ETHNOTECH

QUESTION BANK

MACHINE LEARNING SESSION1, SESSION2, SESSION 3



RNSIT-COE

2023-24

1 X	What is machine learning?
Points:	C An approach for designing hardware components with high computational power
0/1	○ A branch of artificial intelligence that deals with algorithms that can learn from data ✓
	C A method for creating virtual reality simulations
	O A technique for programming computers to perform specific tasks
2 X	Which of the following is NOT a type of machine learning algorithm?
	© Semi-supervised Learning
Points: 0/1	© Supervised Learning
	O Deterministic Learning ✓
	O Unsupervised Learning
3 X	What is the main goal of unsupervised learning?
	C To make predictions based on labeled data
Points: 0/1	© To learn from feedback provided by a teacher
	C To classify data into predefined categories
	○ To discover hidden patterns or structures in data ✓
4 X	Which evaluation metric is commonly used for classification tasks in machine learning?
	© Mean Absolute Error (MAE)
Points: 0/1	© R-squared (R²)
	© Root Mean Squared Error (RMSE)
	○ Accuracy ✓
	What is overfitting in machine learning?
5 X	○ When the model learns noise in the training data and performs poorly on unseen data ✓
Points: 0/1	C When the model underfits the training data
	© When the model performs well on unseen data
	C When the model generalizes well to new data
6 X	Which of the following are types of supervised learning algorithms? (Select all the 3 for full marks)
Points:	□ Decision Trees ✓
0/3	☐ K-Nearest Neighbors
	☐ Support Vector Machines ✓
	☐ K-Means Clustering
	☐ Linear Regression ✓
7 X	What are common methods for handling missing data in machine learning?
Points:	☐ Using predictive models to estimate missing values ✓
0/3	☐ Ignoring missing values during training
	☐ Dropping features with missing values
	☐ Removing observations with missing values ✓
	☐ Replacing missing values with the mean of the column ✓

×	What are the different categ	ories of algorithm classified based on desired outputs?				
Points:	☐ Clustering type ✓					
0/3	☐ Molding type					
	☐ Classification Type ✓					
	☐ Observational Type					
	☐ Regression Type ✓					
×	What are some limitations of	f machine learning techniques? (There are four right answers)				
Points:	☐ Susceptibility to Ove	erfitting 🗸				
0/4	☐ Limited Generalization	to Unseen Data ✓				
	Lack of Interpretability					
	Dependency on Quali	ty of Data ✓				
	□ Difficulty in Handling N	on-Numeric Data				
	Inability to Handle Lar	ge Datasets				
	i					
X	Match the following step by	step process of Machine Learning in the right order of numbers				
Points:						
0/8	Pre-process the Data	- Select -				
	•	Correct answer: 3				
		Correct answer. 5				
	Train the Model	- Select -				
		Correct answer: 6				
		Correct answer. o				
	Fine-tune the Model	- Select -				
	Time-tune the Model	Correct answer: 8				
		Correct answer. o				
	Choose a Model	- Select -				
		Correct answer: 5				
	Define the Problem	- Select -				
	Deline the Flobleth	Correct answer: 1				
	Split the Data	- Select -				
	Opin the Bata	Correct answer: 4				
	Evaluate the Model	- Select -				
	Evaluate the Model	Correct answer: 7				
		- Select -				
	Collect Data	X				
	Concot Data	Correct answer: 2				

```
import csv
            data = [
Points:
               ['Name', 'Age', 'City'],
0/1
               ['John', 30, 'New York'],
               ['Alice', 25, 'San Francisco'],
               ['Bob', 35, 'Los Angeles']
            # Writing to CSV file
                                   _, mode='w', newline=' ') as file:
            with open(___
               writer = csv.writer(file)
               writer.writerows(data)
            print("CSV file created successfully!")
            For the given program, select the appropriate replacements at the missing section
              Pass Nothing
              Just File Name
              C File Path

    ○ File Name with CSV extension ✓
            import csv
X
                                ___, mode='r') as file:
            with open(___
Points:
0/2
               reader = csv.reader(file)
               for row in reader:
                 print(row)
            For the given program, select both the appropriate replacements at the missing section
              ☐ File Path ✓
               ☐ File Name with csv extension ✓
              □ Pass Nothing

☐ File Name

            What does the writerows() function do in Python's csv module?
X
Points:
0/1
               C Reads multiple rows of data from a CSV file
               O Deletes multiple rows from a CSV file
               ○ Writes multiple rows of data to a CSV file ✓
               Writes a single row of data to a CSV file
            What is the purpose of the writerow() function in Python's csv module?
X
Points:
0/1
               O Writes multiple rows of data to a CSV file
               O Deletes a single row of data from a CSV file
               C Reads a single row of data from a CSV file

○ Writes a single row of data to a CSV file 
✓
```



Which of the following statements is true regarding writerows() and writerow() functions?

Points: 0/2

- O writerows() is used for writing a single row of data, while writerow() is used for writing multiple rows of data
- o writerows() is used for writing multiple rows of data, while writerow() is used for writing a single row of data 🗸
- O Both functions can only write a single row of data to a CSV file
- O Both functions are used for reading data from a CSV file

