

Gradio 入门第一课：构建你的第一个AI演示应用

1. Gradio 核心概念

1.1 什么是Gradio?

- 🚀 开源Python包，快速为机器学习模型/API/Python函数构建Web应用
- ✨ 核心优势：
 - 无需前端知识(JavaScript/CSS)
 - 无需服务器托管经验
 - 几行代码创建交互式演示
 - 秒级生成可分享的公网链接

1.2 核心组件：Interface类

- 创建演示应用的"魔法引擎"
- 三要素结构：

```
gr.Interface(  
    fn=你的函数,      # 要包装的核心逻辑  
    inputs=输入组件,  # 用户交互接口  
    outputs=输出组件  # 结果展示方式  
)
```

2. 环境准备与安装

2.1 前提条件

- Python 3.10+ 环境
- 推荐使用虚拟环境

2.2 安装命令

```
pip install --upgrade gradio
```

3. 创建第一个AI应用

3.1 基础代码模板

创建一个问候交互式UI（界面）

```
import gradio as gr # 标准缩写  
  
# 定义核心功能函数  
def greet(name, intensity):
```

```
        return "Hello, " + name + "!" * int(intensity)

# 创建交互界面
demo = gr.Interface(
    fn=greet,                # 关联功能函数
    inputs=["text", "slider"], # 文本输入+滑块
    outputs="text"           # 文本输出
)

# 启动应用
demo.launch()
```

3.2 运行与查看

- 终端执行: `python app.py`
- 浏览器访问: `http://localhost:7860`
- 热重载模式 (修改自动刷新) :

```
gradio demo.py
```

4. Interface 类深度解析

4.1 三大核心参数

参数	说明	示例
<code>fn</code>	包装的函数	<code>fn=greet</code>
<code>inputs</code>	输入组件	<code>inputs=["text", "slider"]</code>
<code>outputs</code>	输出组件	<code>outputs="text"</code>

4.2 组件灵活使用

- 支持30+内置组件 (文本/图像/音频等)
- 两种指定方式:

```
# 方式1: 字符串简写
inputs="text"

# 方式2: 组件实例
inputs=gr.Textbox(label="姓名")
```

4.3 多输入输出处理

```
# 多输入示例
inputs=[gr.Textbox(), gr.Slider(1,10)]

# 多输出示例
outputs=[gr.Textbox(), gr.Image()]
```

5. 分享你的AI应用

5.1 一键分享功能

```
demo.launch(share=True) # 添加share参数
```

- 生成公共链接：🔗 <https://xxx.gradio.live>
- 全球用户可访问，计算仍在本地运行

5.2 分享效果

- 无需服务器配置
- 链接72小时内有效
- 支持密码保护（进阶功能）

6. Gradio 生态概览

组件	用途
<code>gr.Blocks</code>	高级自定义布局
<code>gr.ChatInterface</code>	聊天机器人专用
<code>gradio_client</code>	Python API客户端
Hugging Face Spaces	免费托管平台