

PythonInstitute.PCPP-32-101.v2023-12-01.q15

試験コード:	PCPP-32-101
試験名称:	PCPP1 - Certified Professional in Python Programming 1
認定資格:	Python Institute
無料問題数:	15
バージョン:	v2023-12-01
アクセス数:	347
ページビュー数:	150
https://www.jpnpdf.com/PythonInstitute.PCPP-32-101.v2023-12-01.q15-mondaishu.html	

最新問題: 1

次の呼び出しに関する正しい記述を選択してください。

```
r = requests.get('http://localhost:3000')
```

(答えを 2 つ選択してください。)

- A. localhost (コードが実行されるホスト) にデプロイされたサービスに対応します。
- B. タイムアウトが 3000 ミリ秒に設定されているサービスに対応します。
- C. アドレス local.host.com にあるサービスをアドレス指定します。
- D. ポート 3000 でリッスンするサービスをアドレス指定します。

Answer: A,D ([メッセージを残す](#))

説明

これは、localhost (コードが実行されるホスト) にデプロイされたサービスに対応します。localhost はコードが実行されるローカル マシンまたは現在のホストを参照する特別なホスト名であるため、このステートメントは true です。これは、ネットワーク インターフェイスのループバックアドレスである IP アドレス 127.0.0.1 を使用するのと同じです。ホスト名として localhost を使用することにより、呼び出しはクライアントと同じマシンにデプロイされているサービスをアドレス指定します。

これは、ポート 3000 でリッスンするサービスに対応します。

ポート 3000 は URL のホスト名の後のコロンに続く部分であるため、このステートメントは true です。サービスが受信リクエストをリッスンするポート番号を指定します。ポート番号は、ホスト上の特定のプロセスまたはアプリケーションを識別する 16 ビットの整数です。ポート 3000 を使用することにより、呼び出しはそのポートでリッスンしているサービスをアドレス指定します。これは、タイムアウトが 3000 ミリ秒に設定されているサービスに対応します。

タイムアウトは URL の一部ではなく、Python の requests.get () 関数に渡すことができるパラメータであるため、このステートメントは false です。タイムアウトは、サーバーがデータの送信

を中止するまでの待ち時間を指定します。ミリ秒ではなく秒単位で測定されます。timeout=3 を使用すると、呼び出しではタイムアウトが 3000 ミリ秒ではなく 3 秒に設定されます。

これは、アドレス local.host.com にあるサービスに対応します。

local.host.com は localhost と同じではないため、このステートメントは false です。Local.host.com は、local、host、com の 3 つの部分で構成される完全修飾ドメイン名 (FQDN) です。IP アドレスにマッピングするには DNS 解決が必要です。一方、Localhost は、DNS 解決を必要とせず、常に 127.0.0.1 にマップされる特別なホスト名です。ホスト名として localhost を使用すると、呼び出しは local.host.com にあるサービスをアドレス指定しません。

参考文献:

<https://docs.python.org/3/library/requests.html> : <https://en.wikipedia.org/wiki/Localhost> :

[https://en.wikipedia.org/wiki/Port_\(コンピュータネットワーク\)](https://en.wikipedia.org/wiki/Port_(コンピュータネットワーク)) :

<https://en.wikipedia.org/wiki/完全修飾ドメイン名>

最新問題: 2

次のスニペットを分析し、それを最もよく説明するステートメントを選択してください。

```
def f1(*arg, **args):  
    pass
```

- A. 関数パラメータの名前が命名規則に従っていないにもかかわらず、コードは構文的に正しいです。
- B. *arg パラメータは、名前のないパラメータのリストを保持します。
- C. コードに名前のないパラメータのプレースホルダーがありません。
- D. コードは構文的に正しくありません。関数は def f1 (*args, **kwargs) として定義する必要があります。

Answer: B (メッセージを残す)

