

```

list6.txt
response = openai.chat.completions.create(model="gpt-4o", messages=[{"role": "user", "content": prompt}], tools=tools,
tool_choice="auto") # モデルはfunction calling対応のものを指定

message = response.choices[0].message
messages = [{"role": "user", "content": prompt}]
for _ in range(3):
    if message.tool_calls:
        assistant_message = message.to_dict()
        messages.append(assistant_message)
        for tool_call in message.tool_calls:
            tool_name = tool_call.function.name
            args = json.loads(tool_call.function.arguments)
            # 中略
            mcp_response = requests.post(f"{MCP_BASE_URL}/use_mcp_tool", json={"tool_name": tool_name, "arguments": args})
            mcp_result = mcp_response.json().get("result")
            messages.append({"role": "tool", "content": str(mcp_result), "tool_call_id": tool_call.id})
        # 再度LLMにtool messageを渡して応答生成
        response = openai.chat.completions.create(model="gpt-4-0613", messages=messages, tools=tools, tool_choice="auto")
        message = response.choices[0].message
    else:
        # ツール呼び出しが不要な場合はそのまま返す
        return {"reply": message.content.strip()}
# tool_callsが続く場合はエラーを返す
return {"reply": "ツール呼び出しが3回を超えたため、処理を中断します。"}

```