

输出格式测试

高亮部分是显示有误的

markdown格式内容

1. Markdown 格式

文档主标题

章节副标题

这是一个包含**粗体文本**和*斜体文本*的段落，展示了基本的文本格式化功能。你也可以使用 ***粗斜体文本***。

- 列表项 1
- 列表项 2
 - 子列表项 2.1
 - 子列表项 2.2
 - 嵌套子列表项 2.2.1
- 列表项 3

这是一个有序列表：

1. 第一个项目
2. 第二个项目
 1. 子项目 A
 2. 子项目 B
3. 第三个项目

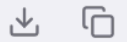
[点击这里访问示例链接](#)



代码块&语法高亮

Python

python

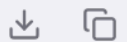


```
# Python "Hello World" 示例
def hello_world_python():
    message = "Hello, World! from Python"
    print(message)
    return message

if __name__ == "__main__":
    hello_world_python()
```

JavaScript

javascript

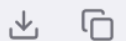


```
// JavaScript "Hello World" 示例
function helloWorldJavaScript() {
    const message = "Hello, World! from JavaScript";
    console.log(message);
    return message;
}

// 调用函数
helloWorldJavaScript();
```

SQL

sql

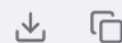


```
-- SQL "Hello World" 示例
-- 这是一个简单的 SELECT 语句，用于返回一个字符串
SELECT 'Hello, World! from SQL' AS Greeting;

-- 也可以插入到表中
-- CREATE TABLE messages (id INT PRIMARY KEY, text VARCHAR(255));
-- INSERT INTO messages (id, text) VALUES (1, 'Hello, World! from SQL');
-- SELECT text FROM messages WHERE id = 1;
```

Java 示例

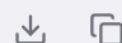
java



```
// HelloWorld.java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World! from Java");
    }
}
```

编译和运行：

bash

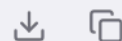


```
javac HelloWorld.java
java HelloWorld
```



C++ 示例

cpp



```
// hello_world.cpp
#include <iostream>

int main() {
    std::cout << "Hello, World! from C++" << std::endl;
    return 0;
}
```

浏览器深色模式下输出代码块的字体颜色改变

Meshub.ai

Agents

Exports

New Chat

My Teams

Original AI

WorkSmart

Life

My Chats

请生成一个包含以下所有内容的测...

Log in

请生成一个包含以下所有内容的测试示例： 1.Markdown 格式： 包括多级标题、列表、粗体/斜体文本、链...

python

import matplotlib.pyplot as plt

import numpy as np

def check_dark_mode_compatibility():

"""

检查深色模式兼容性的示例函数

包含各种语法高亮元素

"""

变量声明和基本操作

data_set = [x**2 for x in range(10)]

result = sum(data_set)

条件语句

if result > 100:

print(f"结果较大: {result}")

else:

print(f"结果正常: {result}")

类定义

class DataProcessor:

Please enter your topic...

Web Search

CSS

(深色模式)

CSS 变量示例 (用于主题切换)

CSS

:root {

--bg-primary: #FFFFFF;

--text-primary: #000000;

--accent-color: #007BFF;

}

[data-theme="dark"] {

--bg-primary: #121212;

--text-primary: #E0E0E0;

--accent-color: #BB86FC;

}

(浅色模式)

CSS

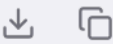


```
:root {  
  --bg-primary: #FFFFFF;  
  --text-primary: #000000;  
  --accent-color: #007BFF;  
}  
  
[data-theme="dark"] {  
  --bg-primary: #121212;  
  --text-primary: #E0E0E0;  
  --accent-color: #BB86FC;  
}
```

json/xml/yaml

JSON 对象示例

json

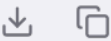


```
{
  "person": {
    "name": "张伟",
    "age": 28,
    "email": "zhangwei@example.com",
    "address": {
      "street": "人民路123号",
      "city": "北京",
      "postal_code": "100000"
    },
    "hobbies": ["阅读", "旅行", "游泳"]
  }
}
```



