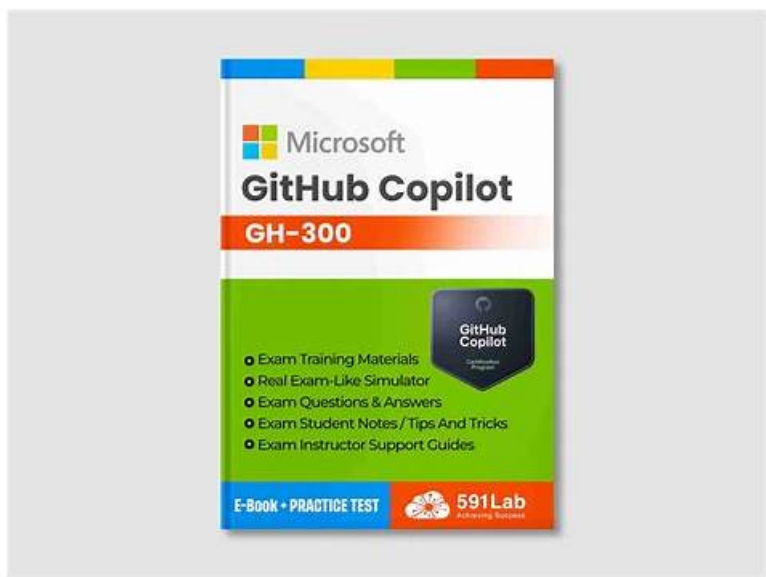


GH-300サンプル問題集、GH-300日本語的中対策



P.S. TopexamがGoogle Driveで共有している無料かつ新しいGH-300ダンプ：<https://drive.google.com/open?id=1FqKF13P4-Rtezo0n5ZHjtz6LsWmAEXJp>

ほとんどの労働者の基準はますます高くなっています。GH-300ガイドの質問にも高い目標を設定しました。私たちのトレーニング資料は、顧客の関心を他のポイントよりも前面に置き、高度なGH-300学習資料に常に取り組んでいます。これまで、最も複雑なGH-300ガイドの質問を簡素化し、簡単な操作システムを設計しました。GH-300試験問題の自然でシームレスなユーザーインターフェイスは、より流暢に成長しました。使いやすさ。

Microsoft GH-300 認定試験の出題範囲：

トピック	出題範囲
トピック 1	ドメイン4: プロンプト作成とプロンプトエンジニアリング このセクションでは、ソフトウェア開発者とAIインタラクショナルデザイナーが、Copilotの出力を最適化するためのプロンプトを効果的に作成するスキルを評価します。プロンプトの構成要素、プロンプトにおける言語の役割、ゼロショット・プロンプトと数ショット・プロンプトの違い、チャット履歴が応答に与える影響といった基礎概念を概説します。プロンプトの設計とトレーニング方法に関するベストプラクティスとエンジニアリング原則についても解説します。
トピック 2	責任: このセクションでは、AI倫理担当者およびリスク管理者のスキルを評価し、AI技術の責任ある倫理的使用について考察します。学習データにおけるバイアスやAI出力の検証の必要性など、生成型AIツールに伴うリスクと限界について解説します。受験者は、バイアス、公平性、プライバシーへの懸念といった潜在的な危害を特定し、倫理的AI原則を適用することでこれらの危害を軽減することで、AIを責任を持って運用する方法を学びます。
トピック 3	GitHub Copilot の仕組みとデータの処理: 機械学習エンジニアとデータプライバシーベシヤリスト向けに設計されたこのセクションでは、Copilot のコードサジェストの背後にあるデータのライフサイクルと処理について説明します。コンテキストの収集、プロンプトの構築、レスポンスの生成、そしてプロキシサービスによる後処理の仕組みを説明します。受験者は、Copilot のデータポリシー、入力データの処理、そしてコンテキストウィンドウのサイズやデータの古さといったサジェストの関連性に影響を与える制限事項を理解します。

トピック 4	GitHub Copilot のプランと機能: このドメインは、プロダクトマネージャーとDevOpsエンジニアを対象としており、Individual、Business、Enterprise といった様々な GitHub Copilot サブスクリプションプランの違いや管理機能について理解することに重点を置いています。Copilot を IDE に統合する方法、コードサジェストの様々なトリガー方法、組織ポリシーの管理、API 経由のサブスクリプション管理、Copilot Chat とナレッジベースの効果的な活用方法についても解説します。また、CLI の使用方法と設定についても学習します。
トピック 5	ドメイン6: GitHub Copilotを使用したテスト: このセクションでは、QAエンジニアとテスト自動化スペシャリストがCopilotを使用して単体テストと統合テストを含むテストを生成する能力を測定します。Copilotがどのようにエッジケースを識別し、アサーションの記述を支援するかを説明します。また、CopilotのサブスクリプションSKUの種類、プライバシーに関する考慮事項、組織におけるコードサジェストの設定、Copilotに関連する設定ファイルについても説明します。
トピック 6	プライバシーの基礎とコンテキスト除外: このドメインはセキュリティエンジニアとコンプライアンス担当者を対象としており、Copilotのテスト提案とセキュリティ最適化を用いたコード品質の向上に焦点を当てています。セキュリティ脆弱性の特定、パフォーマンス向上、リポジトリレベルおよび組織レベルでのコンテンツ除外などのプライバシー機能とその制限事項について解説します。受講者は、重複検出、契約上の保護、セキュリティチェックなどの安全対策メカニズム、そしてコンテキスト除外や提案のギャップといった Copilot の一般的な問題に対するトラブルシューティングガイドについて学習します。

