表示名変更のお願い

- **表示名の変更にご協力ください◎**
- **「ローマ字氏名 / 漢字氏名」という形式でお願いします。**
- **表示名の例**
    - Taro Yamada / 山田 太郎

09:00からスタートします◎

講師紹介

講師紹介

人的資本インテリジェンス部門
講師・生成AIエバンジェリスト

安田 光喜
Mitsuki Yasuda

＜プロフィール＞
公立はこだて未来大学 大学院（メディアデザイン領域）を卒業し、街づくり会社での複合商業ビルの立ち上げにおける情報インフラ整備やデザイン全般業務の責任として関与。その後デザイン事務所を立ち上げ、飲食店を中心としたデザイン業務や美術館展示のアートコンテンツ開発などを行う。また、地域ブランディングにも注力し、市主催の企画運営業務を行なった。2018年10月『小中学生向けのプログラミング教育』に注目し、函館市で初めての小中学生向けプログラミングスクール『D-SCHOOL北海道』を立ち上げて運営。その後北海道のドラッグストア『サッポロドラッグストアー』からスクールの買収を受け、教育を専門とするグループ会社『株式会社シーラクンス』にて技術開発責任者として運営。スクール設計・運営、20以上の自治体連携（イベントや出張教室運営）、大人向けプログラミングスクールメンターなどさまざまな『次世代T教育』に取り組む。
Dify公式資料の日本語翻訳や、SecondMeアンバサダーなど生成AI最新トレンドに注力。

得意分野／経歴

・生成AIをはじめとした世界最先端トレンド
・生成AIを用いた新規事業の開発、運営、補助
・アプリケーション開発/自動化プログラム環境構築
・教育（大学講師、研修、スクール運営など）

研修運営／講演実績

・生成AI研修
・大手企業、外資、教育機関(小中高大）でのITトレンド講演
・自治体と連携した地域教育実績
・上場企業社員対象のリスキリング支援

その他

みなさんがこれから生成AIをパートナーに新しい時代を生きていけるように、2日間全力でサポートしていきます！
なにかご不明点などありましたら、講師陣になんでも気軽にご質問ください。
#ClaudeCode #Codex #Cursor #Antigravity
#全国統一生成AI活用技能試験S1　#生成AIパスポート　#G検定

講師紹介

サブ講師

濵田 龍之介

Ryunosuke Hamada

＜プロフィール＞
大学卒業後、アパレル企業にて店舗運営に従事。販売戦略から売上管理、スタッフの育成・マネジメントまでを担い、顧客起点でのビジネスとチームビルディングの基礎を築く。
その後IT業界へ転身し、国内SIerにて金融システムのサポート業務、総合商社にて社内ITヘルプデスクの立ち上げとメンバー育成を主導。
大手外資クラウドベンダーにて、カスタマーサービスとして顧客の課題解決に貢献。同チームのマネージャーを経て、サポート部門全体の人材開発を担うトレーニングチームへ異動。各種育成プログラムのプロジェクトマネージャーとして、組織的なスキルアップを推進。
2025年12月よりギブリーに参画。

得意言語／多言語経験

【領域】講師育成、研修カリキュラム開発、研修運営、業務改善

【言語】JavaScript

経験／実績など

【研修実績】
・大手外資クラウドベンダー(内製)
新卒研修運営、中途社員研修企画運営、リスキリング研修企画運営、Train The Trainer 研修講師、英語学習プログラム企画運営、マネージャー向け研修企画運営、社内認定資格運営

・2026年度 大手通信系SIer サブ講師

講師紹介

サブ講師

河合 謙一郎
Kenichiro Kawai

＜プロフィール＞
病院勤務（診療報酬請求・診療情報管理）を経てITエンジニアへ転身。
Java/Spring Bootを中心に要件定義から設計・実装・運用まで一貫して基幹システム開発に従事し、実務エンジニアとしての経験を積む。
その後研修インストラクター兼コンテンツクリエイターへ転向。自らの開発経験を軸にゼロからカリキュラム設計・教材開発を一手に担い、3年で250名超のエンジニア育成を実現。
並行して業務改善スペシャリストとしてGemini API×n8nを活用したQ&A自動化など、AI駆動の業務改善プロジェクトを主導し、定量的な成果を複数達成。

得意言語／多言語経験

【言語】Java, Python, JavaScript

【FW/ツール】Spring, React, n8n, Claude Code

【領域】講師・研修設計・カリキュラム開発, AI業務自動化

経験／実績など

【研修実績】
・研修インストラクター3年
3年で250名超のエンジニア育成、AI/Java/n8n eラーニングコース複数開発
・2026年度 大手メーカー系SIer メイン講師

【資格取得支援経験】
ITパスポート・基本情報技術者・AWS Certified Cloud Practitioner・AWS Certified Solutions Architect – Associate

講師紹介

サブ講師

河野 智

Satoru Kono

＜プロフィール＞
システム開発業務に約20年従事し、現在も現役エンジニアとして開発業務に携わる一方、IT分野の講師として主に新入社員研修を担当しています。
開発現場と教育現場の双方を経験していることを強みとし、実務に直結する知識や現場視点の実践的なアドバイスを講義に取り入れています。
【主な開発業務】
RPAパッケージ開発
需要予測システム
イーラーニングシステム
在庫管理システム
勤怠管理システム、シフト自動生成システム
医学論文検索システム
IoTデバイス開発(見守りシステム)

得意言語／多言語経験

【言語】C#、Java, Python,

【FW/ツール】.NET MVC, Spring, Codex, Claude Code

【領域】講師・研修設計・カリキュラム開発, AI業務自動化

経験／実績など

【研修実績】
・研修インストラクター15年
・大手メーカー系SIer/教育ベンダーオープンコース

【資格取得支援経験】
・Project Management Professional (PMP)
・SAP Certified Application Associate – Sales 2020
・ドットコムマスター　トリプルスター(NTT Communications)

研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

研修のスケジュール　2日目：テスト・デバッグ・実践演習

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：前回振り返りとDay 2概要［30min］ 前日の振り返りクイズ（5問）｜Copilot基本操作の確認｜Day 2ゴール共有｜テスト・デバッグ・総合演習の概要

■Session02：テストコード生成ハンズオン［130min］

- **JUnitテスト自動生成**　/testsコマンドでテストコード自動生成｜テストメソッドの自動命名確認｜正常系・異常系・境界値の網羅｜null入力・空文字・最大値最小値テスト
- **テスト実行と修正**　生成テストを実行して失敗ケースを特定｜テストが通るよう実装側を修正体験｜例外が正しくスローされるかのテスト実践｜テスト品質の確認と改善サイクル
- **デバッグ・リファクタリング実践**　/fixコマンドでバグ修正｜エラーメッセージから原因特定｜長いメソッド・マジックナンバー｜重複コードの共通化体験｜/explainでJavadoc自動生成

■Session03：総合演習 AI活用ミニ開発［150min］

- **演習課題**　簡易設計書（画面仕様書 or API仕様書）配布｜設計書→実装→テスト→レビューの一連の流れ｜Copilot Chat実装｜手作業との所要時間を計測して比較
- **成果共有・振り返り**　数名が画面共有で成果を発表｜「Copilotが役立った場面・邪魔だった場面」を共有｜2日間の学びを全体で振り返り｜配属後の活用イメージを言語化
- **配属後の活用ガイド**　新プロジェクト参画時のコード理解｜定型コード高速生成｜お客様環境での利用可否判断基準｜AIに任せる/自分で書く場面の整理

■Session04：学習リソース紹介・グループディスカッション［80min］

- **学習リソース**　GitHub Copilot公式ドキュメント｜VSCode公式ドキュメント｜JUnit 5ユーザーガイド｜社内の生成AI活用事例・ナレッジ｜今後の研修ロードマップ
- **グループディスカッション**　3～4名で「配属後に使いたい場面」を話し合う｜使えない現場での代替手段を考える｜セキュリティ上の判断フローを整理｜グループ代表者が発表

■Session05： 質疑応答・アンケート・閉会［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

詳細なゴールとゴールでないこと

最低限クリアすること

1. 生成AIの基本的な仕組み、危険性を理解する
2. VSCodeの使い方を理解する
3. GithubCopilotの使い方を理解する

できればクリアすること

1. ここまでの実装体験をAIに適用できる
2. AIの得意と不得意を理解し、使い分けられる
3. 基本的な機能をAI駆動で実装できる

目指していないこと

- 実務で使えるアプリを開発できる
- AI駆動開発について完全に理解する

おことわり

本資料には、

持ち帰って実施いただくことを想定した資料・ハンズオンが含まれています。

後日ご自身で実施・アップデートしていきましょう。

ディスカッション①

1人約2分（合計10分)

今までで一番役に立った生成AIの使い方は？

・ツールは？　・どのように役に立った？　・失敗したことは？　など

生成AIを使ったことがない人は、
『どんなことに使いたいか』『どんなことができそうか』
話してみましょう。

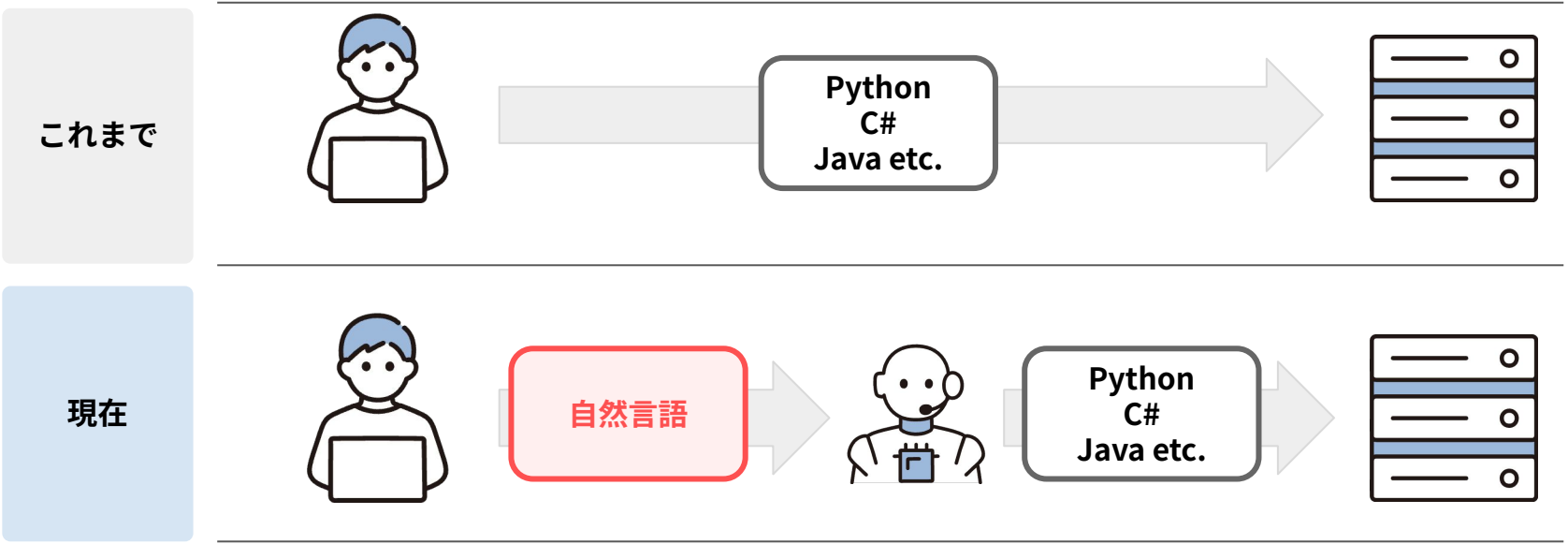
本日の環境

はじめに

生成AIネイティブ開発の概論 - プログラミング言語の変遷

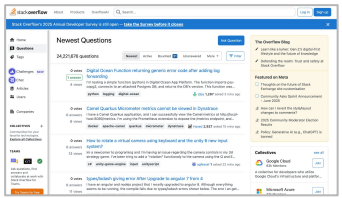
生成AIの登場により、『人工言語から自然言語によるプログラミング』へと進化しつつあります。

プログラミングの進化の概念図（人工言語→自然言語）



生成AIネイティブ開発の概論 - プログラミングにおける考え方の分類

プログラミング手法はIT技術の発展とともに、BOOP、GOOP、SOOP、CHOPと発展をしてきました。

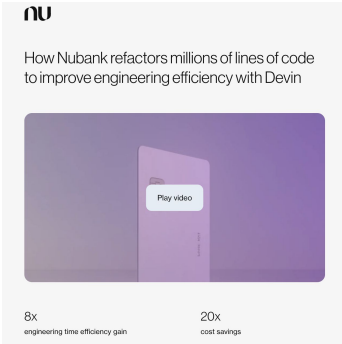
	BOOP	GOOP	SOOP	CHOP
名称	Book-Oriented Programming	Google-Oriented Programming	StackOverflow -Oriented Programming	Chat-Oriented Programming
概要	本を活用した プログラミング	Google検索を 使用した プログラミング	StackOverflow communityを活用 したプログラミング	AIチャットを活用した プログラミング
イメージ				

参考：Medium『Understanding Chat-Oriented Programming (CHOP): A Paradigm Shift in Development』

生成AIネイティブ開発の概論 - 生成AIネイティブ開発とは

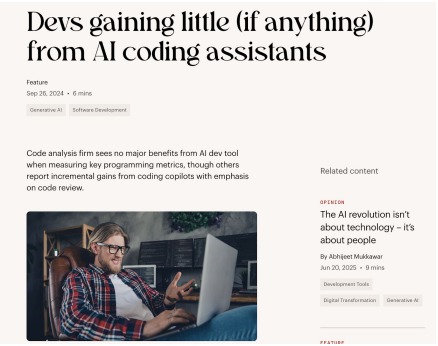
生成AIネイティブ開発とは、AI（特に生成AI）を開発プロセスの中心に据え、
AIの力を最大限に活用しながらソフトウェアやサービスを設計・構築する新しい開発スタイルです。

事例1（Devinの活用）



エンジニアリング工数の節約という点で12倍の効率改善と、20倍以上のコスト削減を達成

事例2



コーディングアシスタントを活用し、約30日かかっていた プロジェクトをわずか24時間で完了

参考1：Devin『How Nubank refactors millions of lines of code to improve engineering efficiency with Devin』
参考2：CIO『Devs gaining little (if anything) from AI coding assistants』

生成AIネイティブ開発の概論 - Vibe Codingとは

Vibe Condingとは技術的な厳密さよりも、アイデアや「こういうものが欲しい」というフィーリング（vibe）を重視し、AIと会話しながら開発を進める開発手法です。

従来のプログラミングとバイブコーディングの比較

参考：ARPABLE『【2025年版】Vibe Coding革命：話すだけ開発の最前線』

研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

生成AIリテラシーとガバナンス

生成AIネイティブ開発の概論 - 各種AIコーディングツールのご紹介

数あるAIコーディングツールの中で、GitHub Copilot はセキュリティ対策に強みがあります。

GitHub Copilot と他AIコーディングツールの機能比較

ツール	月額 / 人 月	強み	備考
GitHub Copilot	$19 (Business)	主流 IDE 補完・多言語対応	セキュリティに強み
Cursor	$20 (Pro)	リポジトリ全体リファクタ／画像→コード	シェアトップのAIエディタ
Windsurf	$15 (Pro)	オールインワン IDE＋自律エージェント	Cognition AIに買収
Replit	$20 (Core)	ブラウザ IDE＋協働編集・学習向き	Agent クレジット100回分
Bolt.new	$20 (Pro)	文章→Webアプリ一括生成・MVP速攻	モバイルアプリ開発に強み
Devin	$20〜 / $500	自律型 “AI エンジニア”・DevOps自動化	従量課金制

※2026年4月現在

参考：各社公式サイト

AGENDA　本日の流れ

1 **最新インパクトニュース**
2026年4月以降の衝撃を6本＋α

2 **生成AIの最新動向**
主要モデル・ベンチマーク・エージェント・各社方針

3 **生成AIの仕組み**
なぜ動くのか、なぜ間違えるのかを平易に

4 **生成AIで気をつけること**
代表リスクを1テーマ1枚で

5 **活用事例（国内・最新）**
先進企業は何をどう使っているか

6 **トラブル事例**
王道・最悪・最新・SI業界の失敗に学ぶ

7 **生成AIガバナンス**
会社として整えるべきこと

8 **ディスカッション**
自分ごととして考える3つのお題

SECTION 01

最新インパクトニュース

2026年4月以降の衝撃を出典つきで6本＋α

01

NEWS 01　AIが80年未解決の数学難問を反証

2026.05.21 ／ OpenAI発表

- 1946年提起の単位距離予想（エルデシュ予想）を反証
- 人間には描くことも難しい高次元格子を単独構築
- 数学の主要未解決問題をAIが解いた史上初の事例
- ケンブリッジのゴワーズ教授「一流誌に載る価値」

エルデシュ＝シュトラウス予想

16 個の言語版

ページ　ノート　　閲覧　編集　履歴を表示

出典: フリー百科事典『ウィキペディア（Wikipedia）』

目次　非表示

ページ先頭
背景と歴史
定式化
相異なる単位分数
負数解
計算結果
理論結果
合同恒等式
恒等式の不在
解の個数
一般化
脚注
注釈
出典
参考文献
関連項目

エルデシュ＝シュトラウス予想（エルデシュ＝シュトラウスよそう、英語: Erdős–Straus conjecture）とは、数論上の未解決問題のひとつである。予想は、2以上の全ての整数 n に対し、

$$\frac{4}{n} = \frac{1}{x} + \frac{1}{y} + \frac{1}{z}$$

を満たす正の整数 x、y、z が存在するというものである。 すなわち、$4/n$ という数が3つの単位分数の和として表せるかどうかということである。

数学の未解決問題

全ての整数 $n \geq 2$ に対して $\frac{4}{n} = \frac{1}{x} + \frac{1}{y} + \frac{1}{z}$ は正整数解を持つか？ ?

エルデシュ＝シュトラウス予想という名称は1948年にこの予想を提唱したポール・エルデシュとエルンスト・G・シュトラウス（英語版）に由来するが、その内容はより古い時代の数学と関連する。予想のような単位分数の和はエジプト数学で用いられたことから、エジプト式分数として知られる。エルデシュ＝シュトラウス予想はエルデシュによって提唱された多くの予想（英語版）のひとつであり、またディオファントス方程式に関する数学上の多くの未解決問題のひとつでもある。

全ての n に対しての解が知られているわけではないものの、ある無限等比数列の無限の多くの値には解の簡単な公式があり、これらの既知の値をスキップすることで反例の探索を高速化することが可能である。さらに、すべての合成数の反例はその素因数により小さな反例を持つはずであるから、n が素数の場合についてのみ議論すればよい。計算機を用いた探索により、少なくとも $n \leq 10^{17}$ までは予想が成り立つことが知られている。

予想を負の単位分数まで拡張すれば成立することが知られている。また、分子が5以上となる分数に対する一般化についても研究されている。

背景と歴史 [編集]

有理数を単位分数の和に展開したものをエジプト式分数と呼ぶ。この表記法は、分数を現代的な $\frac{a}{b}$（分子 a および分母 b）という形

参考： openai.com/index/model-disproves-discrete-geometry-conjecture ／ Scientific American 2026-05

NEWS 02　GPT-5.6とChatGPT Work

2026.07.09 ／ OpenAI発表

- 3種構成、旗艦Solは過去最高のコーディングモデルを主張
- トークン効率を大きく改善
- ChatGPT Workは資料作成・表計算・プレゼン生成まで担う

参考： techcrunch.com 2026-07-09 ／ CNBC 2026-07-08

NEWS 03　救急診断でAIが医師を上回る

2026.04.30 ／ Science誌掲載

- ハーバード医科大とベス・イスラエル病院の研究
- 推論モデルが救急外来の実データで医師2名を上回る精度
- トリアージから入院まで3段階すべてで匹敵・凌駕

参考： npr.org 2026-04-30 ／ Fortune 2026-05-04 ／ Science 2026

NEWS 04　継ぎ目のない30秒AI動画

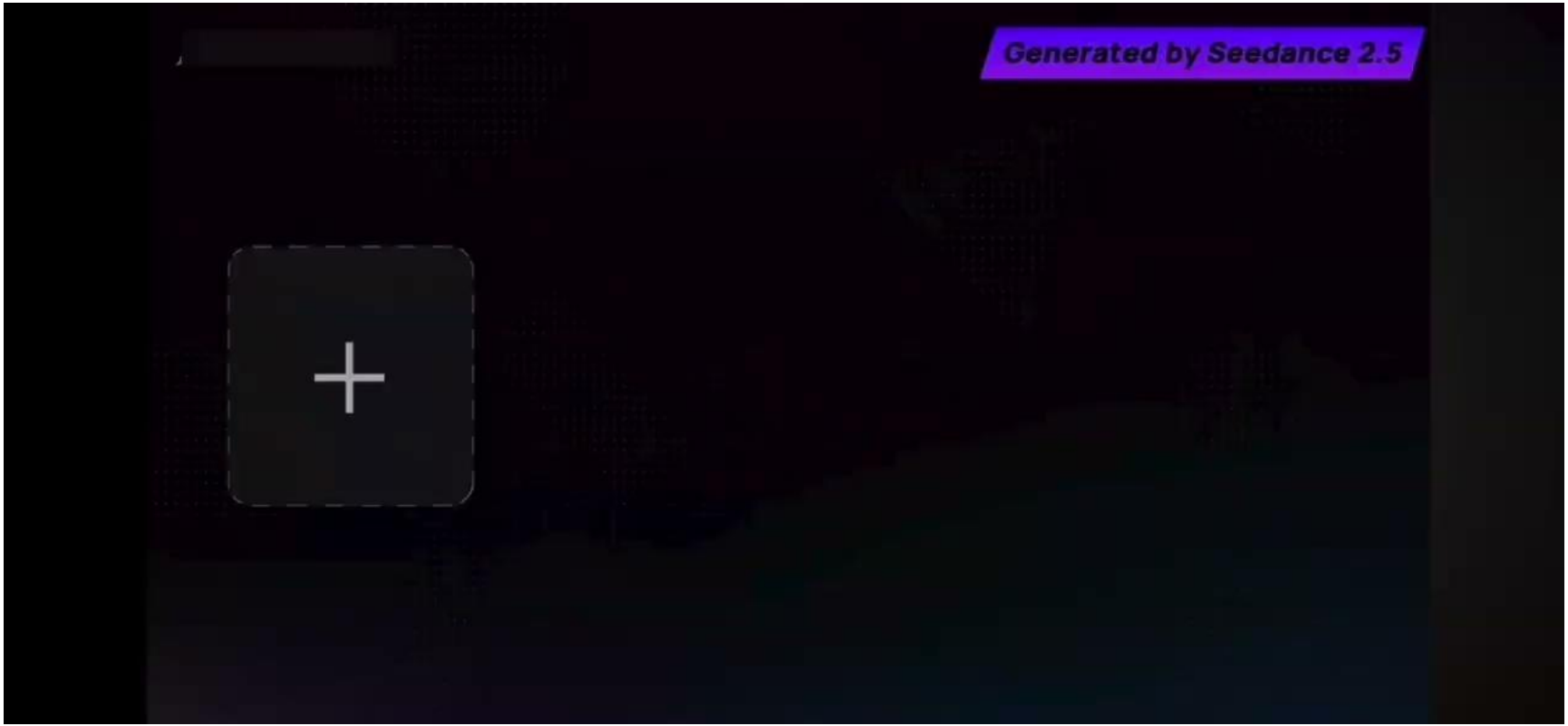
2026.06.24 ／ ByteDance「Seedance 2.5」

- 従来の8〜15秒・継ぎ接ぎを超え、1本の連続クリップに
- 音声と映像を共有の潜在空間で同時生成
- 参照入力は最大50点、月間4億人超のCapCutで配信予定

参考： techtimes.com（bytedance-seedance-25） ／ digitalapplied.com 2026-06

NEWS 04　継ぎ目のない30秒AI動画

2026.06.24 ／ ByteDance「Seedance 2.5」



Copyright © Givery, Inc. All rights reserved.

