

PythonInstitute.PCED-30-02.v2026-07-20.q18

□□□□:	PCED-30-02
□□□□:	PCED - Certified Entry-Level Data Analyst with Python
□□□:	Python Institute
□□ □□ □□□:	18
□□:	v2026-07-20
# □□ □:	133
# □□ □□□:	180
https://www.krdump.com/PythonInstitute.PCED-30-02.v2026-07-20.q18.html	

NEW QUESTION: 1

□ □□□□ □□ □ □□ □□□□ □□□ □□ □□ □□□□ □□□□□.

Club	No. of Students
Chess	25
Drama	40
Robotics	30
Basketball	45
Soccer	25
Guitar	15

□□□ □□□ □□□□ □ □□ □□□ □□ □□□ □□□□□? □□ □□□ □□ □□□ □□.

- A. □ □□□
- B. □□ □□
- C. □□ □□
- D. □□ □□□

Answer: D (LEAVE A REPLY)

□□ □□□□ □ □□□□ □□□ □□ □□ □□ □□ □□ □□□ □□□ □□□□ □ □ □□□□□. □□ □□□□ □ □□□ □□□ □□□ □□□□ □□□□□.

NEW QUESTION: 2

□□ □□□ □□□ □□□ □□ □□ □□□□. □□ □□□ □□ □ □□□ □ □□
□□. □ □□□ □□ □□□ □□□□ □ □□□ □□□□□ □□□□□ □□□. □□□ □
□□ □□□□□.

```
records = [  
    {'name': 'Alice', 'score': 85},  
    {'name': 'Bob', 'score': 72},  
    {'name': 'Alice', 'score': 90},  
    {'name': 'Bob', 'score': 80},  
    {'name': 'Dylan', 'score': 63},  
    {'name': 'Alice', 'score': 95}  
]  
  
totals = {}  
counts = {}  
  
averages = {}  
for name in totals:  
    averages[name] = totals[name] / counts[name]  
  
print(averages)
```

MISSING CODE □□□ □□□□ □□□□ □ □□□ □□ □□□ □□□□ □□□ □□
□□□? □□ □□□ □□ □□□□□.

```
for record in records:  
    name = record['name']  
    score = record['score']  
    totals[name] = totals.get(name, 0) + score  
    counts[name] = counts.get(name, 0) + 1
```

A.

```
for name, score in records.items():  
    totals[name] += score  
    counts[name] += 1
```

B.

```
for record in records:  
    name = record['name']  
    if name not in totals:  
        totals[name] = 0  
        counts[name] = 0  
    totals[name] = score  
    counts[name] += 1
```

C.

```
for record in records:  
    name = record['name']  
    totals[name] = [record['score']]  
    counts[name] = 1
```

D.

Answer: A ([LEAVE A REPLY](#))

□ □□□ □ □□□□ □□□□ □□□ □□□ □□□□, □□□□□ `get()` □□□□ □
 □□□ □□□ □□□ □□□ □□□□ □□□□□. □□ □□□ □□ 0□□ □□□□□□.
 □□ □□ □□ □ □□□ □□□ □□□□ □□□ □ □□□□.

NEW QUESTION: 3

□□□□ [1, 2, 3]□□ □□□□ □□□□. □□□□□□□ `append([4, 5])` □□□□ □□□
 □□. □□ □ □□ □□□□ □□ □□□ □□□?

- A. [1,2,3,4,5]
- B. [1,2,3,[4,5]]
- C. [4,5,1,2,3]
- D. □□

Answer: (SHOW ANSWER)

append() □□□□ □□□ □□ □□□ □□□□□. □□□ □□□ [4, 5]□ □□□ □□ □
 □□□ □□□□ [1, 2, 3, [4, 5]]□ □□□.

NEW QUESTION: 4

□□□□□ □□ □□□ □□□□ □□□ □□ □□□□ □ □□ □□□ □□□□□.

□□□ = [120, 130, 128, 700, 115, 123, 119, 680, 122]

□□□ □ = [12, 15, 13, 50, 11, 14, 10, 55, 13]

□□□ □□□ □□□□:

- □□□(□□□ > 600)□ □□□□□.

- □□□ □□□ □□ □□□□□ □□□□□.

□□ □□ □ □ □□□ □□□□ □□□□ □□□ □□□□□? □□ □□□ □□ □□□□
 □□.

```
import numpy as np

mask = views < 600
v = views[mask]
s = signups[mask]
```

A. corr = np.mean(v, s)

```
import numpy as np
v = np.array(views)
s = np.array(signups)
corr = np.corrcoef(v[s < 600], s)
```

B.

```
import numpy as np

v = np.array(views)
s = np.array(signups)
v = v[v < 600]

corr = np.sum(v, s)
```

C.

```
import numpy as np

v = np.array(views)
s = np.array(signups)
mask = v < 600

corr = np.corrcoef(v[mask], s[mask])
```

D.

Answer: (SHOW ANSWER)

NumPy `np.corrcoef()` function returns the correlation matrix of the input arrays. In this case, we want to calculate the correlation between the `views` and `signups` arrays, but only for values where `views` is less than 600. We can achieve this by creating a mask `mask = v < 600` and then using it to index both arrays: `v[mask]` and `s[mask]`. Finally, we pass these indexed arrays to `np.corrcoef()` to get the correlation coefficient.