>> GH-300サンプル問題集 <<

GH-300日本語的中対策、GH-300最新対策問題

人々は異なる目標がありますが、我々はあなたにMicrosoftのGH-300試験に合格させるという同じ目標があります。この目標を達成するのは、あなたにとってIT分野での第一歩ですが、我々のMicrosoftのGH-300ソフトを開発するすべての意義です。だから、我々は尽力して我々の問題集を多くしてTopexamの専門かたちに研究させてあなたの合格する可能性を増大します。あなたの利用するMicrosoftのGH-300ソフトが最新版のを保証するために、一年間の無料更新を提供します。

Microsoft GitHub Copilot 認定 GH-300 試験問題 (Q36-Q41):

質問 #36

What is zero-shot prompting?

- A. Giving GitHub Copilot examples of the algorithm and outcome you want to use
- B. Telling GitHub Copilot it needs to show only the correct answer
- C. Giving as little context to GitHub Copilot as possible
- D. Only giving GitHub Copilot a question as a prompt and no examples
- E. Giving GitHub Copilot examples of the problem you want to solve

正解: D

解説:

"Zero-shot prompting means asking the model to perform a task without providing examples-only the question or task description is given." This confirms that the correct definition of zero-shot prompting is providing no examples, only a direct prompt.

References: GitHub Copilot and AI prompting documentation.

質問 #37

What types of prompts or code snippets might be flagged by the GitHub Copilot toxicity filter? (Each correct answer presents part of the solution. Choose two.)

- A. Code that contains logical errors or produces unexpected results
- B. Hate speech or discriminatory language (e.g., racial slurs, offensive stereotypes)

- C. Code comments containing strong opinions or criticisms
- D. Sexually suggestive or explicit content

正解: B、D

解説:

GitHub Copilot includes a toxicity filter to prevent the generation of harmful or inappropriate content. This filter flags prompts or code snippets that contain hate speech, discriminatory language, or sexually suggestive or explicit content. This ensures a safe and respectful coding environment.

質問 # 38

Are there any limitations to consider when using GitHub Copilot for code refactoring?

- A. GitHub Copilot understands the context of your entire project and refactors code accordingly.
- B. GitHub Copilot can only be used with a limited set of programming languages.
- C. GitHub Copilot always produces bug-free code during refactoring.
- D. GitHub Copilot may not always produce optimized or best-practice code for refactoring.

正解: D

解説:

"While Copilot can suggest refactoring changes, the code may not always follow best practices or be fully optimized. Developers must review and validate suggestions." This confirms that limitations exist in optimization and best practices, making option A correct. References: GitHub Copilot limitations documentation.

質問 # 39

What configuration needs to be set to get help from Microsoft and GitHub protecting against IP infringement while using GitHub Copilot?

- A. You need to check code suggestions yourself before accepting
- B. Suggestions matching public code to 'blocked'
- C. Enforce blocking of MIT or GPL licensed code
- D. Enable GitHub Copilot license checking

正解: B

解説:

To help protect against IP infringement, you need to configure GitHub Copilot to block suggestions that match public code. This ensures that the generated code is not directly copied from publicly available sources. Reference: GitHub Copilot documentation on IP protection and code filtering.

質問 # 40

How does GitHub Copilot suggest code optimizations for improved performance?

- A. By providing detailed reports on the performance of the codebase.
- B. By automatically rewriting the codebase to use more efficient code.
- C. By analyzing the codebase and suggesting more efficient algorithms or data structures.
- D. By enforcing strict coding standards that ensure optimal performance.

正解: C

解説:

GitHub Copilot suggests code optimizations by analyzing the codebase and recommending more efficient algorithms or data structures.

質問 # 41

.....

GH-300日本語の中対策: https://www.topexam.jp/GH-300_shiken.html

- P.S. TopexamaがGoogle Driveで共有している無料かつ新しいGH-300ダンプ: <https://drive.google.com/open?id=1FqKF13P4-Rtezo0n5ZHjtz6LsWmAEXJp>