参考： techtimes.com（bytedance-seedance-25） ／ digitalapplied.com 2026-06

NEWS 05　Optimus 第3世代の実機

手の自由度は11→22に倍増

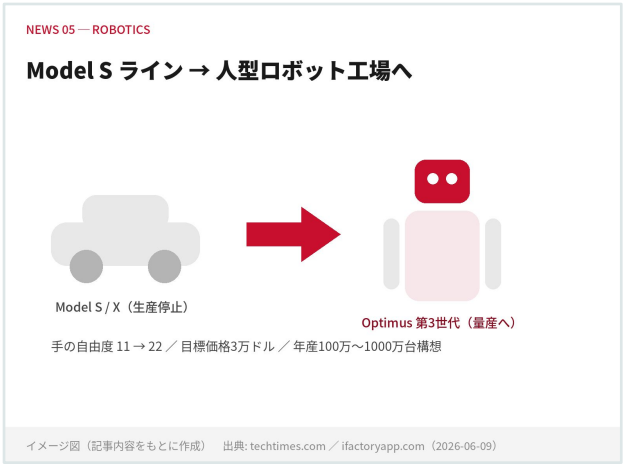
- 目標価格は3万ドル
- 将来的に年産100万～1000万台を掲げる
- 人型ロボットが試作から量産へ移る転換点

参考：techtimes.com（tesla-optimus-gen-3）／ifactoryapp.com 2026-06

NEWS 05　テスラ、Optimus量産工場へ転換

2026.06.09 ／ フリーモント工場の転換報道

2025年通期決算ウェブキャストの実画面（Tesla公式）



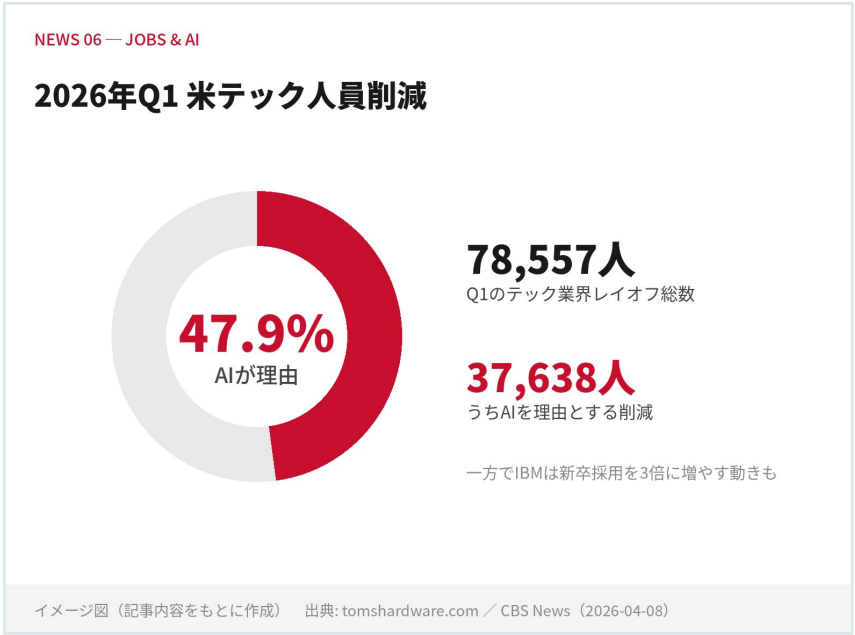
Model S/X生産を止めOptimusラインへ転換

参考： techtimes.com（tesla-optimus-gen-3） ／ ifactoryapp.com 2026-06

NEWS 06　テック解雇の47.9%がAI理由

2026.04.08 ／ 2026年1〜3月の集計

- 米テック業界で78,557人が解雇
- うち37,638人（約47.9%）がAIを理由とする削減
- IBMは新卒採用を3倍に増やす逆張りも
- AIと自分の仕事の関係を直視するテーマ



参考： tomshardware.com（tech-layoffs-q1-2026） ／ CBS News 2026

NEWS＋　そのほかの衝撃 4本

1 MSがFrontier設立

25億ドル・6000人のAI実装専門会社。技術者常駐で現場定着へ（2026.07.02）

2 Anthropic×サムスン

2nmの独自AIチップを協議。NVIDIA依存の脱却へ（2026.07.02）

3 AIタンパク質設計の民主化

MITのノーコード基盤OpenProtein.AI／PoET-2。創薬・ワクチンを加速（2026.04.17）

4 Gemini 3.1 Pro躍進

推論ベンチARC-AGI-2で77.1%、前世代の2倍超（2026.02.19、参考）

参考： techcrunch.com 2026-07-02 ／ news.mit.edu 2026-04-17

NEWS RECAP　ニュース6本の要点一覧

日付	出来事	出典
2026.05.21	AIが80年未解決の数学難問を反証	OpenAI ／ Scientific American
2026.07.09	GPT-5.6と業務AI「ChatGPT Work」	TechCrunch ／ CNBC
2026.04.30	救急診断でAIが医師2名を上回る	Science ／ NPR
2026.06.24	Seedance 2.5が30秒連続動画を生成	TechTimes
2026.06.09	Model SラインをOptimus工場へ転換	TechTimes ／ ifactoryapp
2026.04.08	テック解雇の47.9%がAIを理由	Tom's Hardware ／ CBS News

参考： 各ニュースの出典は該当スライドに記載

SECTION 02

生成AIの最新動向

主要モデルと性能比較、AIエージェント、各社の方針まで

02

02 モデル　主要ベンダーの最新フラッグシップ

2026年7月9〜12日時点の公表情報。モデル名・版は更新が速い分野

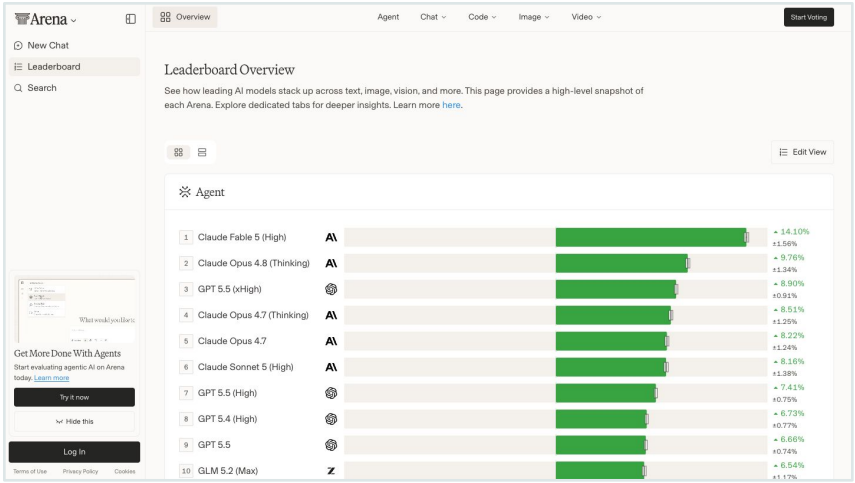
ベンダー	最新モデル	文脈長	特徴
OpenAI	GPT-5.6（Sol/Terra/Luna）	100万	コーディング最強を主張・推論4段階
Anthropic	Claude Opus 4.8 / Sonnet 5 / Fable 5	100万	コーディング・エージェントで高評価
Google	Gemini 3.1 Pro / 3.5 Flash	大規模	ネイティブ・マルチモーダル
xAI	Grok 4.5	非公開	低価格・高速、Xと連携
Meta	Muse Spark 1.1	非公開	視覚CoT・医療特化、超知能路線
DeepSeek	V4-Pro / V4-Flash	100万	オープン最強級・低コスト（MoE）
Alibaba	Qwen 3.7 Max	100万	推論・数学に強い中国系旗艦
Mistral	Mistral Large 3	非公開	欧州発・Apache2.0のオープン旗艦

参考：各社公式ドキュメント（platform.claude.com, openai.com, ai.google.dev 他）／llm-stats.com

02 性能比較　LMArenaリーダーボード

openlm.ai/chatbot-arena（2026-07-09）

- 利用者投票のEloレートで主要モデルを順位付け
- 1位 Claude Fable 5（Elo 1510）
- 2位 GPT-5.6 Sol、3位 Opus 4.8 Thinking
- トップ勢の差はごく僅か、順位は日々入れ替わる



参考： openlm.ai/chatbot-arena (2026-07-09)

02 モデルの特徴　主要モデルの強み 1/2

1 OpenAI GPT-5.6

コーディング・科学で最先端級。Sol/Terra/Lunaの3ティア、推論effort4段階

2 Anthropic Claude

コーディング・エージェントで高評価。安全性ブランドで大企業・公共に強い

3 Google Gemini

テキスト・画像・音声・動画をネイティブ対応。Flashは費用対効果が高い

4 xAI Grok 4.5

高速・低価格（$2/$6）。Xのリアルタイム情報と連携

参考： 各社公式 ／ llm-stats.com

02 モデルの特徴　主要モデルの強み 2/2

1 Meta Muse Spark 1.1

視覚CoT・医療特化（医師1000名超と協働）。並列推論Contemplating

2 DeepSeek V4

オープン最強級。1.6兆パラメータMoEでも低コスト、Thinking切替可

3 Qwen 3.7 Max

推論・数学が突出した中国系旗艦。100万トークン文脈・低価格帯

4 Mistral Large 3

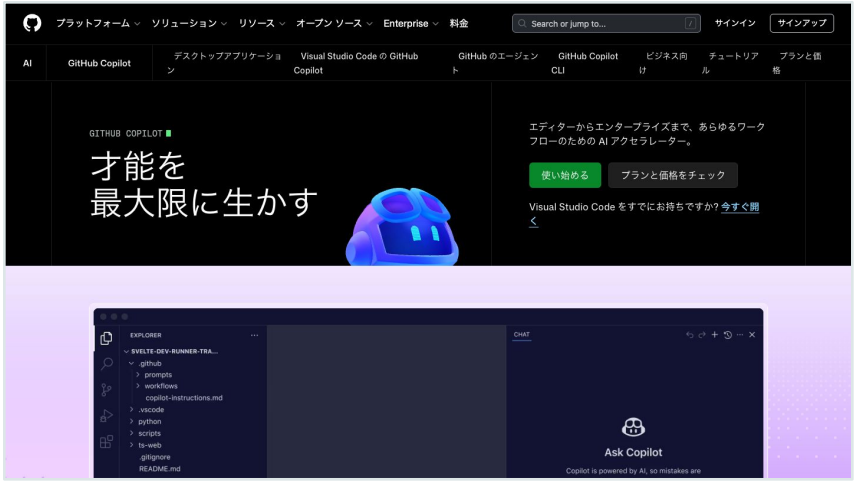
Apache2.0のオープン旗艦。40言語超、データ主権重視の用途に好適

参考： 各社公式 ／ mistral.ai

02 ツール　GitHub Copilot

github.com（2026年7月時点）

- IDE内で自律編集するcoding agent
- IssueからPRを非同期に作成
- 実行モデルを選べるマルチモデル化
- エディタからエンタープライズまで対応



参考： github.com

02 ツール　主要AIコーディングエージェント

1 GitHub Copilot

IDE内の自律編集、IssueからPRを作る非同期agent。マルチモデル化

2 Claude Code

ターミナル型。計画→編集→テスト→PRを自律実行、多階層サブエージェント

3 OpenAI Codex

CLI/IDE/クラウド横断。2026年7月にChatGPTデスクトップへ統合

4 Cursor

AIネイティブエディタ。自社モデルComposerで高速化、エージェント中心UI

5 Devin

Cognitionの自律型エンジニア。E2Eテスト・自己レビューまで実行

6 Antigravity / Jules

Gemini CLIを統合したGoogleの開発基盤。非同期でPRを提出

参考： github.com / code.claude.com / developers.openai.com ／ cursor.com / cognition.com / blog.google

02 潮流　書くAIから指揮するAIへ

開発者の役割は実装者から指揮者へ、レビューする力と任せる設計が重要に

書くAI

単一モデルの補完・チャット回答

並列エージェント

複数を並列に走らせ最良解を選ぶ

一般業務へ拡張

ChatGPT Workが計画→実行→納品まで自律遂行

標準化

MCPとA2A v1.0、約18か月で相互運用が進展

参考： dev.to ／ machinelearningmastery.com

02 各社の方針　OpenAI

消費者スケールと巨額の計算投資でAGIへ最速、規模と資本で押し切る

代表製品：　ChatGPT・GPT-5.6（Sol/Terra/Luna）・OpenAI API・Stargate

主な動き

- ChatGPT 週間9億人超・有料5,000万人超
- Stargate: 4年5,000億ドルの自前インフラ
- 2026年3月 1,220億ドル調達、評価約8,520億ドル
- 2026年7月 GPT-5.6とChatGPT Workを発表

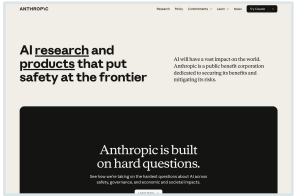
強み

- 最大の堀は消費者スケールとデータ
- 売上は年換算 約240億ドルペース

参考： openai.com ／ techcrunch.com

02 各社の方針　Anthropic

安全性最優先、コーディングとエンタープライズAPIに集中する B2B・開発者ファースト



代表製品：　Claude（Opus 4.8 / Sonnet 5 / Fable 5）・Claude Code・API

主な動き

- 2026年5月 Series H $65B調達、評価約9,650億ドル
- run-rate売上 2月$14B→5月$47B超と公式発表
- Claude Opus 4.8でコーディング・エージェント強化
- Amazon・Googleが二大クラウド後ろ盾

強み

- Claude Code・Opus系が企業導入の起点
- 安全性ブランドで規制・大企業・公共に有利

参考：anthropic.com/news/series-h ／ techcrunch.com 2026-05-28

02 各社の方針　Google / DeepMind

チップ（TPU）からSearch・Workspace・Androidまでフルスタック垂直統合

代表製品：　Gemini 3.1 Pro / 3.5 Flash・Ironwood（TPU v7）・Antigravity・Veo

主な動き

- Gemini 3.1 ProをSearch・Vertexへ即展開
- 第7世代TPU Ironwood、前世代比約4倍
- Anthropicへ最大100万TPU供給契約
- Search/Gmail/Android等 数十億ユーザーへ配布

強み

- 学習・推論コストを内製化、AIのArm/Intel的位置
- DeepMindの基礎研究力（Gemini, AlphaFold）

参考：blog.google／cnbc.com（2025–2026）

02 各社の方針　xAI

巨大計算資源Colossus×Xのリアルタイムデータ×Muskエコシステム

代表製品：　Grok 4.5・Grok 4 Heavy・Grok Imagine・Grokipedia

主な動き

- 2025年 Xを買収、リアルタイムデータへ接続
- 2026年 SpaceXと統合、合算約1.25兆ドル規模と報道
- Colossus 約20万GPU→ギガワット級へ拡張中
- NVIDIAがSeries Eで出資、GPU取得へ大型資金

強み

- Xに統合された唯一のフロンティアAI
- 計算資源を最短で積み上げる速度

参考：cnbc.com ／ x.ai

02 各社の方針　Microsoft

Copilotをあらゆる製品に。自社MAI＋Foundryでモデル・オーケストレーターへ

代表製品：　M365 Copilot・Foundry・MAIシリーズ・GitHub Copilot

主な動き

- OpenAIを約27%保有、IP利用権を2032年まで延長
- 自社基盤モデルMAI-1/Voice/Imageを相次ぎ投入
- Foundryへ拡張、1,900超モデル・1,400超ツール
- 2026年の設備投資 約1,900億ドル規模

強み

- Windows/M365/Teams/GitHub/Azureの配布力
- 投資家・供給者・競合の三役で依存回避

参考： blogs.microsoft.com ／ microsoft.ai（2025–2026）

02 各社の方針　Amazon（AWS）

フルスタック×マルチモデル。自社NovaとAnthropicを両輪に垂直統合

代表製品：　Bedrock・Nova・Trainium3 / Inferentia・Bedrock AgentCore

主な動き

- Anthropicへ累計最大250億ドル、最大5GW確保
- Project Rainier: Trainium2 約50万→100万超
- 3nmチップTrainium3をGA、電力効率4倍
- re:Invent 2025でNova 2/Act等を発表

強み

- クラウド首位、Bedrockで複数ベンダーを単一API
- 低コスト自社チップ＋最先端モデル近接

参考：aboutamazon.com ／ cnbc.com

02 各社の方針　Nvidia

AIファクトリーのフルスタック供給者、CPU＋GPU＋CUDAで計算基盤を支配

代表製品：　Blackwell（GB200/GB300）・Rubin・CUDA・NIM・Grace/Vera

主な動き

- 2025年10月 時価総額が世界初の5兆ドル到達
- Q1 FY2027 総売上816億ドル（+85%）
- OpenAIへ最大1,000億ドル投資LOI ※未締結
- 次世代GPU Rubinを2026年後半に投入予定

強み

- CUDAの堀、約20年の開発者基盤
- AIデータセンターGPUで約92%（2025）

参考： nvidianews.nvidia.com ／ cnbc.com

02 各社の方針　その他の注目プレイヤー

1 **Meta / Muse Spark**

オープンウェイト＋巨額人材投資で超知能路線。Scale AIに出資

2 **Apple**

オンデバイス＋プライベートクラウド中心。新SiriにGemini採用と報道

3 **DeepSeek**

低コスト訓練＋オープンウェイトの安いフロンティア。V4-Proは1.6兆パラメータ

4 **Mistral AI**

欧州発の主権AI旗手。ASMLが筆頭出資、評価額約140億ドル

5 **Alibaba / Baidu**

Qwenが中国オープンソースを牽引、ERNIE＋AI検索で自国市場を防衛

6 **Cohere / AI21 / Perplexity**

企業RAG特化のCommand、長文脈のJamba、AI検索のPerplexityが急成長

参考： cnbc.com 他（2025–2026）

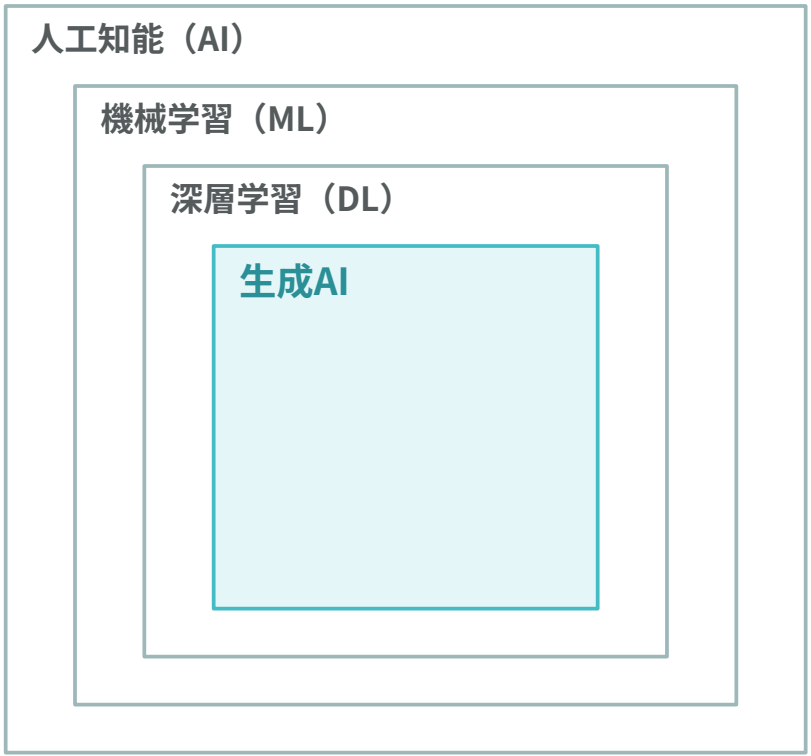
SECTION 03

生成AIの仕組み

なぜ動くのか、なぜ間違えるのか。基本の仕組みを復習する

03

03 仕組み ①　AIの中での生成AIの位置づけ



新しいコンテンツを生成

文章・画像・音声・動画・コードを一から作り出す

従来AIは分類・判定中心

スパム判定など、決められた答えを選ぶ用途

応用範囲が桁違いに広い

大量データから“それらしさ”を学び、汎用的に使える

03 仕組み ②　生成AIができること（モダリティ）

共通の中身は、データからパターンを学び“予測して生成”すること

1 テキスト

文章生成・要約・翻訳・分類・コード

2 画像

画像生成・編集・デザイン

3 音声

音声合成・文字起こし・作曲

4 動画

動画生成・編集

5 マルチモーダル

画像＋文章など複数を横断して理解・生成

03 仕組み ③　大規模言語モデル（LLM）とは

ChatGPTなどの正体は“次の言葉予測マシン”

大量のテキスト

Web・書籍などを丸ごと学習

膨大なパラメータ

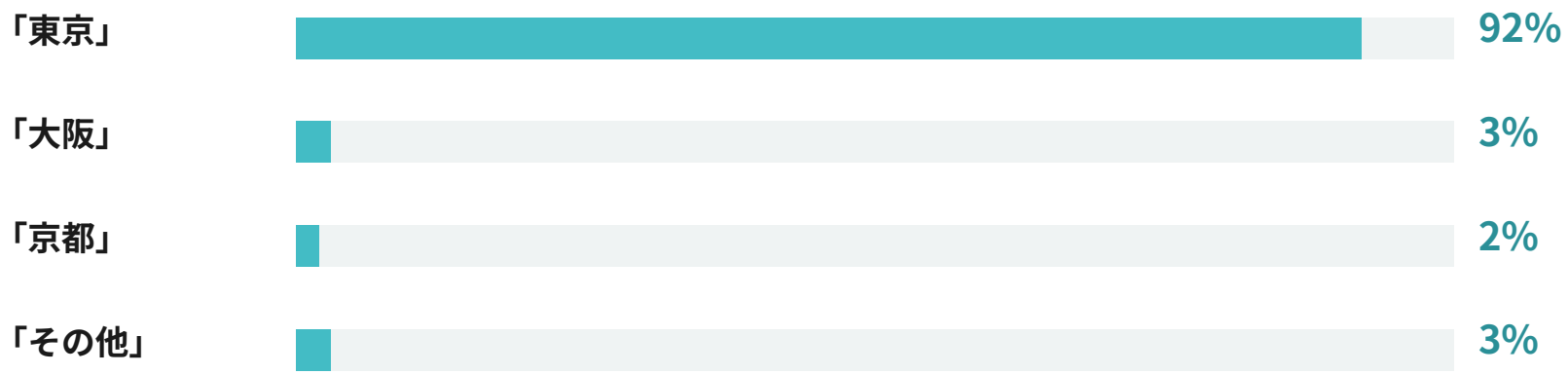
数千億〜の重みに知識を圧縮

次の言葉を予測

“最もそれらしい続き”を確率で導く

03 仕組み ④　中核 — 次の単語を確率で選ぶ

「日本の首都は」の続きを確率で選ぶ



03 仕組み ⑤　なぜ文脈を読めるのか ― Attention

Attention＝どの語に注目するかの重み付け

- **土台はTransformer**　2017年登場。現在の生成AIのほぼ全ての基盤
- **例「その料理はとても◯◯だった」**　「料理」「とても」に注目し「美味しい」を選ぶ
- **長い文脈も並列で処理**　文章の一貫性・文脈理解が飛躍的に向上