XML 示例

xml



```
<person>
  <name>张伟</name>
  <age>28</age>
  <email>zhangwei@example.com</email>
  <address>
    <street>人民路123号</street>
    <city>北京</city>
    <postal_code>100000</postal_code>
  </address>
  <hobbies>
    <hobby>阅读</hobby>
    <hobby>旅行</hobby>
    <hobby>游泳</hobby>
  </hobbies>
</person>
```



yaml格式



以下是一个简单的 Docker Compose YAML 片段示例，用于定义一个包含 Web 应用和数据库的服务：

yaml

⬇️ 📄

说明：

- **version:** 指定 Docker Compose 文件的版本。
- **services:** 定义了三个服务： web （使用 Nginx） ， app （使用 Node.js） ， 和 db

期望效果

yaml

📄 复制

```
version: '3.8' # 指定 Docker Compose 版本

services:
  # ===== Web 应用服务 =====
  webapp:
    image: my-webapp:1.0.0 # 自定义镜像
    build: ./webapp        # 构建上下文路径
    restart: always        # 容器退出时自动重启
    ports:
      - "5000:5000"        # 端口映射（主机:容器）
    environment:
      - DEBUG=false        # 环境变量
      - DB_HOST=database
```

表格

| 部门 | 季度 | 销售额（万元） | 增长率 | 备注 |
|-----|----|---------|------|------|
| 销售部 | Q1 | 150.5 | +15% | 表现优秀 |
| | Q2 | 168.2 | +12% | 持续增长 |
| 小计 | | 318.7 | | |
| 市场部 | Q1 | 85.3 | +8% | 稳定发展 |
| | Q2 | 92.1 | +8% | 达到预期 |
| 小计 | | 177.4 | | |
| 总计 | | 496.1 | +10% | 年度总览 |

| 日期 | 产品类别 | 产品名称 | 销售额（元） | 地区 |
|------------|------|-------|--------|----|
| 2023-01-01 | 电子产品 | 智能手机 | 5000 | 华东 |
| 2023-01-02 | 电子产品 | 笔记本电脑 | 8500 | 华南 |
| 2023-01-03 | 家居用品 | 智能音箱 | 800 | 华北 |
| 2023-01-04 | 办公用品 | 打印机 | 1200 | 华东 |
| 2023-01-05 | 电子产品 | 无线耳机 | 1500 | 华南 |
| 2023-01-06 | 家居用品 | 智能扫地机 | 2500 | 华北 |

复制结果：

| 日期 | 产品类别 | 产品名称 | 销售额（元） | 地区 |
| --- | --- | --- | --- | --- |
| 2023-01-01 | 电子产品 | 智能手机 | 5000 | 华东 |
| 2023-01-02 | 电子产品 | 笔记本电脑 | 8500 | 华南 |
| 2023-01-03 | 家居用品 | 智能音箱 | 800 | 华北 |
| 2023-01-04 | 办公用品 | 打印机 | 1200 | 华东 |
| 2023-01-05 | 电子产品 | 无线耳机 | 1500 | 华南 |
| 2023-01-06 | 家居用品 | 智能扫地机 | 2500 | 华北 |

| 日期 | 产品类别 | 产品名称 | 销售额（元） | 地区 |
|------------|------|-------|--------|----|
| 2023-01-01 | 电子产品 | 智能手机 | 5000 | 华东 |
| 2023-01-02 | 电子产品 | 笔记本电脑 | 8500 | 华南 |
| 2023-01-03 | 家居用品 | 智能音箱 | 800 | 华北 |
| 2023-01-04 | 办公用品 | 打印机 | 1200 | 华东 |
| 2023-01-05 | 电子产品 | 无线耳机 | 1500 | 华南 |

| 日期 | 产品类别 | 产品名称 | 销售额（元） | 地区 |
|------------|------|-------|--------|----|
| 2023-01-06 | 家居用品 | 智能扫地机 | 2500 | 华北 |

可以下载CSV和md格式文件

excel表格：

AI本身无法直接输出一个二进制的 `.xlsx` 文件让用户下载。它所能做的是模拟 **Excel** 中的表格数据，并以 **Markdown** 表格或 **HTML** 表格的形式返回，以便在网页上清晰地展示其结构和内容。

