

# Claude Code 講習 #1 - hooks & GitHub Actions

shi.

# 僕が講習に慣れるためだけにある雑談

- 皆さん、"ccusage"って知っていますか？
- 知っている人も知らなかった人も、一旦ターミナルで `npx -y ccusage` と実行してみて下さい

2025-10-02	- sonnet-4-5	1,079	2,393	87,139	862,143	\$0.62
2025-10-03	- sonnet-4-5	259	4,599	111,739	1,275,694	1,392,201
2025-10-04	- sonnet-4-5	1,482	12,016	423,269	2,339,110	3,075,877
2025-10-05	- sonnet-4-5	2,235	16,042	791,826	7,518,614	8,328,717
2025-10-06	- sonnet-4-5	1,274	7,913	570,097	5,003,124	5,582,408
2025-10-07	- sonnet-4-5	80	813	98,381	581,541	672,815
2025-10-08	- sonnet-4-5	3,726	13,713	1,289,263	9,982,683	11,289,385
2025-10-09	- sonnet-4-5	1,873	9,935	388,790	3,777,783	4,178,381
2025-10-10	- sonnet-4-5	1,273	3,938	283,970	1,635,863	1,924,236
2025-10-11	- claude-3-5-haiku-20241022 - sonnet-4-5	1,027	7,187	312,406	3,881,254	4,121,874
2025-10-12	- claude-3-5-haiku-20241022 - sonnet-4-5	5,810	30,932	1,806,526	10,296,473	12,139,741
2025-10-13	- claude-3-5-haiku-20241022 - sonnet-4-5	10,731	28,916	1,030,481	10,203,783	11,273,911
2025-10-14	- sonnet-4-5	67	3,034	125,624	373,102	501,827
2025-10-15	- sonnet-4-5	368	13,676	571,392	3,030,602	3,616,038
2025-10-16	- haiku-4-5 - sonnet-4-5	112,899	20,757	192,822	818,819	1,145,297
2025-10-17	- haiku-4-5 - sonnet-4-5	91,734	27,859	598,139	3,225,477	3,943,209
2025-10-18	- haiku-4-5 - sonnet-4-5	10,843	2,067	93,427	150,575	256,912
2025-10-19	- haiku-4-5 - sonnet-4-5	16,028	3,786	139,730	523,633	683,177
2025-10-20	- haiku-4-5	5,036	779	4,670	0	10,485
2025-10-21	- haiku-4-5	2,753	358	0	0	3,111
2025-10-23	- claude-3-5-haiku-20241022 - haiku-4-5 - sonnet-4-5	33,602	15,061	505,006	1,697,613	2,251,282
2025-10-24	- haiku-4-5 - sonnet-4-5	36,594	10,971	477,853	2,715,932	3,241,350
2025-10-25	- haiku-4-5 - sonnet-4-5	30	645	32,992	106,362	140,029
2025-10-26	- haiku-4-5 - sonnet-4-5	44,789	19,140	145,479	1,122,018	1,331,426
2025-10-27	- haiku-4-5 - sonnet-4-5	9,398	9,258	318,562	1,052,399	1,389,617
2025-10-28	- haiku-4-5 - sonnet-4-5	26,670	6,915	55,747	187,590	276,922
<b>Total</b>		460,438	945,149	25,341,355	259,592,396	286,339,338
						\$183.31