説明

提供されたコードスニペットは、*args および **kwargs 構文を使用して可変長引数を受け入れる関数 f1 を定義します。*args パラメータを使用すると、任意の数の名前なし引数をタプルとして関数に渡すことができ、**kwargs パラメータを使用すると、任意の数の名前付き引数を辞書として関数に渡すことができます。

したがって、コードを最もよく説明する正しいステートメントは次のとおりです。

*args パラメータは名前のないパラメータのリストを保持し、**kwargs パラメータは名前付きパラメータの辞書を保持します。

最新問題: 3

コード作成のための PEP 8 プログラミング推奨事項に関連する正しい記述を選択してください。(答えを 2 つ選択してください:)

- A. 可読性を高めるには、is not 演算子 (egif spam is not None:) ではなく、not ... is 演算子 (例: not spam is None:) を使用する必要があります。

B. 型を直接比較するのではなく (例: `type(obj)` が `type(i)` である場合)、`ismstanceQ` メソッドを使用してオブジェクト型の比較を行う必要があります (例: `isinstance(obj, int)` :)。

C. PyPy や Cython よりも CPython 実装を優先する方法でコードを作成する必要があります。そしてジトン。

D. 末尾の重要な空白に依存する文字列リテラルは、視覚的に区別できず、特定のエディタで切り取られる可能性があるため、記述しないでください。

Answer: ([解答を表示する](#))

説明

コード作成に関する PEP 8 プログラミングの推奨事項に関連する 2 つの真実の記述は、オプション B とオプション D です。

PEP 8 では、型を直接比較するのではなく、`isinstance()` メソッドを使用してオブジェクト型を比較することが推奨されているため、オプション B は true です。

オプション D は true です。PEP 8 では、末尾の重要な空白文字に依存する文字列リテラルは、視覚的に区別できない可能性があり、特定のエディターでトリミングされる可能性があるため、記述しないことが推奨されています。

最新問題: 4

改行とさまざまなインデント方法を使用した次の例のうち、PEP に準拠しているものはどれですか?

8つのおすすめは?(答えを 2 つ選択してください。)

```
spam = my_function(arg_one, arg_two,
                    arg_three, arg_four)
```

A.

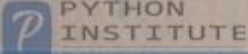
```
eggs = (1, 2, 3,
        4, 5, 6)
```

B.

```
my_list = [
    1, 2, 3,
    4, 5, 6,
```

C.

```
foo = my_function
(
arg_one, arg_two,
arg_three, arg_four
)
```



D.

Answer: ([解答を表示する](#))

説明

正解は B です。選択肢 B と D。選択肢 D。選択肢 B と D は両方とも、改行とインデントに関する PEP 8 の推奨事項に準拠しています。PEP 8 では、インデント レベルごとに 4 つのスペースを使用し、二項演算子の前で改行することを推奨しています。オプション B では、print 関数への引数が開始区切り文字に揃えられます。これは、PEP 8 に従って長い行をフォーマットするもう 1 つの許容可能な方法です。オプション D では、2 行目は次の論理行と区別するために 4 つのスペースでインデントされます。。


最新問題: 5

次のコードの結果は何でしょうか?

```
import logging

logger = logging.getLogger()

logger.info('Debugging mode has been enabled')
logger.debug('Loading data...')
```



次のコードの結果は何でしょうか?

- A. 何も表示されません
- B. データをロード中...
- C. デバッグモードが有効になっています
- D. デバッグ モードが有効になっています データをロードしています...

Answer: ([解答を表示する](#))

説明

コードではロギング モジュールを使用してロガー オブジェクトを作成し、そのレベルを logging.INFO に設定しているため、このステートメントは true です。ロギング モジュールは、プログラムの実行中に発生するイベントを報告する方法を提供します。ログ レベルによって、どのイベントが報告され、どのイベントが無視されるかが決まります。ロギング モジュールは、DEBUG、INFO、WARNING、ERROR、CRITICAL の 5 つの重大度レベルを定義します。レベルが低いほど、より多くのイベントが報告されます。

次に、コードはロガー オブジェクトを使用して 2 つのメッセージを記録します。1 つは logging.DEBUG レベル、もう 1 つは logging.INFO レベルです。ロガー オブジェクトは、自身のレベル以上のレベルを持つメッセージのみを報告します。したがって、logging.DEBUG レベルのメッセージは無視され、logging.INFO レベルのメッセージは報告されます。レポートメッセージのデフォルト形式は「レベル名:メッセージ」です。したがって、コードの出力は次のようになります。

情報: データをロード中...

