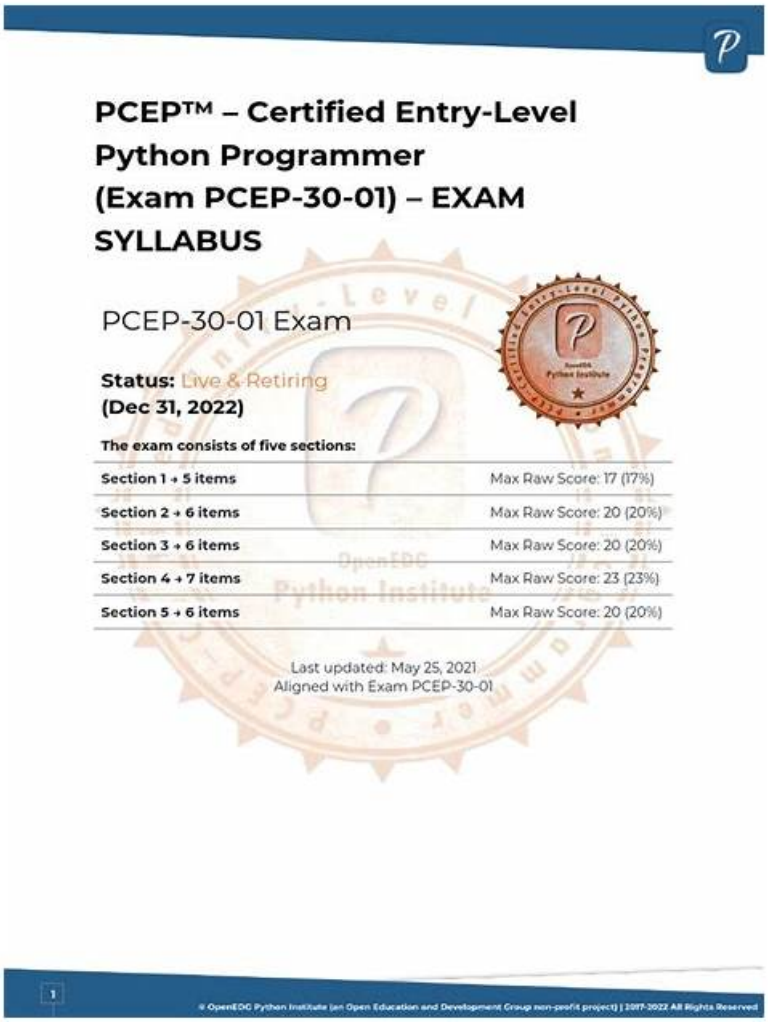
PCEP-30-02試験の準備方法｜認定するPCEP-30-02最新日本語版参考書試験｜素敵なPCEP - Certified Entry-Level Python Programmer試験参考書


PCEP™ – Certified Entry-Level Python Programmer (Exam PCEP-30-01) – EXAM SYLLABUS

PCEP-30-01 Exam

Status: Live & Retiring (Dec 31, 2022)

The exam consists of five sections:

Section 1 • 5 items	Max Raw Score: 17 (17%)
Section 2 • 6 items	Max Raw Score: 20 (20%)
Section 3 • 6 items	Max Raw Score: 20 (20%)
Section 4 • 7 items	Max Raw Score: 23 (23%)
Section 5 • 6 items	Max Raw Score: 20 (20%)

Last updated: May 25, 2021
Aligned with Exam PCEP-30-01

1

© OpenEDG Python Institute (an Open Education and Development Group non-profit project) | 2017-2022 All Rights Reserved

2026年ShikenPASSの最新PCEP-30-02 PDFダンプおよびPCEP-30-02試験エンジンの無料共有：https://drive.google.com/open?id=1e96cG2PR_Y7toNEZQaQUpqgKvvkVHb78

ユーザーのプライバシー保護は、インターネット時代の永遠の問題です。多くの違法ウェブサイトはユーザーのプライバシーを第三者に販売するため、多くの購入者は奇妙なウェブサイトを信じることを嫌います。ただし、PCEP-30-02学習エンジンPCEP-30-02を購入する際に心配する必要はまったくありません。ユーザーの情報が私たちの評判を傷つけているため、ユーザーの情報を決して販売しないことを保証します。

