



第38回 「Introduction to Model Context Protocol」 Anthropic 公式 Claude Academy

Claude Code

資料を取りに行く

公式の教材を、順番に

原石工房

隠れた才能に気づき、花開く

この資料について



原石工房

Anthropic とは関係のない、非公式の資料です

公式の学習サイト Claude Academy の日本語訳です

原文（英語）Introduction to Model Context Protocol /
Accessing resources

前回で資料を定めました。今回は、それを取ってくる側です

全51回の、今日はここ



原石工房

Claude Code 101

1 2 3 4 5 6 7 8 9 10 11 12

Claude Code in Action

13 14 15 16 17 18 19 20 21

スキル

22 23 24 25 26 27

手分け (subagents)

28 29 30 31

MCP 入門

32 33 34 35 36 37 38 39 40

MCP 応用

41 42 43 44 45 46 47 48 49 50 51

第1回～第12回

第13回～第21回

第22回～第27回

第28回～第31回

第32回～第40回

第41回～第51回

今日はここ 第38回

今日のレッスン



原石工房

Introduction to MCP の第7回

原題 Accessing resources

訳 資料を取りに行く

道具を呼んでデータを取りに行かなくてよくなります

① 道具を呼ぶのと、何が違うのか



原石工房

道具を呼んで取りに行かなくてよくなる

道具で取りに行く

呼び出しが1回はさまる

Claude が道具を選ぶ

動かす

結果が返る

資料として渡す

はじめから入っている

@ で指す

中身が頼みごとに入る

そのまま答えられる

「@… には何を書いてある？」と打つと

それを資料への頼みだと見分けて、中身を取ってきます

道具で取りに行くと、呼び出しが1回はさまります

資料として渡すなら、はじめから頼みごとの中に入っています

「@… には何を書いてある？」を、資料への頼みだと見分けます

前後の事情の一部に、はじめからなる

② read_resource を作る



原石工房

原題 Implementing Resource Reading

作るのは、read_resource という働き1つ

まず、要るものを取り込む

```
import json / from pydantic import AnyUrl
```

① サーバーへ頼む

read_resource に住所を渡す

② 返ってきたものを開く

contents の最初の1つを取る

③ 種類を見て、扱いを決める

JSON なら読み解く、それ以外は文字のまま

ふつうは一度に1つしか要らないので、最初のものを取ります

頼んで、開いて、種類で分ける

つなぐ側に、read_resource という
働きを1つ作ります

サーバーへ頼み、返ってきたものを
開き、種類を見て扱いを決めます

ふつうは一度に1つしか要らないの
で、最初のものを取ります

③ 返ってくるものの形



原石工房

原題 Understanding the Response Structure

返ってくるものに入っているもの

実際の中身

文字、またはデータ

データの種類

どう読み解けばよいかを教えてくれる

そのほかの覚え書き

その資料についての情報

種類によって、扱いを変える

application/json なら

JSON として読み解き、形のあるものにして返す

それ以外なら

文字のまま返す

形のあるデータも、ただの文書も、どちらもすんなり扱えます

実際の中身 (文字、またはデータ)

データの種類 (どう読み解けばよいかを教
えてくれる)

その資料についての、そのほかの覚
え書き

種類が、読み方を教えてくれる

④ 種類で、扱いを変える



原石工房

原題 Content Type Handling

返ってくるものに入っているもの

実際の中身

文字、またはデータ

データの種類

どう読み解けばよいかを教えてくれる

そのほかの覚え書き

その資料についての情報

種類によって、扱いを変える

application/json なら

JSON として読み解き、形のあるものにして返す

それ以外なら

文字のまま返す

形のあるデータも、ただの文書も、どちらもすんなり扱えます

application/json なら、JSON として読み解いて返します

それ以外なら、文字のまま返します

形のあるデータも、ただの文書も、どちらもすんなり扱えます

種類が、読み方を教えてくれる

2つの道



原石工房

使う人が @ を打つ

候補が出る

使える資料が並ぶ

選ぶ

中身が入る

頼みごとの中へ、そのまま

AI へ送る

道具の呼び出しは、なし

はじめの前後の事情の一部になる

答えが返る

すぐ

取りに行く往復がない

往復が減るぶん、速くなる

⑤ 使い心地



原石工房

原題 Testing Resource Access

@ を打つと、こうなる

① 使える資料が、候補として出てくる

② 矢印キーとスペースで選べる

③ その資料の中身が、頼みごとの中へそのまま入る

④ 道具を呼ばずに、そのまま AI へ送られる

別途、道具を呼ばせるより、ずっとなめらか
そのデータについて、すぐに答えが返ってきます

@ を打つと、使える資料が候補として出ます

選ぶと、その中身が頼みごとの中へそのまま入ります

道具を呼ばずに、そのまま AI へ送られます

はじめの前後の事情の、一部になる

今回のまとめ



原石工房

作るのは、`read_resource` という働き1つ

頼んで、開いて、種類で分ける

JSON なら読み解き、それ以外は文字のまま

@ で指すと、中身が頼みごとの一部になる

取りに行く往復が減るぶん、答えが速くなる



Introduction to MCP の第8回

原題 Defining prompts

訳 頼み方を定義する

サーバーが差し出す3つめ、よく練られた指示のひな型です

全51回の、39回目です

その往復、なくせませんか



原石工房

隠れた才能に気づき、花開く