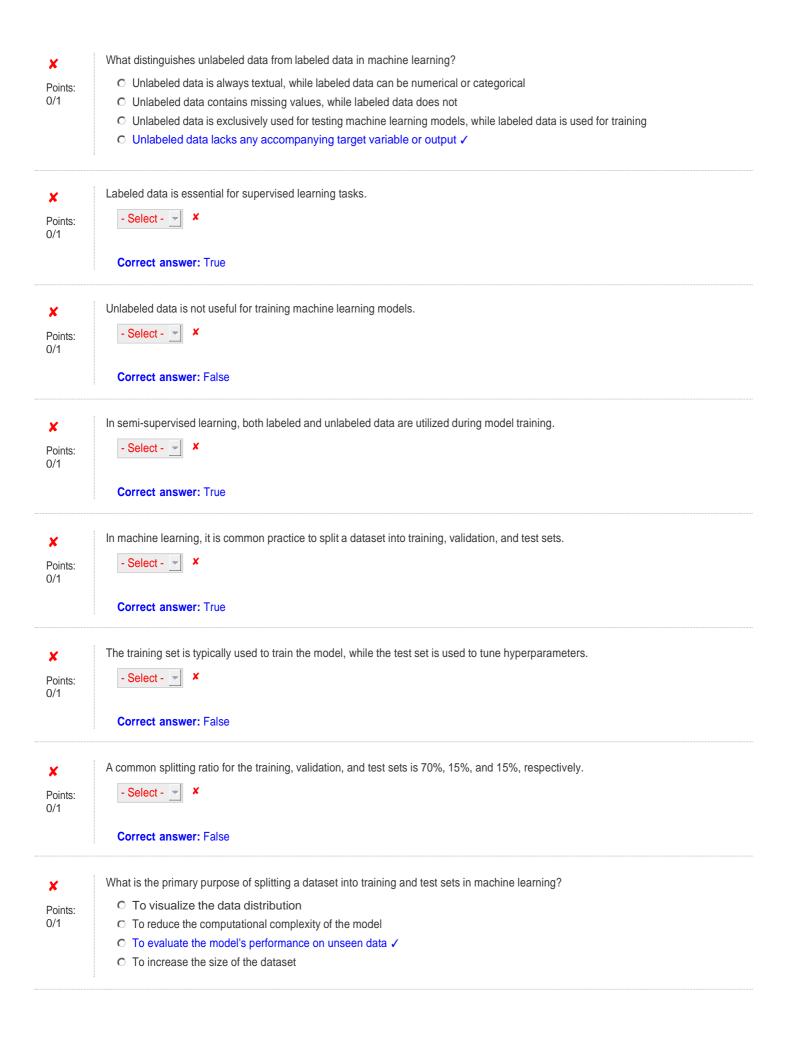
X

Points: 0/1

X

Points: 0/1

×	import csv							
Points:	csv_file_name = 'example.csv'							
0/2	column_to_read = 'Name'							
	with open(csv_file_name, 'r') as csv_file:							
	csv_reader = csv.reader(csv_file)							
	header = next(csv_reader)							
	column_index = header.index(column_to_read)							
	For the given program select the misisng code lines:							
	C for row in csv_reader: print(row[column_index]) ✓							
	C for rowin csv_reader: print(row[row])							
	C for col in csv_reader: for row in csv_reader: print(row[column_index])							
	O for row in csv_reader: print(row[column_to_read])							
×	Which of the following are common forms of data in machine learning? (Select all that apply)							
Points:								
0/5	☐ Text data ✓							
	☐ Audio data ✓							
	☐ Graph data							
	□ Numerical data ✓							
	☐ Categorical data ✓							
	☐ Image data ✓							
×	Data can come from various sources such as:							
Points: 0/1	C Surveys							
0/1	 ○ User-generated content ○ All of the above ✓ 							
	C Scientific experiments							
	© Social media							
	C Sensor data							
×	What is labeled data in the context of machine learning?							
Points:								
0/1	C Data that contains only numerical values							
	C Data that has been processed for visualization purposes							
	○ Data that is accompanied by its corresponding target variable or output ✓							
	 Data that is not suitable for training machine learning models 							
×	Which of the following is an example of labeled data?							
Points:	C A dataset of customer transactions with each transaction labeled as fraudulent or not fraudulent ✓							
0/1	Social media posts without any annotations							
	C A collection of images with no accompanying descriptions							
	C Sensor readings from environmental monitoring stations							



×	Which of the following is a common technique for splitting a dataset into training and test sets?
Points:	
0/1	© Principal Component Analysis
	Mean Squared Error
	C K-Means Clustering
X	What is the purpose of using a validation set in addition to training and test sets?
Points:	○ To increase the size of the training set
0/1	© To select the features for model training
	○ To evaluate the model's performance during training
	To reduce overfitting by tuning hyperparameters ✓
X Points: 0/3	Which of the following splitting ratios is commonly used for dividing a dataset into training and test sets? (Select all the three right answers)
	□ 90% training, 10% test ✓
	□ 80% training, 20% test ✓
	□ 50% training, 50% test
	□ 70% training, 30% test ✓
×	What is the purpose of data validation in machine learning?
Points:	To ensure that the data is accurate and complete ✓
0/1	© To visualize the distribution of the data
	C To prevent overfitting of the model
	○ To reduce the computational complexity of the model
······································	What is the primary goal of data cleaning in machine learning?
X	
Points: 0/1	 ○ To improve the quality of the dataset ✓ ○ To reduce the computational complexity of the model
0/1	To make the dataset more complex
	O To increase the size of the dataset
×	Which of the following are common techniques used in data cleaning?
Points:	
0/2	□ Normalizing data
	☐ Adding noise to the dataset
	☐ Imputing missing values ✓
	☐ Removing duplicates ✓
	What is imputation in the context of data cleaning?
×	What is imputation in the context of data cleaning?
Points:	O Splitting the dataset into training and test sets
0/1	○ Replacing missing values with estimated values ✓
	 Removing outliers from the dataset Adding noise to the dataset
	~ Adding holde to the dataset



Which of the following is a common method for handling outliers during data cleaning?

Points: 0/1

- C Ignoring the outlier during model training
- C Adding the outlier to a separate dataset
- C Replacing the outlier with the mean of the column
- Deleting the entire row containing the outlier



What is the purpose of feature scaling in data cleaning?

Points: 0/1

- C To add noise to the dataset
- To remove features with missing values
- To standardize or normalize the range of features ✓
- C To reduce the number of features in the dataset



What are the advantages of data processing in machine learning?

Points: 0/1

- C Reduced accuracy of predictions
- Increased computational complexity
- C Decreased dataset size



How does data processing contribute to better machine learning outcomes?

Points: 0/1

- By optimizing the training process for faster convergence ✓
- O By enhancing the interpretability of the model
- O By reducing the need for feature engineering
- C By increasing the noise in the dataset



Points: 0/1

	CustomerID	Name	Age	Gender	Address	Income
0	1	John Doe	25.0	Male	123 Main St, City	50000.0
1	2	Alice Smith	32.0	Female	456 Elm St, City	60000.0
2	3	Bob Johnson	NaN	Male	NaN	55000.0
3	4	Sarah Brown	28.0	Female	789 Oak St, City	70000.0
4	5	Michael Lee	42.0	NaN	NaN	65000.0
5	6	NaN	45.0	Male	987 Pine St, City	NaN

Select the appropriate code to handle missing values in the "Age" column by replacing them with the mean age:

a) mean_age = df['Age'].mean()

df['Age'].fillna(mean_age, inplace=True)

- b) df['Age'].fillna(mean_age, inplace=True)
- c) df['Gender'] = df['Gender'].str.lower()
- d) df['Address'] = df['Address'].str.strip()
- e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')