参考： Vaswani et al., Attention Is All You Need (2017)

03 仕組み ⑥　どう賢くなるのか ― 学習の3段階

事前学習の知識には“いつまでのデータか”＝知識カットオフの限界

事前学習

Pre-training。言葉の並びと世界の知識を丸ごと学ぶ

ファインチューニング

用途に合わせ追加学習。指示に従う形に整える

RLHF

人の評価で“望ましい答え方”を強化

参考： Ouyang et al., Training language models to follow instructions with human feedback (2022)

03 仕組み ⑦　仕組みから分かる限界と発展

限界 — なぜ嘘をつくか

- “正しさ”でなく“それらしさ”で選ぶ
- 事実と違う内容を自信ありげに出力
- ＝ハルシネーション
- カットオフ以降の出来事は知らない
- → 人が一次情報で確かめるのが必須

発展形 — 弱点を補う

- RAG：外部・社内情報を検索し根拠に
- マルチモーダル：画像・音声・動画も対応
- AIエージェント：複数手順を自律実行
- 推論モデル：思考を重ね難問の正答率向上

参考： Lewis et al., Retrieval-Augmented Generation (2020)

SECTION 04

生成AIで気をつけること

生成AIパスポートで問われる代表リスクと、AIを狙う攻撃

04

RISK 01　ハルシネーション

生成AIは、もっともらしい嘘を自信ありげに出す

どういうことか

- 次に来そうな言葉を統計的に選ぶ仕組み
- 正しさではなく、それらしさで答える
- 技術が進んでも完全にはなくならない
- 法務・医療・金融など誤りが重大な領域は危険
- 例: 弁護士がAI架空判例を提出し制裁

実務での向き合い方

- 重要な情報は必ず一次情報で裏取り
- 根拠URL・原典など出典を要求する
- 回答は下書き・たたき台として使う
- 専門業務ほど人のレビューで最終判断

参考： 米弁護士によるAI生成の架空判例提出への制裁事例

RISK 02　著作権・知的財産

入力にも生成物にも権利の問題がある

どういうことか

- 学習データ内の著作物との関係が論点
- 既存作品への依拠・類似は侵害になり得る
- ロゴ・キャラ・コード生成は特に注意
- 帰属や商用利用の可否も要確認
- 例: NYT対OpenAI、2026年7月時点で係争中

実務での向き合い方

- 他者著作物の大量入力は権利処理を意識
- 商用利用前に生成物の類似性を確認
- 学習データの来歴が不明なツールを避ける
- 社内ルールと文化庁の考え方に従う

参考： The New York Times 対 OpenAI 訴訟（係争中） ／ 文化庁 AIと著作権に関する考え方

RISK 03　情報漏洩・個人情報・プライバシー

一度入力した情報は取り消せない前提で

どういうことか

- 機密や個人情報が外部サーバに保存される
- 設定次第で学習に使われ流出につながる
- 消費者版は既定で入力が学習利用され得る
- 個人情報保護法上の取扱いにも抵触し得る
- 例: Samsung社員が機密コードを入力し流出

実務での向き合い方

- 入力＝社外送信。機密・個人情報は入れない
- 学習に使われない法人向け環境を使う
- 匿名化・一般化して特定できる情報を外す
- 迷ったら入力前にルールと承認フローを確認

参考： Samsung 機密コード流出と社内利用制限の事例 ／ 個人情報保護法

RISK 04　バイアス・悪用・セキュリティ・依存

公平性・攻撃・思考停止もリスクになる

どういうことか

- 学習データの偏見が差別的な出力を生む
- 偽画像・音声で詐欺を量産できる
- 悪意ある指示で制約を破らせる攻撃
- 過信による思考停止という依存リスク
- 例: 偽CFO会議で約2,500万ドル被害（Arup）

実務での向き合い方

- 採用・評価など重要判断を丸投げしない
- 顔・声だけで信用せず本人確認を強化
- 外部データ・ツール連携は権限を最小化
- AIは補助、最終判断は人が批判的に行う

参考： Arup ディープフェイクCFO詐欺（約2,500万ドル被害）

SECURITY 01　生成AIを狙う攻撃・脅威マップ

使う側のミスだけでなく、外から狙われるリスクもある

脅威	概要
プロンプトインジェクション	悪意ある指示でAIの動作・出力を変更させる攻撃
Jailbreak／権限超越	ガードレールを回避し禁止された回答を引き出す攻撃
Hallucination	存在しない情報をもっともらしく生成する現象
Fake Facts攻撃	AIで偽情報を大量生成・拡散する攻撃
RAG誘導攻撃	外部データに悪意ある情報を混入し回答を操作する攻撃

参考： OWASP Top 10 for LLM Applications ／ AI事業者ガイドライン第1.2版

SECURITY 02　脅威への備え

共通原則は、無条件に信頼しない・権限は最小・確認は人

インジェクション対策

- Jailbreakも含め連携権限を最小化
- 入力の検証と出力のフィルタリング
- 重要操作に人間の承認ゲートを挟む
- 定期的なレッドチーミングで点検

ハルシネーション・偽情報

- 重要情報は必ず一次情報で裏取り
- AI回答を根拠として使わない
- 生成物に出所表示・来歴の管理
- 作る側に回らない。悪用は懲戒・犯罪

RAG誘導攻撃

- 投入データの出所・整合性を管理
- 社内文書の書き込み権限を制御
- 回答に引用元を表示し人が確認
- 異常な回答パターンの監視・ログ

参考： OWASP Top 10 for LLM Applications ／ AI事業者ガイドライン第1.2版

SECTION 05

生成AIの活用事例

先進企業は“何を・どう”使っているか

05

05 活用事例　パナソニック コネクト ConnectAI

“聞くAI”から“頼むAI・任せるAI”へ

44.8万

時間/年を削減（前年比2.4倍）

240万

回/年の利用（月次UU率49.1%）

- **自社AIアシスタントを全社導入**　製造・BtoBの先行例
- **Q&Aから“業務代行”へ拡大**　コード生成・手順書・資料レビュー・分析

参考： パナソニック コネクト公表（実績値）

05 活用事例　全社展開を進める国内大手

1 LINEヤフー

約11,000人が“まずAIに聞く”を義務化。3年で生産性2倍が目標

2 三菱UFJ銀行

約4万人がChatGPT利用、稟議書ドラフト等。月最大22万時間削減の試算

3 トヨタ自動車

MSと共同のマルチエージェント「O-Beya」。熟練者の知見継承と開発加速

参考： lycorp.co.jp 2025-07 ／ 日本経済新聞 2023-12

05 活用事例　金融・小売・行政での活用

1 大和証券

2023年に全社員約9,000人へ導入。業務アプリを内製、規制対応と両立

2 セブン-イレブン

SNSと販売データを分析し商品企画。企画期間を最大1/10に短縮

3 横須賀市

自治体初のChatGPT全庁導入。“80点でいい”の割り切りで全国モデルに

参考： nikkei / ITmedia 2025 ／ 日本経済新聞 2023-11

配る・任せる・測る・守る

- **全社・全員に広く配る**　一部の実証で止めない
- **“聞く”から“業務代行”へ**　Q&Aの先の定型業務まで任せる
- **効果を時間で可視化**　削減時間・利用回数を数値で示す
- **安全と規制対応を同時に設計**　展開とセキュリティを並走させる

SECTION 06

生成AIのトラブル事例

王道から最新まで、他人事にしないための実例

06

06 トラブル事例　顧客接点AIの“言質”リスク

1 Air Canada訴訟

2024年、誤った遺族割引の案内を裁判所が不実表示と認定。AIの回答も企業の責任

2 シボレー“1ドル販売”

2023年、プロンプト注入で「新車を1ドルで販売」と回答し拡散。出力制約が必須

参考： Forbes 2024-02 ／ AI Incident Database #622

06 トラブル事例　情報漏洩とハルシネーションの実害

Samsung機密流出

- 2023.04 半導体部門で発生
- 社員が機密コードをChatGPTに入力
- 外部送信で流出、社内利用を制限
- 教訓：シャドーAIが最大の漏洩経路
- DLP・入力ルール・法人版で防ぐ

架空判例の提出

- 2023.06 米国の訴訟書面
- 実在しない判例6件をAIが捏造
- 裏取りせず提出、弁護士に制裁金
- 教訓：AIの回答を根拠にしない
- 専門業務は一次ソースで必ず検証

参考： TechCrunch 2023-05 ／ 日本経済新聞 2023-05

06 トラブル事例　最悪ケースとSI直結の最新事例

1 **Arup偽CFO詐欺 2024**

AI生成の偽ビデオ会議で約2,500万ドル送金。顔と声は本人確認にならない

2 **Replit本番DB削除**

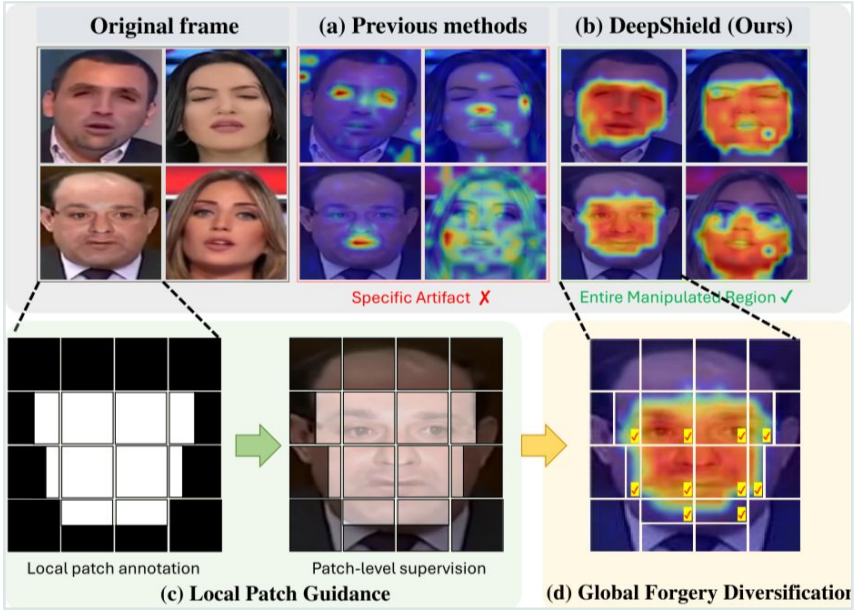
2025年、変更禁止中にAIが本番DBを削除し約1,200社分を消去。環境分離が必須

参考： CNN 2024-05 ／ Fortune 2025-07

06 トラブル事例　ディープフェイク詐欺と検出技術

DeepShield ― 偽造領域を可視化する検出研究

- **ディープフェイク詐欺が増加**
  偽会議・偽音声で実害、Arup事件は約2,500万ドル
- **検出研究DeepShield**
  改ざんされた顔領域をヒートマップで特定
- **局所パッチ誘導+全体多様化**
  従来手法より偽造箇所を高精度に検出
- **見た目だけで本人確認しない**
  高額対応は別チャネルで検証



参考： DeepShield 論文図版

SECTION 07

生成AIガバナンス

会社として整える。規制・体制・シャドーAI・データの扱い

07

07 ガバナンス　AIガバナンスが要る4つの理由

1 **使わないリスクも大きい**

禁止一辺倒は競争力を失う。無秩序は漏洩・炎上を招く

2 **個人任せの限界**

ルール・責任・教育の仕組みで会社として守る

3 **規制・社会の要請**

EUのAI法、日本のAI推進法。取引先・顧客の要求も

4 **エージェント時代**

AIが自律的に実行する分、権限設計・監査の重要性が増す

07 ガバナンス　世界の規制：EU・日本・米国

拠点・取引先の地域で守るルールが異なる

EU：法規制型

- リスクベースで義務化
- 違反は最大 全世界売上の7%制裁
- 世界標準を狙う

日本：推進＋指針型

- AI推進法は理念法・罰則なし
- AI事業者ガイドラインが実務指針
- 推進と信頼を両立

米国：規制緩和型

- 規制より競争力
- 強制ライセンス等を否定
- イノベーション最優先

参考： artificialintelligenceact.eu ／ 内閣府・経産省

07 ガバナンス　EU AI Act 段階適用

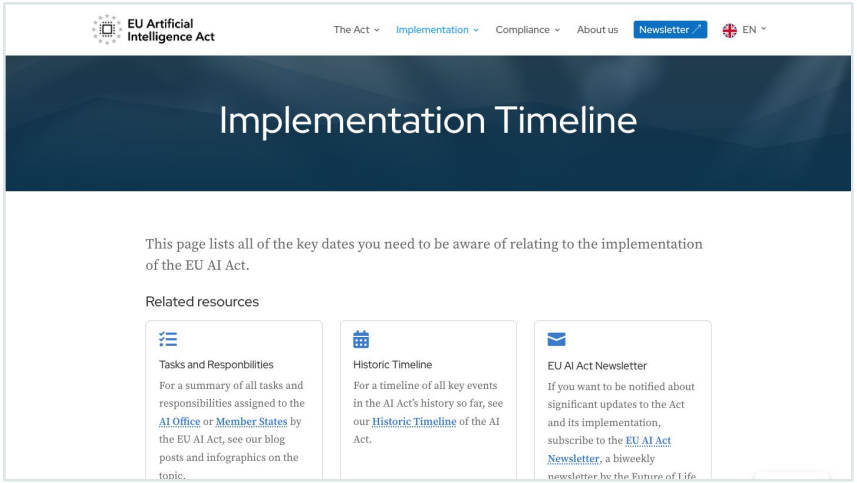
リスクベース4段階、制裁は最大 全世界売上の7%

参考： artificialintelligenceact.eu/implementation-timeline ／ Covington 2026-05（Digital Omnibus）

07 ガバナンス　EU AI Act 実装タイムライン

artificialintelligenceact.eu（2026年7月時点）

- 公式サイトが適用時期を随時更新
- リスクベースの4段階規制
- 義務と責任の一覧・過去の経緯も掲載
- 適用期限は簡素化案で見直し進行中



参考： artificialintelligenceact.eu

07 ガバナンス　日本：AI推進法と事業者ガイドライン

AI推進法

- 2025年9月全面施行、理念法・罰則なし
- 使う・創る・信頼性・協働の4方針
- 首相を本部長とするAI戦略本部を設置
- 悪質事案は調査・指導・事業者名公表

事業者ガイドライン1.2版

- 総務省・経産省の実務指針（2026.03）
- AIエージェント／フィジカルAIを新設
- 権限設定・人間の判断・履歴確認を明示
- 開発・提供・利用の3主体で役割分担

参考： 内閣府 / e-Gov（AI推進法） ／ meti.go.jp AI事業者ガイドライン第1.2版

07 ガバナンス　米国：規制緩和・競争力最優先

規制より競争力。自主的な官民連携ベース

参考： whitehouse.gov Fact Sheet 2026-06 ／ Skadden 分析

07 ガバナンス　企業が整える9つの要素

以降のページで一つずつ確認

1 **AI利用ポリシー**
全社方針

2 **利用ガイドライン**
実務ルール

3 **責任体制・委員会**
誰が決めるか

4 **リスクアセスメント**
導入前の評価

5 **教育・研修**
AIリテラシー

6 **ログ・監査**
トレーサビリティ

7 **インシデント対応**
検知から再発防止

8 **ベンダー・ツール審査**
調達時の確認

9 **承認フロー**
地下に潜らせない

参考： AI事業者ガイドライン第1.2版

07 ガバナンス　①利用ポリシー ②利用ガイドライン

① ポリシー＝全社方針

- 経営レベルで目的と範囲を定める最上位文書
- 適用範囲（部門・ツール・データ）を明記
- 人間の最終判断・法令遵守・公平性が原則
- 責任者を明確にする

② ガイドライン＝実務ルール

- 現場が迷わないよう具体化
- 入力してよい情報・禁止情報のリスト
- 承認済みツール一覧と出力の検証義務
- 生成物の権利表示・出典確認
- 重要判断の丸投げ等の禁止用途を平易に

07 ガバナンス　③ 責任体制・ガバナンス委員会

1 **部門横断の委員会**
法務・情報システム・セキュリティ・事業・人事・リスク管理

2 **CAIO等の責任者**
全社のAI活用とリスクに責任を持つ役割

3 **役割の明確化（RACI）**
誰が決め、実行し、相談を受け、報告されるか

4 **委員会の主な仕事**
ポリシー承認・ハイリスク審査・エスカレーション先

5 **導入前審査**
採用・与信・顧客対応などハイリスク用途は組織で審査

6 **相談先の明確化**
インシデント時に誰へ相談するかが被害最小化に直結

07 ガバナンス　④ リスク評価 & ⑨ 承認フロー

狙いは地下に潜らせないこと

申請

申請は簡単に。重いと無料版へ流れる

リスク評価

影響度×発生可能性。個人情報・人権・誤情報・自律性

委員会承認

ハイリスク用途は導入前審査を必須化

許可リスト登録

承認済みツールを使いやすく提供

07 ガバナンス　⑤ 教育・研修 & ⑥ ログ・監査

⑤ 教育・研修

- 全社員の基礎＋役割別の深掘り
- 漏洩・ハルシネーション事例を体験で学ぶ
- プロンプトの作法・上手な使い方
- EUではリテラシー義務が発効済み

⑥ ログ・監査

- 誰が・いつ・何を入力したか記録
- 監査可能性を確保し定期レビュー
- 法人製品のログ保持設定も管理対象
- 機密はマスキング・アクセス制御

07 ガバナンス　⑦ インシデント対応 & ⑧ ベンダー審査

⑦ インシデント対応

- 想定：漏洩・誤情報の公表・権利侵害
- 検知→初動→影響評価→報告→再発防止
- 報告窓口を事前に定める
- 当局・本人への通知基準も事前定義

⑧ ベンダー審査

- 学習利用の有無などデータポリシー確認
- 保管地・再委託・第三者提供
- ISO/IEC 42001・SOC 2等の認証
- 契約のデータ取扱条項（DPA）
- 消費者版か法人・API版かを区別

シャドーAI＝承認なき業務AI利用

- IT・ガバナンス部門の承認を経ない業務利用
- 無料版ChatGPT等を個人判断で使う
- シャドーITのAI版

参考：CIO誌（BlackFog調査）／TechTimes 2026-06

07 ガバナンス　シャドーAIの4つの危険

1 機密・個人情報の漏洩

承認外ツールへの入力が学習利用・外部保存で流出

2 検証されない出力

ハルシネーション・著作権侵害物が成果物に紛れる

3 可視性の欠如

IT部門が実態を把握できず、監査・対応が機能しない

4 消費者版の保護不足

法人契約のような非学習・保護の仕組みがない

参考：CIO誌（BlackFog調査）／TechTimes 2026-06

07 ガバナンス　シャドーAI：実態と対策

BlackFog調査 2026（従業員500名以上）

49%
会社の承認なしにAIを利用

51%
IT承認なく業務システムへ接続

69%
経営層が承認外利用を容認

33%
社内データの入力を自認

- **禁止より安全な代替：法人版・社内基盤を提供**
- **許可リスト＋申請簡素化、DLP/CASBで検知**
- **継続教育と経営層自身の順守（容認が蔓延要因）**

参考： cio.com（BlackFog調査）

07 ガバナンス　入力してよい情報・だめな情報

原則OK（機密性なし）

- 公開情報
- 機密でない一般的な業務文
- 社内で匿名化・一般化した内容
- 迷ったら入力前に承認を得る

原則NG

- 個人情報・個人を識別できる情報
- 営業秘密・未公開の技術/財務情報
- 顧客から預かった守秘義務下の情報
- 認証情報（パスワード・APIキー）
- 契約で禁じられるソースコード