Python Institute試験に参加するのはあなたに自身のレベルを高めさせるだけでなく、あなたがより良く就職し輝かしい未来を持っています。ShikenPASSのPCEP-30-02資料を利用してから、あなたは短い時間でリラクスで試験に合格することができます。我々が存在するのはあなたの成功を全力で助けるためこそです。

>> PCEP-30-02最新日本語版参考書 <<

信頼的なPCEP-30-02試験ツールの保証購入の安全性-PCEP - Certified Entry-Level Python Programmer

当社のPCEP-30-02ガイド急流を購入するすべての顧客情報は、外部に対して機密情報です。当社から漏洩したプライバシー情報について心配する必要はありません。あなたの名前、電子メール、電話番号で連絡できる人

はすべて社内のメンバーです。お客様から提供されたプライバシー情報は、オンラインサポートサービスでのみ使用でき、専門スタッフによるリモートアシスタンスを提供できます。当社の専門家は、毎日PCEP-30-02試験問題の更新を確認し、お客様に常に情報を提供しています。 PCEP-30-02テストガイドについて質問がある場合は、オンラインでメールまたはお問い合わせください。

Python Institute PCEP-30-02 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• データ コレクション: このセクションでは、リストの構築、インデックス作成、スライス、メソッド、および理解に焦点を当て、タプル、辞書、および文字列について説明します。
トピック 2	• ループ: while、for、range()、ループ制御、ループのネスト。
トピック 3	• 関数と例外：この試験の部分では、関数と呼び出しの定義をカバーします。
トピック 4	• パラメータ、引数、スコープ。また、再帰、例外階層、例外処理などもカバーしています。

Python Institute PCEP - Certified Entry-Level Python Programmer 認定 PCEP-30-02 試験問題 (Q33-Q38):

質問 # 33
Which of the following are the names of Python passing argument styles?
(Select two answers.)

- A. positional
- B. reference
- C. keyword
- D. indicatory

正解：A、C

解説：
Keyword arguments are arguments that are specified by using the name of the parameter, followed by an equal sign and the value of the argument. For example, print (sep='-', end='!') is a function call with keyword arguments. Keyword arguments can be used to pass arguments in any order, and to provide default values for some arguments1.
Positional arguments are arguments that are passed in the same order as the parameters of the function definition. For example, print ('Hello', 'World') is a function call with positional arguments. Positional arguments must be passed before any keyword arguments, and they must match the number and type of the parameters of the function2.
References: 1: 5 Types of Arguments in Python Function Definitions | Built In 2: python - What's the pythonic way to pass arguments between functions ...

質問 # 34
What is true about exceptions and debugging? (Select two answers.)

- A. If some Python code is executed without errors, this proves that there are no errors in it.
- B. The default (anonymous) except branch cannot be the last branch in the try-except block.
- C. A tool that allows you to precisely trace program execution is called a debugger.
- D. One try-except block may contain more than one except branch.

正解：C、D

解説：
Explanation
Exceptions and debugging are two important concepts in Python programming that are related to handling and preventing errors. Exceptions are errors that occur when the code cannot be executed properly, such as syntax errors, type errors, index errors, etc.