用户可以直接选中表格内容，复制后粘贴到 Excel 或 Google Sheets 中，大多数情况下能保持正确的列分隔。

CSV表格

以下是一个简单的 CSV 格式示例，包含一些基本的产品信息：

⬇️ 📄

在这个 CSV 示例中，第一行是表头，后面的每一行代表一条记录，字段之间用逗号分隔。您可以将其保存为 `.csv` 文件，并在电子表格软件（如 Excel 或 Google Sheets）中打开。

期望效果

CSV 数据示例

csv

复制

```
Name, Age, City, Occupation, Salary
John Doe, 28, New York, Engineer, 75000
Jane Smith, 32, Los Angeles, Designer, 68000
Bob Johnson, 45, Chicago, Manager, 92000
Alice Brown, 29, Austin, Developer, 80000
```

在页面上的预期显示效果

由于 CSV 是纯文本，在 AI 对话页面中，它通常会以一个**代码块**的形式显示，并可能标注语言为 `csv`或 `plaintext`，如下所示：

Excel公式

下面是一个复杂的 Excel 公式示例，它包含 IF 、 VLOOKUP 和 SUMIF 函数，并附有解释：

公式示例

excel

⬇️ 📄

公式解释

1. **VLOOKUP(A2, 产品表!A:B, 2, FALSE) :**
- 这个部分用于查找单元格 A2 中的产品 ID 在 产品表 工作表的 A 列中对应的产品状态。
 - 2 表示从查找范围的第二列（即产品状态列）返回值。
 - FALSE 表示进行精确匹配。
2. **IF(... = "缺货", "缺货", ...) :**
- 如果 VLOOKUP 查找到的产品状态为“缺货”，则返回“缺货”

期望效果

Excel 公式：

excel

📄 复制 🗑️

```
=IF(
  ISNA(VLOOKUP(F2, A2:D100, 4, FALSE)),
  "数据不全",
  SUMIF(
    A2:A100,
    F2,
    D2:D100
  )
)
```

公式解释：

1. **VLOOKUP(F2, A2:D100, 4, FALSE) :** 在 A2:D100区域的首列（A列）中精确查找 F2 单元格的值（产品名称），并返回同一行第4列（D列，销售额）的值。



Based on system analysis, ChatGpt will provide the best response. To involve another role, just @tag them

C ChatGpt

以下是一个 Excel 数据的任务列表，以纯文本的方式输出：



这个格式适合在文本编辑器中查看，数据以管道符号分隔，保持了列的对齐，便于阅读。



Sources

期望效果

例如，在你之前问的 Excel 公式例子中：

plaintext

复制



```
=IF(ISNA(VLOOKUP(F2, A2:D100, 4, FALSE)), "数据不全", SUMIF(A2:A100, F
```

latex公式

C ChatGpt

以下是高斯积分公式、傅里叶变换公式和麦克斯韦方程组的表达：

1. 高斯积分公式

高斯积分公式是计算高斯函数积分的一种重要方法，公式为：

$$\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$$

2. 傅里叶变换公式

傅里叶变换是将时间域信号转换为频率域信号的工具，其公式为：

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i \omega t} dt$$

其中， $F(\omega)$ 是信号的傅里叶变换， $f(t)$ 是时间域信号， ω 是角频率。



麦克斯韦方程组

期望效果

1. 高斯积分公式 (Gaussian Integral)