NEW QUESTION: 5

Which of the following is a valid NumPy array? (Select all that apply.)

- A. `np.array([1, 2, 3, 4, 5])`
- B. `np.array([1, 2, 3, 4, 5], dtype='float')`
- C. `np.array([1, 2, 3, 4, 5], dtype='int')`
- D. `np.array([1, 2, 3, 4, 5], dtype='str')`

Answer: C (LEAVE A REPLY)

The correct answer is C. The `np.array()` function in NumPy can create arrays of various data types. In this case, the data type is specified as `'int'`, which is a valid integer data type. The other options are invalid because they either do not specify a data type or specify an invalid data type.

NEW QUESTION: 6

records.txt is a text file containing the following data. Which of the following is a valid NumPy array? (Select all that apply.)

- `np.array([1, 2, 3, 4, 5])`
- `np.array([1, 2, 3, 4, 5], dtype='float')`

- □□ □□ □□□ □□□□ □□□□ □□□.

□□ □ □□□□□ □ □□□ □□□□ □□ □□□ □□□ □□□□□? □□ □□□ □□ □□□□□□.

```
import os
PYTHON INSTITUTE

if os.path.exists("records.txt"):
    f = open("records.txt", "r")
    data = f.readlines()
    f.close()
else:
    data = []
```

A.

```
import os
PYTHON INSTITUTE

if os.path.exists("records.txt"):
    f = open("records.txt", "r")
    data = f.readlines()
else:
    data = []
```

B.

```
import os

try:
    f = open("records.txt", "r")
    data = f.readlines()
    f.close()
except FileNotFoundError:
    data = []
```

C.

```
import os
PYTHON INSTITUTE

with open("records.txt", "r") as f:
    data = f.readlines()
except FileNotFoundError:
    data = []
```

D.

Answer: (SHOW ANSWER)

□ □□□ □□□□ □□□□ □□□□ □□□ □□ □ □□□□ □□□ □□, try-except □ □□□ □□□ □□□ □□ □□ □□□ □□□□ □□□□□. □□□ □□ □ □□ □□ □ □□□ □□□□□.

NEW QUESTION: 7

□□ □□□ □□□ □□ 24□□□ □□□ □□□□ □□□□□.

- A.** CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write.
- B.** XML files are easy to read and write. XML files are easy to read and write. XML files are easy to read and write. XML files are easy to read and write.
- C.** JSON files are easy to read and write. JSON files are easy to read and write. JSON files are easy to read and write. JSON files are easy to read and write.
- D.** SQL files are easy to read and write. SQL files are easy to read and write. SQL files are easy to read and write. SQL files are easy to read and write.

Answer: A (LEAVE A REPLY)

CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write. CSV files are easy to read and write.

NEW QUESTION: 10

Which of the following is a valid way to create a NumPy array? (Select all that apply.)

```
colors = [
    "blue", "green", "blue", "red", "green",
    "blue", "yellow", "red", "purple", "green",
    "blue", "orange", "yellow", "blue", "green"
]
```

Which of the following is a valid way to create a NumPy array? (Select all that apply.)

- NumPy array creation functions like `np.array()` and `np.zeros()`.

- Lists of Python objects like strings, integers, and floats.

Which of the following is a valid way to create a NumPy array? (Select all that apply.)

numpy array creation functions like `np.array()` and `np.zeros()`.

A. Lists of Python objects like strings, integers, and floats.

`len(unique(colors))`

`len(colors)`

numpy array creation functions like `np.array()` and `np.zeros()`.

B. Lists of Python objects like strings, integers, and floats.

`len(unique(colors))`

`color_counts = sum(np.unique(colors))`

numpy array creation functions like `np.array()` and `np.zeros()`.

C. Lists of Python objects like strings, integers, and floats.

`np.unique(colors)`

`color_counts = Counter(set(colors))`

numpy array creation functions like `np.array()` and `np.zeros()`.

D. Lists of Python objects like strings, integers, and floats.

`unique_colors = len(set(colors))`

`color_counts = np.unique(colors)`

Answer: (SHOW ANSWER)

numpy.unique() returns the unique elements of an array, sorted in ascending order.
counter(colors) returns a Counter object representing the frequency of each color.
print(counter(colors).most_common(3))

NEW QUESTION: 11

Which of the following code snippets will output True?

A.

```
a = "2"
b = 3
c = a * b
d = type(c) == int
print(d)
print(isinstance(c, float))
```

B.

```
a = False
b = "True"
c = a + b
d = isinstance(c, str)
print(type(a) == str)
print(d)
```

C.

```
a = 4.0
b = True
c = a + b
d = type(c) == bool
print(d)
print(isinstance(a, int))
```

D.

```
a = 2
b = 3.0
c = a * b
d = isinstance(b, float)
print(type(c) == float)
print(d)
```

Answer: (SHOW ANSWER)

The correct answer is D. The code in D will output True because c is a float (2 * 3.0 = 6.0) and d is True (isinstance(3.0, float) is True).

