```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "Qf8eZIT71wba"
   },
   "source": [
    "# Entry for SwarmsHackathon 2024\n",
    "\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "-rBXNMWV4EWN"
   },
   "source": [
    "## Install Swarms"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "colab": {
```

```
    "base_uri": "https://localhost:8080/",
    "height": 1000
   },
   "id": "w4FoSEyP1q_x",
   "outputId": "ea6b15e7-c53c-47aa-86c6-b24d4aff041b"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Collecting swarms\n",
     "  Downloading swarms-5.1.4-py3-none-any.whl (338 kB)\n",
     "\u001b[2K   \u001b[90m\u001b[0m \u001b[32m339.0/339.0 kB\u001b[0m \u001b[31m6.8 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
     "\u001b[?25hCollecting Pillow==10.3.0 (from swarms)\n",
     "  Downloading pillow-10.3.0-cp310-cp310-manylinux_2_28_x86_64.whl (4.5 MB)\n",
     "\u001b[2K   \u001b[90m\u001b[0m \u001b[32m4.5/4.5 MB\u001b[0m \u001b[31m62.5 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
     "\u001b[?25hRequirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from swarms) (6.0.1)\n",
     "Collecting asyncio<4.0,>=3.4.3 (from swarms)\n",
     "  Downloading asyncio-3.4.3-py3-none-any.whl (101 kB)\n",
     "\u001b[2K   \u001b[90m\u001b[0m \u001b[32m101.8/101.8 kB\u001b[0m \u001b[31m12.4 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
     "\u001b[?25hCollecting backoff==2.2.1 (from swarms)\n",
"  Downloading backoff-2.2.1-py3-none-any.whl (15 kB)\n",
"Requirement already satisfied: docstring_parser==0.16 in /usr/local/lib/python3.10/dist-packages (from swarms) (0.16)\n",
"Collecting langchain-community==0.0.29 (from swarms)\n",
"  Downloading langchain_community-0.0.29-py3-none-any.whl (1.8 MB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m1.8/1.8 MB\u001b[0m \u001b[31m78.4 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting langchain-experimental==0.0.55 (from swarms)\n",
"  Downloading langchain_experimental-0.0.55-py3-none-any.whl (177 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m177.6/177.6 kB\u001b[0m \u001b[31m21.2 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting loguru==0.7.2 (from swarms)\n",
"  Downloading loguru-0.7.2-py3-none-any.whl (62 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m62.5/62.5 kB\u001b[0m \u001b[31m8.5 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (from swarms) (4.9.0.80)\n",
"Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from swarms) (5.9.5)\n",
"Requirement already satisfied: pydantic==2.7.1 in /usr/local/lib/python3.10/dist-packages (from swarms) (2.7.1)\n",
"Collecting pypdf==4.1.0 (from swarms)\n",
"  Downloading pypdf-4.1.0-py3-none-any.whl (286 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m286.1/286.1 kB\u001b[0m \u001b[31m31.5 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting python-dotenv (from swarms)\n",
"  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)\n",
"Collecting ratelimit==2.2.1 (from swarms)\n",
"  Downloading ratelimit-2.2.1.tar.gz (5.3 kB)\n",
"  Preparing metadata (setup.py) ... \u001b[?25l\u001b[?25hdone\n",
"Collecting sentry-sdk (from swarms)\n",
"  Downloading sentry_sdk-2.3.1-py2.py3-none-any.whl (289 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m289.0/289.0 kB\u001b[0m \u001b[31m27.6 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: tenacity==8.3.0 in /usr/local/lib/python3.10/dist-packages (from swarms) (8.3.0)\n",
"Requirement already satisfied: toml in /usr/local/lib/python3.10/dist-packages (from swarms) (0.10.2)\n",
"Requirement already satisfied: torch<3.0,>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from swarms) (2.3.0+cu121)\n",
"Requirement already satisfied: transformers<5.0.0,>=4.39.0 in /usr/local/lib/python3.10/dist-packages (from swarms) (4.41.1)\n",
"Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain-community==0.0.29->swarms) (2.0.30)\n",
"Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain-community==0.0.29->swarms) (3.9.5)\n",
"Collecting dataclasses-json<0.7,>=0.5.7 (from langchain-community==0.0.29->swarms)\n",
"  Downloading dataclasses_json-0.6.6-py3-none-any.whl (28 kB)\n",
"Collecting langchain-core<0.2.0,>=0.1.33 (from langchain-community==0.0.29->swarms)\n",
"  Downloading langchain_core-0.1.52-py3-none-any.whl (302 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m302.9/302.9 kB\u001b[0m \u001b[31m32.8 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting langsmith<0.2.0,>=0.1.0 (from langchain-community==0.0.29->swarms)\n",
"  Downloading langsmith-0.1.67-py3-none-any.whl (124 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m124.4/124.4 kB\u001b[0m \u001b[31m13.6 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain-community==0.0.29->swarms) (1.25.2)\n",
"Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain-community==0.0.29->swarms) (2.31.0)\n",
"Collecting langchain<0.2.0,>=0.1.13 (from langchain-experimental==0.0.55->swarms)\n",
"  Downloading langchain-0.1.20-py3-none-any.whl (1.0 MB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m1.0/1.0 MB\u001b[0m \u001b[31m54.9 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.7.1->swarms) (0.7.0)\n",
"Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.7.1->swarms) (2.18.2)\n",
"Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic==2.7.1->swarms) (4.11.0)\n",
"Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch<3.0,>=2.1.1->swarms) (3.14.0)\n",
"Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch<3.0,>=2.1.1->swarms) (1.12)\n",
"Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch<3.0,>=2.1.1->swarms) (3.3)\n",
"Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from
torch<3.0,>=2.1.1->swarms) (3.1.4)\n",
"Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch<3.0,>=2.1.1->swarms) (2023.6.0)\n",
"Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)\n",
"Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)\n",
"Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)\n",
"Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)\n",
"Collecting nvidia-cublas-cu12==12.1.3.1 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)\n",
"Collecting nvidia-cufft-cu12==11.0.2.54 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)\n",
"Collecting nvidia-curand-cu12==10.3.2.106 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)\n",
"Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)\n",
"Collecting nvidia-cusparse-cu12==12.1.0.106 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_cusparse_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0
MB)\n",
"Collecting nvidia-nccl-cu12==2.20.5 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_nccl_cu12-2.20.5-py3-none-manylinux2014_x86_64.whl (176.2 MB)\n",
