 33

c

Consistency validation Consistency validation checks for consistency in data across different sources or datasets. It ensures that data is uniform and follows predefined rules, contributing to the reliability and accuracy of the collected data.

1.67 Question 34

You are analyzing a dataset containing timestamps for online transactions. Which data validation technique is most appropriate for ensuring the reliability of the timestamps?

a) Data type validation

b) Completeness validation

c) Consistency validation

d) Format validation

1.68 Solution 34

d

Format validation Format validation checks if the timestamps follow the expected format, ensuring the reliability of the data recorded.

1.69 Question 35

You are tasked with validating a dataset containing text data collected from surveys. Which data validation technique should you use to ensure the reliability and accuracy of the text data?

a) Data type validation

b) Completeness validation

c) Cross-reference validation

d) Consistency validation

1.70 Solution 35

d

Consistency validation Consistency validation checks for uniformity and reliability within the dataset, ensuring that the text data is consistent and accurate across all entries.

1.71 Question 36

You are analyzing a dataset for predicting housing prices. The dataset includes features like square footage, number of bedrooms, and neighborhood. What is a recommended approach for handling the "Neighborhood" feature in this dataset for modeling?

a) Convert the 'Keighborhood' feature into ordinal categories based on property value.

b) Use one-hot-encoding to transform the 'Neighborhood' feature into binary columns.

c) Group similar neighborhoods together and create a new categorical feature.

d) Ignore the 'Neighbrhood' feature as it is not relevant for predicting housing prices.

1.72 Solution 36

b

Use one-hot encoding to transform the "Neighborhood" feature into binary columns, is the recommended approach as one-hot encoding preserves the categorical information of the "Neighborhood" feature without introducing ordinality or bias.

1.73 Question 37

You are tasked with writing a Python script to clean a CSV file named "data.csv" containing customer information. The script should remove any rows where the "Age" column has missing values and save the cleaned data to a new file named "cleaned_data.csv". Which of the following code snippets achieves this task?

a)

```
import pandas as pd

data = pd.read_csv('data.csv')
cleaned_data = data.dropna(subset=['Age'])
cleaned_data.to_csv('cleaned_data.csv', index=False)
```

b)

```
import pandas as pd

data = pd.read_csv('data.csv')
cleaned_data = data[data['Age'].notna()]
cleaned_data.to_csv('cleaned_data.csv', index=False)
```

c)

```
import pandas as pd

data = pd.read_csv('data.csv')
cleaned_data = data.drop(['Age'], axis=1, implace=True)
cleaned_data.to_csv('cleaned_data.csv', index=False)
```

d)

```
import pandas as pd

data = pd.read_csv('data.csv')
cleaned_data = data.dropna(subset=['Age'], implace=True)
cleaned_data.to_csv('cleaned_data.csv', index=False)
```

1.74 Solution 37

b

Correctly reads the CSV file, filters out rows with missing values in the "Age" column using the .notna() method, and saves the cleaned data to a new file.

1.75 Question 38

You are presenting a complex heatmap to a non-technical audience that shows the correlation between different variables in a dataset. What is the most effective approach to communicate the key insights from this heatmap?

a) Present the raw data and ask the audience to identify correlations.

b) Start with a summary, focusing on the highest and lowest correlations that matter.

c) Focus on the color scheme to show which variables are strongly correlated.

d) Discuss the mathematical algorithms that power the heatmap.

1.76 Solution 38

b

Starting with a summary that highlights the highest and lowest correlations in the heatmap is the most effective approach to communicate key insights to a non-technical audience. This allows the audience to quickly grasp the most important relationships in the data without getting overwhelmed by the complexity of the heatmap.

