Cmd Markdown 发布第十五次更新 -- 斗罢艰险又出发

`Cmd-Markdown`

Cmd Markdown 命运多舛，距离2016年的最后一次重大更新已经过去了八个年头，距离首次发布的2013年已经过去了十一年，期间几度因为各种非人力可控的因素差一点停罢，好在有各位用户的鼎力支持和本人不言轻易放弃的劲头下，都有惊无险地度过了难关。目前作者本人处理好了私人事务，有充足的时间和精力继续维护和增强这项服务，所有数据和代码都在合法合规的前提下存储在可控安全的服务器上，在过去半年的时间里，查缺补漏，完善了因为年代久远，年久失修的功能，并且在八年之后发布第十五次更新：

1. 增加了对十九种图表绘制的 Markdown 语法支持
2. 提供所有图表的黑白配色主题(Ctrl + Alt + Y)
3. 所有十九种图表可以在 PDF 文件中原样下载输出。

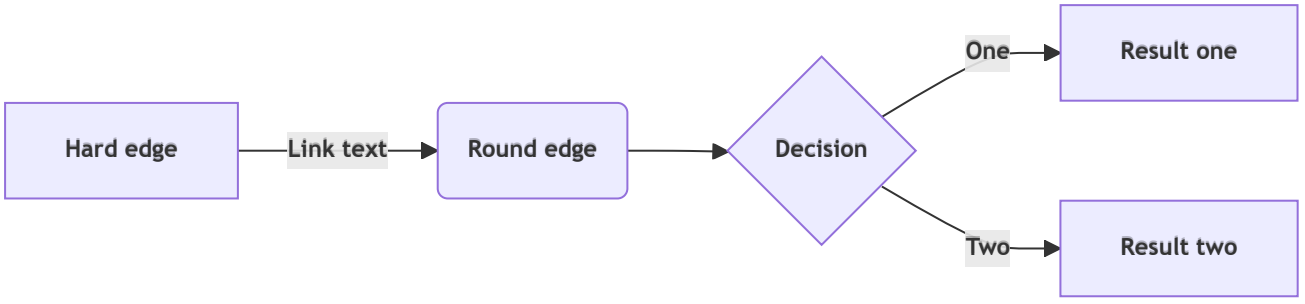
以下是这十九种图表绘制的语法细节，希望这些更新能更好的满足各位的日常使用需要

架构图

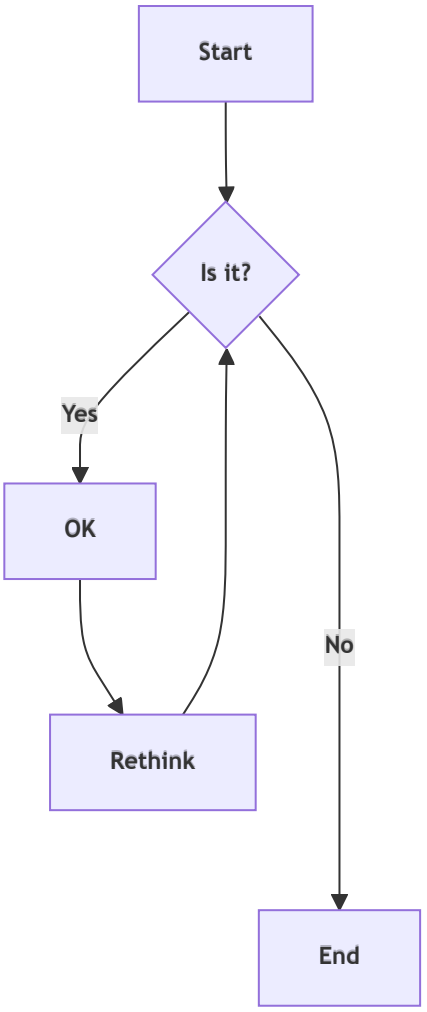
流程图

流程图由节点（几何形状）和边（箭头或线条）组成。Mermaid 代码定义了如何创建节点和边，并适应不同的箭头类型、多方向箭头以及与子图之间的任何链接。
语法参考

```mermaid
flowchart LR
    A[Hard edge] -- Link text --> B(Round edge)
    B --> C{Decision}
    C -- One --> D[Result one]
    C -- Two --> E[Result two]
```