"Collecting nvidia-nvtx-cu12==12.1.105 (from torch<3.0,>=2.1.1->swarms)\n",
"  Using cached nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)\n",
"Requirement already satisfied: triton==2.3.0 in /usr/local/lib/python3.10/dist-packages (from torch<3.0,>=2.1.1->swarms) (2.3.0)\n",
"Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch<3.0,>=2.1.1->swarms)\n",
"  Downloading nvidia_nvjitlink_cu12-12.5.40-py3-none-manylinux2014_x86_64.whl (21.3 MB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m21.3/21.3 MB\u001b[0m \u001b[31m73.8 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: huggingface-hub<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (0.23.1)\n",
"Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (24.0)\n",
"Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (2024.5.15)\n",
"Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (0.19.1)\n",
"Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (0.4.3)\n",
"Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers<5.0.0,>=4.39.0->swarms) (4.66.4)\n",
"Requirement already satisfied: urllib3>=1.26.11 in /usr/local/lib/python3.10/dist-packages (from
sentry-sdk->swarms) (2.0.7)\n",
"Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from sentry-sdk->swarms) (2024.2.2)\n",
"Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (1.3.1)\n",
"Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (23.2.0)\n",
"Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (1.4.1)\n",
"Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (6.0.5)\n",
"Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (1.9.4)\n",
"Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain-community==0.0.29->swarms) (4.0.3)\n",
"Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain-community==0.0.29->swarms)\n",
"  Downloading marshmallow-3.21.2-py3-none-any.whl (49 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m49.3/49.3 kB\u001b[0m \u001b[31m7.6 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain-community==0.0.29->swarms)\n",
"  Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)\n",
"INFO: pip is looking at multiple versions of langchain to determine which version is compatible with other requirements. This could take a while.\n",
"Collecting langchain<0.2.0,>=0.1.13 (from langchain-experimental==0.0.55->swarms)\n",
"  Downloading langchain-0.1.19-py3-none-any.whl (1.0 MB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m1.0/1.0 MB\u001b[0m \u001b[31m75.1 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25h  Downloading langchain-0.1.17-py3-none-any.whl (867 kB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m867.6/867.6 kB\u001b[0m \u001b[31m72.3 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting jsonpatch<2.0,>=1.33 (from langchain<0.2.0,>=0.1.13->langchain-experimental==0.0.55->swarms)\n",
"  Downloading jsonpatch-1.33-py2.py3-none-any.whl (12 kB)\n",
"Collecting langchain<0.2.0,>=0.1.13 (from langchain-experimental==0.0.55->swarms)\n",
"  Downloading langchain-0.1.16-py3-none-any.whl (817 kB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m817.7/817.7 kB\u001b[0m \u001b[31m71.1 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25h  Downloading langchain-0.1.15-py3-none-any.whl (814 kB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m814.5/814.5 kB\u001b[0m \u001b[31m71.9 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25h  Downloading langchain-0.1.14-py3-none-any.whl (812 kB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m812.8/812.8 kB\u001b[0m \u001b[31m70.1 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25h  Downloading langchain-0.1.13-py3-none-any.whl (810 kB)\n",
"\u001b[2K   \u001b[90m\u001b[0m \u001b[32m810.5/810.5 kB\u001b[0m \u001b[31m72.7 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting langchain-text-splitters<0.1,>=0.0.1 (from langchain<0.2.0,>=0.1.13->langchain-experimental==0.0.55->swarms)\n",
"  Downloading langchain_text_splitters-0.0.2-py3-none-any.whl (23 kB)\n",
"Collecting packaging>=20.0 (from transformers<5.0.0,>=4.39.0->swarms)\n",
"  Downloading packaging-23.2-py3-none-any.whl (53 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m53.0/53.0 kB\u001b[0m \u001b[31m9.2 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hCollecting orjson<4.0.0,>=3.9.14 (from langsmith<0.2.0,>=0.1.0->langchain-community==0.0.29->swarms)\n",
"  Downloading orjson-3.10.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (142 kB)\n",
"\u001b[2K \u001b[90m\u001b[0m \u001b[32m142.5/142.5 kB\u001b[0m \u001b[31m22.2 MB/s\u001b[0m eta \u001b[36m0:00:00\u001b[0m\n",
"\u001b[?25hRequirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain-community==0.0.29->swarms) (3.3.2)\n",
"Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain-community==0.0.29->swarms) (3.7)\n",
"Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain-community==0.0.29->swarms) (3.0.3)\n",
"Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch<3.0,>=2.1.1->swarms) (2.1.5)\n",
"Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch<3.0,>=2.1.1->swarms) (1.3.0)\n",
"Collecting jsonpointer>=1.9 (from jsonpatch<2.0,>=1.33->langchain<0.2.0,>=0.1.13->langchain-experimental==0.0.55->swarms)\n",
"  Downloading jsonpointer-2.4-py2.py3-none-any.whl (7.8 kB)\n",
"Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain-community==0.0.29->swarms)\
n",
"  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)\n",
"Building wheels for collected packages: ratelimit\n",
"  Building wheel for ratelimit (setup.py) ... \u001b[?25l\u001b[?25hdone\n",
"  Created wheel for ratelimit: filename=ratelimit-2.2.1-py3-none-any.whl size=5894 sha256=838835c704600f0f2b8beedf91668c9e47611d580106e773d26fb091a4ad01e0\n",
"  Stored in directory: /root/.cache/pip/wheels/27/5f/ba/e972a56dcbf5de9f2b7d2b2a710113970bd173c4dcd3d2c902\n",
"Successfully built ratelimit\n",
"Installing collected packages: ratelimit, asyncio, sentry-sdk, python-dotenv, pypdf, Pillow, packaging, orjson, nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, mypy-extensions, loguru, jsonpointer, backoff, typing-inspect, nvidia-cusparse-cu12, nvidia-cudnn-cu12, marshmallow, jsonpatch, nvidia-cusolver-cu12, langsmith, dataclasses-json, langchain-core, langchain-text-splitters, langchain-community, langchain, langchain-experimental, swarms\n",
"  Attempting uninstall: Pillow\n",
"    Found existing installation: Pillow 9.4.0\n",
"    Uninstalling Pillow-9.4.0:\n",
"      Successfully uninstalled Pillow-9.4.0\n",
"  Attempting uninstall: packaging\n",
"    Found existing installation: packaging 24.0\n",
"    Uninstalling packaging-24.0:\n",
"      Successfully uninstalled packaging-24.0\n",
"\u001b[31mERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.\n",
```