- Select - 🔻 🗶

Correct answer: a

X

Points: 0/1

	CustomerID	Name	Age	Gender	Address	Income
0	1	John Doe	25.0	Male	123 Main St, City	50000.0
1	2	Alice Smith	32.0	Female	456 Elm St, City	60000.0
2	3	Bob Johnson	NaN	Male	NaN	55000.0
3	4	Sarah Brown	28.0	Female	789 Oak St, City	70000.0
4	5	Michael Lee	42.0	NaN	NaN	65000.0
5	6	NaN	45.0	Male	987 Pine St, City	NaN

Select the appropritate code to remove rows with missing values in the dataset:

- a) mean_age = df['Age'].mean()
 - df['Age'].fillna(mean_age, inplace=True)
- b) df.dropna(inplace=True)
- c) df['Gender'] = df['Gender'].str.lower()
- d) df['Address'] = df['Address'].str.strip()
- e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')
 - Select 🔻 🗶

Correct answer: b



	CustomerID	Name	Age	Gender	Address	Income
0	1	John Doe	25.0	Male	123 Main St, City	50000.0
1	2	Alice Smith	32.0	Female	456 Elm St, City	60000.0
2	3	Bob Johnson	NaN	Male	NaN	55000.0
3	4	Sarah Brown	28.0	Female	789 Oak St, City	70000.0
4	5	Michael Lee	42.0	NaN	NaN	65000.0
5	6	NaN	45.0	Male	987 Pine St, City	NaN

Select the appropritate code to convert the "Gender" column to lowercase for consistency:

- a) mean_age = df['Age'].mean()
 - df['Age'].fillna(mean_age, inplace=True)
- b) df['Age'].fillna(mean_age, inplace=True)
- c) df['Gender'] = df['Gender'].str.lower()
- d) df['Address'] = df['Address'].str.strip()
- e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')
 - Select 🔻 🗶

Correct answer: c

X Points:

	CustomerID	Name	Age	Gender	Address	Income
0	1	John Doe	25.0	Male	123 Main St, City	50000.0
1	2	Alice Smith	32.0	Female	456 Elm St, City	60000.0
2	3	Bob Johnson	NaN	Male	NaN	55000.0
3	4	Sarah Brown	28.0	Female	789 Oak St, City	70000.0
4	5	Michael Lee	42.0	NaN	NaN	65000.0
5	6	NaN	45.0	Male	987 Pine St, City	NaN

Select the appropritate code to remove leading and trailing whitespaces in the "Address" column:

- a) mean_age = df['Age'].mean()
 - df['Age'].fillna(mean_age, inplace=True)
- b) df['Age'].fillna(mean_age, inplace=True)
- c) df['Gender'] = df['Gender'].str.lower()
- d) df['Address'] = df['Address'].str.strip()
- e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')
 - Select 🔻 🗶

Correct answer: d



	CustomerID	Name	Age	Gender	Address	Income
0	1	John Doe	25.0	Male	123 Main St, City	50000.0
1	2	Alice Smith	32.0	Female	456 Elm St, City	60000.0
2	3	Bob Johnson	NaN	Male	NaN	55000.0
3	4	Sarah Brown	28.0	Female	789 Oak St, City	70000.0
4	5	Michael Lee	42.0	NaN	NaN	65000.0
5	6	NaN	45.0	Male	987 Pine St, City	NaN

Select the appropritate code to convert numeric values in the "Name" column to NaN (missing values):

- a) mean_age = df['Age'].mean()
 - df['Age'].fillna(mean_age, inplace=True)
- b) df['Age'].fillna(mean_age, inplace=True)
- c) df['Gender'] = df['Gender'].str.lower()
- d) df['Address'] = df['Address'].str.strip()
- e) df['Name'] = pd.to_numeric(df['Name'], errors='coerce')
 - Select -

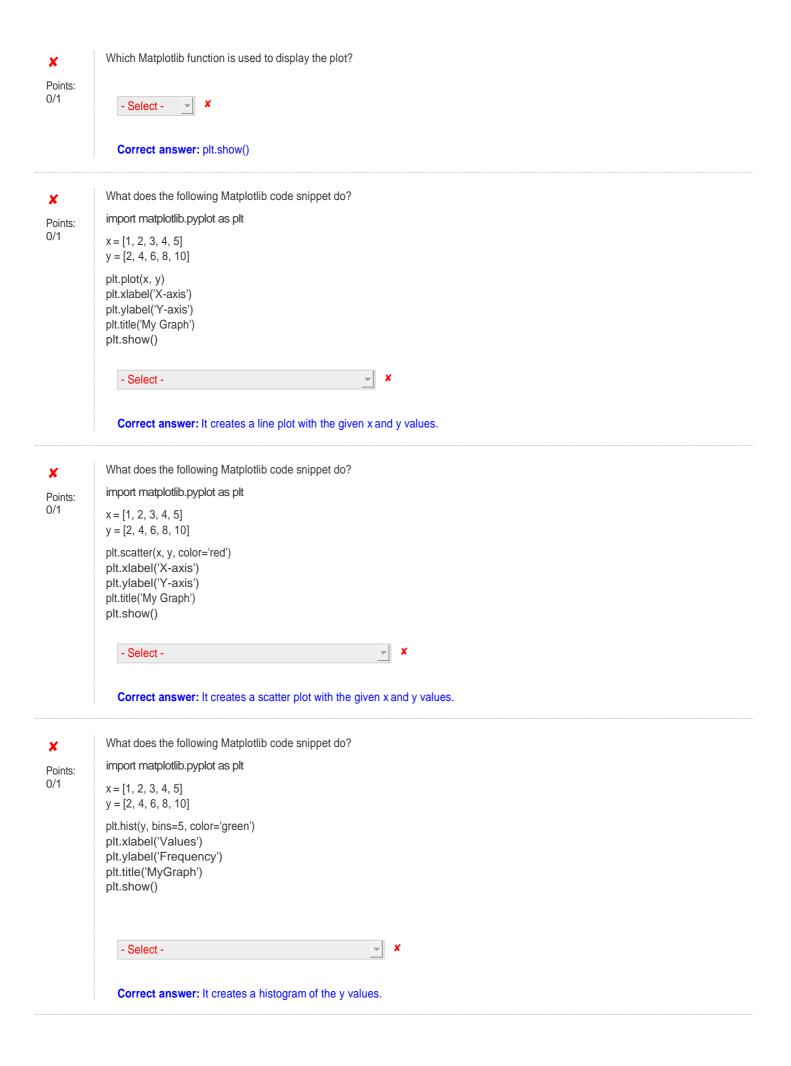
Correct answer: e

What is a key feature of NumPy? Points: 0/1 O Data manipulation and analysis through DataFrames C Efficient handling of large datasets Plotting of data visualizations What is a key feature of Pandas? X C Implementation of advanced mathematical functions Points: C Efficient handling of large datasets 0/1 Plotting of data visualizations Data manipulation and analysis through DataFrames What is a key feature of Matplotlib? X C Implementation of advanced mathematical functions Points: 0/1 C Efficient handling of large datasets O Data manipulation and analysis through DataFrames Plotting of data visualizations What is a key feature of SciPy? X Points: 0/1 O Data manipulation and analysis through DataFrames C Efficient handling of large datasets ○ Implementation of advanced mathematical functions ✓ Plotting of data visualizations Which library is primarily used for scientific computing and mathematical operations in Python? a X Matplotlib Points: 0/1 SciPy
✓ Pandas **ONumPy** import numpy as np data =____([Points: 0/1 [1, 2, 3], [4, 5, 6], [7, 8, 9]1) np.savetxt("Filename.csv", data, delimiter=',') The given data need to be converted to array, select the missing code to be get the program running properly. O np.random.rand()