Debugging is the process of finding and fixing errors in the code, using various tools and techniques. Some of the facts about exceptions and debugging are:
A tool that allows you to precisely trace program execution is called a debugger. A debugger is a program that can run another program step by step, inspect the values of variables, set breakpoints, evaluate expressions, etc. A debugger can help you find the source and cause of an error, and test possible solutions. Python has a built-in debugger module called pdb, which can be used from the command line or within the code. There are also other third-party debuggers available for Python, such as PyCharm, Visual Studio Code, etc12 If some Python code is executed without errors, this does not prove that there are no errors in it. It only means that the code did not encounter any exceptions that would stop the execution. However, the code may still have logical errors, which are errors that cause the code to produce incorrect or unexpected results. For example, if you write a function that is supposed to calculate the area of a circle, but you use the wrong formula, the code may run without errors, but it will give you the wrong answer. Logical errors are harder to detect and debug than syntax or runtime errors, because they do not generate any error messages. You have to test the code with different inputs and outputs, and compare them with the expected results34 One try-except block may contain more than one except branch. A try-except block is a way of handling exceptions in Python, by using the keywords try and except. The try block contains the code that may raise an exception, and the except block contains the code that will execute if an exception occurs. You can have multiple except blocks for different types of exceptions, or for different actions to take. For example, you can write a try-except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except ZeroDivisionError: # handle the ZeroDivisionError exception except: # handle any other exception This way, you can customize the error handling for different situations, and provide more informative messages or alternative solutions5 The default (anonymous) except branch can be the last branch in the try-except block. The default except branch is the one that does not specify any exception type, and it will catch any exception that is not handled by the previous except branches. The default except branch can be the last branch in the try-except block, but it cannot be the first or the only branch. For example, you can write a try-except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except: # handle any other exception This is a valid try-except block, and the default except branch will be the last branch. However, you cannot write a try-except block like this:
try: # some code that may raise an exception except: # handle any exception This is an invalid try-except block, because the default except branch is the only branch, and it will catch all exceptions, even those that are not errors, such as KeyboardInterrupt or SystemExit. This is considered a bad practice, because it may hide or ignore important exceptions that should be handled differently or propagated further. Therefore, you should always specify the exception types that you want to handle, and use the default except branch only as a last resort5 Therefore, the correct answers are A. A tool that allows you to precisely trace program execution is called a debugger. and C. One try-except block may contain more than one except branch.

質問 # 35
What is the expected result of the following code?

- A. 0
- B. The code will cause an unhandled
- C. 1
- D. 2

正解：B

解説：
The code snippet that you have sent is trying to use a list comprehension to create a new list from an existing list. The code is as follows:
my_list = [1, 2, 3, 4, 5] new_list = [x for x in my_list if x > 5]
The code starts with creating a list called "my_list" that contains the numbers 1, 2, 3, 4, and 5. Then, it tries to create a new list called "new_list" by using a list comprehension. A list comprehension is a concise way of creating a new list from an existing list by applying some expression or condition to each element. The syntax of a list comprehension is:
new_list = [expression for element in old_list if condition]
The expression is the value that will be added to the new list, which can be the same as the element or a modified version of it. The element is the variable that takes each value from the old list. The condition is an optional filter that determines which elements will be included in the new list. For example, the following list comprehension creates a new list that contains the squares of the even numbers from the old list:
old_list = [1, 2, 3, 4, 5, 6] new_list = [x ** 2 for x in old_list if x % 2 == 0] new_list = [4, 16, 36] The code that you have sent is trying to create a new list that contains the elements from the old list that are greater than 5. However, there is a problem with this code. The problem is that none of the elements in the old list are greater than 5, so the condition is always false. This means that the new list will be empty, and the expression will never be evaluated. However, the expression is not valid, because it uses the variable x without defining it. This will cause a NameError exception, which is an error that occurs when a variable name is not found in the current scope. The code does not handle the exception, and therefore it will terminate with an error message.

The expected result of the code is an unhandled exception, because the code tries to use an undefined variable in an expression that is never executed. Therefore, the correct answer is D. The code will cause an unhandled exception.
Reference: Python - List Comprehension - W3SchoolsPython - List Comprehension - GeeksforGeeksPython Exceptions: An Introduction - Real Python

質問 # 36
Arrange the binary numeric operators in the order which reflects their priorities, where the top-most position has the highest priority and the bottom-most position has the lowest priority.

正解：

解説：
Explanation
The correct order of the binary numeric operators in Python according to their priorities is:
Exponentiation (**)
Multiplication (*) and Division (
Addition (+) and Subtraction (
This order follows the standard mathematical convention of operator precedence, which can be remembered by the acronym PEMDAS (Parentheses, Exponents, Multiplication/Division, Addition/Subtraction). Operators with higher precedence are evaluated before those with lower precedence, but operators with the same precedence are evaluated from left to right. Parentheses can be used to change the order of evaluation by grouping expressions.
For example, in the expression 2 + 3 * 4 ** 2, the exponentiation operator (**) has the highest priority, so it is evaluated first, resulting in 2 + 3 * 16. Then, the multiplication operator (*) has the next highest priority, so it is evaluated next, resulting in 2 + 48. Finally, the addition operator (+) has the lowest priority, so it is evaluated last, resulting in 50.
You can find more information about the operator precedence in Python in the following references:
6. Expressions - Python 3.11.5 documentation
Precedence and Associativity of Operators in Python - Programiz
Python Operator Priority or Precedence Examples Tutorial

質問 # 37
Python Is an example of which programming language category?

