



第33回 「Introduction to MCP」
Anthropic 公式 Claude Academy

Claude Code

つなぐ側

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 / MCP clients

前回「MCP とは何か」の続きです

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

第1回～第12回

Claude Code in Action

13 14 15 16 17 18 19 20 21

第13回～第21回

スキル

22 23 24 25 26 27

第22回～第27回

手分け (subagents)

28 29 30 31

第28回～第31回

MCP 入門

32 33 34 35 36 37 38 39 40

第32回～第40回

MCP 応用

41 42 43 44 45 46 47 48 49 50 51

第41回～第51回

今日はここ 第33回

今日のレッスン



原石工房

Introduction to MCP の第2回

原題 MCP clients

訳 つなぐ側

つなぐ側は、自分のところと MCP サーバーのあいだの、橋になります

① つなぎ方を選ばない

原題 Transport Agnostic Communication

つなぐ側は、あいだの橋

言葉のやりとりと、取り決めの細かいところを引き受ける
だから、自分の作っているものが、それを抱えずに済みます

つなぎ方は、選ばない

決まった入り口と出口
同じ機械で動かす。いちばんよくある形

HTTP
ウェブで使われている、ふつうのやりとり

WebSockets
つなぎっぱなしでやりとりする形

そのほかの網のつなぎ方
組み方しだいで選べます

どの道でも、同じように話せる

言葉のやりとりと、取り決めの細かいところを引き受けてくれます

いちばんよくあるのは、同じ機械で動かし、決まった入り口と出口で話す形です

HTTP、WebSockets、そのほかの網のつなぎ方も使えます

② やりとりする言葉

原題 MCP Message Types

やりとりする言葉は、2組

ListToolsRequest

どんな道具を持っていますか

→ ListToolsResult 使える道具の一覧

CallToolRequest

この道具を、この中身で動かして

→ CallToolResult その結果

どちらも、つなぐ側からサーバーへ尋ね、返事を受け取る形

MCP の取り決めで、言葉の種類が定められています

尋ねる言葉と、動かす言葉

ListToolsRequest どんな道具を持っていますか、と尋ねます

ListToolsResult 使える道具の一覧が返ります

CallToolRequest と
CallToolResult 道具を動かして、結果を受け取ります

③ 全体の流れ（行き）



原石工房

原題 How It All Works Together

「どんな置き場所を持っていますか」と尋ねると

① 使う人が、自分のところへ問いを出す

② 自分のところが、つなぐ側へ「使える道具は」と尋ねる

③ つなぐ側が ListToolsRequest を送り、一覧を受け取る

④ 自分のところが、問いと道具を Claude へ送る

⑤ Claude が「道具を呼ぶ必要がある」と決める

⑥ つなぐ側が CallToolRequest を送り、サーバーが GitHub を呼ぶ

続きは、結果が戻って Claude が答えをまとめるまで

段は多いが、受け持ちははっきり分かれている

使う人が問いを出す → 自分のところが、つなぐ側へ道具を尋ねる

つなぐ側が ListToolsRequest を送り、一覧を受け取る

自分のところが、問いと道具を Claude へ送り、Claude が道具を呼ぶと決める

③ 全体の流れ（戻り）



原石工房

結果が、戻ってくる

⑦ GitHub がデータを返す

⑧ サーバーを通じて CallToolResult として戻る

⑨ 自分のところが、道具の結果を Claude へ送る

⑩ Claude が、データを使って答えをまとめる

⑪ 自分のところが、Claude の答えを使う人へ返す

つなぐ側が、込み入ったところを覆い隠してくれる
だから、自分の作っているものの筋道に集中できます

つなぐ側が CallToolRequest を送り、サーバーが GitHub の口を呼ぶ

GitHub がデータを返し、
CallToolResult として戻る

Claude が答えをまとめ、自分のところが使う人へ返します

この流れが、この先ずっと出てくる

ひと続きの流れ



原石工房

使う人

問いを出す

「どんな置き場所を持っていますか」

自分のところ

道具を知り、Claude へ渡す

つなぐ側に尋ね、一覧を受け取る

つなぐ側

MCP の言葉で話す

ListTools と CallTool の2組

MCP サーバー

外の口を実際に呼ぶ

GitHub からデータを受け取る

行って、戻る。12の段に分かれている

④ 段は多いが、受け持ちは分かれている



原石工房

12の段に分かれていますが、それぞれの部品の受け持ちははっきりしています

つなぐ側が、サーバーとのやりとりの込み入ったところを覆い隠します

だから、自分の作っているものの筋道に集中できます

今回のまとめ



原石工房

つなぐ側は、自分のところとサーバーのあいだの橋

つなぎ方は選ばない。同じ機械でも、網ごしでもよい

やりとりする言葉は2組。道具を尋ねる組と、動かす組

流れは12の段。行って、戻る

込み入ったところは、つなぐ側が覆い隠してくれる



Introduction to MCP の第3回

原題 Defining tools with MCP

訳 道具を定義する

公式の Python の道具立てを使って、実際に道具を作ります

全51回の、34回目です

その橋は、何を隠してくれていますか



原石工房

隠れた才能に気づき、花開く