Your grade: 80%

Your latest: 80% • Your highest: 80% • To pass you need at least 80%. We keep your highest score.

Next item →

1. A data analyst is going to use the library `addup` and wants to give the library the alias au. Which of the following pieces of code should the analyst use? — 1 / 1 point

- `apply alias au to addup`
- `alias au as addup`
- `rename au to addup`
- (selected) `import addup as au`

Correct
That's right! Using an alias in place of a library name can be more convenient than typing out the entire library name.

2. A data analyst is using Pandas to analyze recent trends in their social media campaigns. They have recent data saved in a CSV named "smcdata". Which of the following pieces of code would allow them to import this data? — 1 / 1 point

- import_csv as ('smcdata.csv')
- `pandas.read_csv('smcdata')`
- `pandas.import_csv('smcdata.csv')`
- (selected) `pandas.read_csv('smcdata.csv')`

Correct
That's right! This code will import the campaign data to a DataFrame object which can be used with the Pandas library.

3. If you want to preview the first 25 rows of a DataFrame assigned the variable `df`, which of the following methods should you use? — 1 / 1 point

- (selected) `df.head(25)`
- `head.df(25)`
- `df.first25()`
- `df.prev(25)`

Correct
That's right! `head()` is a special function attached to the DataFrame object.

4. When using Pandas, what is a series object? — 1 point

- A preview of the first five rows of a DataFrame
- A random sample pulled from a DataFrame
- A single array of data, like a column of data from a DataFrame
- (selected) A single horizontal slice, like a row of data from a DataFrame

Incorrect
Not quite. Review the video Working with Pandas Series & DataFrames.

5. You need to identify a subset of car models with a fuel efficiency of 30 mpg or higher. The DataFrame is assigned the variable `df` and the applicable information is stored in the mpg column. Which of the following masks should you use? — 1 / 1 point

- (selected) `my_mask_high_eff = df['mpg'] >= 30`
- `my_mask_high_eff = df['mpg'] <= 30`
- `my_mask_high_eff = df['fuel'] == 30`
- `my_mask_high_eff = df['mpg'] == 30`

Correct
That's right! It's helpful to create masks that use conditionals to find a subset of data.

6. When scrubbing data in Python, what is one example of an issue that should be fixed early in the process? — 1 point

- Duplicate data
- Unsupportive data
- (selected) Strange data
- Missing data

Incorrect
Not quite. Review the video What is Scrubbing?

7. Which of the following methods can you use to return a version of the DataFrame with a row or column removed? — 1 / 1 point

- (selected) `drop()`
- `delete()`
- `remove()`
- `pull()`

Correct
That's right! The drop() method will return a version of the DataFrame with rows or columns removed, leaving the original DataFrame unchanged.

8. Your dataset has multiple rows that are identical. This scenario is expected, but you only need to retain the final instance of the duplicate row. Which of the following methods should you use? — 1 / 1 point

- (selected) `duplicated(keep='last')`
- `duplicated(keep='final')`
- `duplicated(keep=False)`
- `duplicated(keep='none')`

Correct
That's right! The duplicated() method will return a series of True or False, and by default will not consider the first instance of a row a duplicate.

9. You are scrubbing a DataFrame and want to find out the data types of each column. Which of the following methods can you use? — 1 / 1 point

- `type()`
- `learn()`
- (selected) `info()`
- `kind()`

Correct
That's right! The `info()` method provides information about the entire DataFrame including things like memory usage and the number of non-null values.

10. You are scrubbing a DataFrame (`df`) and notice that a number of age values for subscribers are negative. You know from context that people's ages should be all positive integers and these instances are typos. You want to replace negative values in `df.subscribers` with positive values in your dataset. Which of the following pieces of code will change those negatives to positives? — 1 / 1 point

-
```
negative_ages = df.ages < 0
df.ages[negative_ages] = invert[df.ages[negative_ages]
```
-
```
negative_ages = df.ages < 0
df.ages[negative_ages] = pos df.ages[negative_ages]
```
-
```
negative_ages = df.ages = -10
df.ages[negative_ages] = -1 * df.ages[negative_ages]
```
- (selected)
```
negative_ages = df.ages < 0
df.subscribers[negative_ages] = -1 * df.ages[negative_ages]
```

Correct
That's right! This code identifies the values less than one, then uses a calculation to change the values to a positive equivalent.