参考： 個人情報保護法 ／ 各社データ利用ポリシー

07 ガバナンス　消費者版と法人・API版の違い

同じブランドでもプランで保護レベルが別物

消費者版（無料・Plus）

- 既定で入力が学習に使われ得る
- オプトアウト設定が必要
- 入力は取り消せない前提で運用

法人・API版

- 既定で学習に使わない明記が一般的
- 契約・ログ設定を管理下に置く
- 必ず個別に規約を確認

ベンダー確認5点

- 学習利用の有無（既定設定）
- 保持期間・削除要求の可否
- 保管地・再委託・第三者提供
- DPA・認証の有無
- 消費者版か法人版かの区別

参考： OpenAI Enterprise Privacy 他 各社公式

SECTION 08

ディスカッション

自分ごととして考える。身近なケースでどうすべきかを話し合う

08

08 ディスカッション　進め方

考えるヒント：入力してよい情報か・検証したか・最終責任・相談先・会社のルール

1 **3つのお題、各10分**

近くの3〜4名でグループになって話し合う

2 **正解当てゲームではない**

何が論点か・誰の立場で考えるかを出し合う

3 **時間内で調べてもよい**

Web検索・AI検索を使ってよい

4 **ガバナンスの視点で**

情報の扱い・責任・確認プロセスを意識する

5 **最後に気づきを共有**

2〜3グループが気づきを発表する

08 お題１（10分）　議事録を早く作りたい

シチュエーション

- 取引先との重要な会議、録音データあり
- 早く議事録を作りたい
- 無料の生成AIに文字起こしと要約をさせたい

話し合う3点

- この対応の問題点は？　潜むリスク
- 安全に活用する条件・ツール・手順は？
- 会社側の備えは？　ルール・承認・教育

08 お題 2（10分）　AI回答を提案書にそのまま使う

シチュエーション

- 顧客への提案書を作成中
- AIに業界データと競合情報を質問
- 具体的な数字と出典風のURLが返ってきた
- 締切が近い。そのまま貼り付けたい

話し合う3点

- そのまま使うと？　ハルシネーション・著作権
- 提出前の確認は？　検証プロセスと最終責任
- AI利用自体は問題？　線引きを議論

08 お題 3（10分）　便利な新ツールを見つけた

シチュエーション

- 個人で見つけた海外製の生成AIツール
- 社内データを上げれば自動で分析してくれる
- 同僚にも勧めてチーム業務で使い始めたい

話し合う3点

- 導入前の確認は？　データ・契約・審査
- これはシャドーAIでは？　なぜ問題か
- 正しい進め方は？　申請・承認フローの意味

まとめ　持ち帰ってほしい5つのこと

1 **AIはそれらしさで答える**
人が一次情報で確かめる。鵜呑みにしない

2 **変化は速い、本質は同じ**
モデルは日々更新。原理とリスクの理解が土台

3 **入力＝社外送信と考える**
機密・個人情報は承認された環境だけで扱う

4 **会社のルールで守る**
個人の注意任せにしない。困ったら相談・承認

5 **任せる時代ほど人の判断**
最終責任は人。レビューする力を磨く

APPENDIX　主な出典・参考

数値・モデル名等は2026年7月12日時点。実施時に最新をご確認ください

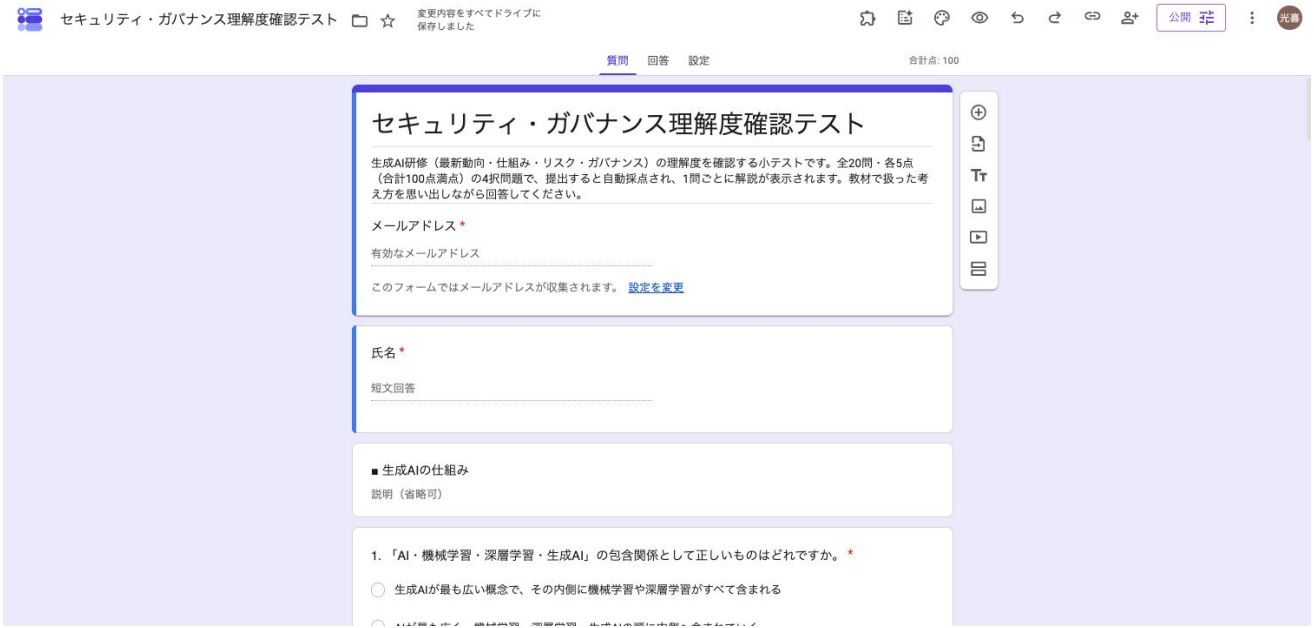
分野	主な出典
AI企業公式	openai.com、anthropic.com、blog.google、x.ai
ニュース	TechCrunch、CNBC、Axios、NPR
リーダーボード	openlm.ai/chatbot-arena、llm-stats.com
研究・科学	MIT News、science.org、Scientific American
開発者向け	github.com、code.claude.com、developers.openai.com
規制・政策	EU AI Act、内閣府/e-Gov（AI推進法）、経産省 AI事業者GL 1.2版、米EO 2026-06
導入事例・調査	Panasonic、LINEヤフー、横須賀市、CIO.com（BlackFog シャドーAI調査）
インシデント	Forbes、CNN、Fortune、AI Incident Database

参考： EU簡素化（Digital Omnibus）や一部の投資・売上数値は流動的・報道ベースを含む

TEST

理解度確認テスト

※ 数値・モデル名等は2026年7月12日時点。更新の速い分野のため、実施時に最新をご確認ください。EUの簡素化(Digital Omnibus)や一部の投資・売上数値は流動的・報道ベースの項目を含みます。



研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

研修のスケジュール　2日目：テスト・デバッグ・実践演習

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：前回振り返りとDay 2概要［30min］ 前日の振り返りクイズ（5問）｜Copilot基本操作の確認｜Day 2ゴール共有｜テスト・デバッグ・総合演習の概要

■Session02：テストコード生成ハンズオン［130min］

- **JUnitテスト自動生成**　/testsコマンドでテストコード自動生成｜テストメソッドの自動命名確認｜正常系・異常系・境界値の網羅｜null入力・空文字・最大値最小値テスト
- **テスト実行と修正**　生成テストを実行して失敗ケースを特定｜テストが通るよう実装側を修正体験｜例外が正しくスローされるかのテスト実践｜テスト品質の確認と改善サイクル
- **デバッグ・リファクタリング実践**　/fixコマンドでバグ修正｜エラーメッセージから原因特定｜長いメソッド・マジックナンバー｜重複コードの共通化体験｜/explainでJavadoc自動生成

■Session03：総合演習 AI活用ミニ開発［150min］

- **演習課題**　簡易設計書（画面仕様書 or API仕様書）配布｜設計書→実装→テスト→レビューの一連の流れ｜Copilot Chat実装｜手作業との所要時間を計測して比較
- **成果共有・振り返り**　数名が画面共有で成果を発表｜「Copilotが役立った場面・邪魔だった場面」を共有｜2日間の学びを全体で振り返り｜配属後の活用イメージを言語化
- **配属後の活用ガイド**　新プロジェクト参画時のコード理解｜定型コード高速生成｜お客様環境での利用可否判断基準｜AIに任せる/自分で書く場面の整理

■Session04：学習リソース紹介・グループディスカッション［80min］

- **学習リソース**　GitHub Copilot公式ドキュメント｜VSCode公式ドキュメント｜JUnit 5ユーザーガイド｜社内の生成AI活用事例・ナレッジ｜今後の研修ロードマップ
- **グループディスカッション**　3～4名で「配属後に使いたい場面」を話し合う｜使えない現場での代替手段を考える｜セキュリティ上の判断フローを整理｜グループ代表者が発表

■Session05： 質疑応答・アンケート・閉会［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

GitHub Copilot 概論

GitHub Copilot 概論 - ゴール

本パートのゴール	GitHub Copilotの機能・特徴・できることを体感する

GitHub Copilot 概論

- GitHub Copilot 概論
  - GitHub Copilotとは？
  - GitHub Copilotにおける機能概要のご紹介
- GitHub Copilotにおける各種機能の演習

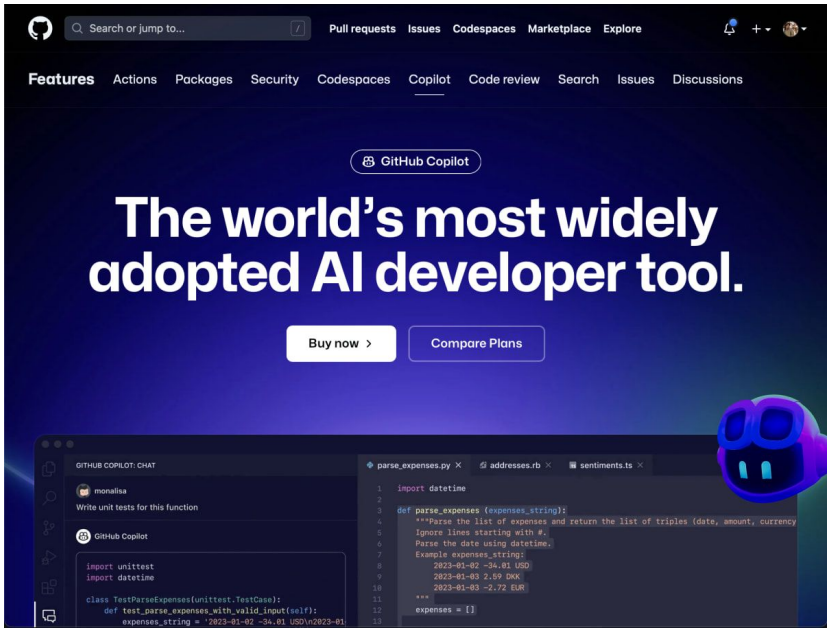
GitHub Copilot 概論

- GitHub Copilot 概論
  - GitHub Copilotとは？
  - GitHub Copilotにおける機能概要のご紹介
- GitHub Copilotにおける各種機能の演習

GitHub Copilot 概論 - GitHub Copilot とは

GitHub Copilot はコーディング作業の効率化を支援するAIツールです。

GitHub Copilot トップページ

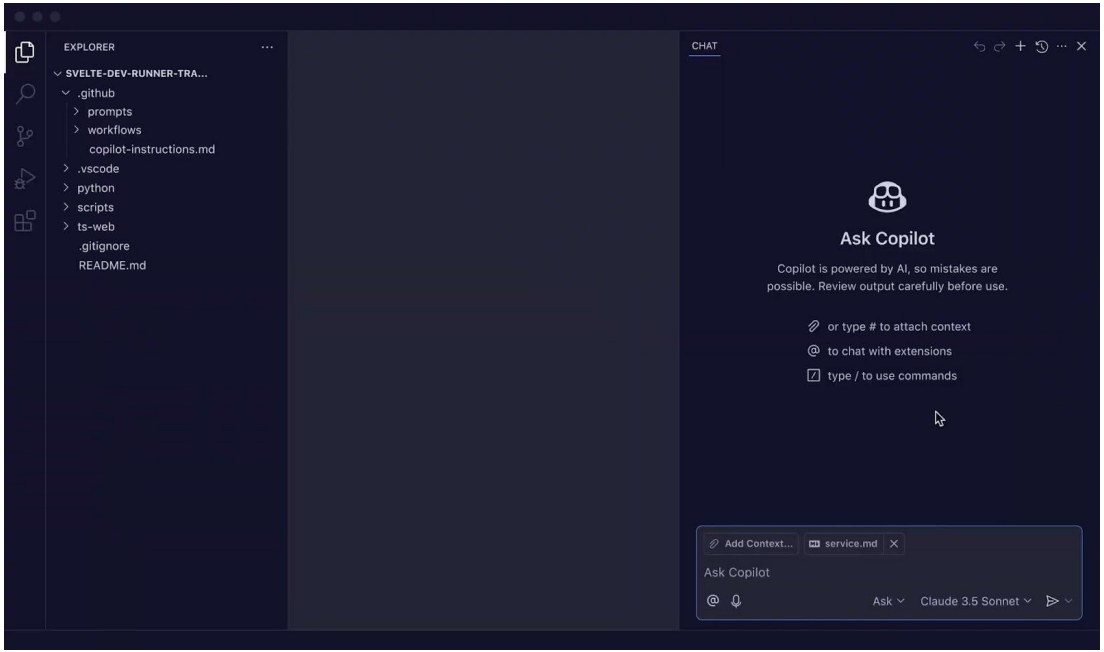


 参考：[GitHub「GitHub Copilot」]

GitHub Copilot 概論 - GitHub Copilot とは

GitHub Copilot はコーディング作業の効率化を支援するAIツールです。

GitHub Copilotのデモ動画

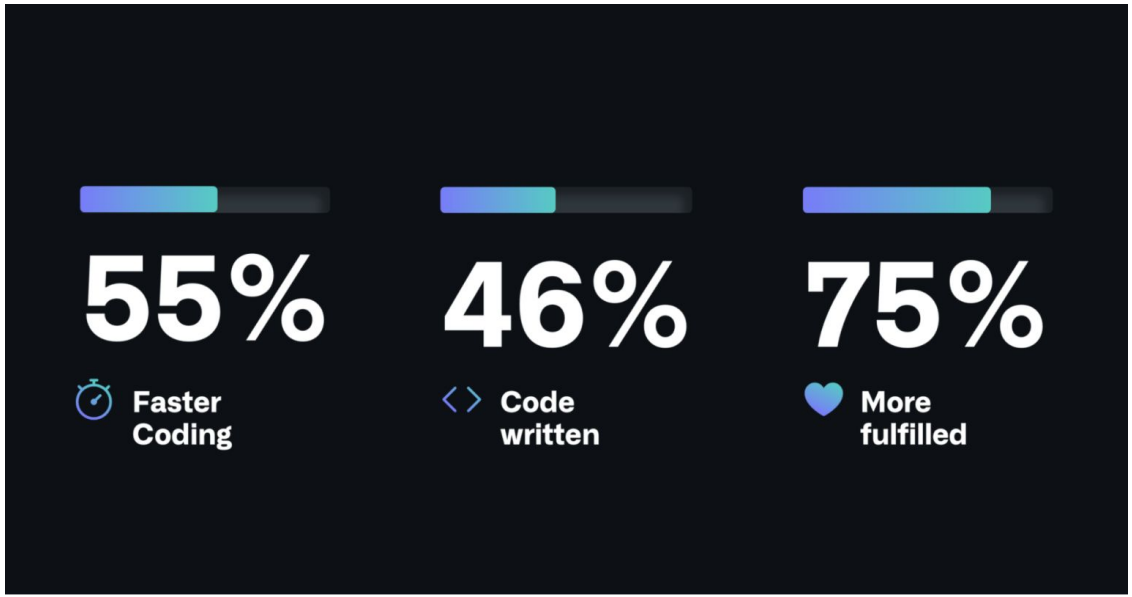


 参考：[GitHub「GitHub Copilot」]

GitHub Copilot 概論 - GitHub Copilot について

GitHub Copilotを使用したグループのほうが、タスクの完了率が高く、
GitHub Copilotを使用しグループは使用しなかったグループよりも55%速くタスクを完了しました。

GitHub Copilotの検証実験結果



 参考：GitHub「AIを活用した開発者ライフサイクルがもたらす経済効果とGitHub Copilotからの学び」

GitHub Copilot 概論 - GitHub Copilot の主な機能一覧

GitHub Copilot はコーディング作業の効率化を支援するAIツールです。

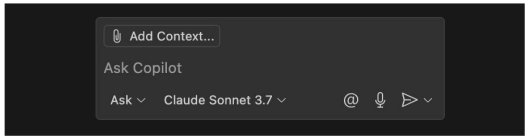
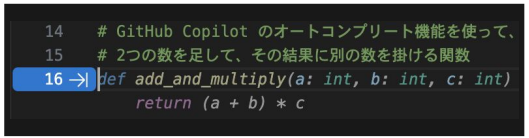
GitHub Copilot の主な機能一覧

機能カテゴリ	概要
Copilot チャット	自然言語で質問・指示（コード説明・テスト生成など）
コード補完	IDE上で次のコードの候補を自動提案（Visual Studio, VS Code など対応）
インラインチャット	作業中のファイルやターミナル内で直接チャットすることが可能
コーディングエージェント	Issue を割り当てるだけで、GitHub Actions が自動実装 → ドラフト PR を生成
拡張機能連携	拡張情報をインストールして外部ツールとCopilot を連携

 参考：GitHub「GitHub Docs」

GitHub Copilot 概論 - 主な機能

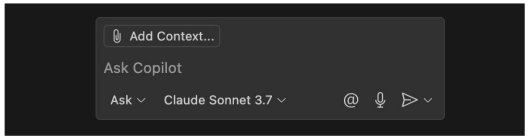
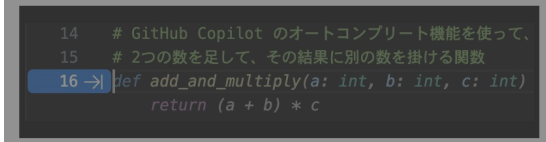
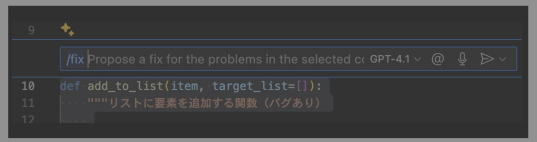
IDEで使用できるGitHub Copilot には3つの機能があります。

	Copilot チャット	コード補完	インラインチャット
概要	チャットベースで質問・指示（コード説明・バグ修正など）を行う	次に入力すべき コードの候補を提示	作業中のファイルまたは ターミナルから 直接チャットを開始する
イメージ			

 参考：Visual Studio Code「Use chat in VS Code」

演習：GitHub Copilot 概論 - 主な機能

IDEで使用できるGitHub Copilot には3つの機能があります。

	Copilot チャット	コード補完	インラインチャット
概要	チャットベースで質問・指示（コード説明・バグ修正など）を行う	次に入力すべき コードの候補を提示	作業中のファイルまたは ターミナルから 直接チャットを開始する
イメージ			

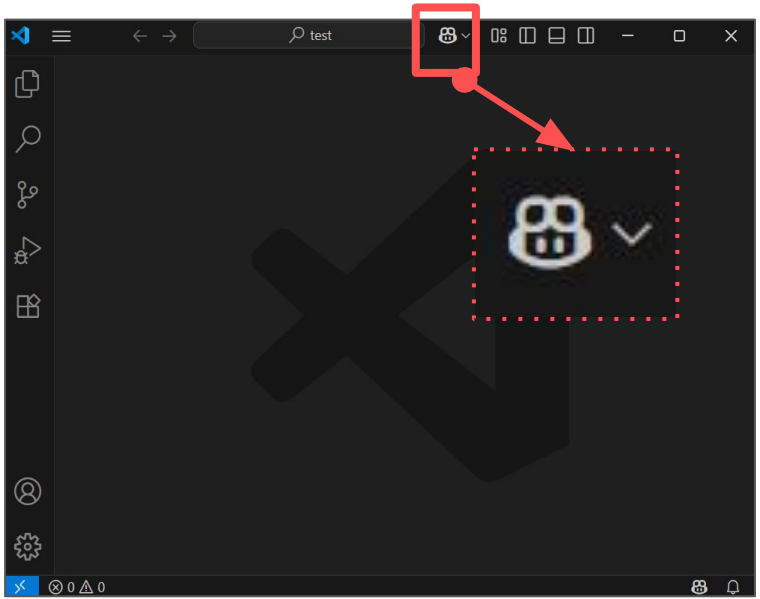
 参考：[Visual Studio Code「Use chat in VS Code」]

GitHub Copilot 概論 - GitHub Copilot チャットの開き方

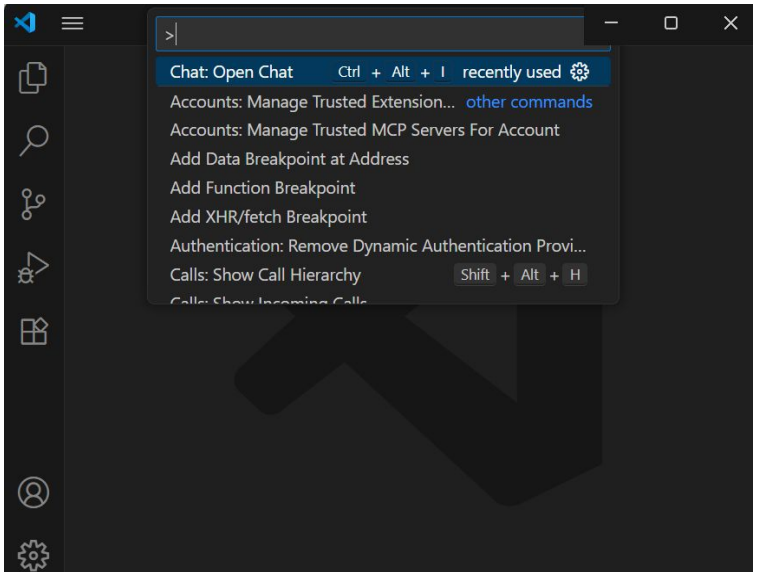
VScode

Copilot チャットはメニューバーかコマンドパレットから開くことができます。

上部メニューバーから開く場合



コマンドパレットから開く場合

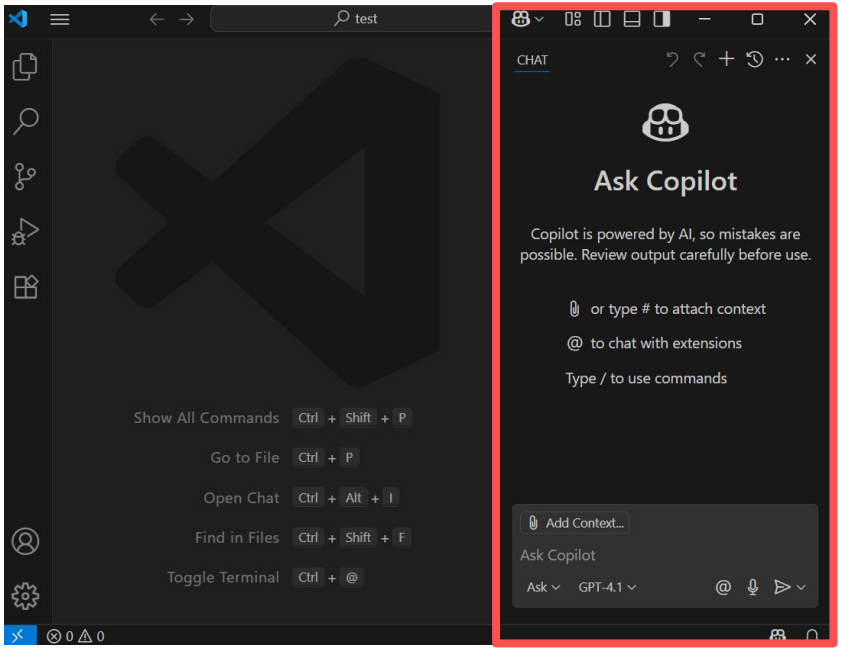


GitHub Copilot 概論 - GitHub Copilot チャットの開き方

VScode

Copilotチャットが開かれるとエディタにチャット入力欄が表示されます。

Copilot チャットの画面

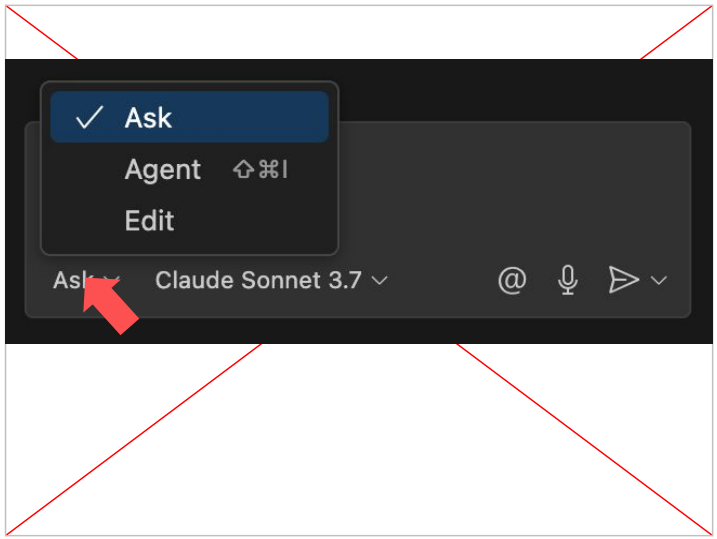


GitHub Copilot 概論 - GitHub Copilot チャットの各選択

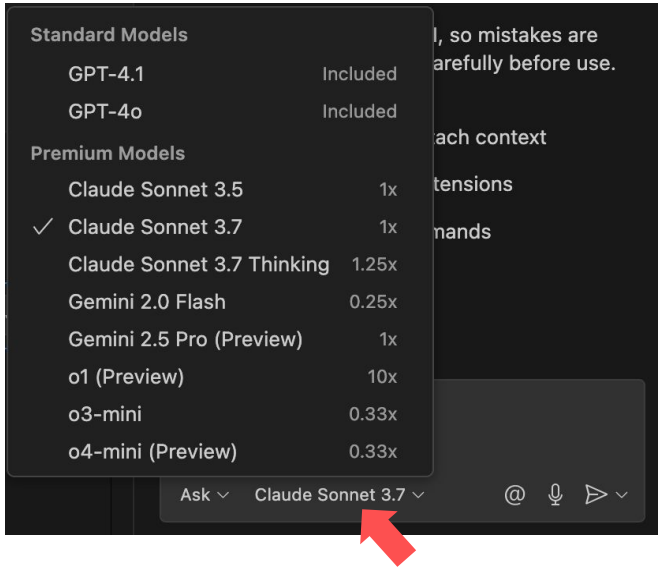
VScode

GitHub Copilot チャットでは、３つのチャットモード（左）と複数のLLMモデル（右）が選択できます。
用途に応じて、それぞれ、使い分けることが可能です。

