

欢迎加入 Jina AI 开源社区 解锁数据应用的可能性



官方网站

<https://jina.ai/>



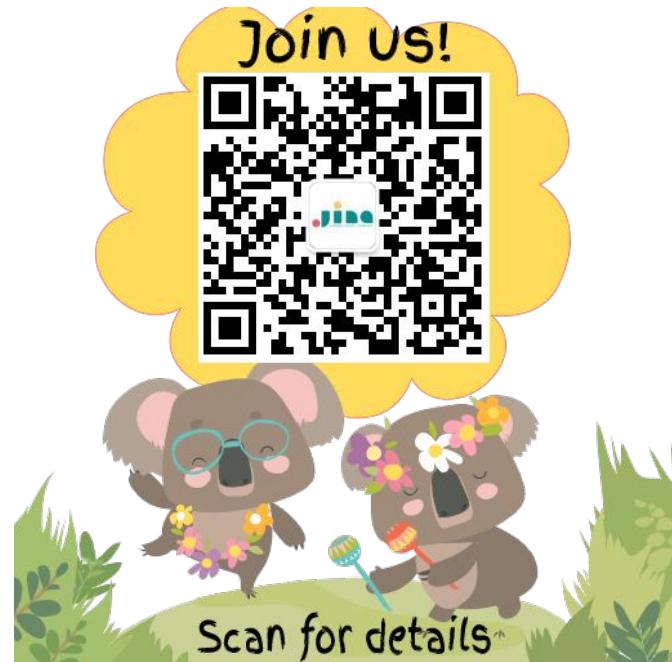
GitHub

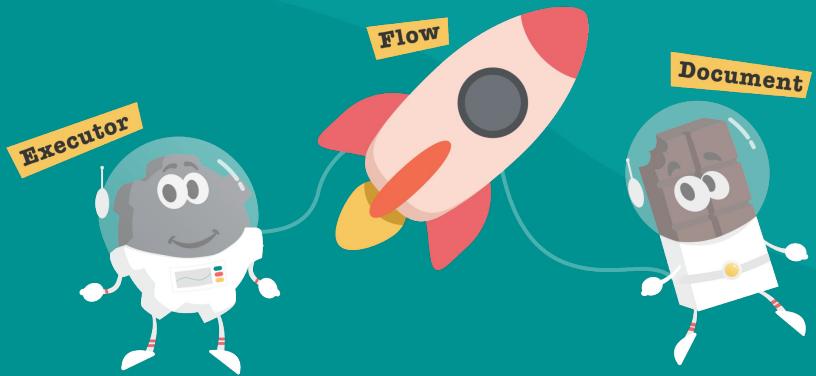
<https://github.com/jina-ai>



加入全球社区

<https://slack.jina.ai/>





Python 装饰器指南

J-tech Talk 02

Ziniu Y.

jina



About Jina AI



Jina AI 成立于 2020 年 2 月，是一家业内领先的开源企业，致力于通过人工智能和深度学习技术，打造下一代云原生神经搜索框架，帮助开发者和企业简化非结构化数据的搜索难题。

Jina AI 总部位于德国柏林，在北京、深圳、巴塞罗那等地均设有办公室，海外员工人数比例超过三分之二。

2021 年，Jina AI 完成了由 Canaan Partners 领投的 3000 万美元 A 轮融资，截至目前融资总额达 3900 万美元。

Canaan

GGV CAPITAL

SAP.
io

YUNQI
PARTNERS

MANGO
CAPITAL



DocArray

将非结构化数据, 统一成同一种数据结构



Jina

适用于所有数据类型的云原生神经搜索框架



Finetuner

微调深度神经网络, 优化神经搜索任务中的向量



Hub

分享和发现神经搜索应用的可复用组件



CLIP-as-service

基于 CLIP 的图像和文本跨模态编码服务



JCloud

神经搜索系统的云端部署及管理平台



NOW

图像搜索的无代码解决方案

Creator of Neural Search

Contributor to Open Source

主要内容

02 什么是装饰器

嗯。。这是可以说的吗

02 Python装饰器的用法

一起写点代码吧

03 装饰器在Jina产品中的应用

以CLIP-as-service为例

什么是装饰器

01

What is a decorator?