1.77 Question 39

In Python, when using the pandas library to clean a DataFrame that has several columns with missing values, which function would you use to fill these gaps with the column's mean value?

a) `df.fillna(df.mean())`

b) `df.replace(np.nan, df.mean())`

c) `df.dropna()`

d) `df.mean(fillna=True)`

1.78 Solution 39

a

`df.fillna(df.mean()) df.fillna(df.mean())` is the correct method to fill missing values with the mean of each column in a pandas DataFrame. This function specifically targets missing values (NaN) and replaces them with the mean value of the respective column.

1.79 Question 40

How can a data analyst in Python ensure that a pandas DataFrame df does not contain any null or NaN values before proceeding with further data analysis?

a) `df.isnull().any()`

b) `df.notnull().all()`

c) Both A and B can be used effectively

d) `df.validate_null()`

1.80 Solution 40

c

Both A and B can be used effectively Both df.isnull().any() and df.notnull().all() are effective methods for checking the presence of null or NaN values in a DataFrame. The former checks if there are any null values, and the latter verifies that all values are not null.

1.81 Question 41

You are tasked with loading a CSV file into a Pandas DataFrame and performing basic data exploration. Which code snippet correctly accomplishes this task assuming the CSV file is named "data.csv" and contains comma-separated values?

```
import pandas as pd
```

a) `df = pd.read_csv('data.csv')`

b) `df = pd.read_excel('data.csv')`

c) `df = pd.read_table('data.csv')`

d) `df = pd.load_csv('data.csv')`

1.82 Solution 41

a

A uses the read_csv() function from the Pandas library to load the CSV file "data.csv" into a DataFrame named df, which is the correct approach for loading CSV data into Pandas.

1.83 Question 42

When using Matplotlib in Python, what is the main purpose of the plt.subplots() function?

a) To create a single plot within a larger grid

b) To save the current plot to an image file.

c) To create a 3D plot.

d) To create multiple sub-plot in a single figure.

e) To add interactive features like zoom and pan.

1.84 Solution 42

d

The plt.subplots() function in Matplotlib is used to create multiple sub-plots within a single figure. This allows for the visualization of multiple plots in a grid-like format, making it easier to compare different data visualizations.

1.85 Question 43

In Matplotlib, which function is commonly used to display images?

a) plt.bar()

b) plt.pie()

c) plt.scatter()

d) plt.line()

e) plt.imshow()

1.86 Solution 43

e

The plt.imshow() function in Matplotlib is commonly used to display images. It is specifically designed to show images as 2D arrays, making it the ideal choice for visualizing images in Python.

1.87 Question 44

You are dealing with a complex dataset involving numerous features from different domains, such as financial metrics, user activity, and geographical data. You are tasked with creating a report that synthesizes this information for non-technical stakeholders. Which data visualization technique would be best suited to represent the multi-domain features in a digestible format?

a) Radar Chart

b) Dashboard with Multiple 2D plots.

c) Boxplot

d) Heatmap

e) 3D Scatter Plot

1.88 Solution 44

b

A dashboard with multiple 2D plots would be the best choice for representing multi-domain features in a digestible format for non-technical stakeholders. This visualization technique allows for the simultaneous display of various features in separate plots, making it easier for stakeholders to compare and analyze different aspects of the data at a glance

---

1.89 Question 45

You're using the sqlite3 library in Python to fetch data from an SQLite database. You want to get all the records from a table named "Employees" where the "Age" is greater than 30 and sort them by "Name" in ascending order. Which SQL query accomplishes this?

a) `SELECT * FROM Employees IF Age > 30 ORDER BY Name ASC;`

b) `SELECT * FROM Employees WHERE Age > 30 ORDER BY Name ASC;`

c) `SELECT * FROM Employees WHERE Age > 30 SORT BY Name ASC;`

d) `SELECT * FROM Employees WHERE Age > 30 ORDER BY Age ASC;`

1.90 Solution 45

b

This SQL query correctly selects all records from the "Employees" table where the "Age" is greater than 30 and sorts them by the "Name" column in ascending order using the ORDER BY clause with ASC keyword.

Creado por:

Isabel Maniega