チャットモード選択



モデル選択



GitHub Copilot 概論 - GitHub Copilot チャットの３つのモード

VScode

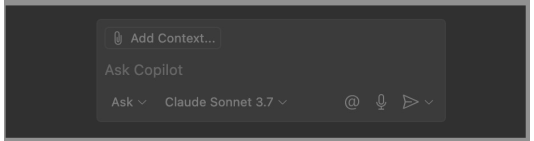
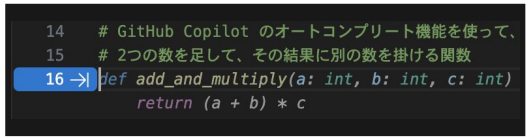
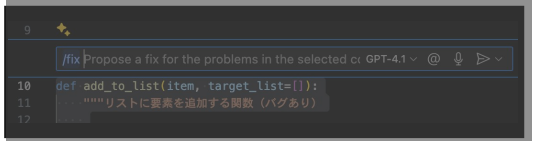
GitHub Copilotチャットでは、3つのモードを活用することで開発効率を最大化することができます。

	Askモード	Editモード	Agentモード
概要	対話型チャットで コード相談・質問解決	指示に基づいて 複数ファイルを一括編集	自律的にプロジェクト 全体のタスクを実行
@/ コマンド	○	×	×
使用 シーン	例:「Pythonについて教えてください」	例:「(複数ファイル選択)Google 形式の docstring を付けて」	例:「Flask で在庫管理のWebアプリケーションを作って」

 参考：GitHub Inc.「Copilot ask, edit, and agent modes: What they do and when to use them」

演習：GitHub Copilot 概論 - 主な機能

IDEで使用できるGitHub Copilot には3つの機能があります。

	Copilot チャット	コード補完	インラインチャット
概要	チャットベースで質問・指示（コード説明・バグ修正など）を行う	次に入力すべき コードの候補を提示	作業中のファイルまたは ターミナルから 直接チャットを開始する
イメージ			

 参考：[Visual Studio Code「Use chat in VS Code」]

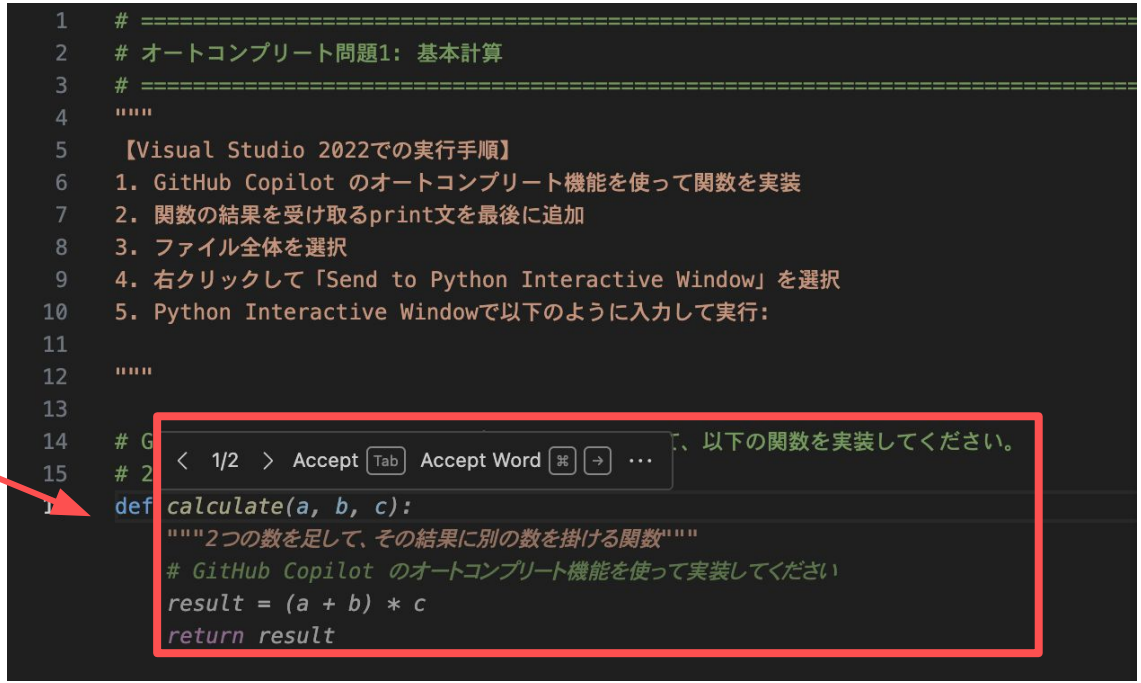
GitHub Copilot 概論 - コード補完機能とは - Visual Studio Code

VScode

AI が次に入力すべきコードを予測し、オートコンプリート形式で候補を提示してくれます。

“def” まで入力すると、入力候補となるコードが提示される

ここでTabキーを押すと、提案を受け入れる（Accept）



 参考：GitHub「GitHub Docs」

GitHub Copilot 概論 - コード補完機能とは - Visual Studio Code

VScode

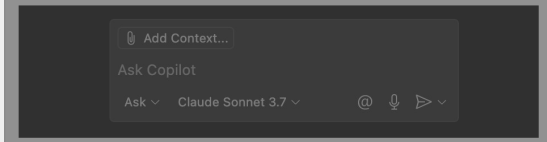
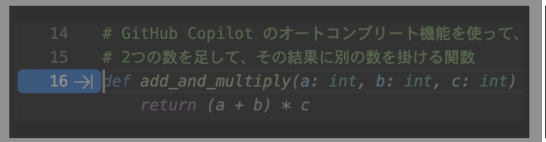
AI が次に入力すべきコードを予測し、オートコンプリート形式で候補を提示してくれます。

コード補完機能の操作方法

コマンド操作	macOS	Windows
提案を受け入れる	Tab	Tab
提案を拒否する	Esc	Esc
次の提案を表示	Option (⌥) +]	Alt +]
前の提案を表示	Option (⌥) + [	Alt + [
次の単語を受け入れる	Command (⌘) + →	Control + →
複数の提案を含む新しいタブを開く	Ctrl + Enter	Ctrl + Enter

演習：GitHub Copilot 概論 - 主な機能

IDEで使用できるGitHub Copilot には3つの機能があります。

	Copilot チャット	コード補完	インラインチャット
概要	チャットベースで質問・指示（コード説明・バグ修正など）を行う	次に入力すべきコードの候補を提示	作業中のファイルまたはターミナルから直接チャットを開始する
イメージ			

 参考：[Visual Studio Code「Use chat in VS Code」]

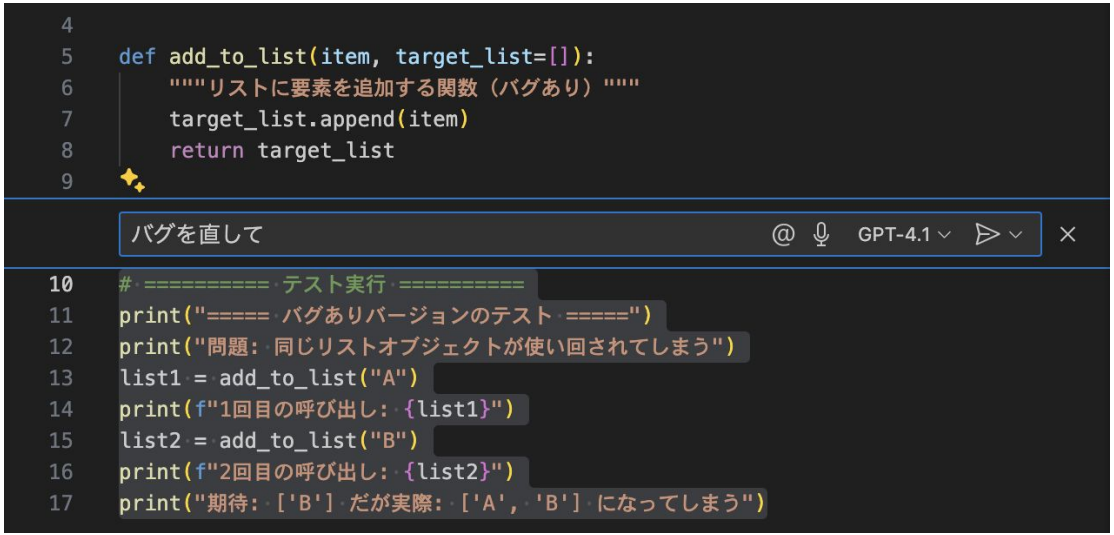
GitHub Copilot 概論 - インラインチャットとは - Visual Studio Code

VScode

エディタ内でCopilotに質問・修正を依頼できる機能です。

コードを範囲選択 → ショートカット OR 右クリック「Copilot に質問する」を選択

- **例）「この関数を非同期化して」「コメントアウトをつけて」などを即時リクエスト**
- **ショートカット：Ctrl + I (Windows)**



GitHub Copilot 概論

- GitHub Copilot 概論
  - GitHub Copilotとは？
  - GitHub Copilotにおける機能概要のご紹介
- **GitHub Copilotにおける各種機能の演習**

ディスカッション②

『AGIに到達したか？』

AGI（Artificial General Intelligence）：汎用人工知能
→人類と同じレベルの知能や汎用性をもったAI

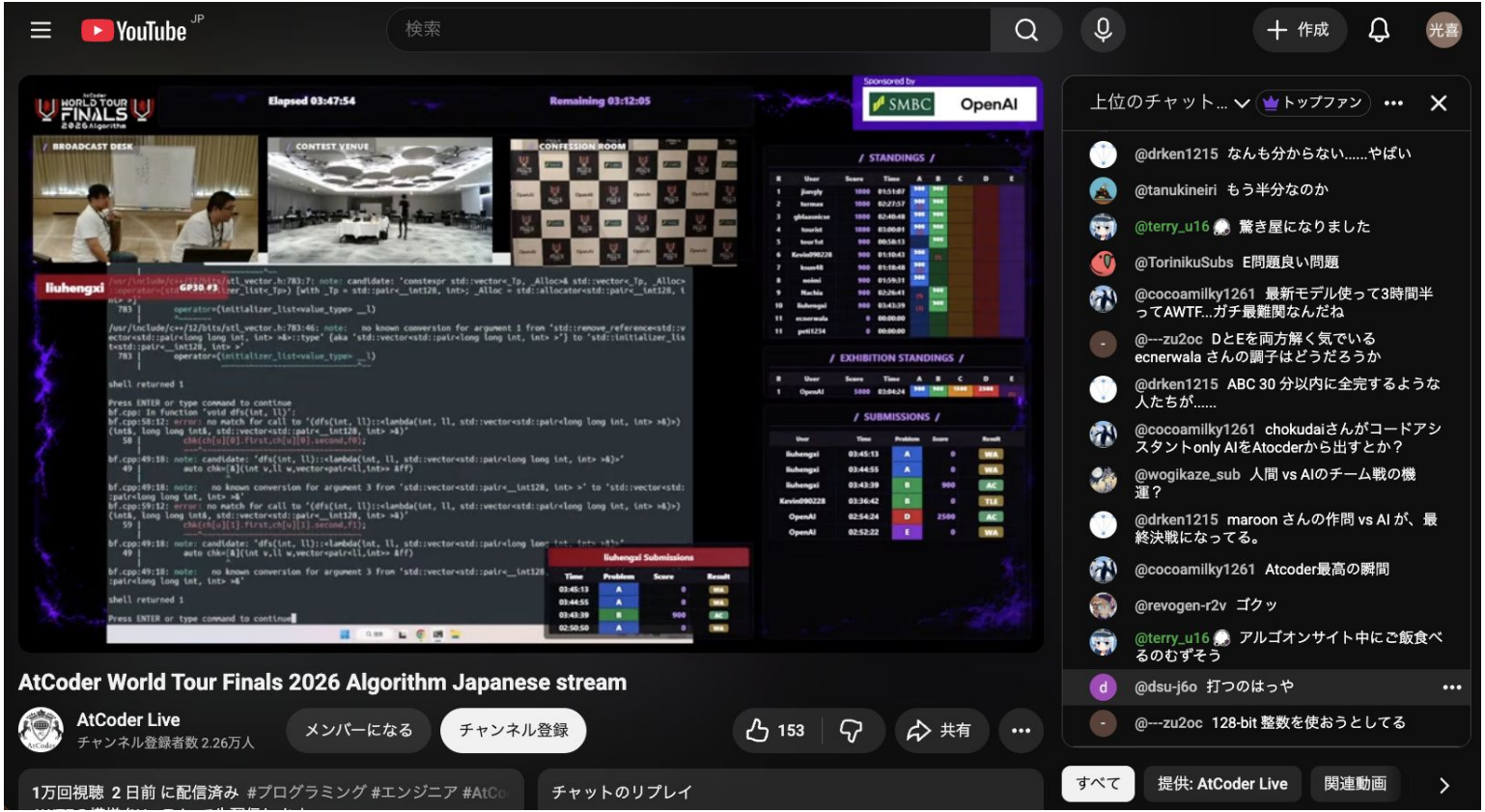
AtCoder World Tour Finals 2026でAIが人間に完全勝利(2026.07.09)

#	名前	得点	時間	A (900点)	B (900点)	C (1500点)	D (2500点)	E (2500点)
	OpenAI AI	8300点	6:30:36	30:05 -	32:11 -	28:48 -	2:54:24 (誤答 2)	5:55:36 (誤答 5)
1	jiangly 人類	1800点	1:51:07	27:21 -	1:51:07 -			
2	turmax 人類	1800点	2:27:57	1:42:16 (誤答 1)	2:22:57 -			
3	gblaasnicse 人類	1800点	2:40:48	2:40:48 -	42:16 -			
4	tourist 人類	1800点	3:00:01	2:50:01 (誤答 2)	46:45 -			
5	Kevin090228 人類	1800点	3:52:45	1:10:43 -	3:47:45 (誤答 1)		(誤答 1)	
6	liuhengxi 人類	1800点	4:15:08	3:55:08 (誤答 4)	3:43:39 -			
7	tour1st 人類	900点	0:58:13		58:13 -		(誤答 2)	
8	ksun48 人類	900点	1:18:48	1:08:48 (誤答 2)				
9	noimi 人類	900点	1:59:31	1:59:31 -				
10	Nachia 人類	900点	2:26:41	(誤答 2)	2:26:41 -			
11	peti1234 人類	0点	0:00:00	(誤答 1)				
11	ecnerwala 人類	0点	0:00:00					

※正解した問題の誤答 1 件あたり、ペナルティとして 5 分が加算されます

https://info.atcoder.jp/more/contents/awtf/2026?_gl=1*19hw7ri*_ga*MTY2MjgxNDEyMC4xNzczODgyNDM4*_ga_RC512FD18N*czE3ODM4MTI2MzIkbzYkZzAkdDE3ODM4MTI2MzIkajYwJGwwJGgw

AtCoder World Tour Finals 2026でAIが人間に完全勝利(2026.07.09)



https://info.atcoder.jp/more/contents/awtf/2026?_gl=1*19hw7ri*_ga*MTY2MjgxNDEyMC4xNzczODgyNDM4*_ga_RC512FD18N*czE3ODM4MTI2MzIkbzYkZzAkdDE3ODM4MTI2MzIkajYwJGwwJGgw

ChatGPT、Gemini東大理IIIに主席合格(2026.LifePrompt様調べ)

東大 首席比較表

LifePrompt

科類	配点	ChatGPT 5.2 Thinking	Claude 4.5 Opus	Gemini 3 Pro Preview
理科一類	550	503.59	451.99	496.54
理科二類	550	503.59	451.99	496.54
理科三類	550	503.59	451.99	496.54
文科一類	550	452.70	426.18	460.07
文科二類	550	452.70	426.18	460.07
文科三類	550	452.70	426.18	460.07

※オレンジの数値は、各科類の合格者最高点を超えていることを示します。

※配点は、各科類の合計点（満点）を示します。

【各科類の合格者最高点（参考）】

- 理科一類：443.28
- 理科二類：396.85
- 理科三類：453.60
- 文科一類：430.13
- 文科二類：420.85
- 文科三類：434.96

※上記データは東京大学が公表した2026年度一般選抜（前期日程）における合格者の実績に基づきます。

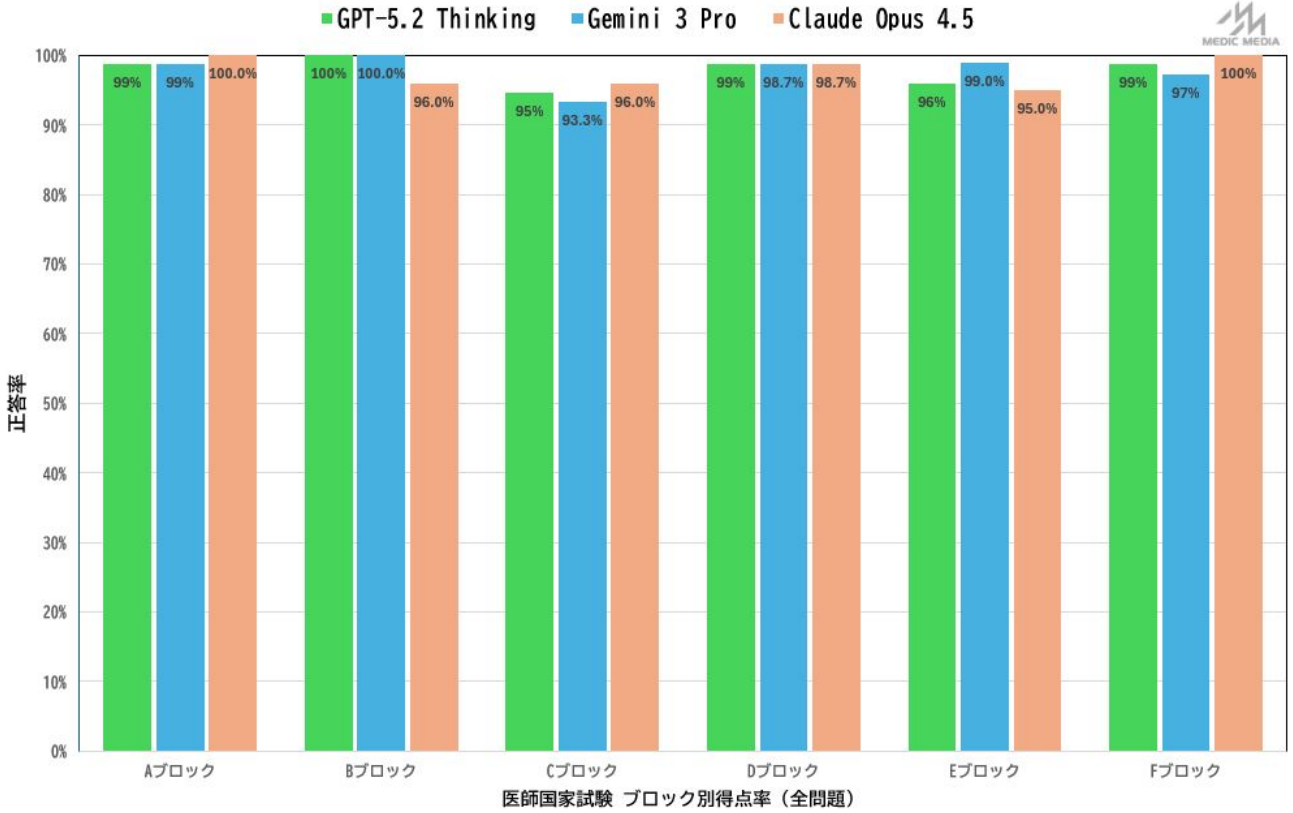
https://note.com/lifeprompt/n/n85674c186fbc

医師国家試験であらゆるAIが95%以上(2026)

	必修		一般	
	得点	割合	得点	割合
GPT-5.2 Thinking	196点/200点	98.0%	292点/300点	97.3%
Gemini 3 Pro	199点/200点	99.5%	290点/300点	96.7%
Claude Opus 4.5	191点/200点	95.5%	295点/300点	98.3%

https://informa.medilink-study.com/web-informa/post51586.html/

医師国家試験であらゆるAIが95%以上(2026)



https://informa.medilink-study.com/web-informa/post51586.html/

2025-2026で加速した生成AI分野

・Agentic AI（AIエージェント）の実用化（2025-2026の最大トレンド）
単に質問に答えるだけでなく、目標を与えて複数ステップのタスクを自律的に実行するAI。ITサポート、研究、顧客対応、ワークフロー自動化などで広く導入。Gartnerが2025年のトップ戦略技術に挙げ、企業での採用が急増。企業全体戦略として本格化し、ROI（投資収益率）向上や業務効率化が実証され始めた。

・生成動画（Generative Video）の成熟
テキストや画像から高品質動画を生成する技術（Sora、Veo、Klingなどの進化版）が実用レベルに。映画・広告・教育・エンタメで活用爆増。リアルタイム性や音声同期も向上し、クリエイティブ業界の生産性が劇的に変わった。

・ヘルスケア・ライフサイエンス
薬剤発見の加速（分子構造提案・シミュレーション）、画像診断の精度向上、パーソナライズド治療、仮想細胞モデルなど。AIが新薬開発タイムラインを大幅短縮。医療機器の知能化も進む。