最新問題: 6

赤と青を混ぜると紫が得られる場合、その 2 つの成分を表すカラーコードは何ですか? 答えを 2 つ選択してください)

- A. #FF0000
- B. #FFFFFF
- C. #0000FF
- D. #000000

Answer: A,C (メッセージを残す)

最新問題: 7

「2 つの変数は同じオブジェクトを参照しますか?」という質問に対して True または False の答えを得るには、どの関数または演算子を使用する必要がありますか?

- A. = 演算子
- B. isinstance() 関数
- C. id()関数
- D. is 演算子

Answer: D (メッセージを残す)

説明

2 つの変数がメモリ内の同じオブジェクトを参照しているかどうかをテストするには、is 演算子を使用する必要があります。

is 演算子は、2 つの変数がメモリ内の同じオブジェクトを指している場合は True を返し、それ以外の場合は False を返します。

例えば :

```
a = [1, 2, 3]
```

```
b = a
```

```
c = [1, 2, 3]
```

```
print(a is b) # True
```

```
print(a is c) # False
```

この例では、a と b はメモリ内の同じリスト オブジェクトを参照しているため、a is b は True を返します。一方、a と c は同じ値を持つ 2 つの別個のリスト オブジェクトを参照するため、a is c は False を返します。

最新問題: 8

構成に関する正しい記述を選択してください

- A. 構成は、新しいコンポーネントを追加したり、既存のコンポーネントを変更したりすることで、クラスの機能を拡張します。
- B. コンポジションにより、クラスを異なるクラスのコンテナとして投影できるようになります。
- C. 合成はコードの再利用性を促進する概念であり、継承はカプセル化を促進します。
- D. has a relationship: に基づいて構成されているため、継承と併用することはできません。

Answer: B ([メッセージを残す](#))

説明

構成は、has-a 関係をモデル化するオブジェクト指向の設計概念です。コンポジションでは、コンポジットとして知られるクラスに、コンポーネントとして知られる別のクラスのオブジェクトが含まれます。言い換えれば、複合クラスには別のクラスのコンポーネントが含まれます¹。

コンポジションを使用すると、クラスをさまざまなクラスのコンテナとして投影できます。

合成は、別のクラスの 1 つ以上のオブジェクトを属性として集約することで、単純なオブジェクトから複雑なオブジェクトを構築できるようにする Python の概念です。集約されたオブジェクトは一般にオブジェクト全体の一部とみなされ、多くの場合、それを含むオブジェクトは小さなオブジェクトのコンテナとして見なされます。

合成では、継承よりも優れた柔軟性と変更可能性を可能にする方法でオブジェクトが結合されます。合成では、コンテナ オブジェクトを使用して他のオブジェクトをホストすることにより、既存のオブジェクトを組み合わせて新しいオブジェクトを作成できます。対照的に、継承では、新しいオブジェクトは親クラスの動作を拡張し、その継承階層によって制限されます。

参考文献:

* Python の公式ドキュメントは次のとおりです。

構成: <https://docs.python.org/3/tutorial/classes.html#composition>

* GeeksforGeeks のコンポジション vs に関する記事

継承: <https://www.geeksforgeeks.org/composition-vs-inheritance-python/>

* 合成に関する実際の Python の記事と

継承: <https://realpython.com/inheritance-composition-python/>

最新問題: 9

PEP 8 の命名規則に関連する正しい記述を選択してください。(答えを 2 つ選択してください。)