- A. interpreted
- B. machine
- C. assembly
- D. compiled

正解：A

解説：
Explanation
Python is an interpreted programming language, which means that the source code is translated into executable code by an interpreter at runtime, rather than by a compiler beforehand. Interpreted languages are more flexible and portable than compiled languages, but they are also slower and less efficient. Assembly and machine languages are low-level languages that are directly executed by the hardware, while compiled languages are high-level languages that are translated into machine code by a compiler before execution.

質問 # 38
.....

- 有効的なPCEP-30-02最新日本語版参考書 - 合格スムーズPCEP-30-02試験参考書 | 一番優秀なPCEP-30-02ブロンズ教材 PCEP - Certified Entry-Level Python Programmer □ ➠ www.mogiexam.com □に移動し、[PCEP-30-02]を検索して無料でダウンロードしてくださいPCEP-30-02学習関連題
- PCEP-30-02試験の準備方法丨便利なPCEP-30-02最新日本語版参考書試験丨更新するPCEP - Certified Entry-Level Python Programmer試験参考書 □ ▸ www.goshiken.com ◂で使える無料オンライン版➽ PCEP-30-02 □ の試験問題PCEP-30-02復習範囲
- 試験の準備方法-最新なPCEP-30-02最新日本語版参考書試験-効率的なPCEP-30-02試験参考書 □ ⇛ www.mogiexam.com ⇚は、➥ PCEP-30-02 □を無料でダウンロードするのに最適なサイトですPCEP-30-02試験勉強過去問
- PCEP-30-02入門知識 □ PCEP-30-02日本語参考 □ PCEP-30-02専門知識訓練 □ “ PCEP-30-02 ”の試験問題は □ www.goshiken.com □で無料配信中PCEP-30-02日本語版試験勉強法
- PCEP-30-02学習関連題 □ PCEP-30-02入門知識 □ PCEP-30-02日本語参考 □ 「 jp.fast2test.com 」で【 PCEP-30-02 】を検索して、無料で簡単にダウンロードできますPCEP-30-02日本語版問題集
- PCEP-30-02トレーリングサンプル ↖ PCEP-30-02復習資料 □ PCEP-30-02日本語版問題集 □ 「 www.goshiken.com 」で▷ PCEP-30-02 ◁を検索し、無料でダウンロードしてくださいPCEP-30-02資格模擬
- PCEP-30-02テストトレーニング □ PCEP-30-02科目対策 □ PCEP-30-02模擬問題 □ ☀ www.xhs1991.com □☀□は、▸ PCEP-30-02 ◂を無料でダウンロードするのに最適なサイトですPCEP-30-02資格取得
- Python Institute PCEP-30-02 Exam | PCEP-30-02最新日本語版参考書 - 有効な評判の良いウェブサイトPCEP-30-02試験参考書 □ ➠ www.goshiken.com □にて限定無料の【 PCEP-30-02 】問題集をダウンロードせよPCEP-30-02学習関連題
- PCEP-30-02資格取得 □ PCEP-30-02日本語参考 □ PCEP-30-02日本語問題集 □ URL ➽ www.mogiexam.com □をコピーして開き、{ PCEP-30-02 }を検索して無料でダウンロードしてくださいPCEP-30-02資格トレーリング
- 一番優秀なPCEP-30-02最新日本語版参考書試験-試験の準備方法-素晴らしいPCEP-30-02試験参考書 □ 《 www.goshiken.com 》で使える無料オンライン版➤ PCEP-30-02 □ の試験問題PCEP-30-02科目対策
- 試験の準備方法-素晴らしいPCEP-30-02最新日本語版参考書試験-最高のPCEP-30-02試験参考書 □ ☀ www.jpshiken.com □☀□で✔ PCEP-30-02 □✔□を検索し、無料でダウンロードしてくださいPCEP-30-02資格トレーリング
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, Disposable vapes

2026年ShikenPASSの最新PCEP-30-02 PDFダンプおよびPCEP-30-02試験エンジンの無料共有：https://drive.google.com/open?id=1e96cG2PR_Y7toNEZQaQUpqgKvvkVHb78