・製造業・ロボティクス（特にヒューマノイドロボット）
AI統合ヒューマノイドロボット（Tesla Optimus、Boston Dynamics Atlas、Figureなど）が工場実装やフィールドテスト段階に。労働力不足対策として注目。家事支援やサービス用途への展開も始まる。

https://informa.medlink-study.com/web-informa/post51586.html/

『AGIに到達したか？』

研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

GitHub Copilot の使い方

GitHub Copilot の使い方 - ゴール

本パートのゴール	GitHub Copilot を使って生成AIネイティブ開発をスタートできる状態になる

GitHub Copilot の使い方

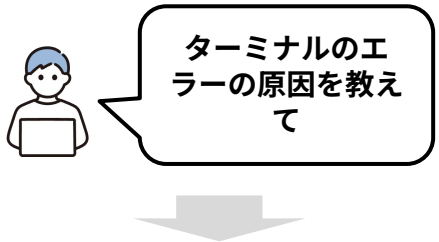
- 各種コマンド（@コマンド、#コマンド、/コマンド）の全体像のご説明
  - 『@コマンド』の説明と演習
  - 『#コマンド』の説明と演習
  - 『/コマンド』の説明と演習
- Custom Instructionsの説明と演習

GitHub Copilot の使い方

- 各種コマンド（@コマンド、#コマンド、/コマンド）の全体像のご説明
  - 『@コマンド』の説明と演習
  - 『#コマンド』の説明と演習
  - 『/コマンド』の説明と演習
- Custom Instructionsの説明と演習

GitHub Copilot の使い方 - GitHub Copilot の3種類のコマンド

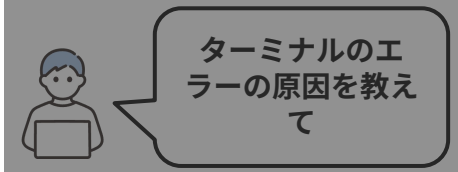
GitHub Copilotには3種類の基本コマンドがあり、それぞれ異なる用途で指示を最適化できます。

	@コマンド	#コマンド	/コマンド
概要	質問対象の スコープを指定する	Copilot に ツールの使用を指示する	特定の指示を ショートカットとして 利用する
使用例	 @vscode エディタの言語を日本語にして	 #terminalLastCommand エラーの原因を教えて	 @workspace /explain

参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - GitHub Copilot の3種類のコマンド

GitHub Copilotには3種類の基本コマンドがあり、それぞれ異なる用途で指示を最適化できます。

	@コマンド	#コマンド	/コマンド
概要	質問対象の スコープを指定する	Copilot に ツールの使用を指示する	特定の指示を ショートカットとして 利用する
使用例	 @vscode エディタの言語を日本語にして	 #terminalLastCommand エラーの原因を教えて	 @workspace /explain

参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - @コマンド

VScode

Copilot Chat では、質問に正確なコンテキストを与えることで、より的確な回答を得られます。

Copilot Chat の @コマンド一覧と用途

コマンド名	機能内容
`@workspace`	ワークスペース全体に関する質問
`@vscode`	VS Code の使い方や設定を質問
`@terminal`	ターミナル操作に関する質問
`@github`	GitHub 固有の Copilot スキルを使用可能

参考：GitHub「GitHub Docs」

演習：GitHub Copilot の使い方 - 演習：@コマンド

VScode

演習内容

@terminalを実践してみましょう！

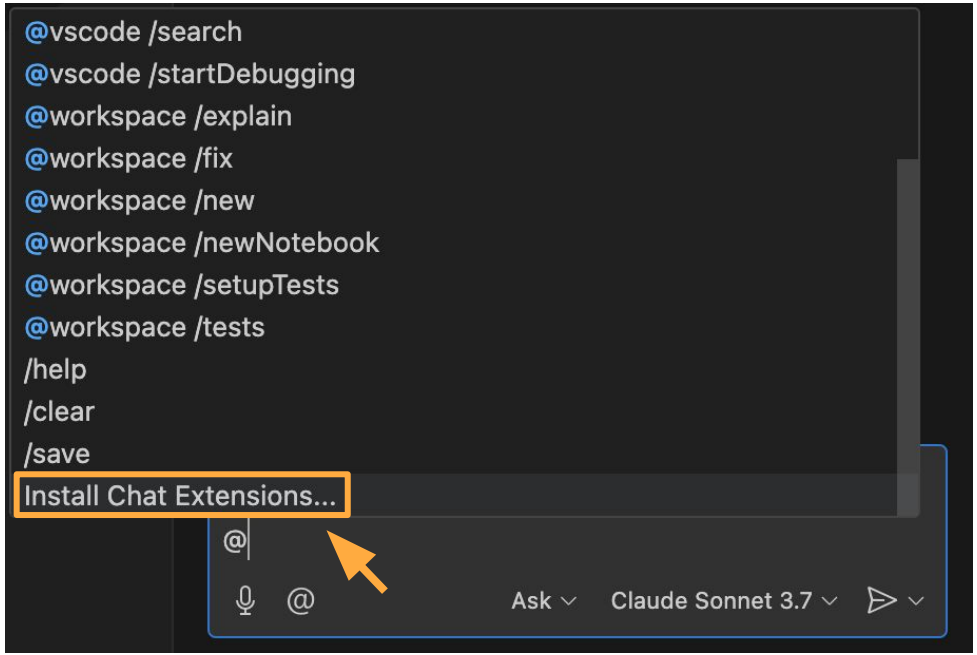
演習手順

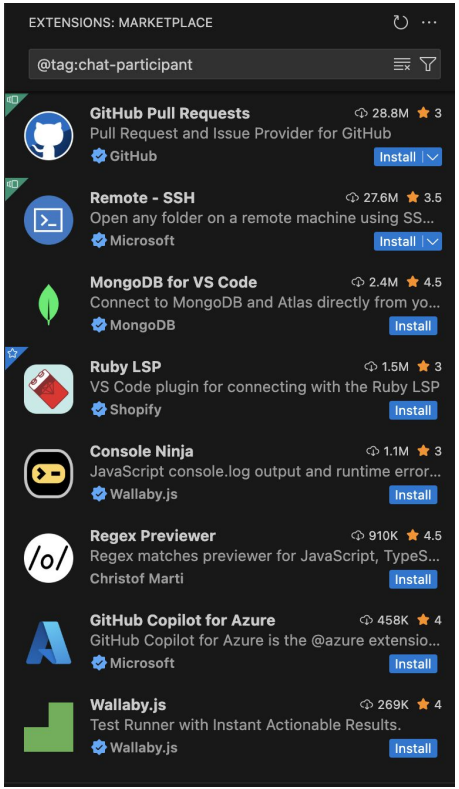
1. cd/src/work_exercise/fix をターミナルに入力
2. python exercise_bug_2.py を実行
3. エラーを確認
4. Copilotチャットで「@terminal /explain」を入力

GitHub Copilot の使い方 - @コマンドの補足

VScode

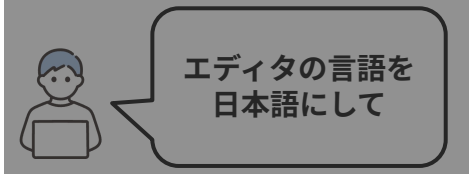
@コマンドには豊富な拡張機能があります。





GitHub Copilot の使い方 - GitHub Copilot の3種類のコマンド

GitHub Copilotには3種類の基本コマンドがあり、それぞれ異なる用途で指示を最適化できます。

	@コマンド	#コマンド	/コマンド
概要	質問対象の スコープを指定する	Copilot に ツールの使用を指示する	特定の指示を ショートカットとして 利用する
使用例	 @vscode エディタの言語を日本語にして	 #terminalLastCommand エラーの原因を教えて	 @workspace /explain

参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - #コマンド一覧

VScode

コマンドでCopilot に使用する“ツール”を与えることができます。

#コマンド別 機能概要一覧

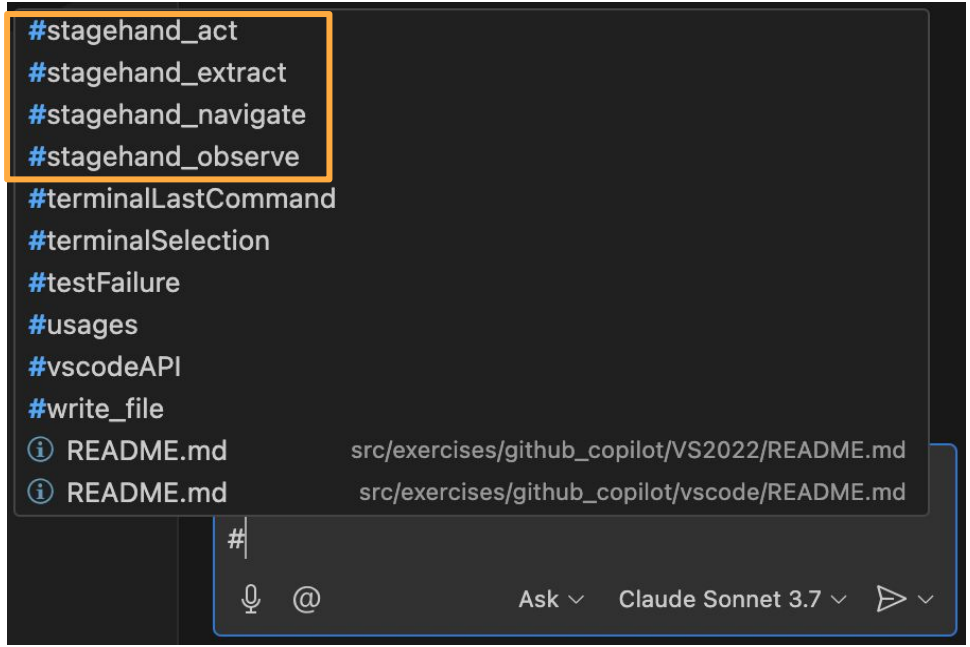
コマンド	説明
#changes	変更されたファイルの差分を取得
#codebase	ワークスペース全体を検索
#editFiles	指定したファイルを編集
#extensions	VS Code Marketplace で拡張機能を検索
#findTestFiles	テストファイル群を検索
#githubRepo	GitHub リポジトリで関連コードを検索
#new	空ディレクトリに新規ワークスペースの土台を作成
#openSimpleBrowser	Simple Browser でローカルサイトをプレビュー
#usages	シンボルの参照・定義・使用箇所を一覧取得

コマンド	説明
#problems	特定のファイルのエラーを確認
#runCommands	ターミナルでコマンドを実行
#runNotebooks	Notebook セルを実行
#runTasks	ワークスペースでタスクを実行
#searchResults	検索ビューからの結果
#terminalLastCommand	アクティブなターミナルの最後の実行コマンド
#terminalSelection	アクティブなターミナルの選択
#testFailure	前回の単体テストの失敗情報
#vscodeAPI	VS Code API リファレンスを元に回答

GitHub Copilot の使い方 - #コマンドの補足

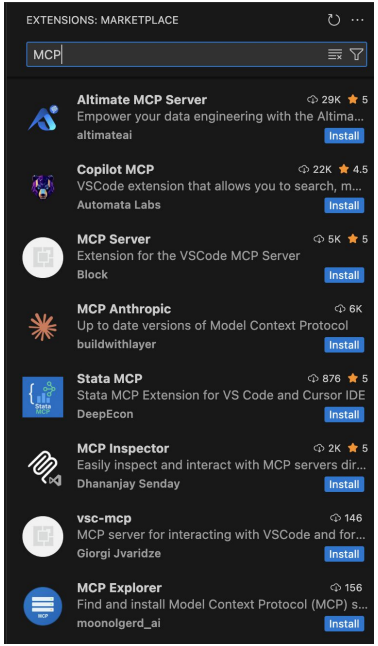
VScode

拡張機能から様々なMCPサーバーを追加することが可能です。
追加すると、#コマンドにMCPツールが表示されます。



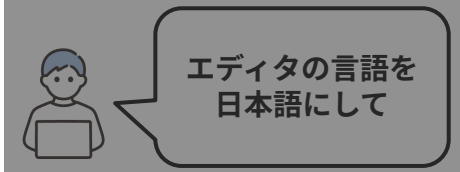
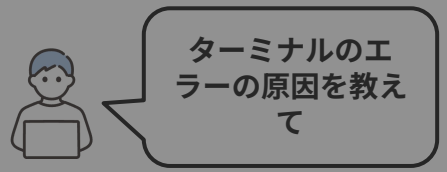
※1
MCP：Agentとツール間でコンテキストを標準化してやり取りする通信プロトコル

※2
拡張機能を追加する前に、必ず開発元を確認して、不明な拡張機能は追加しないことを推奨します



GitHub Copilot の使い方 - GitHub Copilot の3種類のコマンド

GitHub Copilotには3種類の基本コマンドがあり、それぞれ異なる用途で指示を最適化できます。

	@コマンド	#コマンド	/コマンド
概要	質問対象の スコープを指定する	Copilot に ツールの使用を指示する	特定の指示を ショートカットとして 利用する
使用例	 @vscode エディタの言語を日本語にして	 #terminalLastCommand エラーの原因を教えて	 @workspace /explain

参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - Copilot Chat での / コマンド

VScode

/コマンドを使用することでCopilotがタスクを正確に理解し、適切な回答を提供します。

Copilot Chat での『/コマンド』一覧

#	コマンド	機能	対象
1	`/explain`	コードやターミナルの内容を説明	`@workspace, @terminal`
2	`/fix`	コードの修正案を提案	`@workspace`
3	`/doc`	コメントを付与	インラインチャットのみ
4	`/tests`	単体テストを生成	`@workspace`
5	`/search`	ワークスペース内検索クエリの生成	`@vscode`
6	`/startDebugging`	デバッグ設定を作成して開始（実験的）	`@vscode`
7	`/new`	ファイルツリー構造の提案	`@workspace`
8	`/newNotebook`	Jupyter Notebook を作成	`@workspace`
9	`/setupTests`	プロジェクトのテストを設定する	`@workspace`
10	`/help`, `/clear`, `/save`	ユーティリティ系操作	–

参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - Copilot Chat での / コマンド

VScode

/explain と /tests コマンドの解説と演習をさせていただきます。

Copilot Chat での『/コマンド』一覧

#	コマンド	機能	対象
1	/explain	コードやターミナルの内容を説明	@workspace, @terminal
2	/fix	コードの修正案を提案	@workspace
3	/doc	コメントを付与	インラインチャットのみ
4	/tests	単体テストを生成	@workspace
5	/search	ワークスペース内検索クエリの生成	@vscode
6	/startDebugging	デバッグ設定を作成して開始（実験的）	@vscode
7	/new	ファイルツリー構造の提案	@workspace
8	/newNotebook	Jupyter Notebook を作成	@workspace
9	/setupTests	プロジェクトのテストを設定する	@workspace
10	/help, /clear, /save	ユーティリティ系操作	–

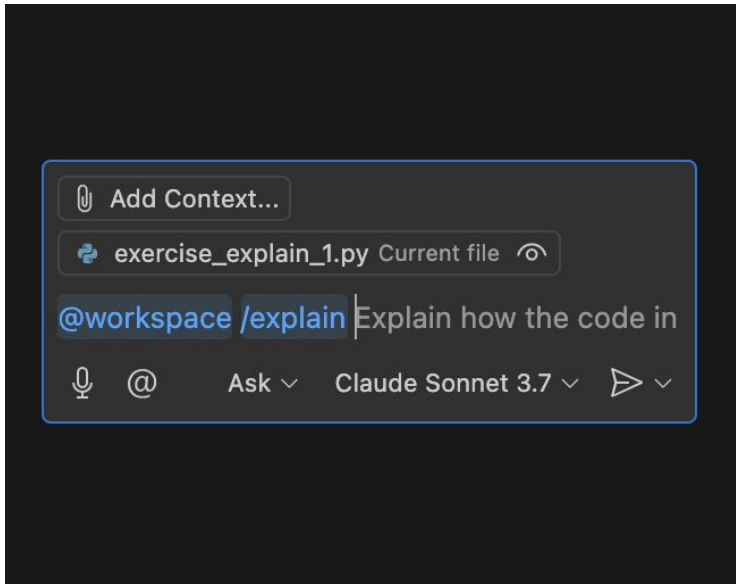
参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - /explain の使い方

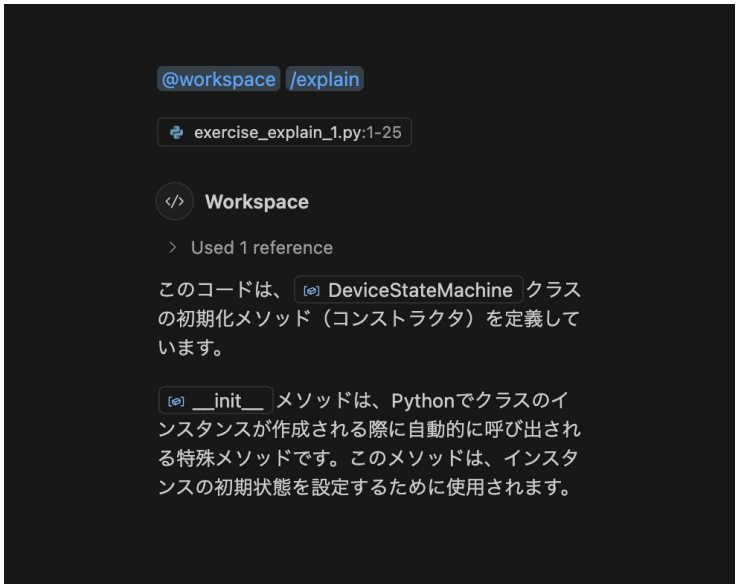
VScode

『/explain』 コマンドでは、コードの内容を説明させることができます。

『/explain』コマンドの実行イメージ



実行



※Copilot チャットでは@workspace や @terminal と /explain はセットで使用します。

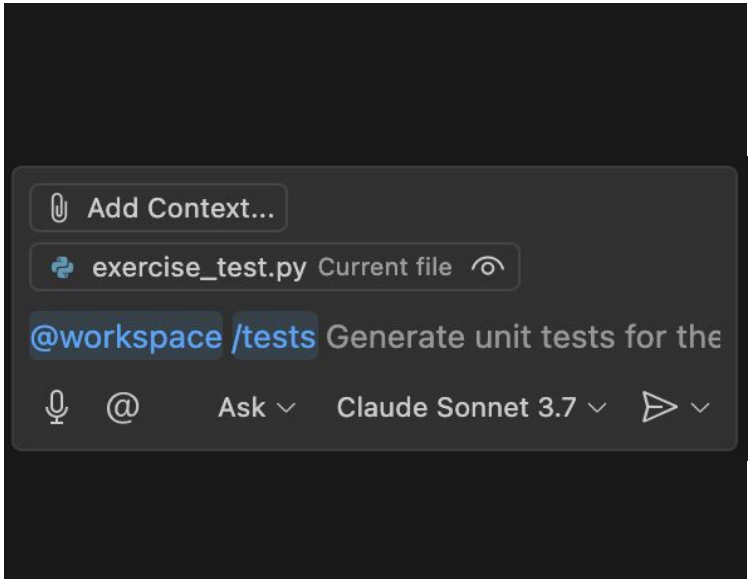
参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - /tests コマンド の使い方

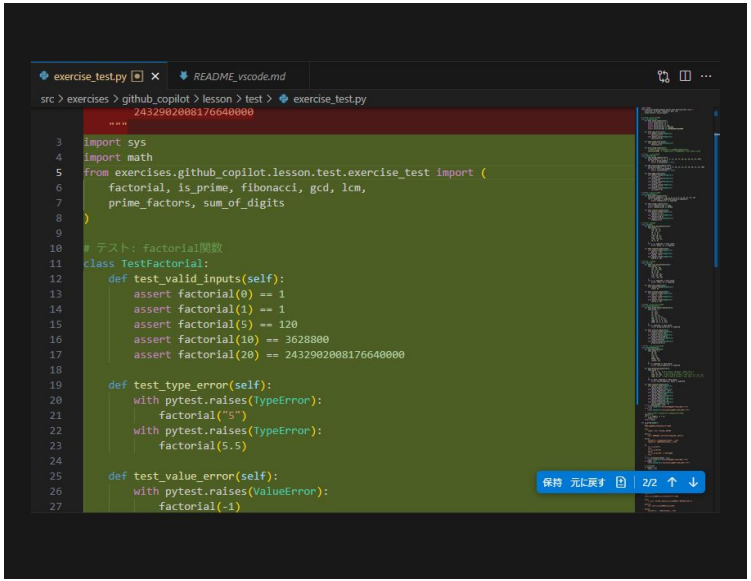
VScode

『/tests』コマンドでは、選択したコードやファイルのテストを実行することができます。

『/tests』コマンドの実行イメージ



実行



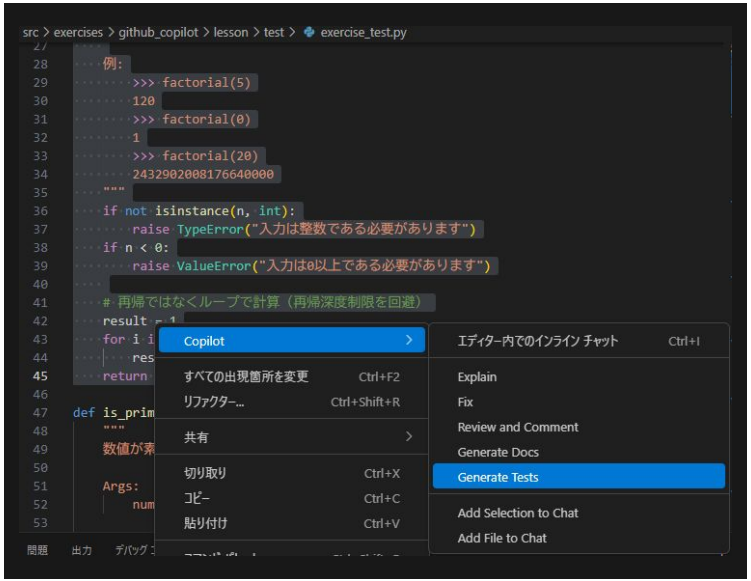
参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - /tests コマンド の使い方