# ccusageとは？

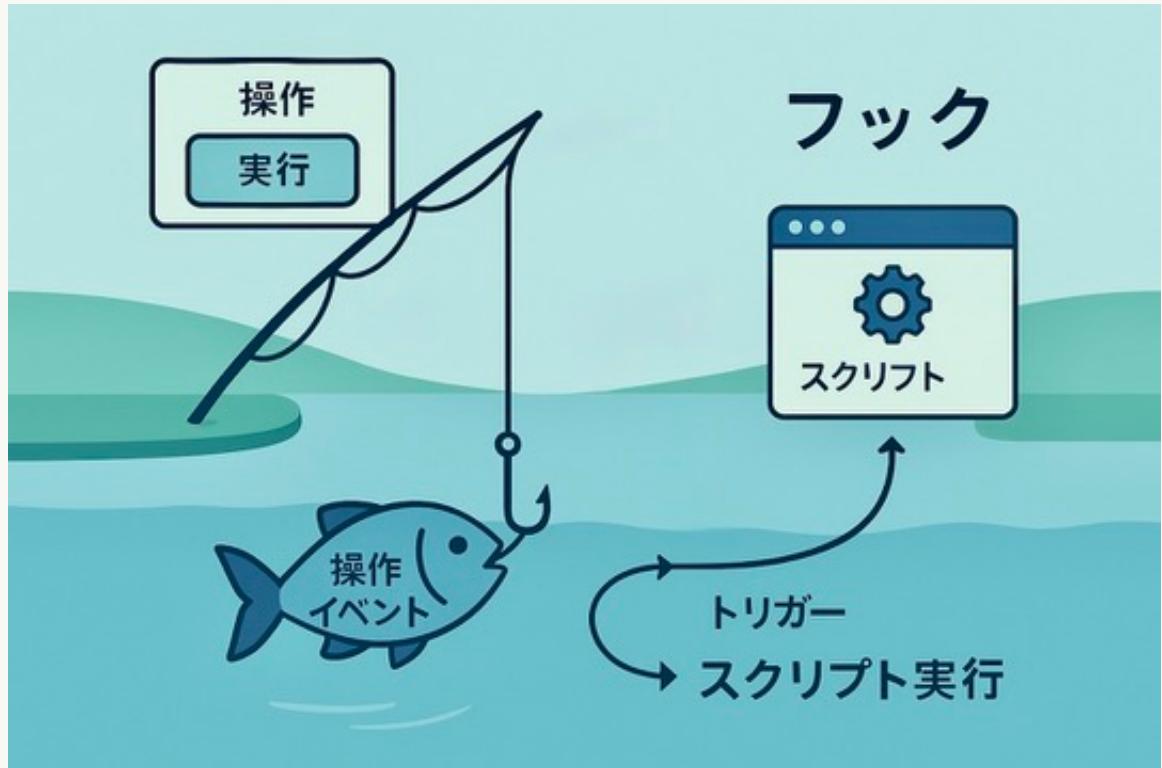
- Claude CodeでのAPI課金状況を表示するツール
- Pro / Maxプランで、APIを直接使用せずに定額で利用している人も含む
- 今まで使った量が、APIの料金で可視化される

# やっと本題、、、の前に

- 僕のネット環境は最悪です！(冗談ではありません)
  - たまに聞き取れないくらいに音質が悪くなります。
  - 一回プライベートでボイチャしてたら、ネットが落ちました
- 聞き取れなくて、聞き返すことがそれなりにあると思いますが、しょうがないことだと思っていただけだと嬉しいです。
  - 他意とかは全くないので、そこは安心してください
- Mac環境で実行します
- ターミナルでClaude Codeを操作する前提です

# フックについて

- 見てもらうのが一番早いです
- 今回紹介する3つのフックを見せます



## フックのイメージ

- 釣りの「フック」みたいに、釣り針が魚を引っかけるイメージ
- 操作の前後に、自動で走らせるスクリプトをトリガーさせることができる機能
- 操作のトリガーの対象となるフックが池に入つて、それに魚が食いついてスクリプトが自動で実行されるイメージ

# フックの具体例① - 音を鳴らすフック

## Bash command

```
touch sample_hello.py sample_calculator.js sample_data.json
3つのサンプルファイルを作成

Do you want to proceed?
> 1. Yes
  2. Yes, and don't ask again for touch commands in /Users/shin/Documents/programming/coding_agent/clause_code
  3. No, and tell Claude what to do differently (esc)
```

- Notificationのフックを設定しよう
  - ユーザーの許可が必要になったときに実行するやつ (Notificationは通知の意味)
  - 上の画像のこと
- これから、この画面が表示されたら音が鳴るようにフックの設定をしていきます！

# フックの基本

- json っていうファイル形式のテキストに保存します
  - 設定が保存される場所:
    - ユーザーの設定(いつでも有効): `~/.claude/settings.json`
    - プロジェクトの設定 (そのフォルダごとの設定) : `./.claude/settings.json`
    - プロジェクトの設定 (コミットされない) : `./.claude/settings.local.json`

