



第34回 「Introduction to MCP」
Anthropic 公式 Claude Academy

Claude Code

道具を定義する

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 /
Defining tools with MCP

ここから、実際に手を動かす回に入ります

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34回



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 の道具立てを使うと、サーバー作りはぐっと簡単になります

① サーバーを立ち上げる



原石工房

原題 Setting Up the MCP Server

立ち上げるのは、たった1行

```
from mcp.server.fastmcp import FastMCP
mcp = FastMCP("DocumentMCP", log_level="ERROR")
```

この例で作るもの

書類を読む道具
番号から、中身を返す

書類を書き換える道具
探して、置き換える

書類は、覚えのうえの簡単な一覧として持つ
鍵が書類の番号、値がその中身

入り組んだ JSON の決め書きは、書きません

たった1行で立ち上げられます

この例で作るのは、書類を読む道具
と、書き換える道具の2つ

書類は、覚えのうえの簡単な一覧と
して持ちます

重い仕事は、道具立てに任せる

② 飾りで道具を定める

原題 Tool Definition with Decorators

飾り（デコレーター）で、道具を定める

```
@mcp.tool(  
    name="read_doc_contents",  
    description="書類の中身を読んで、文字として返す"  
)  
def read_document(  
    doc_id: str = Field(description="読む書類の番号")  
)
```

飾りが決めるもの

道具の名前と、説明

関数の欄が決めるもの

要る中身（引数）

Field が決めるもの

その欄が何を待っているか

型の目印と欄の説明から、正しい決め書きが自動でできます

関数を書けば、道具になる

JSON の決め書きを手で書く代わりに、型の目印と欄の説明を書きます

道具立てが、Claude に分かる正しい決め書きを、自動で作ります

飾りが名前と説明を、関数の欄が要る中身を定めます

③ 読む道具を作る



原題 Creating a Document Reader Tool

飾り（デコレーター）で、道具を定める

```
@mcp.tool(  
    name="read_doc_contents",  
    description="書類の中身を読んで、文字として返す"  
)  
def read_document(  
    doc_id: str = Field(description="読む書類の番号")  
)
```

飾りが決めるもの

道具の名前と、説明

関数の欄が決めるもの

要る中身（引数）

Field が決めるもの

その欄が何を待っているか

型の目印と欄の説明から、正しい決め書きが自動でできます

関数を書けば、道具になる

番号から書類の中身を読んで、返します

Field が、その欄が何を待っているのかを Claude に説明します

見つからないときは、その場で知らせます

④ 書き換える道具を作る



原石工房

原題 Building a Document Editor Tool

書き換える道具は、3つを受け取る

`doc_id`

書き換える書類の番号

`old_str`

置き換える前の文字。空白まで、そのまま合うこと

`new_str`

代わりに入れる文字

書類が見つからないときの備えも、中に入っています

あとは、素直な文字の置き換えをするだけ

第17回のフックと同じで、合わなければ何も起きません

そのまま合うことが、条件

受け取るのは3つ。書類の番号、探す文字、置き換える文字

探す文字は、空白まで、そのまま合うことが条件です

書類が見つからないときの備えも、中に入っています

書く量の違い



原石工房

手で書く

JSON の決め書きを自分で

欄の名前と型を、1つずつ書く

説明文を、別に書く

確かめも、自分で足す

道具の登録も、自分で

飾りを使う

Python の関数を定めるだけ

型の目印が、そのまま型になる

Field の説明が、そのまま説明に

確かめは、自動

登録も、自動

同じ道具が、ずっと短く書ける

⑤ この道具立ての良いところ



原題 Key Benefits of the SDK Approach

この道具立てを使う良いところ

- ・ JSON の決め書きを、手で書かなくてよい
- ・ 型の目印が、確かめを自動でやってくれる
- ・ はっきりした欄の説明が、使い方を Claude に分からせる
- ・ 困りごとの備えが、Python のふだんのやり方となじむ
- ・ 道具の登録が、飾りによって自動で起きる

入り組んだ決め書きを書く作業から
素直な関数を定めることへ、作り変わる

作るのも、手入れするのも、楽になる

JSON の決め書きを、手で書かなくてよい

型の目印が、確かめを自動でやってくれる

はっきりした欄の説明が、使い方を Claude に分からせる

⑤ 残りの2つ



原石工房

困りごとの備えが、Python のふだんのやり方となじむ

道具の登録が、飾りによって自動で起きる

入り組んだ決め書きを書く作業から、素直な関数を定めることへ、作り変わります

今回のまとめ



原石工房

サーバーは、1行で立ち上がる

道具は、飾りを付けた関数として定める

型の目印と Field の説明から、決め書きが自動でできる

書き換えは、空白まで、そのまま合うことが条件

作るのも、手入れするのも、ずっと楽になる



Introduction to MCP の第4回

原題 The server inspector

訳 動作を覗く道具

作ったサーバーが、思ったとおりに動いているかを見る回です

全51回の、35回目です

その道具、何を待っていると書いてありますか



原石工房

隠れた才能に気づき、花開く