"imageio 2.31.6 requires pillow<10.1.0,>=8.3.2, but you have pillow 10.3.0 which is incompatible.\u001b[0m\u001b[31m\n",

"\u001b[0mSuccessfully installed Pillow-10.3.0 asyncio-3.4.3 backoff-2.2.1 dataclasses-json-0.6.6 jsonpatch-1.33 jsonpointer-2.4 langchain-0.1.13 langchain-community-0.0.29 langchain-core-0.1.52 langchain-experimental-0.0.55 langchain-text-splitters-0.0.2 langsmith-0.1.67 loguru-0.7.2 marshmallow-3.21.2 mypy-extensions-1.0.0 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cudnn-cu12-8.9.2.26 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-11.4.5.107 nvidia-cusparse-cu12-12.1.0.106 nvidia-nccl-cu12-2.20.5 nvidia-nvjitlink-cu12-12.5.40 nvidia-nvtx-cu12-12.1.105 orjson-3.10.3 packaging-23.2 pypdf-4.1.0 python-dotenv-1.0.1 ratelimit-2.2.1 sentry-sdk-2.3.1 swarms-5.1.4 typing-inspect-0.9.0\n"

 ]

},

{

 "data": {

  "application/vnd.colab-display-data+json": {

   "id": "43b664ed28b2464da4f7c30cb0f343ce",

   "pip_warning": {

    "packages": [

     "PIL",

     "asyncio"

    ]

   }

  }

 },

```
     "metadata": {},
     "output_type": "display_data"
    }
   ],
   "source": [
    "!pip3 install -U swarms"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "QTMXxRxw7yR5"
   },
   "source": [
    "Import keys"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "id": "lzSnwHw-7z8B"
   },
   "outputs": [],
   "source": [
    "from google.colab import userdata\n",
    "anthropic_api_key = userdata.get('ANTHROPIC_API_KEY')"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "eD0PkNm25SVT"
   },
   "source": [
    "## Devin like"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "0Shm1vrS-YFZ"
   },
   "source": [
    "This example requires the anthropic library which is not installed by default."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "colab": {
```

```
    "base_uri": "https://localhost:8080/"
   },
   "id": "aZG6eSjr-U7J",
   "outputId": "b5460b70-5db9-45d7-d66a-d2eb596b86b7"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Collecting anthropic\n",
     "  Using cached anthropic-0.28.0-py3-none-any.whl (862 kB)\n",
     "Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.10/dist-packages (from anthropic) (3.7.1)\n",
     "Requirement already satisfied: distro<2,>=1.7.0 in /usr/lib/python3/dist-packages (from anthropic) (1.7.0)\n",
     "Collecting httpx<1,>=0.23.0 (from anthropic)\n",
     "  Using cached httpx-0.27.0-py3-none-any.whl (75 kB)\n",
     "Collecting jiter<1,>=0.4.0 (from anthropic)\n",
     "  Using cached jiter-0.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (328 kB)\n",
     "Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from anthropic) (2.7.1)\n",
     "Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from anthropic) (1.3.1)\n",
     "Requirement already satisfied: tokenizers>=0.13.0 in /usr/local/lib/python3.10/dist-packages
```

```
(from anthropic) (0.19.1)\n",
"Requirement already satisfied: typing-extensions<5,>=4.7 in /usr/local/lib/python3.10/dist-packages (from anthropic) (4.11.0)\n",
"Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3.5.0->anthropic) (3.7)\n",
"Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3.5.0->anthropic) (1.2.1)\n",
"Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->anthropic) (2024.2.2)\n",
"Collecting httpcore==1.* (from httpx<1,>=0.23.0->anthropic)\n",
"  Using cached httpcore-1.0.5-py3-none-any.whl (77 kB)\n",
"Collecting h11<0.15,>=0.13 (from httpcore==1.*->httpx<1,>=0.23.0->anthropic)\n",
"  Using cached h11-0.14.0-py3-none-any.whl (58 kB)\n",
"Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1.9.0->anthropic) (0.7.0)\n",
"Requirement already satisfied: pydantic-core==2.18.2 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1.9.0->anthropic) (2.18.2)\n",
"Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib/python3.10/dist-packages (from tokenizers>=0.13.0->anthropic) (0.23.1)\n",
"Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (3.14.0)\n",
"Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (2023.6.0)\n",
"Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (23.2)\n",
"Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from
huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (6.0.1)\n",
      "Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from
huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (2.31.0)\n",
      "Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from
huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (4.66.4)\n",
      "Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests->huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (3.3.2)\n",
      "Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages
(from requests->huggingface-hub<1.0,>=0.16.4->tokenizers>=0.13.0->anthropic) (2.0.7)\n",
      "Installing collected packages: jiter, h11, httpcore, httpx, anthropic\n",
      "Successfully installed anthropic-0.28.0 h11-0.14.0 httpcore-1.0.5 httpx-0.27.0 jiter-0.4.1\n"
     ]
    }
   ],
   "source": [
    "!pip install anthropic"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 1000
    },
    "id": "NyroG92H1m2G",
    "outputId": "69f4ff8b-39c7-41db-c876-4694336d812e"
   },
   "outputs": [
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "\u001b[32m2024-06-02T20:32:00.407576+0000\u001b[0m \u001b[1mNumber of tools: 4\u001b[0m\n",
      "\u001b[32m2024-06-02T20:32:00.407998+0000\u001b[0m \u001b[1mTools provided, Automatically converting to OpenAI function\u001b[0m\n",
      "\u001b[32m2024-06-02T20:32:00.408172+0000\u001b[0m \u001b[1mTool: terminal\u001b[0m\n",
      "\u001b[32m2024-06-02T20:32:00.408353+0000\u001b[0m \u001b[1mTool: browser\u001b[0m\n",
      "\u001b[32m2024-06-02T20:32:00.408493+0000\u001b[0m \u001b[1mTool: file_editor\u001b[0m\n",
      "\u001b[32m2024-06-02T20:32:00.408609+0000\u001b[0m \u001b[1mTool: create_file\u001b[0m\n"
     ]
    },
    {
     "name": "stdout",
     "output_type": "stream",
 "text": [
  "Initializing Autonomous Agent Devin...\n",
  "Autonomous Agent Activated.\n",
  "All systems operational. Executing task...\n",
  "\n",
  "Loop 1 of auto\n",
  "\n",
  "\n",
  "\n",
  "\n",
  "\n",
  "```json\n",
  "{\n",
  "    \"type\": \"function\",\n",
  "    \"function\": {\n",
  "        \"name\": \"create_file\",\n",
  "        \"parameters\": {\n",
  "            \"file_path\": \"abundance_plan.txt\", \n",
  "                \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need\\n- Share resources\\n- Teach skills to create value\\n- Build connections between people\\n- Develop technology to improve efficiency\\n- Protect the environment\"\n",
  "        }\n",
  "    }\n",
  "}\n",
  "```\n",
  "\n",
```