# フックを設定してみよう - 音を鳴らすフック

- まずNotificationのフック
- 実際に設定してみます
- いっしょにやってみてもいいと思います。英語で書かれているので
- 追加するBash コマンド (Mac) : `afplay /System/Library/Sounds/Glass.aiff`
- 追加するBash コマンド (Windows) : `powershell.exe -Command  
'[System.Media.SystemSounds]::Hand.Play()'`

# フックを設定してみよう - 音を鳴らすフック

- 同じ流れで、Claude Codeがタスクが完了したときにも音が鳴るようにしてみよう
- "Stop"っていう名前のフックがあります
- Claude Codeが実行しているタスクが完了して、全ての出力が終わって止まったとき

## フックの具体例② - 自動でコミット

- さっき設定したStopのフックを使えば、Claude Codeが止まったときに自動でコミット作業もやってくれるようになります
- 今回はシェルスクリプトを実行する形で実装しようかなと思います
  - 僕は普段サブエージェントを実行させてますが、サブエージェントを解説するときにまとめて紹介します

# 他にもいろいろあるフック

- PreToolUse
  - ツールの実行前に自動的にトリガーされる処理です
- PostToolUse
  - ツールの実行後に自動的にトリガーされる処理です
- UserPromptSubmit
  - ユーザーがプロンプトを送ったときにトリガーされる処理です

# 他にもいろいろあるフック

- SessionStart
  - 新しく Claude Code を起動したときに自動的にトリガーされる処理です
- オートコンパクトが発動する前に自動的にトリガーされる処理です
  - SessionEnd
- セッションが終了するに自動的にトリガーされる処理です

# GitHub Actions

# GitHub Actionsとは？

- 基本的なイメージはフックと結構似てる
- あるイベント（この場合、プルリクや@メンション）をトリガーに、いろんなスクリプトを実行してくれる
  - CI/CDって呼ばれてたりする（ニュアンスは同じ）
  - 継続的に連携して、継続的にデプロイする仕組み
- GitHubのサーバー上で実行される（なので、無料枠があるがそれを超えたら有料）
  - パブリックリポジトリだったら無料
  - プライベートリポジトリの無料枠は、月に2000分使える

# 設定方法 - 1

- 一応推奨されてる"github cli"をインストールします
- インストールコマンド（Mac）：`brew install gh`
- インストールコマンド（Windows）：`winget install --id GitHub.cli`
- この後、`gh auth login` でGitHubアカウントにログインします

# 設定方法 - 2

- あとは `/install-github-app` をclaude code上で実行すれば設定できる、、、はず
- 初めての人がほとんどだと思うので、その人たちはGitHubのアカウントと連携から初めてください
- YouTubeの動画では、api-keyの設定をしないとできなかつたって人がいた（おそらくWindowsだから？）
- 何の問題もなく設定できれば、Claudeの定額プランの中で実行できるはず

# 動いている様子

- @claude のメンション
  - MCPサーバーのモジュール化処理に変更・Issue #1 · shin902/serena-modular-mcp
- プルリクのレビュー
  - feat: Implement category-based MCP modularization by shin902 · Pull Request #3 · shin902/serena-modular-mcp · GitHub

# 補足 / 注意点

- 今回はこの講習のためだけに作ったIssueなのでCLAUDE.mdは作成しなかったが、あつたらそれも読み込んでくれるはず
- 久しぶりに使うと、「実行に失敗しました」的な文言が表示されるかも
  - `/install-github-app` をもう1回実行すればいいです
- ブランチ関係は気をつけること
  - @claudeでメンションして、新しいタスクを開始しちゃうと、またブランチを作成しちゃって1からになるから
- 複数人でのシェアはやらない / 組織での使用はAPI課金を使用しましょう
  - 利用規約違反ととらえられる可能性があります

# 参考文献

- Claude Codeの開発効率を3倍に引き上げる7つの上級者向けTips 【バイブルコーディング/Vibe Coding】 - YouTube
- 【待望の】 公式ClaudeCodeActionがMAXプラン内に対応！これで面倒なことをする必要はありません！ - YouTube
- Claude Code等のクラウド型AIエージェントのおかげで、Githubでbranchをつくってmergeするというスタイルがようやくわかって嬉しい - YouTube