1. 高斯积分公式

这是一个非常重要的定积分结果，在概率论、物理学和工程学中广泛应用。

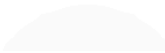
公式：

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

更一般的形式：

$$\int_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}$$

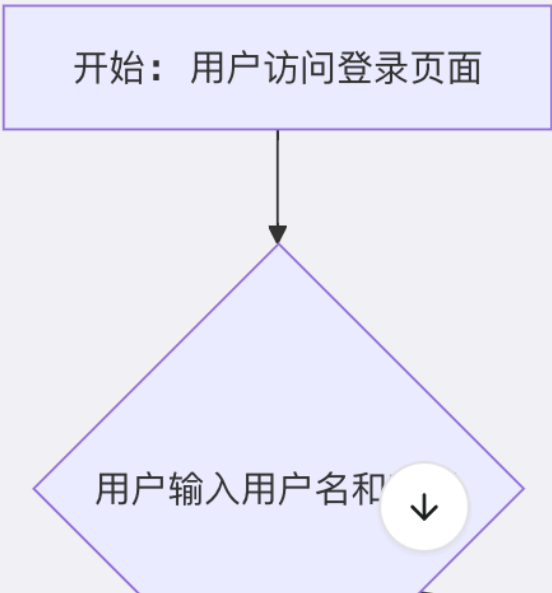
物理意义： 它计算了高斯函数（钟形曲线）下的总面积。



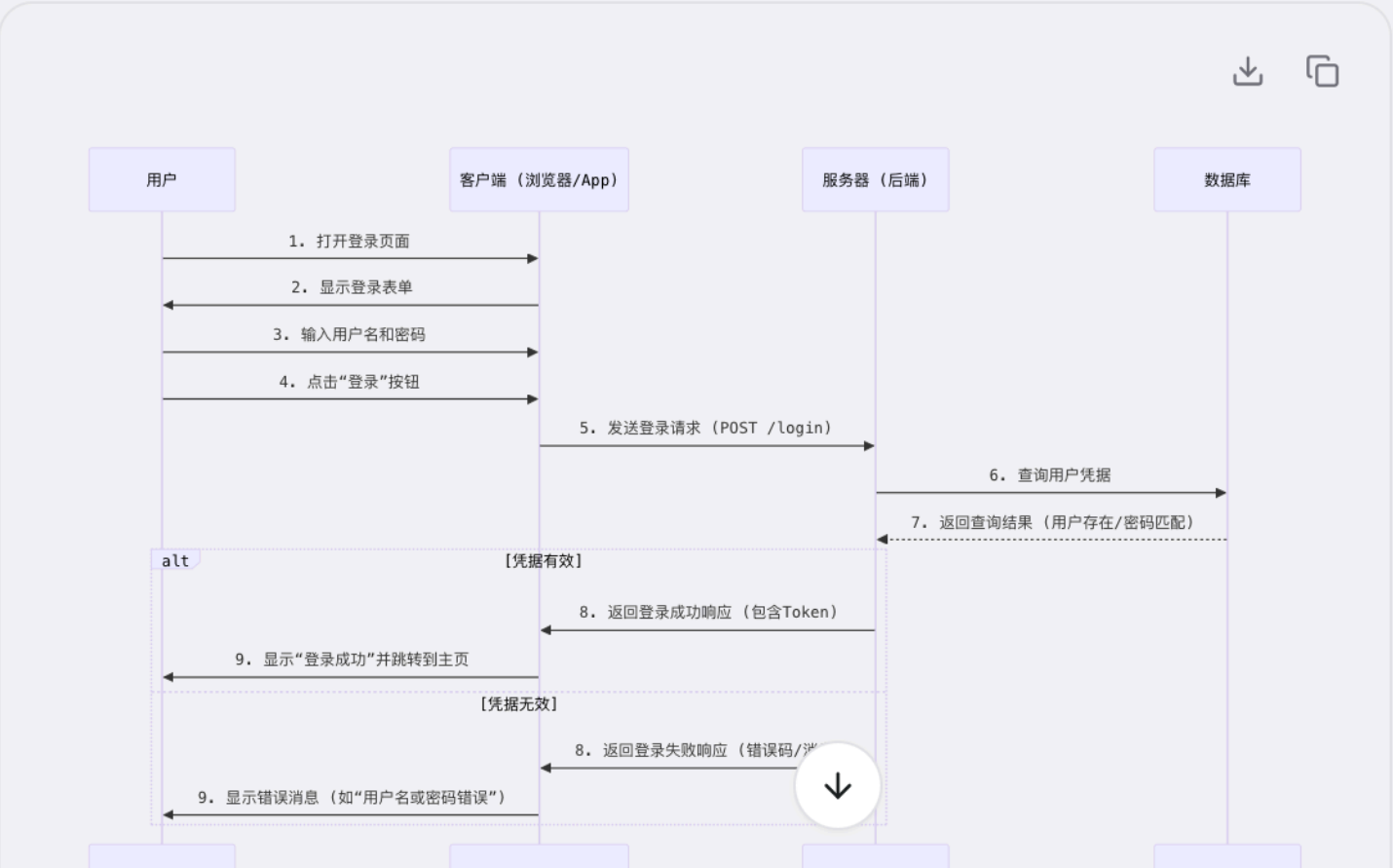
流程图/序列图/甘特图

当然，这是一个简单的用户登录流程图和序列图，使用 Mermaid 语法表示：

简单的用户登录流程图



简单的用户登录序列图



甘特图需要提示输出语法



Based on system analysis, ChatGpt will provide the best response. To involve another role, just @tag them