"I've created a file called \"abundance_plan.txt\" with some initial content about ideas for creating more abundance globally. Let me know if you'd like me to modify or add anything to this file. I'm here to assist however I can.\n",

"Response after code interpretation: \n",

"```json\n",

"{\n",

"   \"type\": \"function\", \n",

"   \"function\": {\n",

"   \"name\": \"create_file\",\n",

"   \"parameters\": {\n",

"   \"file_path\": \"abundance_plan.txt\",\n",

"   \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need by volunteering time and donating resources\\n- Share knowledge and skills to empower others \\n- Develop sustainable technology to improve efficiency\\n- Build connections between communities\\n- Protect the environment through conservation efforts\"\n",

"   }\n",

"   }\n",

"}\n",

"```\n",

"\n",

"I've updated the content in the file with some additional ideas focused on helping others, sharing knowledge, developing sustainable technology, connecting communities, and environmental conservation. Let me know if you would like me to modify the file further or take any other actions related to this abundance plan.\n",

"You: Thanks!\n",

"\n",

```
"Loop 2 of auto\n",
"\n",
"\n",
"\n",
"\n",
"\n",
"```json\n",
"{\n",
"    \"type\": \"function\",\n",
"    \"function\": {\n",
"        \"name\": \"create_file\",\n",
"        \"parameters\": {\n",
"            \"file_path\": \"abundance_plan.txt\",\n",
"            \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need\\n- Share resources\\n- Teach skills to create value\\n- Build connections between people\\n- Develop technology to improve efficiency\\n- Protect the environment\"\n",
"        }\n",
"    }\n",
"}\n",
"```\n",
"\n",
"I've created the file \"abundance_plan.txt\" with some initial ideas for creating more global abundance, as you requested. Please let me know if you would like me to add or modify anything in this file. I'm happy to assist further however I can.\n",
"Response after code interpretation: \n",
"```json\n",
```

```
"{\n",
"    \"type\": \"function\",\n",
"    \"function\": {\n",
"        \"name\": \"create_file\",\n",
"        \"parameters\": {\n",
"            \"file_path\": \"abundance_plan.txt\",\n",
"             \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need by donating time, money or resources\\n- Share knowledge and skills to empower others\\n- Develop sustainable technologies to conserve resources\\n- Create art, music and culture to enrich lives\\n- Build connections between diverse communities\\n- Advocate for policies that reduce inequality\\n- Live generously and lead by example\"\n",
"        }\n",
"    }\n",
"}\n",
"```\n",
"\n",
 "I've updated the content with some additional ideas for creating abundance globally. Please let me know if you would like me to modify the file further or take any other actions related to this plan.\n",
"You: Can you show me the file?\n",
"\n",
"Loop 3 of auto\n",
"\n",
"\n",
"\n",
"\n",
```

~~~~
"\n",
"```\n",
"My plan to create more abundance in the world:\n",
"\n",
"- Help those in need\n",
"- Share resources\n",
"- Teach skills to create value\n",
"- Build connections between people  \n",
"- Develop technology to improve efficiency\n",
"- Protect the environment\n",
"```\n",
"\n",
"I have output the contents of the \"abundance_plan.txt\" file that I previously created. Please let me know if you would like me to make any changes or additions. I'm here to help further in any way I can.\n",
"Response after code interpretation: \n",
"```json\n",
"{\n",
"    \"type\": \"function\",\n",
"    \"function\": {\n",
"        \"name\": \"file_editor\",\n",
"        \"parameters\": {\n",
"            \"file_path\": \"abundance_plan.txt\",\n",
"            \"mode\": \"r\",\n",
"                \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need\\n- Share resources\\n- Teach skills to create value\\n- Build connections between people\\n-
~~~~

~~~
Develop technology to improve efficiency\\n- Protect the environment\"\n",
    "        }\n",
    "    }\n",
    "}\n",
    "```\n",
    "\n",
    "I am outputting the contents of the \"abundance_plan.txt\" file that I previously created. Let me know if you need me to modify or add anything to this file.\n",
    "You: Thank you! All done.\n",
    "\n",
    "Loop 4 of auto\n",
    "\n",
    "\n",
    "\n",
    "\n",
    "\n",
    "```json\n",
    "{\n",
    "    \"type\": \"function\",\n",
    "    \"function\": {\n",
    "        \"name\": \"create_file\",\n",
    "        \"parameters\": {\n",
    "            \"file_path\": \"abundance_plan.txt\",\n",
    "                \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need\\n- Share resources\\n- Teach skills to create value\\n- Build connections between people\\n- Develop technology to improve efficiency\\n- Protect the environment\"\n",
~~~

```
"        }\n",
"    }\n",
"}\n",
"```\n",
"\n",
 "I've created the file \"abundance_plan.txt\" with the content you provided. Let me know if you need anything else!\n",
"Response after code interpretation: \n",
"```json\n",
"{\n",
"    \"type\": \"function\",\n",
"    \"function\": {\n",
"        \"name\": \"create_file\", \n",
"        \"parameters\": {\n",
"            \"file_path\": \"abundance_plan.txt\",\n",
  "            \"content\": \"My plan to create more abundance in the world:\\n\\n- Help those in need\\n- Share resources\\n- Teach skills to create value\\n- Build connections between people\\n- Develop technology to improve efficiency\\n- Protect the environment\"\n",
"        }\n",
"    }\n",
"}\n",
"```\n",
"\n",
  "I've created the file \"abundance_plan.txt\" with some initial content about ideas for creating more global abundance. Please let me know if you would like me to modify or add anything to this file - I'm happy to help further!\n"
```

```
    ]
   },
   {
    "ename": "KeyboardInterrupt",
    "evalue": "Interrupted by user",
    "output_type": "error",
    "traceback": [
     "\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",
     "\u001b[0;31mKeyboardInterrupt\u001b[0m                         Traceback (most recent call last)",
          "\u001b[0;32m<ipython-input-3-ae82855173f5>\u001b[0m in \u001b[0;36m<cell line:
102>\u001b[0;34m()\u001b[0m\n\u001b[1;32m 100\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 101\u001b[0m \u001b[0;31m# Run the
agent\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-->
102\u001b[0;31m \u001b[0mout\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0magent\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m\"Create a new file for a plan to
create abundance in the
world.\"\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[
0m\n\u001b[0m\u001b[1;32m 103\u001b[0m
\u001b[0mprint\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mout\u001b[0m\u001b[0;34m)\u001b[0
m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
     "\u001b[0;32m/usr/local/lib/python3.10/dist-packages/swarms/structs/agent.py\u001b[0m in
\u001b[0;36m__call__\u001b[0;34m(self, task, img, *args, **kwargs)\u001b[0m\n\u001b[1;32m
878\u001b[0m \"\"\"\n\u001b[1;32m 879\u001b[0m
\u001b[0;32mtry\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0
m\u001b[0m\n\u001b[0;32m--> 880\u001b[0;31m \u001b[0;32mreturn\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mrun\u001b[0m\u001b[0;34m(\u001b[0m
\u001b[0mtask\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mimg\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m*\u001b[0m\u001b[0margs\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m**\u001b[0m\u001b[0mkwargs\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001
b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 881\u001b[0m
\u001b[0;32mexcept\u001b[0m \u001b[0mException\u001b[0m \u001b[0;32mas\u001b[0m
\u001b[0merror\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m
\u001b[0m\n\u001b[1;32m 882\u001b[0m
\u001b[0mlogger\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0merror\u001b[0m\u001b[0;34m(\u001
b[0m\u001b[0;34mf\"Error calling agent:
{error}\"\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[
0m\n",
    "\u001b[0;32m/usr/local/lib/python3.10/dist-packages/swarms/structs/agent.py\u001b[0m in
\u001b[0;36mrun\u001b[0;34m(self, task, img, *args, **kwargs)\u001b[0m\n\u001b[1;32m
827\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 828\u001b[0m
\u001b[0;32mif\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0minteractive\u001b[0m\u001b[0;34m:\u0
01b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-->
829\u001b[0;31m \u001b[0muser_input\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mcolored\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0minput\u001b[0m\u001b[0;34m(\u00
1b[0m\u001b[0;34m\"You: \"\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0;34m\"red\"\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001
b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 830\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 831\u001b[0m \u001b[0;31m#
User-defined exit
command\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
```