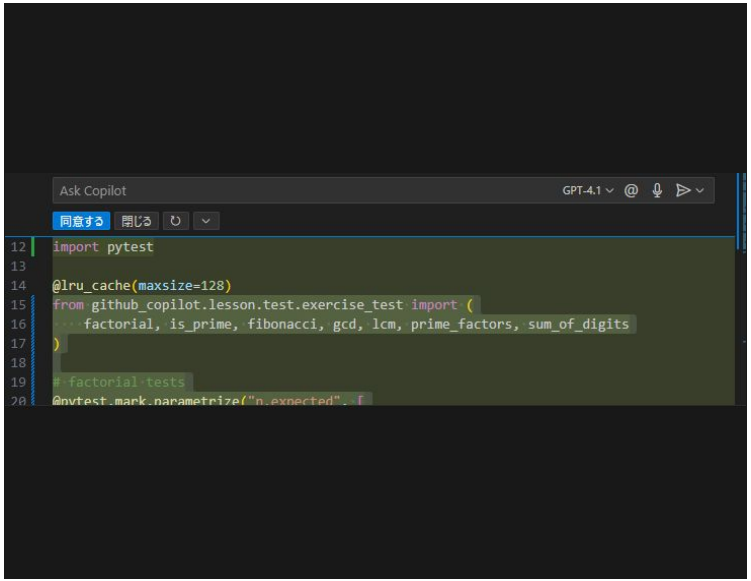
VScode

コードを範囲選択して、右クリック → Generate Tests を実行します。

『/tests』コマンドの実行イメージ



実行



例： インラインチャットでGenerate Test を実行

参考：GitHub「GitHub Docs」

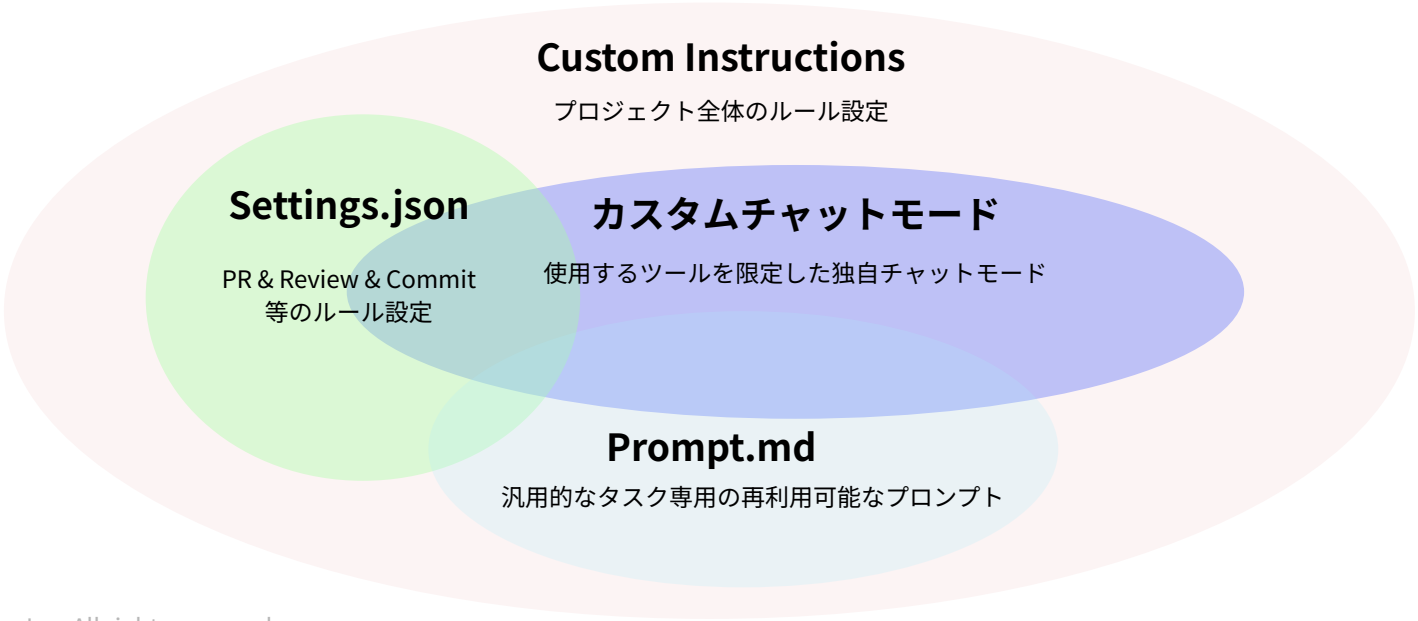
生成AI概論

- 各種コマンド（@コマンド、#コマンド、/コマンド）の全体像のご説明
  - 『@コマンド』の説明と演習
  - 『#コマンド』の説明と演習
  - 『/コマンド』の説明と演習
- **Custom Instructionsの説明と演習**

GitHub Copilot の使い方 - Copilot のカスタマイズ：全体のイメージ

4つのカスタマイズ機能により、プロジェクト全体のルールから個別タスクまで、チームの開発スタイルに合わせたAIコーディング環境を構築できます。

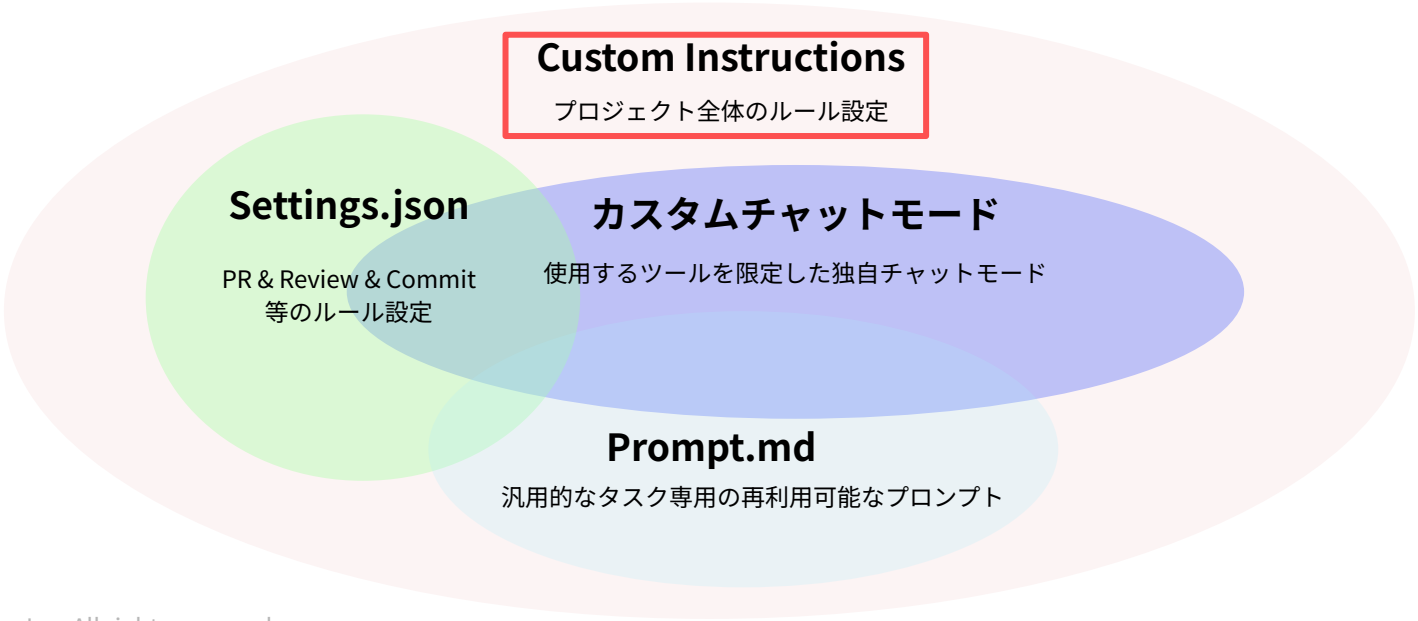
GitHub Copilot のカスタマイズ機能の全体像



GitHub Copilot の使い方 - Copilot のカスタマイズ：全体のイメージ

4つのカスタマイズ機能により、プロジェクト全体のルールから個別タスクまで、チームの開発スタイルに合わせたAIコーディング環境を構築できます。

GitHub Copilot のカスタマイズ機能の全体像



GitHub Copilot の使い方 - Custom Instructions

GitHub CopilotのCustom Instructionsとは
AIの提案をプロジェクトやチームのルールに合わせてカスタマイズできる機能です。

GitHub Copilot - Custom Instructions 公式ドキュメント説明

参考：[GitHub Docs「GitHub Copilot のリポジトリ カスタム命令を追加する」]

GitHub Copilot の使い方 - Custom Instructionsの活用シーン

**Custom Instructionsを活用することでチーム開発から個人開発まで、
様々なシーンで開発の生産性を向上させることができます。**

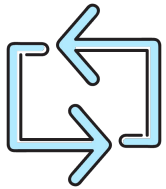
チーム内での設定

**チームのコーディング規約や
使用技術、プロジェクト固有の
情報を理解させる**

個人最適化

**個人の好み（応答言語、スタイル
など）を反映させる**

コンテキスト繰り返し利用



**繰り返し同じコンテキストを
チャットで説明する手間を省く**

GitHub Copilot の使い方 - Custom Instructionsの設定方法

リポジトリカスタム指示：.github/copilot-instructions.md を設定する

	説明
ファイル形式	• ルートに .github/copilot-instructions.md を作成（.github フォルダが無ければ新規作成）。
記述形式	• Markdown形式で、自然言語で指示を記述。 • 指示は短く、自己完結型の文章が望ましい。 • 読みやすさのために空白行で区切っても、段落としてまとめてもOK。
前提条件	• ファイルが存在する状態で、VS Code／Visual Studio などの 「カスタム指示を有効にする」 オプションを ON（既定は OFF）。 • ※設定画面に該当チェックボックスあり。
利用確認方法	• Copilot Chat で質問 → 応答下部 References に copilot-instructions.md が表示されていれば適用完了。

参考：GitHub「GitHub Docs」

演習：GitHub Copilot の使い方 - .github/copilot-instructions.md

次のシナリオを読み、`.github/copilot-instructions.md` に、どのような情報を書けば Copilot の提案精度を高められるかを実践してみましょう。

プロンプト例

`.github/copilot-instructions.md` を以下の参考資料を元にして作成してください。

参考資料：

- Zoom のチャットにテキストを貼り付けます。

GitHub Copilot の使い方 - 複数のCustom Instructions設定

用途別にCustom Instructionsファイルを作成して、Agentに参照させることができます

Custom Instructions の例

```
.github/
    instructions/
        general.instructions.md
        nextjs15app.instructions.md
        api_document.instructions.md
        progress.md
```

解説

- 複数 instructions.md を活用
  - フレームワーク別・言語別・API仕様別に用意
  - Agentが適切な情報を自動で読み込み、毎回の指示入力が不要
- progress.md で進捗を可視化
  - 完了タスク／残タスクを簡潔に更新
  - チャットウィンドウを替えても Agent が最新状況を継承
- 得られるメリット
  - チーム内外で同じドキュメントを共有でき、認識ズレを防止
  - 新メンバーもファイルを見るだけで背景を把握
  - 会話のたびに情報を引き継ぐため、開発効率と回答精度が向上

参考：Visual Stuido Code「April 2025 (version 1.100)」

研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

ガバナンス基礎

ガバナンス基礎 - ゴール

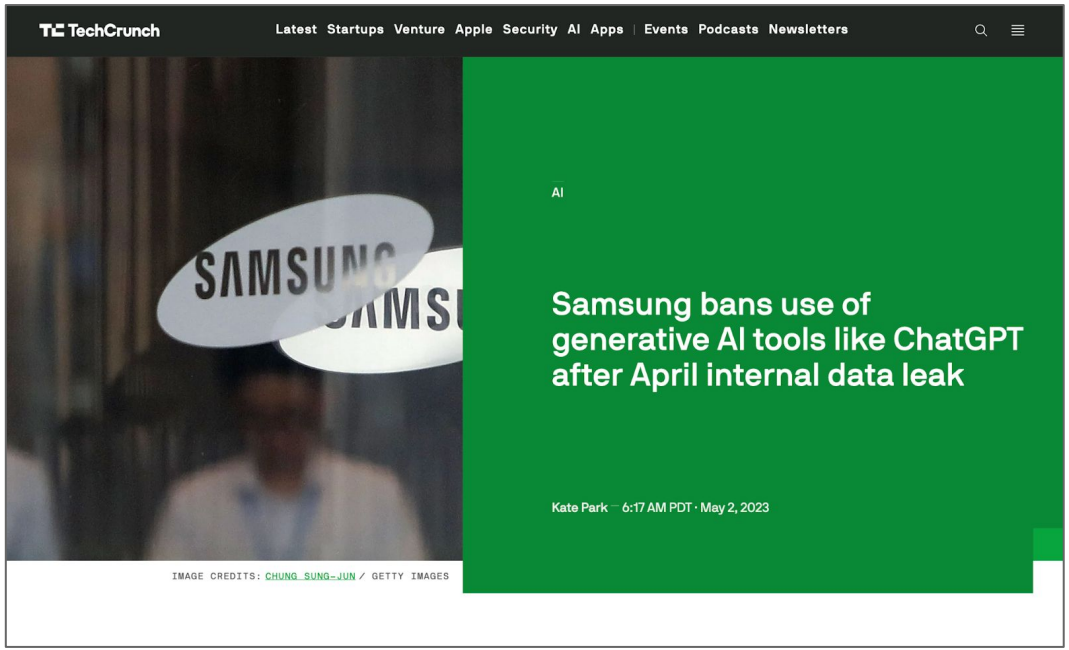
本パートのゴール	AIコーディングにおける注意事項と その対策を抑える

ガバナンス基礎 - 項目

- 事例紹介：Samsung社員がChatGPTに社内機密を誤送信
- GitHub Copilot によるセキュリティ対策
- AI と著作権について

ガバナンス基礎 - 事例：Samsung社員がChatGPTに社内機密を誤送信

生成AIの明確なルール未整備によって、対応が後手に回った事例です。



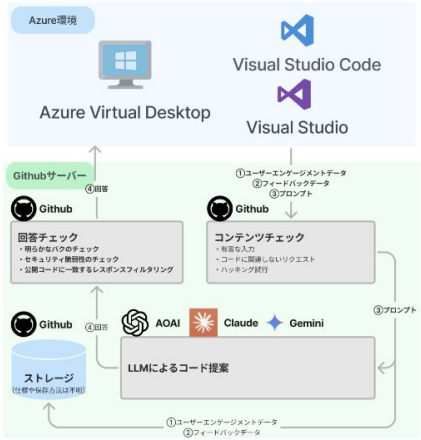
サムスンは社内データ漏洩を受け、ChatGPTなど生成AIツールの社内利用を一時禁止しました。

参考：TechCrunch「Samsung bans use of generative AI tools like ChatGPT after April internal data leak」

ガバナンス基礎 - GitHub Copilot のセキュリティ

暗号化 × 最小保持 × 厳格権限制御によって、Copilotのコード流出リスクを低減します。

データのフローおよびデータの概要、各保存期間、暗号化については下記の通りです。



送信されるデータの種類と保存期間

データの種類	データの概要	データの保存期間
① ユーザーエンゲージメントデータ	• 仮名化されたID • 承認/却下された補完 • エラーメッセージ • システムログ • 製品使用状況メトリクス	2年間保存される
② フィードバックデータ	• AIの回答のユーザー評価 • ユーザーのコメント	本来の目的に必要な期間保存
③ プロンプト	• コードの入力内容 • チャットの入力内容 • 関連するコンテキスト	IDE 経由：保存されない IDE以外：28 日間保持
④ 回答	• AI生成コード • AIのチャット応答	IDE 経由：保存されない IDE以外：28 日間保持

出典：GitHub Trust Centerなどのドキュメントを元に作成

データフローの要約

- すべての経路で TLS、保存時は FIPS 140-2 準拠で暗号化。
- ①②は品質向上のためのみ長期保管、③④はリアルタイム処理後に速やかに消去。
- アクセス権はロールベース＋多要素認証で厳格に制御。

※**仮名化ID**：ユーザーを特定できないようハッシュ化した内部識別子。利用傾向分析用で、個人情報は含まれない。
※**承認／却下補完**：Copilot 提案をそのまま挿入すれば「承認」、無視・削除・編集すれば「却下」と記録。モデル改善の指標になる操作ログ。

参考：GitHub「GitHub Copilot Trust Center」

ガバナンス基礎 - 文化庁 AIと著作権

生成AI活用には、社内ルール整備と侵害時の対応策を事前に用意しておくことが重要です。

3-1-2 著作権侵害 に対する適切な予防措置及び対応の検討

AI利用者としては、利用するAIシステム・サービスに応じて、著作権侵害を生じさせ ない適正な方法で生成AIを利用できるよう、生成AIの利用に関する内部的ルールの 策定等、著作権侵害に対する適切な予防措置を講じることが望まれます。 また、仮に生成AIの利用により著作権侵害が生じた場合に講じるべき対応（例えば、 事案に関する情報共有、AIシステム・サービスの停止・復旧、権利者への対応、原因解明、 再発防止措置等）についても、あらかじめ想定し検討しておくことが望まれます。[1]

[1]: 文化庁　AIと著作権に関する チェックリスト＆ガイダンスより抜粋

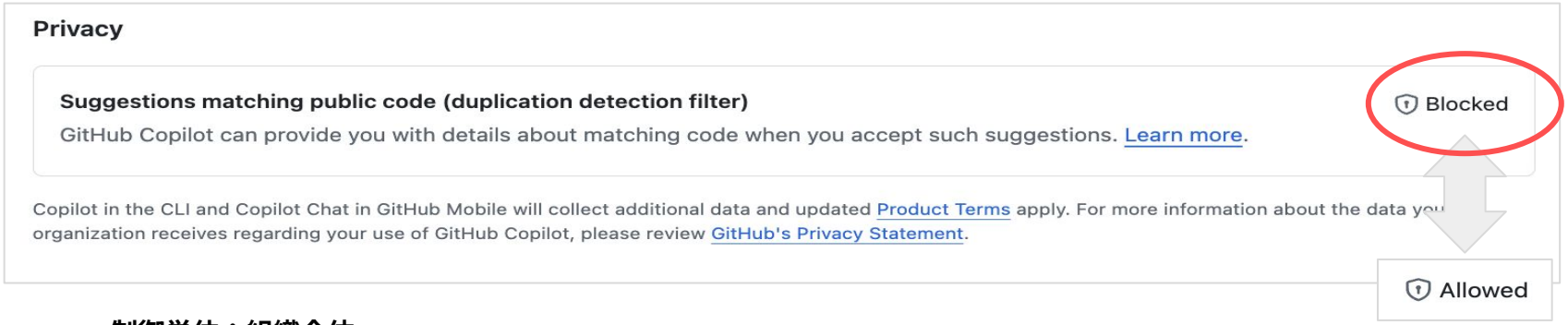
※本資料は法的見解を保証するものではありません。具体的な法的事案は法律専門家にご相談ください。

参考：文化庁「AIと著作権に関する チェックリスト＆ガイダンス」

ガバナンス基礎 - GitHub Copilot : パブリックコード制限

Copilot の回答がパブリックコードと類似する場合、出力を制限することができる機能です。

1. ✔️ GitHub のダッシュボードの右上アカウントをクリック ⇒ Setting をクリック
2. ✔️ 左側サイドメニューから Copilot をクリック ⇒ Features を選択
3. ✔️ Privacy の項目の Suggestions matching public code の項目で「Blocked」を選択



- 制御単位：組織全体
- Blocked：パブリックコードと類似した出力を制限
- Allowed：出力制限なし

ガバナンス基礎 - GitHub Copilot ：Microsoftの補償体制

万一の著作権訴訟でも、Microsoft が弁護・賠償を全面補償します。（ガードレール必須）

引用箇所	抜粋	意味
Microsoft公式ブログ「Copilot Copyright Commitment」	*“specifically for **paid versions of Microsoft commercial Copilot services** … **It also includes GitHub Copilot**”* (blogs.microsoft.com)	**「有償版の商用 Copilot サービス」が補償対象であり、その中に GitHub Copilot も含まれると明記。**
GitHub Copilot Trust ガイド	*“**GitHub Copilot Business** and **GitHub Copilot Enterprise** customers are entitled to IP indemnification …”* (resources.github.com)	**GitHub側の契約条項でも、Business／Enterprise プラン利用者に対して著作権等の補償（indemnity）を提供すると記載。**

※ガードレール：パブリックコードの出力制限アクティブ化

ガバナンス基礎 - AI法施行（令和7年6月4日）

政府は、国・企業・研究機関・自治体・国民に対し、AI研究開発と活用の“責務”と“透明性”を明文化しました。

章・条文	主な対象	ポイント（要約）
第1条 目的	—	AI研究開発と活用を総合的・計画的に推進し、国民生活と経済発展に寄与する。
第3条 基本理念	共通	①行政・産業の効率化 / ②国際競争力の向上 / ③透明性と適正性 / ④国際協調 を掲げる。
第4条 国の責務	国	総合的施策の策定・実施、行政分野でのAI活用を促進。
第5条 地方公共団体の責務	自治体	地域特性を踏まえた自主施策を策定し、国と役割分担。
第6条 研究開発機関の責務	大学・研究所	研究・成果普及・人材育成に努め、国・自治体施策に協力。
第7条 活用事業者の責務	企業等	AIによる業務高度化・新産業創出を推進し、公的施策に協力。
第8条 国民の責務	国民	AIへの理解と関心を深め、公的施策に協力。
第9条 連携の強化	産官学	国・自治体・研究機関・企業の連携を強化する施策を国が講ずる。
第14〜15条 人材・教育	国	多分野人材の確保・育成と国民向けAI教育を推進。

※本資料は法的見解を保証するものではありません。具体的な法的事案は法律専門家にご相談ください。

参考：[e-Gov法令検索「人工知能関連技術の研究開発及び活用の推進に関する法律」]

ガバナンス基礎 - まとめ

法規・ガイドライン・社内ルールを押さえ、リスクを管理しながら生成 AIで生産性を最大化することが重要です。

- ✔️ **Microsoft Copilot／GitHub Copilot（商用版）は補償付き**
  - 著作権侵害で訴えられても弁護・賠償を補填〈ガードレール有効化が条件〉

- ✔️ **文化庁ガイドライン**：生成AIによる著作権侵害の予防策＋侵害発生時の対応計画を事前策定
- ✔️ **AI法施行**：国・自治体・研究機関・企業・国民に責務／透明性と適正性を要求
- ✔️ **社内実務**
  - 利用ポリシーと審査フローを整備
  - 自動フィルタ＋人手レビューで再発防止
  - 全社員向け AI リテラシー教育を継続

