

# GitHub Actions クイックリファレンス

ワークフロー、トリガー、ジョブ、シークレット、キャッシュ、アーティファクト

## ワークフローの基本

### 最小ワークフロー

```
# .github/workflows/ci.yml
name: CI
on: push
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - run: echo "Hello from CI"
```

### 主要なコンセプト

<b>Workflow</b>	自動化を定義する <code>.github/workflows/</code> の YAML ファイル
<b>Event</b>	ワークフローを起動するトリガー (push、PR、schedule など)
<b>Job</b>	同じランナーで実行されるステップのセット
<b>Step</b>	個別のタスク - コマンドを実行するかアクションを使用
<b>Runner</b>	ジョブを実行する VM ( <code>ubuntu-latest</code> 、 <code>macos-latest</code> 、 <code>windows-latest</code> )
<b>Action</b>	<code>uses:</code> で参照する再利用可能なコードの単位

### トリガー

#### よく使うイベント

```
on:
  push:
    branches: [main]
  pull_request:
    branches: [main]
  schedule:
    - cron: "0 6 * * 1" # 毎週月曜 UTC 6 時
  workflow_dispatch: # 手動トリガー
```

#### イベントフィルター

<b>branches:</b>	特定のブランチのみトリガー
<b>paths:</b>	一致するファイルが変更された場合のみトリガー
<b>tags:</b>	タグのプッシュでトリガー ( <b>v*</b> )
<b>types: [opened, synchronize]</b>	PR のアクティビティタイプをフィルタ
<b>branches-ignore:</b>	特定のブランチを除外
<b>paths-ignore:</b>	特定のファイルパスを除外

### ジョブとステップ

#### ジョブの設定

```
jobs:
  test:
    runs-on: ubuntu-latest
    needs: build # build ジョブに依存
    if: github.ref == 'refs/heads/main'
    timeout-minutes: 10
    steps:
      - uses: actions/checkout@v4
      - run: npm test
```

### ステップの種類

<b>run:</b>	シェルコマンドを実行
<b>uses:</b>	公開されたアクションを使用
<b>with:</b>	アクションに入力を渡す
<b>name:</b>	UI での表示名
<b>id:</b>	<code>steps.&lt;id&gt;.outputs</code> でステップ出力を参照
<b>if:</b>	条件付き実行
<b>continue-on-error: true</b>	ステップが失敗してもジョブを失敗させない

### アクション

#### アクションの使用

```
steps:
  - uses: actions/checkout@v4
  - uses: actions/setup-node@v4
  with:
    node-version: 20
  - uses: ./github/actions/my-action # ローカルアクション
```

#### 人気のアクション

<b>actions/checkout@v4</b>	リポジトリのコードをチェックアウト
<b>actions/setup-node@v4</b>	Node.js をインストール
<b>actions/setup-python@v5</b>	Python をインストール
<b>actions/upload-artifact@v4</b>	ビルドアーティファクトをアップロード
<b>actions/download-artifact@v4</b>	別ジョブのアーティファクトをダウンロード
<b>actions/cache@v4</b>	実行間で依存関係をキャッシュ
<b>actions/github-script@v7</b>	GitHub API クライアントで JS を実行

### 環境変数

#### 変数の設定

```
env: # ワークフローレベル
  NODE_ENV: production
jobs:
  build:
    env: # ジョブレベル
      CI: true
    steps:
      - run: echo "$MY_VAR" # ステップレベル
        env:
          MY_VAR: hello
```

#### デフォルト変数

<b>github.sha</b>	ワークフローをトリガーしたコミット SHA
<b>github.ref</b>	ブランチまたはタグの ref ( <code>refs/heads/main</code> )
<b>github.repository</b>	オーナー/リポジトリ名
<b>github.actor</b>	ワークフローをトリガーしたユーザー
<b>github.event_name</b>	ワークフローをトリガーしたイベント
<b>runner.os</b>	ランナーの OS ( <b>Linux</b> 、 <b>macOS</b> 、 <b>Windows</b> )

### シークレット

#### シークレットの使用

```
steps:
  - run: deploy --token "$TOKEN"
    env:
      TOKEN: ${ secrets.DEPLOY_TOKEN }
  - uses: some/action@v1
    with:
      api-key: ${ secrets.API_KEY }
```

### シークレットのルール

<b>secrets.GITHUB_TOKEN</b>	リポジトリにスコープされた自動生成トークン
<b>Settings → Secrets</b>	リポジトリまたは組織設定でシークレットを追加
<b>マスキング</b>	シークレット値はログで自動的にマスクされる
<b>環境シークレット</b>	デプロイ環境にスコープされる
<b>組織シークレット</b>	組織内のリポジトリ間で共有

### マトリクス戦略

#### マトリクスビルド

```
jobs:
  test:
    strategy:
      matrix:
        os: [ubuntu-latest, macos-latest]
        node: [18, 20]
    runs-on: ${ matrix.os }
    steps:
      - uses: actions/setup-node@v4
        with:
          node-version: ${ matrix.node }
```

#### マトリクスオプション

<b>matrix:</b>	展開する変数の組み合わせを定義
<b>include:</b>	マトリクスに追加の組み合わせを加える
<b>exclude:</b>	特定の組み合わせを削除
<b>fail-fast: false</b>	1 つが失敗しても他のジョブを継続
<b>max-parallel: 2</b>	同時実行するマトリクスジョブを制限

### キャッシュ

#### 依存関係をキャッシュ

```
- uses: actions/cache@v4
  with:
    path: ~/.npm
    key: npm-${ hashFiles('package-lock.json') }
    restore-keys: npm-
```

#### 組み込みキャッシュ

```
- uses: actions/setup-node@v4
  with:
    node-version: 20
    cache: npm # npm/yarn/pnpm を自動キャッシュ
- uses: actions/setup-python@v5
  with:
    python-version: "3.12"
    cache: pip # pip を自動キャッシュ
```

### アーティファクト

#### アップロードとダウンロード

```
- uses: actions/upload-artifact@v4
  with:
    name: build-output
    path: dist/
    retention-days: 7
- uses: actions/download-artifact@v4
  with:
    name: build-output
```

# GitHub Actions クイックリファレンス

## アーティファクトの注意点

<b>retention-days</b>	N 日後に自動削除 (デフォルト 90)
<b>path</b>	アップロードするファイルまたはディレクトリ (グロブ対応)
<b>ジョブ間</b>	1つのジョブでアップロード、 <b>needs:</b> で別ジョブからダウンロード
<b>compression-level</b>	0 (なし) ~ 9 (最大)、デフォルト 6

## よくあるパターン

### 条件付きデプロイ

```
- name: Deploy to production
  if: github.ref == 'refs/heads/main'
  run: ./deploy.sh
- name: Post PR comment
  if: github.event_name == 'pull_request'
  run: gh pr comment $PR --body "Build passed"
```

### 便利な式

<b>success()</b>	前のすべてのステップが成功した場合に true
<b>failure()</b>	前のいずれかのステップが失敗した場合に true
<b>always()</b>	ステータスに関わらず実行 (クリーンアップ)
<b>cancelled()</b>	ワークフローがキャンセルされた場合に true
<b>contains(github.ref, 'release')</b>	文字列の含有チェック
<b>startsWith(github.ref, 'refs/tags')</b>	文字列のプレフィックスチェック
<b>hashFiles('**/lock*')</b>	ファイルの SHA-256 (キャッシュキー用)