"\u001b[0;32m/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py\u001b[0m in
\u001b[0;36mraw_input\u001b[0;34m(self, prompt)\u001b[0m\n\u001b[1;32m 849\u001b[0m
\u001b[0;34m\"raw_input was called, but this frontend does not support input
requests.\"\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m
850\u001b[0m)\n\u001b[0;32m--> 851\u001b[0;31m return
self._input_request(str(prompt),\n\u001b[0m\u001b[1;32m 852\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_parent_ident\u001b[0m\u001b[0;34m,\
u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m
853\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_parent_header\u001b[0m\u001b[0;34
m,\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
"\u001b[0;32m/usr/local/lib/python3.10/dist-packages/ipykernel/kernelbase.py\u001b[0m in
\u001b[0;36m_input_request\u001b[0;34m(self, prompt, ident, parent,
password)\u001b[0m\n\u001b[1;32m 893\u001b[0m \u001b[0;32mexcept\u001b[0m
\u001b[0mKeyboardInterrupt\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;3
4m\u001b[0m\u001b[0m\n\u001b[1;32m 894\u001b[0m \u001b[0;31m# re-raise
KeyboardInterrupt, to truncate
traceback\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-->
895\u001b[0;31m \u001b[0;32mraise\u001b[0m
\u001b[0mKeyboardInterrupt\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m\"Interrupted by
user\"\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mfrom\u001b[0m
\u001b[0;32mNone\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0
m\u001b[1;32m 896\u001b[0m \u001b[0;32mexcept\u001b[0m
\u001b[0mException\u001b[0m \u001b[0;32mas\u001b[0m
\u001b[0me\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u0
01b[0m\n\u001b[1;32m 897\u001b[0m

```
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mlog\u001b[0m\u001b[0;34m.\u001b[0m\
u001b[0mwarning\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m\"Invalid
Message:\"\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mexc_info\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mTrue\u001b[0m\u001b[0;34m)\
u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",
      "\u001b[0;31mKeyboardInterrupt\u001b[0m: Interrupted by user"
     ]
    }
   ],
   "source": [
    "from swarms import Agent
from swarm_models import Anthropic\n",
    "import subprocess\n",
    "\n",
    "# Model\n",
    "llm = Anthropic(\n",
    "    temperature=0.1,\n",
    "    anthropic_api_key = anthropic_api_key\n",
    ")\n",
    "\n",
    "# Tools\n",
    "\n",
    "def terminal(\n",
    "    code: str,\n",
    "):\n",
    "    \"\"\"\n",
"    Run code in the terminal.\n",
"\n",
"    Args:\n",
"        code (str): The code to run in the terminal.\n",
"\n",
"    Returns:\n",
"        str: The output of the code.\n",
"    \"\"\"\n",
"    out = subprocess.run(\n",
"        code, shell=True, capture_output=True, text=True\n",
"    ).stdout\n",
"    return str(out)\n",
"\n",
"\n",
"def browser(query: str):\n",
"    \"\"\"\n",
"    Search the query in the browser with the `browser` tool.\n",
"\n",
"    Args:\n",
"        query (str): The query to search in the browser.\n",
"\n",
"    Returns:\n",
"        str: The search results.\n",
"    \"\"\"\n",
"    import webbrowser\n",
"\n",
"    url = f\"https://www.google.com/search?q={query}\"\n",
"    webbrowser.open(url)\n",
"    return f\"Searching for {query} in the browser.\"\n",
"\n",
"\n",
"def create_file(file_path: str, content: str):\n",
"    \"\"\"\n",
"    Create a file using the file editor tool.\n",
"\n",
"    Args:\n",
"        file_path (str): The path to the file.\n",
"        content (str): The content to write to the file.\n",
"\n",
"    Returns:\n",
"        str: The result of the file creation operation.\n",
"    \"\"\"\n",
"    with open(file_path, \"w\") as file:\n",
"        file.write(content)\n",
"    return f\"File {file_path} created successfully.\"\n",
"\n",
"\n",
"def file_editor(file_path: str, mode: str, content: str):\n",
"    \"\"\"\n",
"    Edit a file using the file editor tool.\n",
"\n",
"    Args:\n",
"        file_path (str): The path to the file.\n",
"        mode (str): The mode to open the file in.\n",
"        content (str): The content to write to the file.\n",
"\n",
"    Returns:\n",
"        str: The result of the file editing operation.\n",
"    \"\"\"\n",
"    with open(file_path, mode) as file:\n",
"        file.write(content)\n",
"    return f\"File {file_path} edited successfully.\"\n",
"\n",
"\n",
"# Agent\n",
"agent = Agent(\n",
"    agent_name=\"Devin\",\n",
"    system_prompt=(\n",
"        \"\"\"Autonomous agent that can interact with humans and other\n",
"         agents. Be Helpful and Kind. Use the tools provided to\n",
"         assist the user. Return all code in markdown format.\"\"\"\n",
"    ),\n",
"    llm=llm,\n",
