

GitHub Enterpriseは、ソフトウェアのコード管理・共有、レビュー機能、課題管理など、チーム開発に適した機能を備えたGitのソフトウェアバージョン管理サービスです。

Gitの基本機能であるバージョン管理以外の機能も豊富に取り揃えており、開発業務の効率化を実現いたします。

「AIによるコーディング支援」や、「脆弱性のあるコード検出する機能」等の開発者に必要となる機能が次々と実装されています。

標準機能

<h3><u>Pull Request</u></h3> <p>ソースコードのレビューをブラウザ上で行えます。</p>	<h3><u>Issue/Milestone/Projects</u></h3> <p>リアルタイムにIssueの進捗管理ができます。</p>
<h3><u>機能制御</u></h3> <p>リポジトリのアクセス権限を細かく制御できます。</p>	<h3><u>Wiki/GitHub Pages</u></h3> <p>コンテンツをWebページとして公開できます。</p>
<h3><u>コード検索</u></h3> <p>様々な種類の情報を検索できます。</p>	<h3><u>Insights</u></h3> <p>リポジトリに発生した情報を参照できます。</p>
<h3><u>認証</u></h3> <p>SAML認証や多要素認証が行えます。</p>	<h3><u>監査ログ</u></h3> <p>ユーザのログインや操作を証跡として残せます。</p>

Pull Request

ソースコードのレビューをブラウザ上で行えます。ソースコードの差分が自動的に表示され、レビュアと担当者間でコメントのやり取りをしながら修正を取り込みます。

The screenshot shows a GitHub Pull Request interface. Callouts highlight the following features:

- 誰がどこに対してレビューしたのかなどレビュー状態を確認**: A callout pointing to the 'Conversation' tab, indicating that users can track who reviewed and when.
- 差分が色付きで表示**: A callout pointing to the code diff view, indicating that changes are color-coded (green for additions, red for deletions).
- コメントでレビュー結果を記録**: A callout pointing to the 'Review changes' section, indicating that review results are recorded in comments.

オプション機能

GitHub Actions	GitHub Advanced Security
GitHub Copilot	GitHub Packages
GitHub Codespaces	Git LFS

GitHub Actions プッシュなどのイベントをトリガーとしてビルド、テスト、デプロイのパイプラインを自動化できます。

The screenshot shows the GitHub Actions interface for a workflow named 'cpplint'. The workflow file is defined in YAML:

```
1 name: cpplint
2
3 on:
4   push:
5     branches: [ "main" ]
6   pull_request:
7     branches: [ "main" ]
8
9 jobs:
10  cpplint:
11    runs-on: ubuntu-latest
12    steps:
13      - uses: actions/checkout@v3
14      - uses: actions/setup-python@v4
15        with:
16          python-version: '3.x'
17      - run: pip install cpplint
18      - run: cpplint --recursive .
```

Callouts explain the workflow configuration and execution:

- YAML構文を使用してワークフローを定義
- ワークフローは、イベントのトリガーや実行条件、ジョブなどを定義
- 実行履歴もログとして保持でき、GitHub上からジョブごとの実行履歴を確認

GitHub Copilot 膨大なGitHubのパブリックリポジトリでトレーニングされたCopilotがAIペアプログラマーとなり、最適なソースコードをリアルタイムに提案します。

The screenshot shows a code editor with a Python function definition. Copilot has suggested the following code:

```
1 # fizzbuzz.py
2
3 def fizzbuzz(n):
4     if n % 15 == 0:
5         return 'FizzBuzz'
6     elif n % 3 == 0:
7         return 'Fizz'
8     elif n % 5 == 0:
9         return 'Buzz'
10    else:
11        return str(n)
```

Callouts explain Copilot's capabilities:

- 関数名やコメントをすることでCopilotが推奨コードを自動生成
- Copilotコード自動生成部分
- ユーザーが望むコードを予測して、AIが推奨コードを生成
- 複数の推奨コードが提案可能
- 様々な言語（Python、Ruby、C++、Javaなど）に対応

※本リーフレット中の会社名、商品名は各社の商標、または登録商標です。 ※GitHubは、GitHub, Inc.の米国およびその他の国における商標または登録商標です。 ※本文中および図中では、TMマーク、®マークは表記していません。 ※製品の仕様は、改良のため、予告なく変更する場合があります。 ※本製品を輸出される場合には、外国為替及び外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。なお、ご不明な場合は、当社担当営業にお問い合わせください。 ※本リーフレット中の情報は、作成時点のものであります。

株式会社 日立ソリューションズ
www.hitachi-solutions.co.jp

本リーフレット掲載商品・サービスの詳細情報
www.hitachi-solutions.co.jp/github/