○ np.linspace()○ np.data()○ np.array() ✓

×	What is the purpose of the <pre>numpy.savetxt()</pre> function?
Points: 0/1	 C To load data from a text file into a numpy array. C To save a numpy array to a text file. ✓ C To concatenate two numpy arrays along a specified axis. C To perform element-wise multiplication on two numpy arrays.
X Points: 0/1	What is the purpose of the numpy.loadtxt() function? C To convert a text file into a numpy array. ✓ C To concatenate two numpy arrays along a specified axis. C To save a numpy array to a text file. C To perform element-wise multiplication on two numpy arrays.
X Points: 0/1	What is the code line to save a NumPy array to a text file using numpy.savetxt()? - Select - Correct answer: np.savetxt("data.txt", arr, delimiter=",")
X Points: 0/1	What is the code line to load data from a text file into a NumPy array using numpy.loadtxt()? - Select - Correct answer: np.loadtxt("data.txt", delimiter=",")
Points: 0/1	What is the code line to save a NumPy array to a text file with a specified header using numpy.savetxt()? - Select - Correct answer: np.savetxt("data.txt", arr, header="Header")
X Points: 0/1	What is the code line to skip the first row while loading data from a text file into a NumPy array using numpy.loadtxt()? - Select - Correct answer: np.loadtxt("data.txt", skiprows=1)
Points: 0/1	What is the correct way to create a Pandas DataFrame from a dictionary? - Select - Correct answer: pd.DataFrame.from_dict(dict)
Points: 0/1	What is the correct way to create a Pandas DataFrame from a CSV file? - Select - Correct answer: pd.read_csv("data.csv")

×	import pandas as pd							
Points:	data = {'Name': ['John', 'Alice', 'Bob', 'Sarah'],							
0/1	'Age': [25, 32, 28, 35], 'City': ['New York', 'San Francisco', 'Los Angeles', 'Chicago']}							
	df = pd.DataFrame(data)							
	You have to write the data in to a csv file using pands, select the approriate missing code lines							
	- Select -							
	Correct answer: df.to_csv('output.csv', index=False)							
×	for given							
Points:	x = [1, 2, 3, 4, 5]							
0/1	y = [2, 4, 6, 8, 10]							
	What type of plot is most suitable for visualizing the relationship between the given x and y data?							
	- Select - 💌 🗶							
	Correct answer: Scatter plot							
×	for given							
Points:	x = [1, 2, 3, 4, 5]							
0/1	y = [2, 4, 6, 8, 10]							
	Which Matplotlib function is used to plot a bargraph?							
	- Select - 💌 🗴							
	Correct answer: plt.bar()							
×	for given							
Points:	x = [1, 2, 3, 4, 5]							
0/1	y = [2, 4, 6, 8, 10]							
	Which Matplotlib function is used to plot a linegraph?							
	- Select - 💌 🗴							
	Correct answer: plt.plot()							
×	What is the purpose of adding labels to the axes in a plot?							
Points:								
0/1	- Select -							
	Correct answer: To provide context and interpretation to the plotted data							



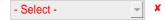


Points: 0/1 What does the following Matplotlib code snippet do?

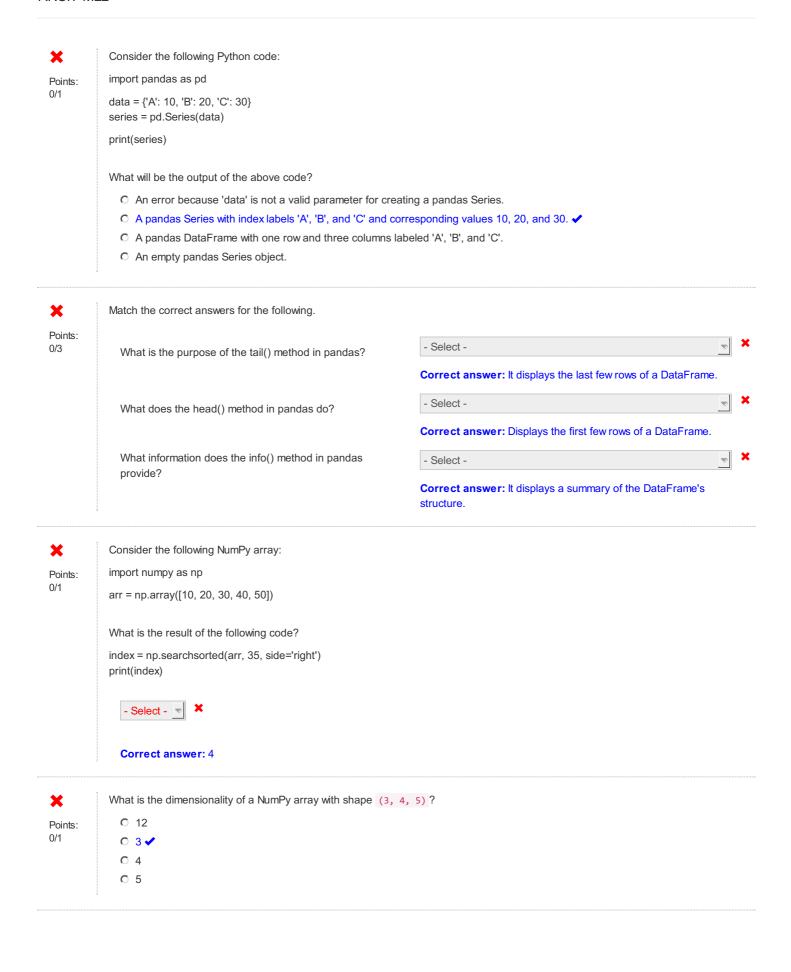
import matplotlib.pyplot as plt

labels = ['A', 'B', 'C', 'D'] sizes = [20, 30, 40, 10]

plt.pie(sizes, labels=labels) plt.title('My Chart') plt.show()



Correct answer: It creates a pie chart.