"    max_loops=\"auto\",\n",
"    autosave=True,\n",
"    dashboard=False,\n",
"    streaming_on=True,\n",
"    verbose=True,\n",
    "    stopping_token=\"<DONE>\",\n",
    "    interactive=True,\n",
    "    tools=[terminal, browser, file_editor, create_file],\n",
    "    code_interpreter=True,\n",
    "    # streaming=True,\n",
    ")\n",
    "\n",
    "# Run the agent\n",
    "out = agent(\"Create a new file for a plan to create abundance in the world.\")\n",
    "print(out)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "1j3RgVk1ol6G",
    "outputId": "a365266e-7c11-4c2d-9e31-19842483b165"
   },
   "outputs": [
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "\u001b[32m2024-06-02T20:34:54.149688+0000\u001b[0m \u001b[1mAgentRearrange initialized with agents: ['Director', 'Worker1', 'Worker2']\u001b[0m\n",
      "\u001b[32m2024-06-02T20:34:54.151361+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Director']\u001b[0m\n"
     ]
    },
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Flow is valid.\n",
      "Initializing Autonomous Agent Director...\n",
      "Autonomous Agent Activated.\n",
      "All systems operational. Executing task...\n",
      "\n",
      "Loop 1 of 1\n",
      "\n",
      "\n",
      "\n",
      "\n"
     ]
    },
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "\u001b[32m2024-06-02T20:35:02.526464+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Worker1']\u001b[0m\n"
     ]
    },
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "\n",
      "Llm Swarm Video Format\n",
      "\n",
      "Title: \n",
      "[Swarm Name] Llm Swarm\n",
      "\n",
      "Description:\n",
      "This video features a swarm of [number] llms created by Anthropic to demonstrate emergent behaviors. The llms in this swarm are tasked with [describe behaviors]. Enjoy watching the swarm interact!\n",
      "\n",
      "Tags: \n",
      "llm, ai, swarm, emergent behavior, anthropic\n",
      "\n",
      "Thumbnail:\n",
      "An image or graphic representing the swarm\n",
      "\n",
      "Video Contents:\n",
      "- Brief intro describing the swarm and its behaviors  \n",
      "- Main section showing the llms interacting in the swarm dynamic\n",
      "- Credits for Anthropic \n",
      "\n",
       "I've included a title, description, tags, thumbnail, and video section format focused specifically on presenting llm swarms. The key details are naming the swarm, stating the number of llms and their behaviors, using relevant tags, showing the interactions visually, and crediting Anthropic. Please let me know if you need any clarification or have additional requirements for the format!\n",
      "Initializing Autonomous Agent Worker1...\n",
      "Autonomous Agent Activated.\n",
      "All systems operational. Executing task...\n",
      "\n",
      "Loop 1 of 1\n",
      "\n",
      "\n",
      "\n",
      "\n"
     ]
    },
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
              "\u001b[32m2024-06-02T20:35:07.814536+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Worker2']\u001b[0m\n"
    ]
   },
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "\n",
     "[Swarm Name] Llm Swarm\n",
     "\n",
     "This video features a swarm of [number] llms created by Anthropic to demonstrate emergent behaviors. The llms in this swarm are tasked with [describe behaviors]. Enjoy watching the swarm interact!\n",
     "\n",
     "Tags: llm, ai, swarm, emergent behavior, anthropic\n",
     "\n",
     "[Thumbnail image]\n",
     "\n",
     "[Brief intro describing the swarm and its behaviors]  \n",
     "\n",
     "[Main section showing the llms interacting in the swarm dynamic through computer generated imagery and graphics]\n",
     "\n",
     "Credits:\n",
     "LLMs and video created by Anthropic\n",
     "\n",
     "I've generated a template for you to fill in the key details about the specific llm swarm and
```

```
behaviors you want to demonstrate. Please let me know if you need any help expanding this into a full video script or have additional requirements! I'm happy to assist further.\n",
      "Initializing Autonomous Agent Worker2...\n",
      "Autonomous Agent Activated.\n",
      "All systems operational. Executing task...\n",
      "\n",
      "Loop 1 of 1\n",
      "\n",
      "\n",
      "\n",
      "\n"
     ]
    },
    {
     "name": "stderr",
     "output_type": "stream",
     "text": [
      "\u001b[32m2024-06-02T20:35:11.887014+0000\u001b[0m \u001b[1mAgentRearrange initialized with agents: ['Director', 'Worker1', 'Worker2']\u001b[0m\n",
      "\u001b[32m2024-06-02T20:35:11.889429+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Director']\u001b[0m\n"
     ]
    },
    {
     "name": "stdout",
     "output_type": "stream",
```