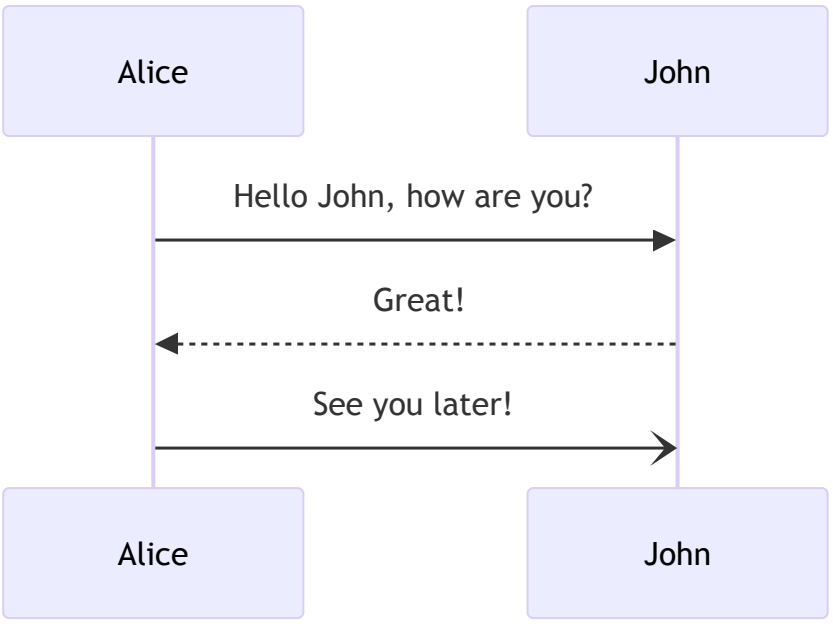

```mermaid
flowchart TD
    A[Start] --> B{Is it?}
    B -- Yes --> C[OK]
    C --> D[Rethink]
    D --> B
    B -- No ----> E[End]
```



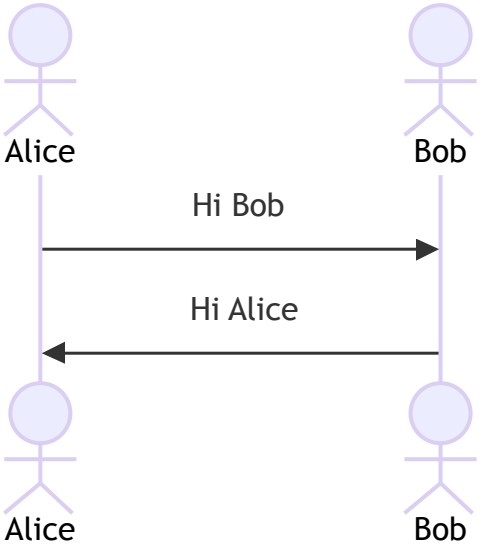
序列图

序列图是一种交互图，显示进程如何相互操作以及按什么顺序操作。
语法参考

```
```mermaid
sequenceDiagram
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
    Alice-)John: See you later!
```
```




```mermaid
sequenceDiagram
    actor Alice
    actor Bob
    Alice->>Bob: Hi Bob
    Bob->>Alice: Hi Alice
```