Points: 0/1

Consider the following Python code: import pandas as pd data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'], 'Age': [25, 30, 35, 40]} df = pd.DataFrame(data) result = df.loc[[0, 2]]print(result) What will be the output of the above code? C Name Alice Age 25 Name Charlie Age 35 Name: 0, dtype: object Name Alice Age 25 Name: 0, dtype: object Name Charlie Age 35 Name: 2, dtype: object Name Alice Age 25 Name Charlie

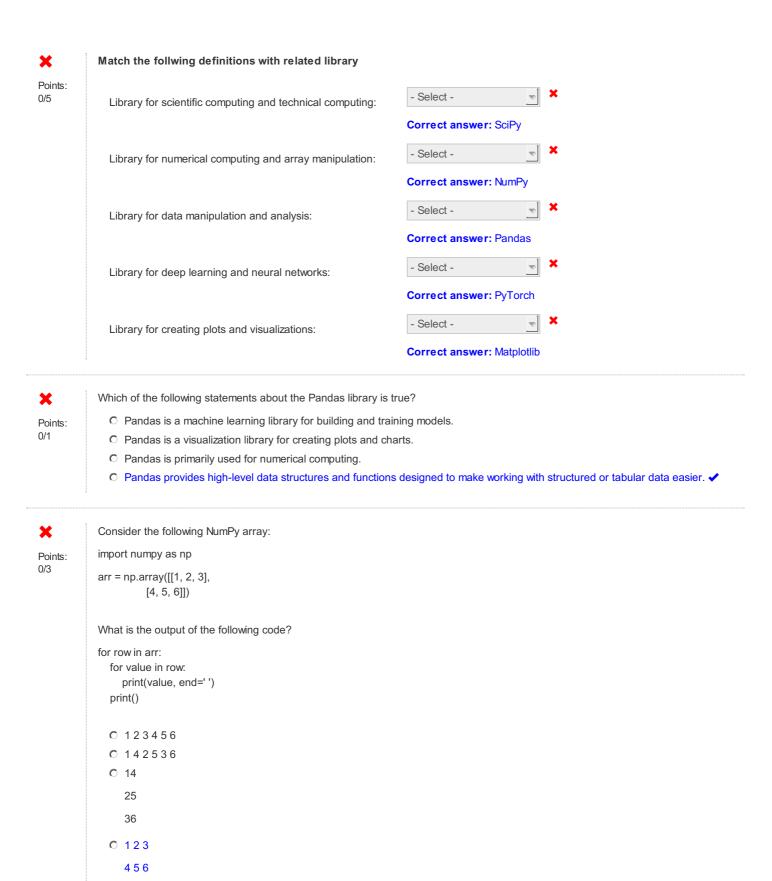


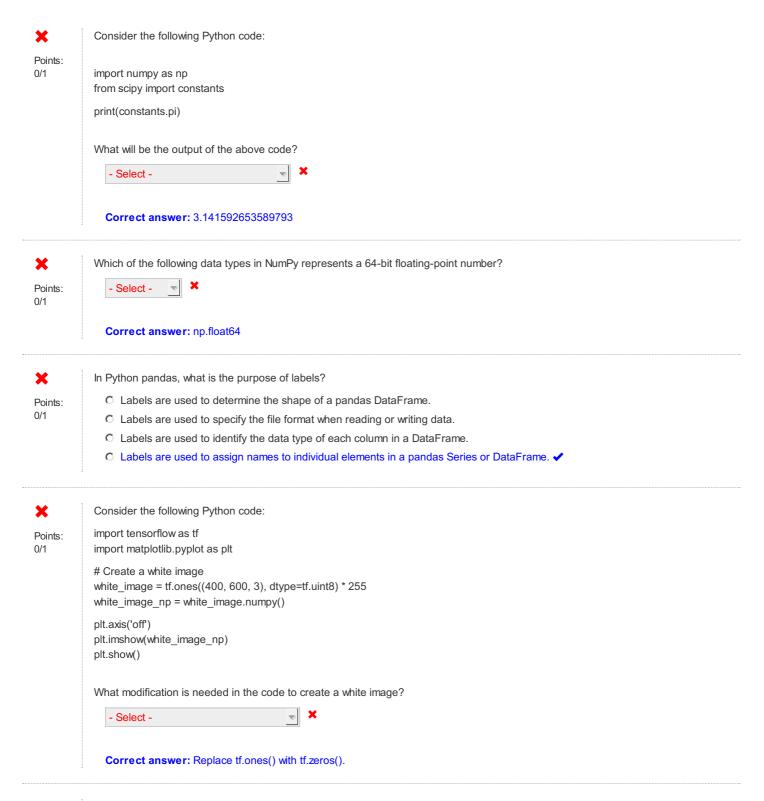
Consider the following NumPy array:

C Error: DataFrame has no attribute 'loc'