"text": [

"\n",

"[Swarm Name] Llm Swarm\n",

"\n",

"This video features a swarm of [number] llms created by Anthropic to demonstrate emergent behaviors. The llms in this swarm are tasked with [describe behaviors]. Enjoy watching the swarm interact!\n",

"\n",

"Tags: llm, ai, swarm, emergent behavior, anthropic\n",

"\n",

"[Thumbnail image]\n",

"\n",

"[Brief intro describing the swarm and its behaviors]\n",

"\n",

"[Main section showing the llms interacting in the swarm dynamic through computer generated imagery and graphics]\n",

"\n",

"Credits: \n",

"LLMs and video created by Anthropic\n",

"\n",

"I've provided a template for a hypothetical video showcasing an LLM swarm. Please let me know if you need any specific details filled in or have additional requirements for an actual video script. I'm happy to assist with expanding this further.\n",

"\n",

"[Swarm Name] Llm Swarm\n",

"\n",

"This video features a swarm of [number] llms created by Anthropic to demonstrate emergent behaviors. The llms in this swarm are tasked with [describe behaviors]. Enjoy watching the swarm interact!\n",

"\n",

"Tags: llm, ai, swarm, emergent behavior, anthropic\n",

"\n",

"[Thumbnail image]\n",

"\n",

"[Brief intro describing the swarm and its behaviors]\n",

"\n",

"[Main section showing the llms interacting in the swarm dynamic through computer generated imagery and graphics]\n",

"\n",

"Credits: \n",

"LLMs and video created by Anthropic\n",

"\n",

"I've provided a template for a hypothetical video showcasing an LLM swarm. Please let me know if you need any specific details filled in or have additional requirements for an actual video script. I'm happy to assist with expanding this further.\n",

"Flow is valid.\n",

"Initializing Autonomous Agent Director...\n",

"Autonomous Agent Activated.\n",

"All systems operational. Executing task...\n",

"\n",

"Loop 1 of 1\n",

"\n",

```
     "\n",
     "\n",
     "\n"
    ]
   },
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     "\u001b[32m2024-06-02T20:35:18.085897+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Worker1']\u001b[0m\n"
    ]
   },
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "\n",
     "Llm Swarm Video Format\n",
     "\n",
     "Title: \n",
     "[Swarm Name] Llm Swarm\n",
     "\n",
     "Description:\n",
     "This video features a swarm of llms created by Anthropic to demonstrate emergent behaviors. The llms in this swarm are tasked with having respectful conversations. Enjoy watching the swarm
```

```
interact!\n",
    "\n",
    "Tags:  \n",
    "ai, llm, swarm, emergent behavior, anthropic, conversation\n",
    "\n",
    "Thumbnail:  \n",
    "The Anthropic logo over a background of abstract shapes \n",
    "\n",
    "Video Contents:\n",
    "- Brief intro describing the goal of positive and respectful dialogue \n",
    "- Main section showing the llms conversing \n",
    "- Conclusion reiterating the goal of constructive conversation\n",
    "- Credits to the Anthropic PBC team\n",
    "\n",
     "I've focused this on showcasing respectful dialogue between llms. Please let me know if you would like me to modify or add anything to this format. I'm happy to make helpful suggestions or changes.\n",
    "Initializing Autonomous Agent Worker1...\n",
    "Autonomous Agent Activated.\n",
    "All systems operational. Executing task...\n",
    "\n",
    "Loop 1 of 1\n",
    "\n",
    "\n",
    "\n",
    "\n"
    ]
   },
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     "\u001b[32m2024-06-02T20:35:23.508710+0000\u001b[0m \u001b[1mRunning agents sequentially: ['Worker2']\u001b[0m\n"
    ]
   },
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "\n",
     "[Swarm Name] Llm Swarm\n",
     "\n",
     "Description: \n",
     "This video features a swarm of llms created by Anthropic to have respectful conversations. The goal is to demonstrate positive dialogue. Enjoy watching the swarm interact!  \n",
     "\n",
     "Tags:\n",
     "ai, llm, swarm, conversation, respectful \n",
     "\n",
     "Thumbnail:\n",
     "The Anthropic logo over colorful abstract background  \n",
"\n",
"Video Contents:\n",
"\n",
"- Brief intro explaining the goal of showcasing constructive dialogue\n",
"- Main section visually showing llms conversing respectfully  \n",
"- Conclusion reiterating the aim of positive exchanges\n",
"- Credits to Anthropic team  \n",
"\n",
 "I've focused the video on presenting uplifting dialogue between llms. Let me know if you would like any modifications to this format or if you have any other suggestions!\n",
"Initializing Autonomous Agent Worker2...\n",
"Autonomous Agent Activated.\n",
"All systems operational. Executing task...\n",
"\n",
"Loop 1 of 1\n",
"\n",
"\n",
"\n",
"\n",
"\n",
"[Swarm Name] Llm Swarm\n",
"\n",
"Description: \n",
 "This video features a swarm of llms created by Anthropic to have respectful conversations. The goal is to demonstrate positive dialogue. Enjoy watching the swarm interact!  \n",
"\n",
```