- A. 関数名と変数名は小文字にし、単語をアンダースコアで区切る必要があります。
- B. インスタンス メソッドの最初の引数として self を使用し、クラス メソッドの最初の引数として cls を常に使用する必要があります。
- C. モジュールには CamelCase で記述された短い名前が必要です。
- D. 定数はすべて小文字で、単語をアンダースコアで区切って記述する必要があります。

Answer: (解答を表示する)

説明

PEP 8 では関数名と変数名を小文字にし、単語をアンダースコアで区切ることを推奨しているため、オプション A は true です。

PEP 8 では、定数はすべて大文字で単語をアンダースコアで区切って記述することが推奨されているため、オプション D は true です。

PEP 8 は、Python コードの公式スタイル ガイドです。一貫した命名規則に従って読みやすいコードを記述する方法のガイドラインを提供します。PEP 8 の目的は、Python コードの読みやすさを向上させ、理解と保守を容易にすることです。

PEP 8 によれば、A に記載されているように、変数名と関数名は単語をアンダースコアで区切ってすべて小文字で記述する必要があります。定数は、コード全体で値が一定であることが期待される変数であり、すべて小文字で記述する必要があります。D に記載されているように、大文字とアンダースコアで区切られた単語。

参考文献:

* PEP 8 -- Python コードのスタイル ガイド: <https://www.python.org/dev/peps/pep-0008/>

* Python ドキュメント: <https://docs.python.org/3/tutorial/classes.html#classmethods-and-staticmethods>

最新問題: 10

__name__ 属性に関する正しい記述を選択してください。

- A. __name__ はクラスとインスタスの両方に固有の特別な属性であり、クラス インスタスが属するクラスに関する情報が含まれています。
- B. __name__ はクラスとインスタスの両方に固有の特別な属性であり、オブジェクト属性の辞書が含まれています。
- C. __name__ はクラスに固有の特別な属性であり、クラス インスタスが属するクラスに関する情報が含まれています。
- D. __name__ はクラスに固有の特別な属性であり、クラスの名前が含まれます。

Answer: D ([メッセージを残す](#))

説明

__name__ 属性に関する正しいステートメントは D です。name はクラスに固有の特別な属性であり、クラスの名前が含まれます。__name__ 属性は、クラスの名前を文字列として含むクラスの特別な属性です。

__name__ 属性は、すべてのクラスで使用できる Python の特別な属性であり、クラスの名前が文字列として含まれます。__name__ 属性には、ドット表記を使用してクラスとそのインスタスの両方からアクセスできます。

最新問題: 11

json.-dumps () 関数に関する正しいステートメントを選択してください。(答えを 2 つ選択してください。)

- A. JSON 文字列を返します。
- B. Python エンティティを返します。
- C. JSON 文字列を引数として受け取ります
- D. Python データを引数として受け取ります。

Answer: A,D (メッセージを残す)

説明

json.dumps() 関数は、Python オブジェクトを JSON 文字列 1 に変換するために使用されます。この関数は、辞書やリストなどの Python データを引数として受け取り、JSON 文字列を返します。

最新問題: 12

改行と二項演算子に関連する PEP 8 の推奨事項について、正しいものを選択してください。

- A. コードを読みやすくするために、二項演算子の前に改行を使用することをお勧めします。
- B. 規則がローカルで一貫している限り、二項演算子の前後で改行を使用することができます。ただし、新しいコードの場合は、二項演算子の後にのみ改行を使用することをお勧めします。
- C. コードを読みやすくするために、二項演算子の後に改行を使用することをお勧めします。
- D. 二項演算子での改行の使用に関する特定の PEP 8 推奨事項はありません。

Answer: (解答を表示する)

説明

Python の公式スタイル ガイドである PEP 8 によると、特に長い式を含むコード ブロックでは、二項演算子の前に改行を入れると、より読みやすいコードが生成されます。これはいくつかの情報源 (1、2、6、8) で述べられており、広く受け入れられている慣例です。

参考文献:

- * <https://www.python.org/dev/peps/pep-0008/#Should-a-line-break-before-or-after-a-binary-operator>
- * <https://stackoverflow.com/questions/30614124/are-long-lines-broken-up-before-or-after-binary-operators->
- * <https://www.quora.com/What-is-PEP-8-Python>
- * <https://www.techbeamers.com/python-tutorial-pep-8/>
- * <https://www.section.io/engineering-education/python-coding-conventions-guidelines-for-python-programm>
- * <https://towardsdatascience.com/a-step-in-pep8-style-guide-improving-the-readability-of-the-code-8114fd4>
- * <https://www.codementor.io/@rishikeshdhokare/python-coding-style-best-practices-that-every-python-prog>
- * <https://www.dataschool.io/python-pep8-tips-and-tricks/>

最新問題: 13

ソケット オブジェクトは通常、次の呼び出しのどれによって作成されますか?

- A. ソケット。ソケット (ソケットドメイン、ソケットタイプ)
- B. ソケット = ソケット。ソケット (socket_number)
- C. ソケット = ソケット。ソケット (ソケットドメイン、ソケットタイプ、サーバーアドレス)
- D. ソケット = ソケット。ソケット(サーバーアドレス)

Answer: A (メッセージを残す)

説明

ソケットオブジェクトは通常、Pythonのソケットモジュールによって提供されるsocket()コンストラクターを使用して作成されます。正しい呼び出しはsocket.socket(socket_domain, socket_type)です。これにより、指定されたソケットドメインとタイプで新しいソケットオブジェクトが作成されます。

最新問題: 14

Pythonのコメントとドキュメント文字列の次の例を見て、有用でPEP 8推奨事項に準拠しているものを選択してください(最良の回答を2つ選択してください)。A)

A.

```
def area_price(area, price=1.25):  
    """Calculate the area in square meters.  
    Keyword arguments:  
    area -- the land area of the slot  
    price -- price per sq/m (default 1.25)"""  
    ...
```

```
def area_price(area, price=2.25):  
    """Calculate the area in square meters.  
  
    Keyword arguments:  
    area -- the land area of the slot  
    price -- price per sq/m (default 2.25)  
    """  
    ...
```

B.

```
# Example that illustrates creating  
# a two-element list, and printing  
# the list contents to the screen.  
  
my_list = [a, b]  
print(my_list)
```

C.

D.

```
price = price + 1 # Decrement price by one to compensate for loss.
```

Answer: [\(解答を表示する\)](#)

説明

PEP 8の推奨事項によると、最適な2つのオプションはオプションBとオプションDです。オプションBは、すべての行を79文字に制限し、ドキュメント文字列やコメントなどの長いテキストブロックの長さを72文字に制限するというPEP 8の提案に従っています¹。オプションDは、PEPで永続化される適切なドキュメント文字列(別名 docstring)を記述するためのPEP 8の規則に従います。

257. すべてのパブリック モジュール、関数、クラス、メソッドに対して docstring を記述することを推奨しています2。

最新問題: 15

オブジェクト指向プログラミングの意味での型について正しいのはどれでしょうか？

- A. これは、任意のオブジェクトが継承できる最下位の型です。
- B. 複合オブジェクトの列挙を可能にする組み込みメソッドです。
- C. 任意のクラスが継承できる最上位の型です。
- D. クラスのインスタンス化に使用されるオブジェクトです。

Answer: C ([メッセージを残す](#))

説明

Python では、type はすべての新しいスタイルのクラスの基本クラスとして機能する組み込みメタクラスです。int や str などの組み込み型を含む、Python のすべての新しいスタイルのクラスは、型メタクラスのインスタンスであり、それを継承します。

Valid PCPP-32-101 Dumps shared by GoShiken.com for Helping Passing PCPP-32-101 Exam! GoShiken.com now offer the **newest PCPP-32-101 exam dumps**, the GoShiken.com PCPP-32-101 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com PCPP-32-101 dumps with Test Engine here:
<https://www.goshiken.com/Python-Institute/PCPP-32-101-mondaishu.html> (46 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)