Age 35 dtype: object

Points: 0/1





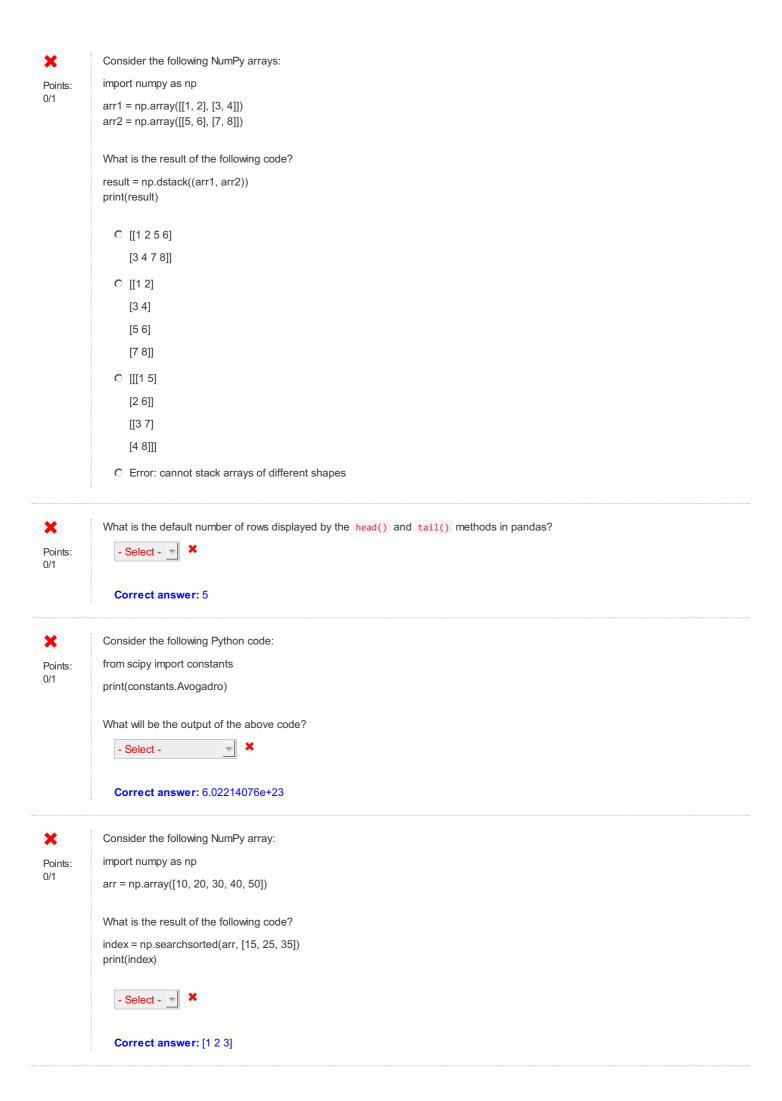
×

What is the difference between a copy and a view of a NumPy array?

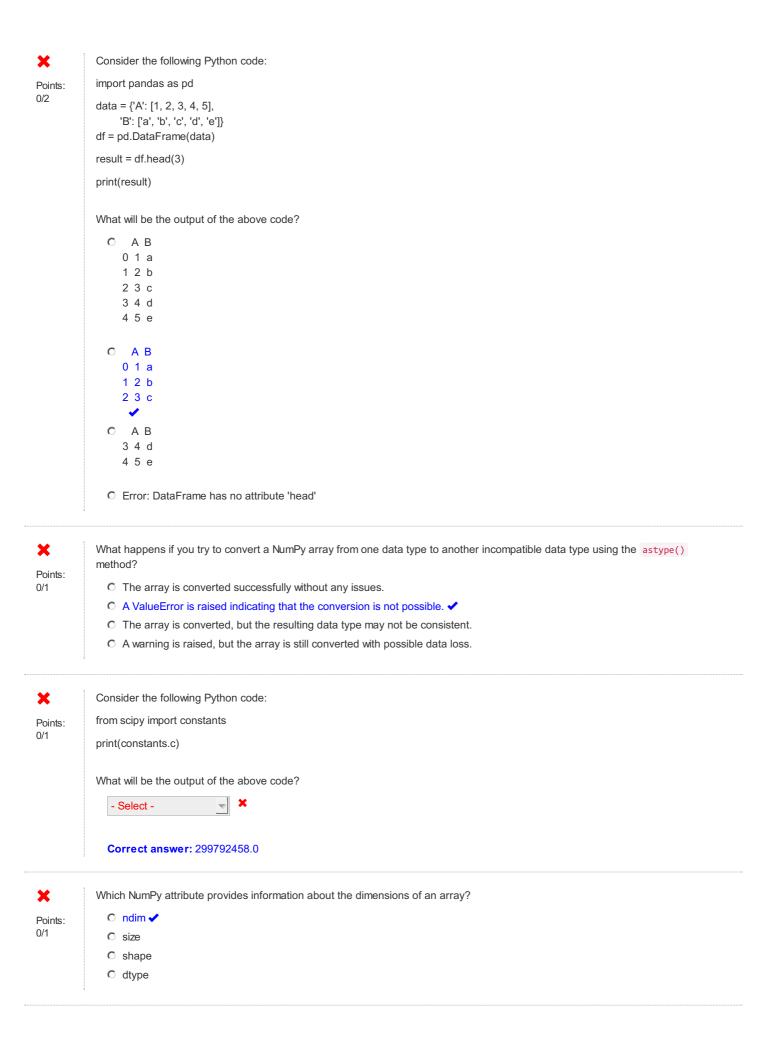
Points: 0/1

- O A copy always has a different shape than the original array, while a view has the same shape.
- O A copy is created using the copy() method, while a view is created using the view() method.
- O A copy shares the same data memory as the original array, while a view has its own data memory.
- Modifying a copy does not affect the original array, while modifying a view affects the original array.

 ✓



```
What is the shape of the following NumPy array?
Points:
0/1
            import numpy as np
            arr = np.array([1, 2, 3, 4, 5])
              O (5,) 🗸
              O (1,)
              C (1, 5)
              C (5, 1)
×
            Which method is used to display the first few rows of a DataFrame in pandas?
              C show()
Points:
              ○ head() 
              C display()
              C preview()
            What is the dimensionality of a NumPy scalar?
              O 0 🗸
Points:
0/1
              O 3
              O 2
              0 1
×
            Consider the following Python code:
            import pandas as pd
Points:
0/1
            data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
                 'Age': [25, 30, 35, 40]}
            df = pd.DataFrame(data)
            result = df.loc[:, 'Name']
            print(result)
            What will be the output of the above code?
              Alice
                 Bob
                 Charlie
                 David
              C Error: DataFrame has no attribute 'loc'
              0
                 2
                 3
              O 0 Alice
                 1 Bob
                 2 Charlie
                 3 David
                 Name: Name, dtype: object
```



```
Consider the following Python code:
            import pandas as pd
Points:
0/1
            data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
                 'Age': [25, 30, 35, 40]}
            df = pd.DataFrame(data)
            result = df.loc[1]
            print(result)
            What will be the output of the above code?
              Name Bob
                 Age 30
                 Name: 1, dtype: object
                  •
              C Bob
              C 30
              Alice
              C Labels are case-sensitive in pandas.
```



Which of the following statements about labels in pandas is true?

Points: 0/1

- C Labels are only used for visualization purposes and have no impact on data manipulation.
- C Labels must always be unique within a pandas DataFrame or Series. ✓
- C Labels are optional and not required for indexing in pandas.