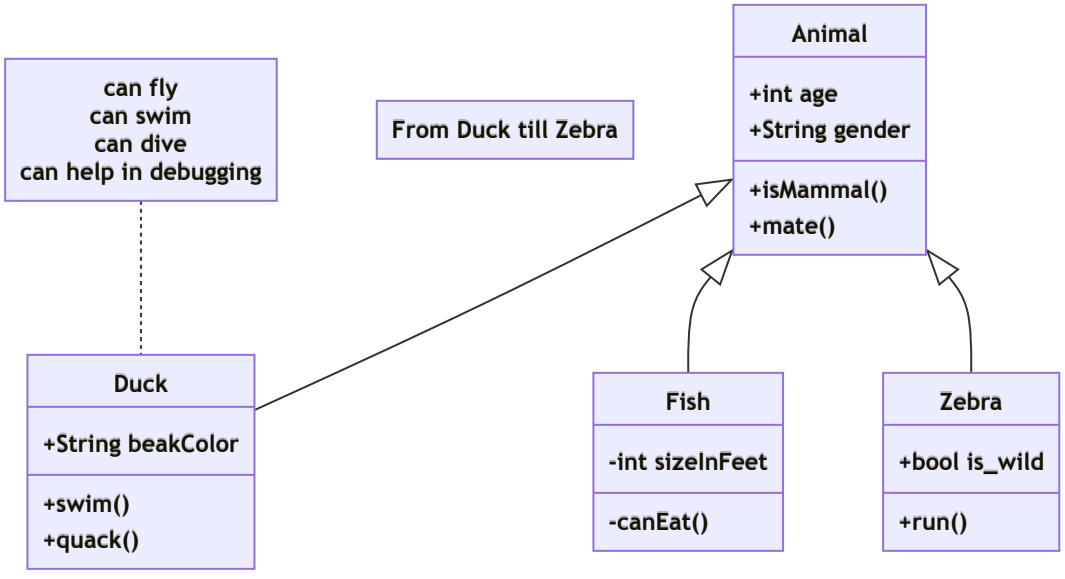
类图

在软件工程中，统一建模语言 (UML) 中的类图是一种静态结构图，它通过显示系统的类、其属性、操作（或方法）以及对象之间的关系来描述系统的结构。

语法参考

```mermaid
---
title: Animal example
---
classDiagram
    note "From Duck till Zebra"
    Animal <|-- Duck
    note for Duck "can fly\ncan swim\ncan dive\ncan help in debugging"
    Animal <|-- Fish
    Animal <|-- Zebra
    Animal : +int age
    Animal : +String gender
    Animal: +isMammal()
    Animal: +mate()
    class Duck{
        +String beakColor
        +swim()
        +quack()
    }
    class Fish{
        -int sizeInFeet
        -canEat()
    }
    class Zebra{
        +bool is_wild
        +run()
    }
```

Animal example

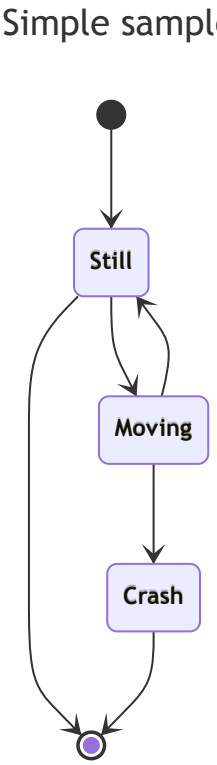


状态图

状态图是计算机科学和相关领域中用来描述系统行为的一种图表。状态图要求所描述的系统由有限数量的状态组成；有时确实如此，而有时这是一种合理的抽象。
语法参考

```
```mermaid
---
title: Simple sample
---
stateDiagram-v2
    [*] --> Still
    Still --> [*]

    Still --> Moving
    Moving --> Still
    Moving --> Crash
    Crash --> [*]
```
```