NEW QUESTION: 12

Which of the following code snippets will output 6.0?

NEW QUESTION: 14

□□ data.csv □□□ □ □□ □□ □□□□. f-string□ □□□□ □ □□ □□□ □□□□
□ □□□□ □□□ □□□□ □□ □□□ □□□□ □□□.
□□□□ □□□□ □□□□ □□ □□□□□ □□ □□ □□□□ □□□?

- A. □□ = line.split().strip(',')
- B. □□ = line.strip().split(',')
- C. □□ = line.replace(', ', '|').split('|')
- D. □□ = line.split(',').trim()

Answer: B ([LEAVE A REPLY](#))

□□ □□□ □□ □□ □□□ □□□□, □□-□□ □□ □□□ □□□□ □□ □□□□ □
□ □□ □□□ □□ □□□□ □□□□.

NEW QUESTION: 15

□□ □□□□ □□□□ □:

□□□ = "2024 □□ □□□"

□□ □□ □ □□□ □□ □□ □□□□□? (□ □□ □□□□□□.)

- A. text.capitalize()□ Final□ □□□□ □□□, text.find("L")□ True□ □□□□□.
- B. text[6:8].isdigit()□ True□ □□□□ text.endswith("Final")□ True□ □□□□□.
- C. text.isalpha()□ True□ □□□□ text.find("2024")□ True□ □□□□□.
- D. text[-6].isdigit()□ True□ □□□□, text.startswith("r")□ True□ □□□□□.
- E. text[0:6].isalpha()□ True□ □□□□ text.find("2024")□ 6□ □□□□□.
- F. text.capitalize()□ Report2024Final□ REPORT2024FINAL□ □□□□ text.find("2024")
□ 6□ □□□□□.

Answer: A,B,E ([LEAVE A REPLY](#))

□□□□ □□□[6:8]□ □□□ □□□□ "20"□□ □□□□ "Final"□ □□□□. □□ 6□
□□□ □□ □□□□ "Report"□□ "2024"□ □□□ 6□□ □□□□□.

NEW QUESTION: 16

□□ □□□ □□□ □□ □□□□ □□ □□ □□□□ □□□□□.

□□□ □□□□ □□□ □□□□ □□□ □□ □□□ □□□ □□□ □□□ □□ □□□
□□□□□? □□ □□□ □□□ □□□□□□.

- A. □□□□ □□ □□□ □□ □□ □□□ □□□□ □□□□ □□ □□□ □□□ □ □□
□ □□□.
- B. □□ □□ □□ □ □□□ □□ □□ □□ □□□ □□□ □□□□ □□□□ □□□□ □
□□□□.
- C. □□□□ □□□ □□□ □□ □□□ □□□ □□□□, □□□ □□□□□ □□□□ □
□□ □□□□, □□□ □□□□ □□□□□.
- D. □□ □□□ □□ □□□ □□□□□ □□□ □□ □□□ □□□ □□□ □□□□□□.

Answer: ([SHOW ANSWER](#))

Python is a high-level, interpreted, interactive, object-oriented programming language. It features a dynamic type system and automatic memory management. It is designed to be easy to learn and use, and it has a large and growing standard library.

PCED-30-02 is a Python certification exam. It is a 90-minute exam with 50 questions. The exam is divided into two sections: a multiple-choice section and a short-answer section. The multiple-choice section consists of 40 questions, and the short-answer section consists of 10 questions. The exam is available in English and Spanish.

<https://www.dumptop.com/Python-Institute/PCED-30-02-dump.html> (52 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 17

Python is a high-level, interpreted, interactive, object-oriented programming language. It features a dynamic type system and automatic memory management. It is designed to be easy to learn and use, and it has a large and growing standard library.

- A. ☐
- B. ☐
- C. ☐
- D. ☐

Answer: (SHOW ANSWER)

Python is a high-level, interpreted, interactive, object-oriented programming language. It features a dynamic type system and automatic memory management. It is designed to be easy to learn and use, and it has a large and growing standard library.

NEW QUESTION: 18

Python is a high-level, interpreted, interactive, object-oriented programming language. It features a dynamic type system and automatic memory management. It is designed to be easy to learn and use, and it has a large and growing standard library.

- A. 49
- B. 35
- C. 19
- D. 28

Answer: C (LEAVE A REPLY)

Python is a high-level, interpreted, interactive, object-oriented programming language. It features a dynamic type system and automatic memory management. It is designed to be easy to learn and use, and it has a large and growing standard library.

PCED-30-02 is a Python certification exam. It is a 90-minute exam with 50 questions. The exam is divided into two sections: a multiple-choice section and a short-answer section. The multiple-choice section consists of 40 questions, and the short-answer section consists of 10 questions. The exam is available in English and Spanish.

DumpTop PCED-30-02 □□ □□□ □□□□□□□□ □□□ □□□□□□□. □□□
□ □□□ □□□□ □□ DumpTop PCED-30-02 □□□ □□□□□.

<https://www.dumptop.com/Python-Institute/PCED-30-02-dump.html> (52 Q&As Dumps,

30%OFF Special Discount: **KrDump**)