×

Consider the following NumPy arrays:

Points: 0/1

import numpy as np

```
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

```
result = np.hstack((arr1, arr2))
print(result)
```

O [[[1 2]

[3 4]]

[[5 6]

[7 8]]]

C [[1 2]

[3 4]

[5 6]

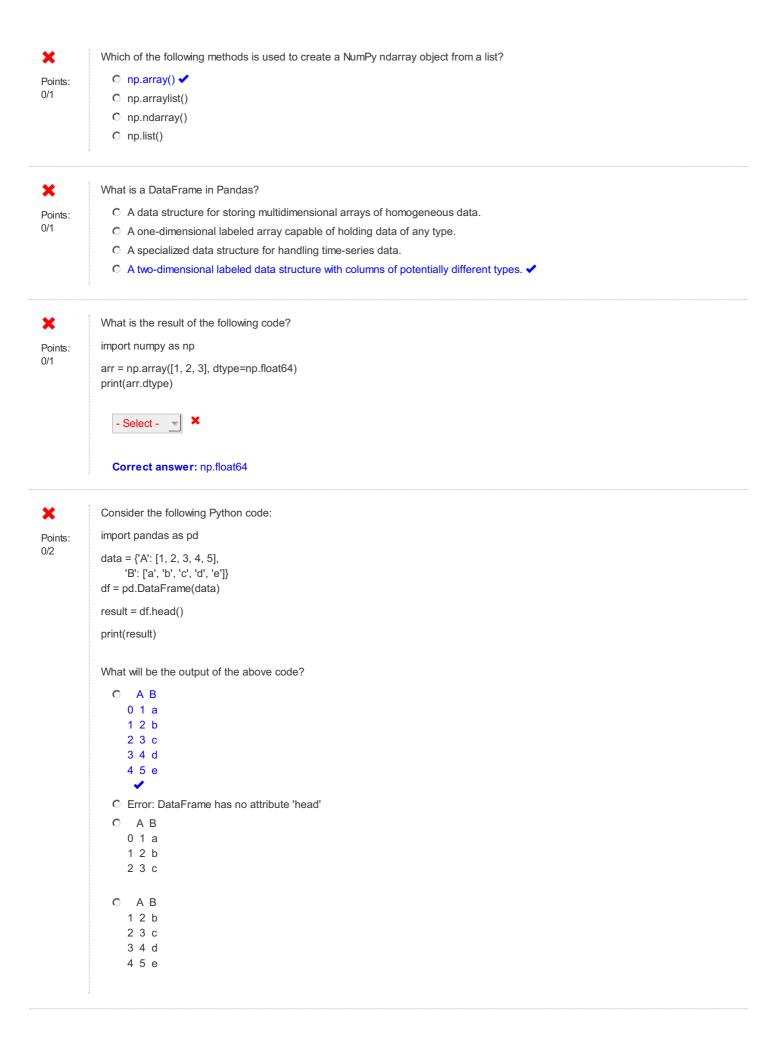
[7 8]]

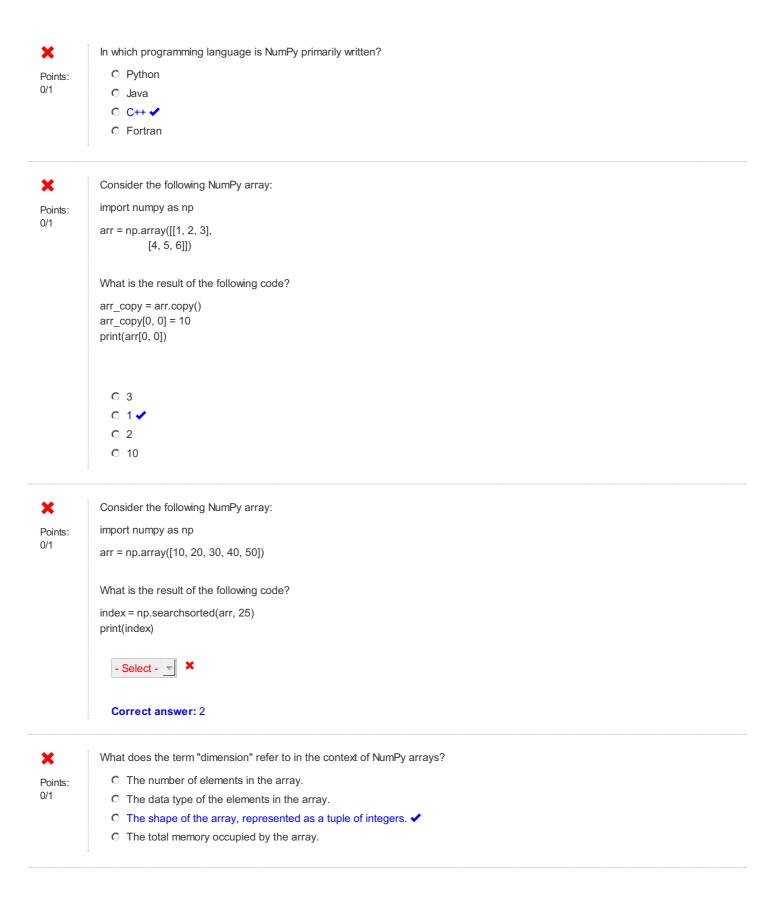
O [[1 2 5 6]

[3 4 7 8]]



C Error: cannot stack arrays of different shapes





```
Consider the following NumPy arrays:
            import numpy as np
Points:
0/1
            arr1 = np.array([[1, 2], [3, 4]])
            arr2 = np.array([[5, 6], [7, 8]])
            What is the result of the following code?
            result = np.stack((arr1, arr2))
            print(result)
              C [[[1 2]
                  [3 4]]
                  [[5 6]
                  [7 8]]]
               C Error: cannot stack arrays of different shapes
               C [[1256]
                  [3 4 7 8]]
              O [[1 2]
                  [3 4]
                  [5 6]
                  [7 8]]
×
            Consider the following Python code:
            from scipy import constants
Points:
0/1
            print(constants.G)
            What will be the output of the above code?
               - Select -
               Correct answer: 6.67430e-11
X
            Consider the following NumPy array:
            import numpy as np
Points:
0/1
            arr = np.array([1.5, 2.7, 3.8])
            What is the data type of the array after converting it to integers using the astype() method?
               - Select - 🔻
               Correct answer: np.int64
×
            Why is NumPy faster than lists for numerical computations in Python?
               O NumPy arrays support dynamic resizing, resulting in faster data manipulation compared to lists.
Points:
0/1
               O NumPy arrays use less memory than lists, resulting in faster computation.
```

O NumPy provides built-in parallel processing capabilities, enabling faster execution of numerical operations.

efficient memory access. 🗸

O NumPy arrays store elements of the same data type in contiguous memory locations, allowing for vectorized operations and



What does a one-dimensional NumPy array represent?

Points: 0/1

- C A row in a two-dimensional array.
- C A matrix in a three-dimensional array.
- C A column in a two-dimensional array.
- A single sequence of elements.



Which of the following statements is true regarding the dimensions of NumPy arrays?

Points: 0/1

- C The dimensions of a NumPy array must be explicitly specified during initialization.
- O NumPy arrays can have an unlimited number of dimensions.
- A scalar in NumPy is considered to have zero dimensions.
- The dimensions of a NumPy array are limited to a maximum of five.



Consider the following NumPy array:

Points: 0/1

```
import numpy as np
arr = np.array([[1, 2, 3, 4],
          [5, 6, 7, 8],
          [9, 10, 11, 12]])
```

What is the result of the following slicing operation?