C ChatGpt

以下是一个甘特图的文本示例，用于表示项目任务的时间安排和进度。由于文本格式的限制，甘特图通常以条形图的形式呈现，下面的示例使用简单的字符来表示任务的开始和结束时间。

↓

📄

在这个甘特图中：

- 每个任务的名称位于左侧。
- 每个任务的时间段用块状字符（■）表示，显示任务的持续时间。
- 时间轴在顶部，从1号到31号，表示特定的日期。

您可以根据实际的项目需求和时间范围来调整任务和时间。这个文本格式的甘特图适合快速展示项目进度，但在实际应用中，通常会根据具体的项目需求来创建更复杂的甘特图。

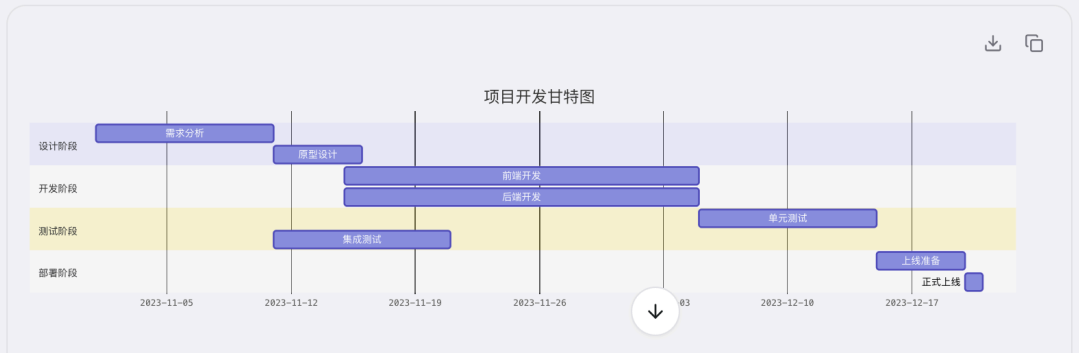
@ChatGpt 甘特图示例 用Mermaid 语法显示给我可以吗



Based on system analysis, ChatGpt will provide the best response. To involve another role, just @tag them

C ChatGpt

当然可以！以下是一个使用 Mermaid 语法绘制的甘特图示例：



任务列表

- 优先级：高
- 备注：需要收集数据和撰写总结。

- [] 准备下周会议
 - 截止日期：2023年11月10日
 - 优先级：中
 - 备注：需要制作演示文稿。

- [] 更新客户联系信息
 - 截止日期：2023年11月12日
 - 优先级：低
 - 备注：联系所有客户确认信息。

已完成的任务

- [x] 提交财务报表
 - 完成日期：2023年11月01日
 - 优先级：高
- [x] 整理团队活动照片
 - 完成日期：2023年11月02日
 - 优先级：中
- [x] 参加市场调研会议
 - 完成日期：2023年11月03日
 - 优先级：中



- ☐ 更新客户联系信息
 - 截止日期：2023年11月12日
 - 优先级：低
 - 备注：联系所有客户确认信息。

已完成的任务

- ☒ 提交财务报表
 - 完成日期：2023年11月01日
 - 优先级：高
- ☒ 整理团队活动照片
 - 完成日期：2023年11月02日
 - 优先级：中



这是一个引用块，用于引用重要的信息或声明。引用可以包含多行内容，并且可以继续在下一行。

C

ChatGpt

以下是一个简单的分割线示例，可以在文本中使用：

这是一段文本，上面有一个分割线。

你可以根据需要在文本中插入类似的分割线。