什么是装饰器



What is a decorator?

什么是装饰器？

Python 装饰器是一种函数或类，可以在不修改原函数代码及功能的前提下，增添新的功能。

常见的使用场景有生成日志、性能测试、计算时间、处理端点等。



Decorator highlights

装饰器的特点

- Inputs: callable; returns: callable
- 在被修饰的函数前、后进行操作
- 轻量、琐碎任务，使核心功能锦上添花

Tips

- 简单易用，命名应一目了然
- 防止多重嵌套带来的副作用

装饰器的用法

02



Demo

代码演示

[示范代码链接](#)

装饰器在 **Jina** 产品中的应用

03



一起看点真家伙

CLIP-as-service

基于 CLIP 的图像和文本跨模态编码服务

Demo server>> grpcs://demo-cas.jina.ai:2096



```
pip install clip_server  
pip install clip_client
```

<https://clip-as-service.jina.ai>

<https://github.com/jina-ai/clip-as-service>



CLIP-as-service

指定处理端点

```
  @requests(on='/rank')
  async def rank(self, docs: 'DocumentArray', parameters: Dict, **kwargs):
      ...

  @requests
  async def encode(self, docs: 'DocumentArray', parameters: Dict = {}, **kwargs):
      ...
```

https://github.com/jina-ai/clip-as-service/blob/main/server/clip_server/executors/clip_torch.py#L85
<https://docs.jina.ai/fundamentals/executor/executor-methods/>

函数重载

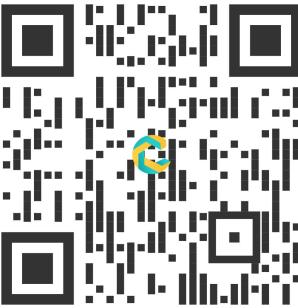
```
❶ @overload
❷ def encode(
❸     self, content: Iterable[str], *, batch_size: Optional[int] = None,
❹         show_progress: bool = False, parameters: Optional[dict] = None,
❺ ) → 'np.ndarray':
❻     ...
❼
➋ @overload
⌃ def encode(
⌄     self, content: Union['DocumentArray', Iterable['Document']], *,
⌅         batch_size: Optional[int] = None, show_progress: bool = False, parameters: Optional[dict] = None,
⌆ ) → 'DocumentArray':
⌇     ...
⌈
⌉ def encode(self, content, **kwargs):
    # Real implementation
```

https://github.com/jina-ai/clip-as-service/blob/main/server/clip_server/executors/clip_torch.py#L85



CLIP-as-service

启动时参数配置



```
    @click.command(name='clip-as-service benchmark')
    @click.argument('server')
    @click.option(
        '--batch_sizes',
        multiple=True,
        type=int,
        default=[1, 8, 16, 32, 64],
        help='number of batch',
    )
    @click.option(
        '--num_iter', default=10, help='number of repeat run per experiment (must > 2)'
    )
    @click.option(
        "--concurrent_clients",
        multiple=True,
        type=int,
        default=[1, 4, 16, 32, 64],
        help='number of concurrent clients per experiment',A
    )
    @click.option("--image_sample", help='path to the image sample file')
    def main(server, batch_sizes, num_iter, concurrent_clients, image_sample):
        # Real implementation
```

<https://github.com/jina-ai/clip-as-service/blob/main/scripts/benchmark.py#L87>

单元测试参数配置



```
@pytest.mark.parametrize('protocol', ['grpc', 'http', 'websocket', 'other'])
@pytest.mark.parametrize('jit', [True, False])
def test_protocols(port_generator, protocol, jit, pytestconfig):
    from clip_server.executors.clip_torch import CLIPExecutor
    if protocol == 'other':
        with pytest.raises(ValueError):
            Client(server=f'{protocol}://0.0.0.0:8000')
        return
    f = Flow(port=port_generator(), protocol=protocol).add(
        uses=CLIPExecutor, uses_with={'jit': jit}
    )
    with f:
        c = Client(server=f'{protocol}://0.0.0.0:{f.port}')
        c.profile()
```

https://github.com/jina-ai/clip-as-service/blob/main/tests/test_simple.py#L10

.jina



加点广告:)