```
arr_slice = arr[1:, :2]
```

- C array([[1, 2, 3], [5, 6, 7], [9, 10, 11]])
- © array([[2, 3, 4], [6, 7, 8], [10, 11, 12]])
- C array([[1, 2], [5, 6], [9, 10]]) ✓
- C array([[5, 6], [9, 10]])



Consider the following NumPy arrays:

Points: 0/1

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
```

import numpy as np

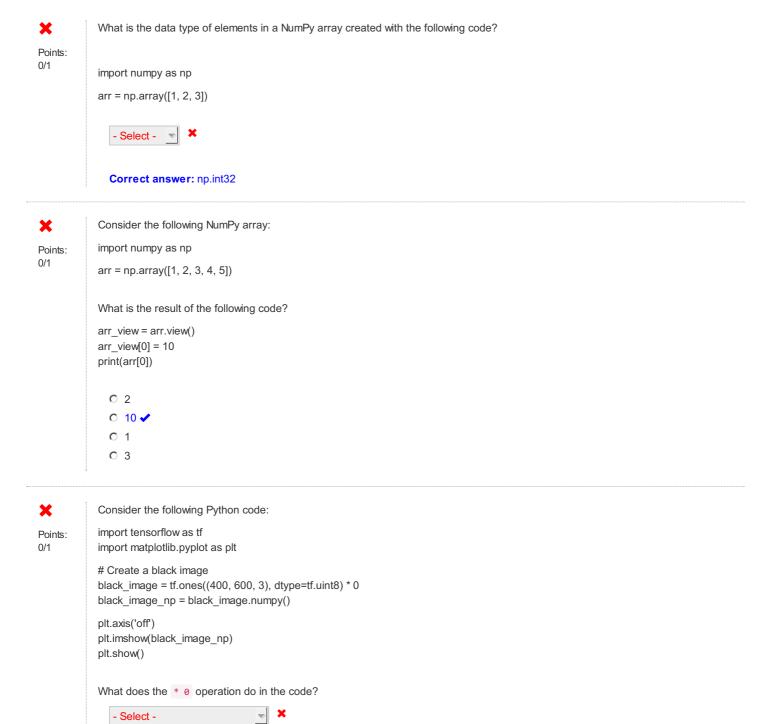
What is the result of the following code?

```
result = np.concatenate((arr1, arr2))
print(result)
```

- C Error: cannot concatenate arrays of different dimensions
- O [123456] <
- O [[1 2 3]

[4 5 6]]

C [[1 2 3 4 5 6]]



Correct answer: Sets all pixel values to 0.

```
×
```

Points: 0/2

```
Consider the following Python code:
```

```
import pandas as pd
```

data = {'A': [1, 2, 3, 4, 5],

'B': ['a', 'b', 'c', 'd', 'e']}

df = pd.DataFrame(data)

result = df.tail()

print(result)

What will be the output of the above code?

- C AB
 - 3 4 d
 - 4 5 e
- О АВ
 - 0 1 a
 - 1 2 b
 - 2 3 c
 - 3 4 d
 - 4 5 e
- C AB
 - 0 1 a
 - 1 2 b
 - 2 3 c
- O AB
 - 1 2 b
 - 2 3 c
 - 3 4 d
 - 4 5 e





Which of the following statements is true regarding creating a NumPy ndarray object from a list?

Points: 0/1

- C The resulting ndarray object has a fixed size and cannot be modified.
- All elements in the list must have the same data type for the conversion to succeed.
- C The resulting ndarray object has a data type of object.
- ${\color{red} \bullet} \quad \text{The np.ndarray() function is preferred over np.array() for creating arrays from lists.}$



What is the output of the following code?

Points: 0/1

import numpy as np

arr = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]]) print(arr.ndim)

- 0
- O 1
- O 2
- O 3 🗸



What is the result of executing the following code?

Points: 0/1

```
import numpy as np

my_list = [1, 2, 3, 4, 5]

my_array = np.array(my_list)

print(my_array.dtype)

C float64
C object
C ndarray
C int64 ✓
```



Consider the following NumPy array:

Points: 0/1

What is the dimensionality of the array arr?

- O 6
- O 2 🗸
- O 3
- 0 1



Consider the following NumPy arrays:

Points: 0/1

```
import numpy as np
```

```
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
```

What is the result of the following code?

result = np.vstack((arr1, arr2)) print(result)

- C [[1 2]
 - [3 4]
 - [5 6]
 - [7 8]]
 - 1
- C [[[1 2]
 - [3 4]]
 - [[5 6]
 - [7 8]]]
- C [[1 2 5 6]
 - [3 4 7 8]]
- C Error: cannot stack arrays of different shapes

×	Use the correct NumPy method to change the shape of an array from 1-D to 2-D. (Select all the correct three options)
Points:	
0/3	arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
	newarr = arr.reshape()
	☐ (1,4)
	□ (4,3) ✓
	□ (2,6) ✓
	□ (3,4) ✓
×	What is the result of executing the following code?
Points: 0/1	
0/1	import numpy as np
	my_tuple = (1, 2, 3, 4, 5)
	my_array = np.array(my_tuple) print(my_array.dtype)
	print(niy_array.utype)
	O ndarray
	C float64
	○ int64 ✔
	C object
×	What is the output of the following code?
	what is the output of the following code:
Points: 0/1	
	import numpy as np
	arr = np.array([1, 2, 3, 4], ndmin=5)
	print('shape of array :', arr.shape)
	O shape of array: (1, 1, 1, 1, 4) ✓
	O shape of array: (1, 1, 2, 2, 4)
	C shape of array: (1, 2, 3, 4, 4)
	C shape of array: (4, 4, 3, 2, 1)
×	Consider the following NumPy array:
Points:	
0/1	import numpy as np
	arr = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
	What is the dimensionality of the array arr?
	O 2
	C 3 ✓
	C 1
	C 4
×	What functionality does SciPy provide?
Points:	☐ Statistical functions. ✓
0/3	☐ Linear algebra operations. ✔
	☐ Optimization algorithms. ✔
	☐ Creating pixel coloring