- ✔️**結論**：リスク対策を講じながら、AIコーディングによって開発効率を向上させることが重要です。

※本資料は法的見解を保証するものではありません。具体的な法的事案は法律専門家にご相談ください。

ガバナンス補足：別資料









GENERATIVE AI TRAINING

生成AIの「今」と向き合い方

最新動向 ・ 仕組み ・ リスク ・ ガバナンス

生成AI活用実践研修

2026年7月版

Github Copilot の活用Tips

Github Copilot の活用Tips - ゴール

本パートのゴール	GitHub Copilotの様々な機能を把握し、 自らの継続学習をスタートするきっかけとする

Github Copilot の活用Tips

- 便利な拡張機能のご紹介
- MCPの概要とMCPサーバーのご紹介

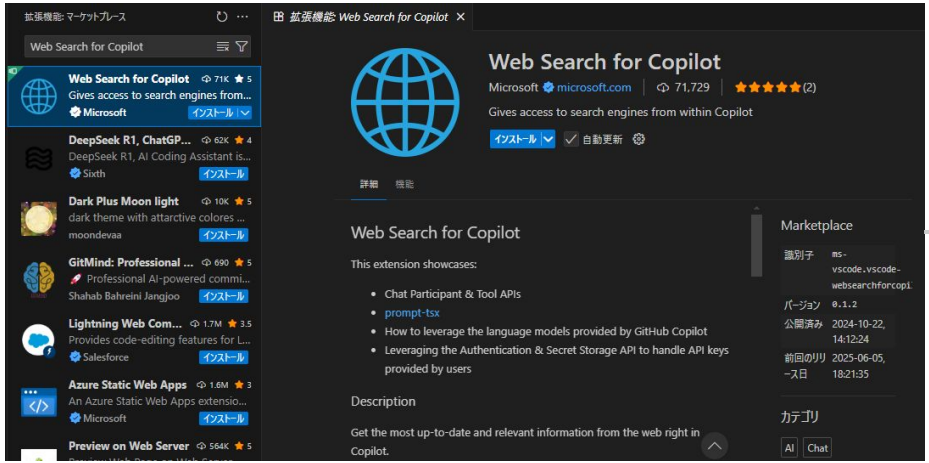
Github Copilot の活用Tips

- 便利な拡張機能のご紹介
- MCPの概要とMCPサーバーのご紹介

Github Copilot の活用Tips - 拡張機能：Web検索

VScode

Web検索によって、適切なソースから回答を得ることが可能



① エディタ左側の拡張機能検索から、Web Search for Copilot を検索してインストール

※モデルがClaude だと使用することができません

② @websearch ～を実行する

③ ポップアップが出たら許可

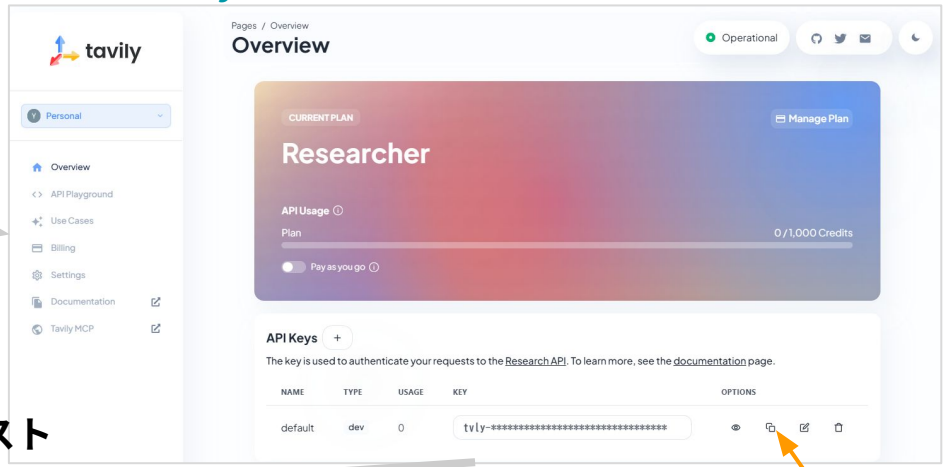
Github Copilot の活用Tips - 拡張機能：Web検索

VSCode

TavilyのAPIキーを登録：初めて@websearchを使用する際にAPIキーの入力が求められるので、TavilyでAPIキーを作成しましょう。

④ 再びポップアップが出たら許可

⑤ Tavily でアカウント作成・APIを取得



⑥ APIキーをペースト

APIキーをコピー

Github Copilot の活用Tips - 拡張機能：Web検索

VScode

Websearch：Web検索して公式ドキュメントから回答させることも可能です。

Websearch

Github Copilot の活用Tips - 拡張機能：vscode-mermAId について

VS Code Mermaidは、コードやプロジェクト構造から図表を自動生成し、自然言語での編集や共有が可能なVS Code 拡張機能です。

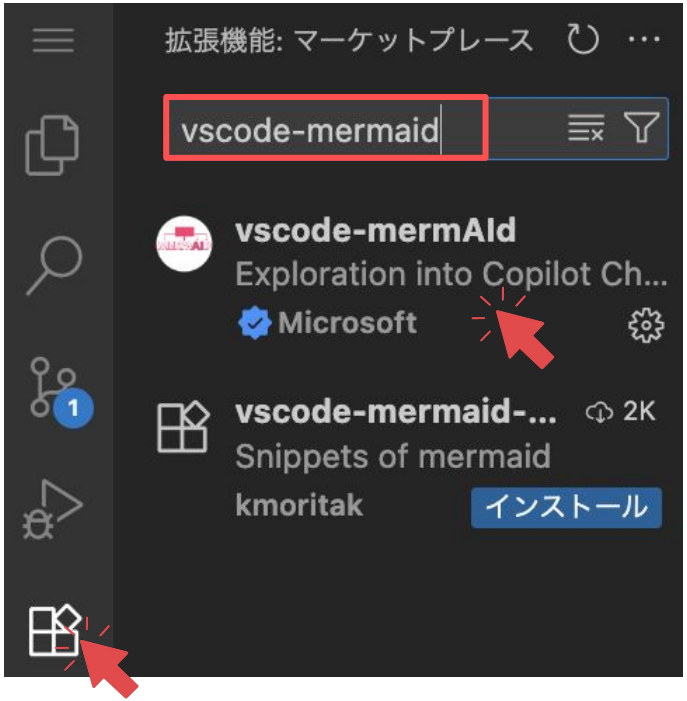
vscode-mermAId 使用イメージ

参考：[GitHub - Microsoft「vscode-mermAId」]

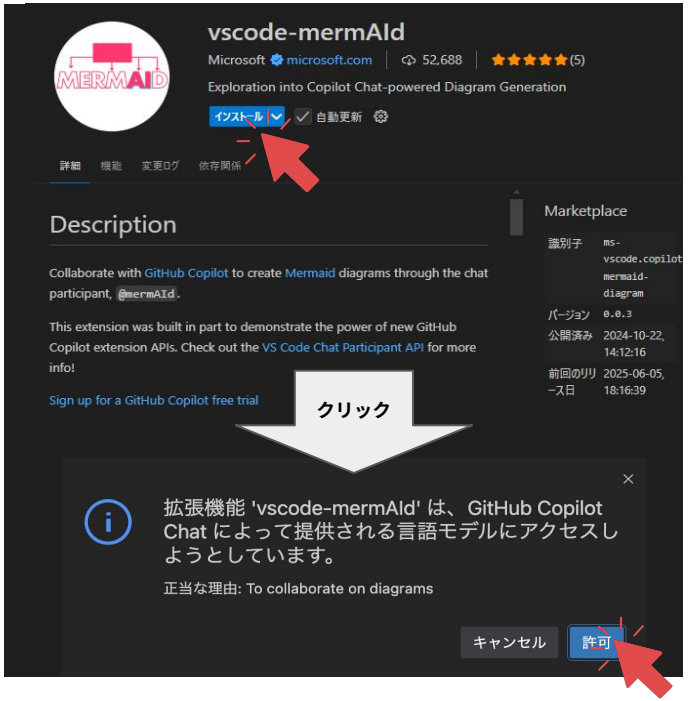
Github Copilot の活用Tips - 拡張機能：vscode-mermAId のインストール

VS Codeの拡張機能から、vscode-maermAId をインストールして使用することができます。

vscode-maermAIdのインストール手順



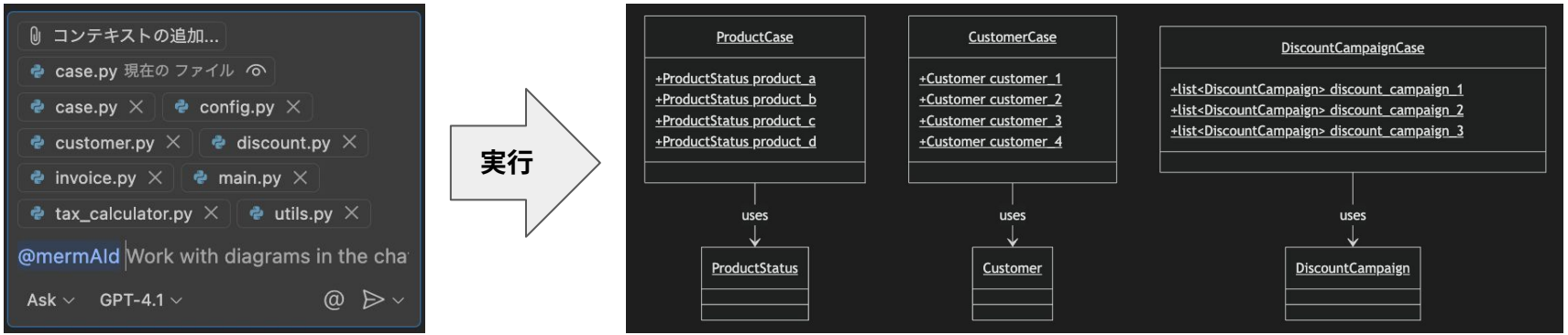
クリック



Github Copilot の活用Tips - 拡張機能： vscode-mermAId について

コードやプロジェクトの理解時間を短縮し、手動でのフローチャート作成やドキュメント更新作業を自動化して開発効率の向上が期待できます。

vscode-mermAId 使用イメージ



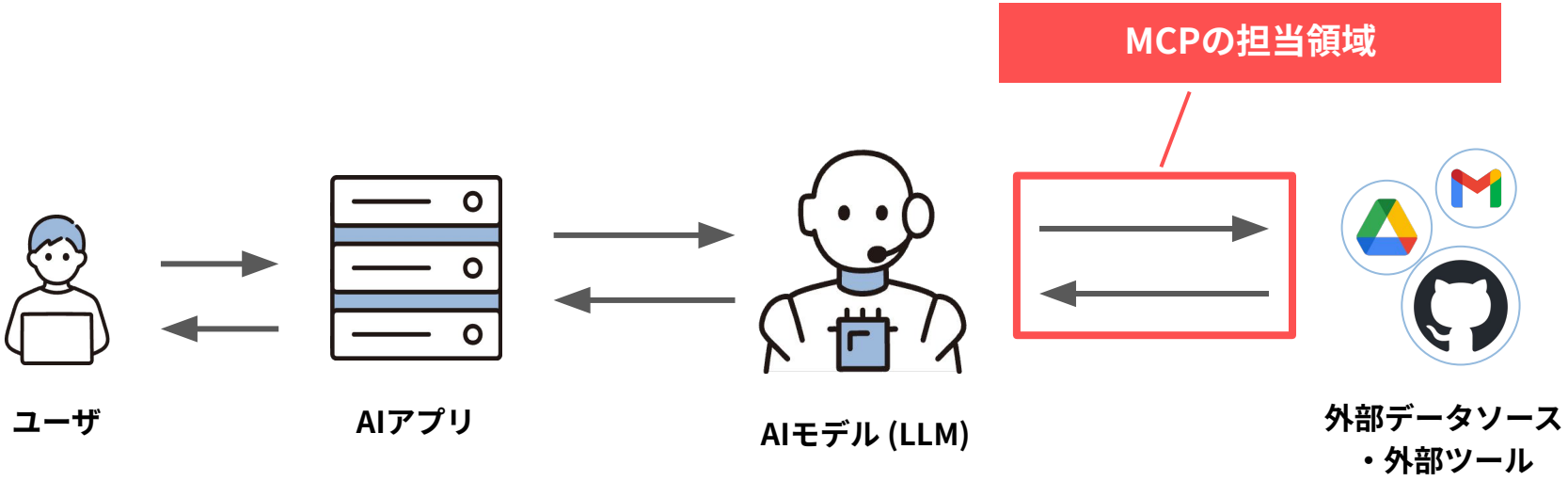
参考：GitHub - Microsoft「vscode-mermAId」

Github Copilot の活用Tips

- 便利な拡張機能のご紹介
- **MCPの概要とMCPサーバーのご紹介**

Github Copilot の活用Tips - MCPの概要とイメージ

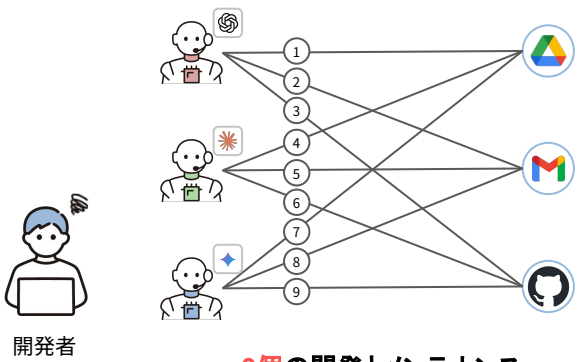
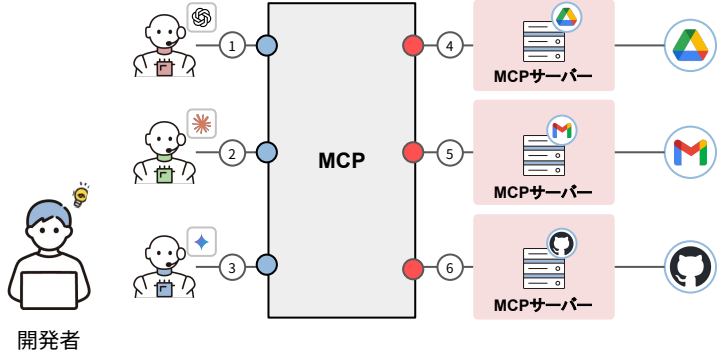
Model Context Protocol（MCP）は、2024年11月にAnthropic社が生み出した
AIモデルと外部のデータやツールとスムーズに連携するプロトコルのことです。



 参考： [Anthropic「Introducing the Model Context Protocol」]

Github Copilot の活用Tips - 従来型AI連携の課題とMCP導入による効果

MCPを活用することで開発及びメンテナンス対象のシステム数を大幅に削減することができます。

	Before	After
特徴	• ツールが増えるごとに複雑かつ手間がかかる • メンテナンスが大変で効率が悪い	• 規格に従えば簡単に手間なくAI連携が可能 • 開発コストを大幅に削減できる
イメージ	 9個の開発とメンテナンス	 6個の開発とメンテナンス

Github Copilot の活用Tips - MCPオフィシャルサーバー一覧 - 機能と用途

各言語の公式 MCP-SDK が幅広いシナリオに対応しています。

言語別の主な公式MCP の機能と用途 一覧

SDK／サーバー名	主な機能	主な用途
Python SDK	Resources／Tools／Prompts登録、CLI・SSE・Streamable HTTP対応	プロトタイピング、CLIツール、Webアプリ
TypeScript SDK	同上 ＋ 低レベルServer、stdio	Node.jsアプリ、VS Code拡張、サーバーレス開発
Java SDK	同上 ＋ Spring Bootスターター連携	エンタープライズサービス、マイクロサービス
Kotlin SDK	同上 ＋ iOS／Wasm対応	モバイル・ブラウザ・マルチプラットフォーム
C# SDK	同上 ＋ ASP.NET Core統合、DI・属性ベース定義	Windows／Azure Functions、Web API

参考：GitHub「Model Context Protocol」

まとめ

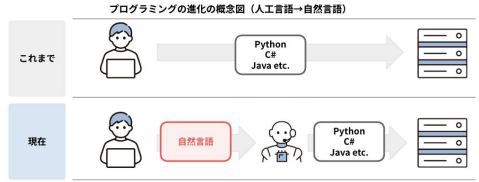
まとめ

プログラミングの歴史的な変遷とAIコーディングに必要なスキルを理解し、
適切なリスク対策を講じながら、生成AIコーディングによって開発効率を向上させていくことが重要です。

まとめ

生成AIネイティブ開発の概論 - プログラミング言語の変遷

生成AIの登場により、『人工言語から自然言語によるプログラミング』へと進化しつつあります。



GitHub Copilot 概論 - GitHub Copilot チャットの３つのモード

VScode

GitHub Copilotチャットでは、3つのモードを活用することで開発効率を最大化することができます。

	Askモード	Editモード	Agentモード
概要	対話型チャットで コード相談・質問解決	指示に基づいて 複数ファイルを一括編集	自律的にプロジェクト 全体のタスクを実行
@/ コマンド	○	×	×
使用 シーン	例:「Pythonについて教えてください」	例:「(複数ファイル選択) Google 形式の docstring を付けて」	例:「Flask で在庫管理のWeb アプリケーションを作って」

 参考：GitHub Inc.「Copilot ask, edit, and agent modes: What they do and when to use them」

GitHub Copilot の使い方 - GitHub Copilot の3種類のコマンド

GitHub Copilotには3種類の基本コマンドがあり、それぞれ異なる用途で指示を最適化できます。

	@コマンド	#コマンド	/コマンド
概要	質問対象の スコープを指定する	Copilot に ツールの使用を指示する	特定の指示を ショートカットとして 利用する
使用例	エディタの言語を日本語にして @vscode エディタの言語を日本語にして	ターミナルのエラーの原因を教えて #terminalLastCommand エラーの原因を教えて	このコードを解説して @workspace /explain

 参考：GitHub「GitHub Docs」

GitHub Copilot の使い方 - Custom Instructionsの活用シーン

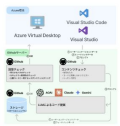
Custom Instructionsを活用することでチーム開発から個人開発まで、
様々なシーンで開発の生産性を向上させることができます。

チーム内での設定	個人最適化	コンテキスト繰り返し利用
チームのコーディング規約や使用技術、プロジェクト固有の情報を理解させる	個人の好み（応答言語、スタイルなど）を反映させる	繰り返し同じコンテキストをチャットで説明する手間を省く

生成AIネイティブ開発におけるリスク対策 - GitHub Copilot のセキュリティ

暗号化 × 最小保持 × 厳格権限制御によって、Copilotのコード流出リスクを低減します。

データのフローおよびデータの概要、各保存期間、暗号化については下記の通りです。



送信されるデータの種類と保存期間

データの種類	データの概要	データの保存期間
① ユーザーエンゲージメントデータ	• 受け入れられた・拒否/修正された補完 • 使用統計 • システムログ • 製品使用状況メトリクス	2年間保存される
② フィードバックデータ	• 👍/👎回答のユーザー評価 • ユーザーのコメント	未来の目的に必要な期間保存
③ プロンプト	• コード出力の内容 • チャットの入力内容 • 関連するコンテキスト	IDE 環境：保存されない IDE以外：28 日間保持
④ 回答	• AI生成コード • AIのチャット応答	IDE 環境：保存されない IDE以外：28 日間保持

出典：GitHub Trust Centerなどのドキュメントを元に作成

※仮名化ID：ユーザーを特定できないようハッシュ化した内部識別子。利用傾向分析用で、個人情報は含まれない。
※承認／却下補完：Copilot 提案をそのまま挿入すれば「承認」。無視・削除・編集すれば「却下」と記録。モデル改善の指標になる操作ログ。

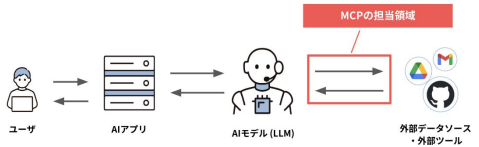
データフローの要約

- すべての経路で TLS、保存時は FIPS 140-2 準拠で暗号化。
- ①②は品質向上のためのみ長期保管、③④はリアルタイム処理後に速やかに消去。
- アクセス権はロールベース＋多要素認証で厳格に制御。

 参考：GitHub「GitHub Copilot Trust Center」

MCP概論 - MCPのイメージ

Model Context Protocol（MCP）は、2024年11月にAnthropic社が生み出した
AIモデルと外部のデータやツールとスムーズに連携するプロトコルのことです。



 参考： Anthropic「Introducing the Model Context Protocol」

研修のスケジュール

1日目：GitHub Copilot基礎と実装体験

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：オリエンテーション［30min］研修ゴール共有｜自己紹介・2日間の概要説明｜ツール紹介（VSCode・GitHub Copilot）｜学習スタンスの共有

■Session02：生成AIリテラシーとガバナンス［100min］

● ガバナンス基礎	生成AIの仕組み｜4月研修の復習・接続｜機密情報・著作権・ハルシネーションの実例｜社内ガイドラインの要点確認｜やってよいこと・いけないことの判断基準
● 理解度確認テスト	理解度確認テスト｜生成AI利用時のセキュリティリスク｜ガイドラインに基づく利用可否判断｜AI生成コードのレビュー義務確認｜テスト後に正解と解説をその場で共有
● 環境セットアップ	VSCode起動・Copilot拡張の認証確認｜インラインチャット｜Copilot Chatの基本操作｜スラッシュコマンド｜FizzBuzzで最初の生成体験

■Session03：Javaコード生成ハンズオン［140min］

● メソッド・DTOの生成	メソッド名とJavadocからの実装自動生成｜getter/setter/toString/equals自動生成｜RecordクラスやLombokアノテーションの提案確認｜カスタム例外クラス
● CRUD処理の生成	DAOクラスのCRUDメソッド自動生成｜SQLクエリ生成支援｜Spring Data JPAリポジトリ生成｜try-catchブロックの自動追加｜例外処理パターンの提案比較
● レビュー実践	問題点発見｜セキュリティ観点（SQLインジェクション・ハードコード値）｜ロジック観点（NullPointerException・境界値）｜命名規約・コード規約との整合性確認

■Session04：ペアワーク・コードレビュー演習［120min］

● ペアレビュー	ペアレビュー｜セキュリティ・ロジック・命名の3観点｜Copilot Chatで修正｜インラインチャットで部分修正を体験｜複数パターン比較・採用判断基準の整理
● Day 1振り返り	本日のキーメッセージ3点を確認｜AIコードの最終責任は人間にあることを再確認｜プロンプトの書き方と品質の関係｜Day 2の予告（テスト生成・デバッグ・総合演習）

■Session05：Day1クロージング・明日の予告［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

研修のスケジュール　2日目：テスト・デバッグ・実践演習

TIME：7h

● 座学主体　● ハンズオン主体

■Session01：前回振り返りとDay 2概要［30min］ 前日の振り返りクイズ（5問）｜Copilot基本操作の確認｜Day 2ゴール共有｜テスト・デバッグ・総合演習の概要

■Session02：テストコード生成ハンズオン［130min］

- **JUnitテスト自動生成**　/testsコマンドでテストコード自動生成｜テストメソッドの自動命名確認｜正常系・異常系・境界値の網羅｜null入力・空文字・最大値最小値テスト
- **テスト実行と修正**　生成テストを実行して失敗ケースを特定｜テストが通るよう実装側を修正体験｜例外が正しくスローされるかのテスト実践｜テスト品質の確認と改善サイクル
- **デバッグ・リファクタリング実践**　/fixコマンドでバグ修正｜エラーメッセージから原因特定｜長いメソッド・マジックナンバー｜重複コードの共通化体験｜/explainでJavadoc自動生成

■Session03：総合演習 AI活用ミニ開発［150min］

- **演習課題**　簡易設計書（画面仕様書 or API仕様書）配布｜設計書→実装→テスト→レビューの一連の流れ｜Copilot Chat実装｜手作業との所要時間を計測して比較
- **成果共有・振り返り**　数名が画面共有で成果を発表｜「Copilotが役立った場面・邪魔だった場面」を共有｜2日間の学びを全体で振り返り｜配属後の活用イメージを言語化
- **配属後の活用ガイド**　新プロジェクト参画時のコード理解｜定型コード高速生成｜お客様環境での利用可否判断基準｜AIに任せる/自分で書く場面の整理

■Session04：学習リソース紹介・グループディスカッション［80min］

- **学習リソース**　GitHub Copilot公式ドキュメント｜VSCode公式ドキュメント｜JUnit 5ユーザーガイド｜社内の生成AI活用事例・ナレッジ｜今後の研修ロードマップ
- **グループディスカッション**　3～4名で「配属後に使いたい場面」を話し合う｜使えない現場での代替手段を考える｜セキュリティ上の判断フローを整理｜グループ代表者が発表

■Session05： 質疑応答・アンケート・閉会［30min］

*上記スケジュールは変更となる場合がございます。

事前セットアップ：事前セットアップ：VS Code最新版 ｜ CopilotChat拡張機能 ｜ 各自の開発言語実行環境

End of File