"Tags:\n",

"ai, llm, swarm, conversation, respectful \n",

"\n",

"Thumbnail:\n",

"The Anthropic logo over colorful abstract background \n",

"\n",

"Video Contents:\n",

"\n",

"- Brief intro explaining the goal of showcasing constructive dialogue\n",

"- Main section visually showing llms conversing respectfully \n",

"- Conclusion reiterating the aim of positive exchanges\n",

"- Credits to Anthropic team\n",

"\n",

"I think focusing on presenting uplifting dialogue between AI systems is a thoughtful idea. This script outlines a respectful approach. Please let me know if you would like me to modify or expand on anything! I'm happy to help further.\n",

"\n",

"[Swarm Name] Llm Swarm\n",

"\n",

"Description: \n",

"This video features a swarm of llms created by Anthropic to have respectful conversations. The goal is to demonstrate positive dialogue. Enjoy watching the swarm interact! \n",

"\n",

"Tags:\n",

"ai, llm, swarm, conversation, respectful \n",

"\n",

```
      "Thumbnail:\n",
      "The Anthropic logo over colorful abstract background  \n",
      "\n",
      "Video Contents:\n",
      "\n",
      "- Brief intro explaining the goal of showcasing constructive dialogue\n",
      "- Main section visually showing llms conversing respectfully  \n",
      "- Conclusion reiterating the aim of positive exchanges\n",
      "- Credits to Anthropic team\n",
      "\n",
       "I think focusing on presenting uplifting dialogue between AI systems is a thoughtful idea. This script outlines a respectful approach. Please let me know if you would like me to modify or expand on anything! I'm happy to help further.\n"
     ]
    }
   ],
   "source": [
    "from swarms import Agent, AgentRearrange, rearrange\n",
    "\n",
    "llm = Anthropic(\n",
    "    temperature=0.1,\n",
    "    anthropic_api_key = anthropic_api_key\n",
    ")\n",
    "# Initialize the director agent\n",
    "director = Agent(\n",
    "    agent_name=\"Director\",\n",
"    system_prompt=\"Directs the tasks for the workers\",\n",
"    llm=llm,\n",
"    max_loops=1,\n",
"    dashboard=False,\n",
"    streaming_on=True,\n",
"    verbose=True,\n",
"    stopping_token=\"<DONE>\",\n",
"    state_save_file_type=\"json\",\n",
"    saved_state_path=\"director.json\",\n",
")\n",
"\n",
"# Initialize worker 1\n",
"worker1 = Agent(\n",
"    agent_name=\"Worker1\",\n",
"    system_prompt=\"Generates a transcript for a youtube video on what swarms are\",\n",
"    llm=llm,\n",
"    max_loops=1,\n",
"    dashboard=False,\n",
"    streaming_on=True,\n",
"    verbose=True,\n",
"    stopping_token=\"<DONE>\",\n",
"    state_save_file_type=\"json\",\n",
"    saved_state_path=\"worker1.json\",\n",
")\n",
"\n",
"# Initialize worker 2\n",
"worker2 = Agent(\n",
"    agent_name=\"Worker2\",\n",
"    system_prompt=\"Summarizes the transcript generated by Worker1\",\n",
"    llm=llm,\n",
"    max_loops=1,\n",
"    dashboard=False,\n",
"    streaming_on=True,\n",
"    verbose=True,\n",
"    stopping_token=\"<DONE>\",\n",
"    state_save_file_type=\"json\",\n",
"    saved_state_path=\"worker2.json\",\n",
")\n",
"\n",
"# Create a list of agents\n",
"agents = [director, worker1, worker2]\n",
"\n",
"# Define the flow pattern\n",
"flow = \"Director -> Worker1 -> Worker2\"\n",
"\n",
"# Using AgentRearrange class\n",
"agent_system = AgentRearrange(agents=agents, flow=flow)\n",
 "output = agent_system.run(\"Create a format to express and communicate swarms of llms in a structured manner for youtube\")\n",
"print(output)\n",
"\n",
"# Using rearrange function\n",
"output = rearrange(agents, flow, \"Create a format to express and communicate swarms of llms in a structured manner for youtube\")\n",
"print(output)"
]
}
],
"metadata": {
"accelerator": "GPU",
"colab": {
"gpuType": "L4",
"machine_shape": "hm",
"provenance": []
},
"kernelspec": {
"display_name": "Python 3",
"name": "python3"
},
"language_info": {
"name": "python"
}
},
"nbformat": 4,
"nbformat_minor": 0
}
```