文档问答机器人

DocsQA开放公测

<https://jina.ai/docsQA-console/>

- 快速为代码仓库的文档构建问答机器人
- 整理汇总社区问题，积累社区知识
- 已部署在全部 Jina 全家桶文档以及 fastapi、kornia 等多个开源项目

Ask our docs!

You can ask questions about our docs. Try:

- Does Jina support Kubernetes?
- How can I traverse a nested DocumentArray?
- What are the basic concepts in Jina?

what is jina?



Powered by Jina



<https://github.com/jina-ai/discoart>



DISCOART

jina

"ocean beach front view in Van Gogh style"



Github

jina-ai/dalle-flow

jina

*"the statue of liberty wearing a
vr headset"*



Github

jina-ai/dalle-flow

<https://github.com/jina-ai/dalle-flow>

DALL·E FLOW

jina

Jina



Jina



Github < jina-ai/discoart



“一行代码的艺术”

DISCOART发布

<https://github.com/jina-ai/discoart>



A beautiful painting of clouds in the blue sky by Hayao Miyazaki, graphic art, wide angle landscape, 8k, anime nature digital art wallpaper



欢迎加入 Jina AI 开源社区 解锁数据应用的可能性



官方网站

<https://jina.ai/>



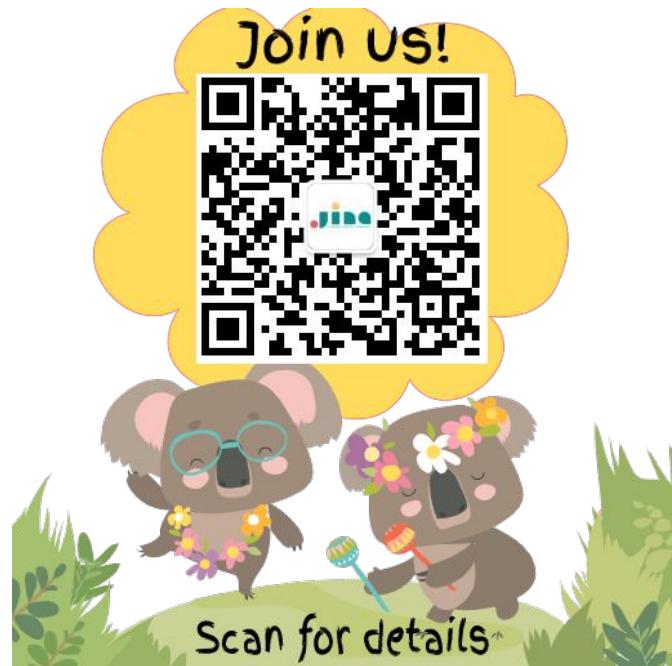
GitHub

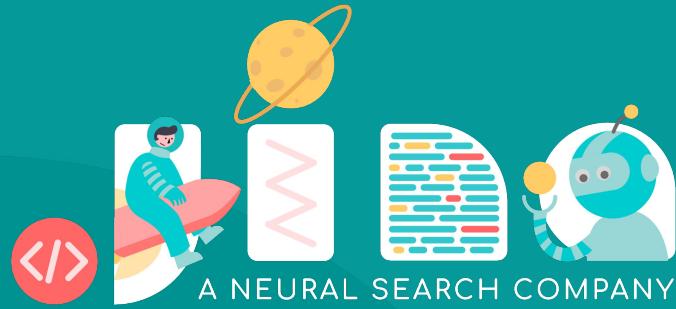
<https://github.com/jina-ai>



加入全球社区

<https://slack.jina.ai/>





A NEURAL SEARCH COMPANY

谢谢 !



Global



jina.ai



@ JinaAI_



get.jina.ai



slack.jina.ai