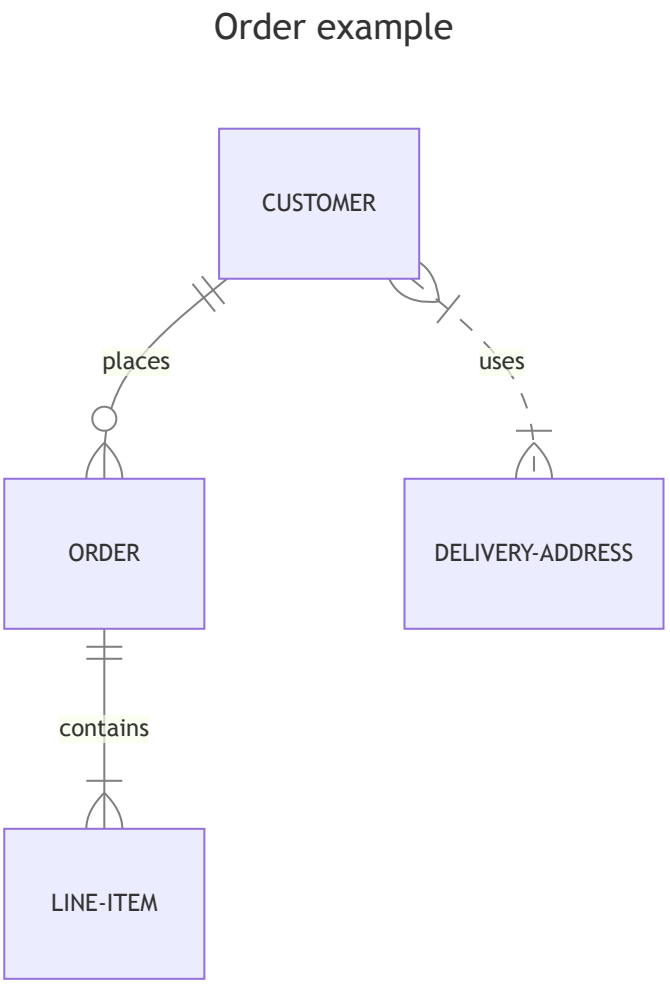


实体关系图

基本ER关系模型由实体类型（对相关事物进行分类）组成，并指定实体（这些实体类型的实例）描述之间可能存在的关系

语法参考

```mermaid
---
title: Order example
---
erDiagram
    CUSTOMER ||--o{ ORDER : places
    ORDER ||--|{ LINE-ITEM : contains
    CUSTOMER }|..|{ DELIVERY-ADDRESS : uses
```

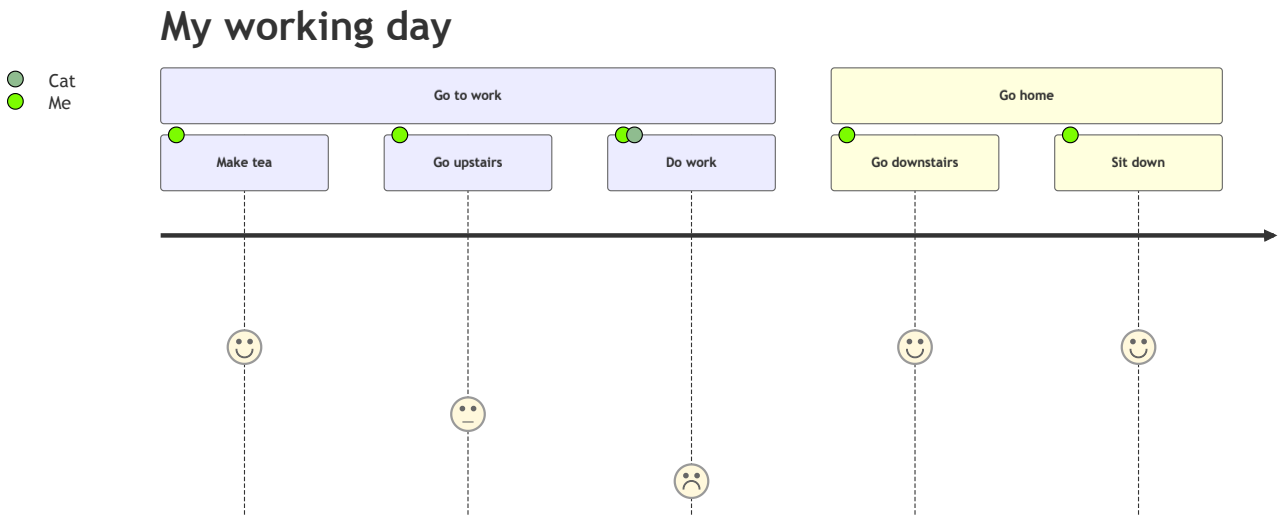


用户旅程图

用户旅程详细描述了不同用户在系统、应用程序或网站中完成特定任务的具体步骤。此技术显示当前（原样）用户工作流程，并揭示未来工作流程的改进领域。

语法参考

```mermaid
journey
    title My working day
    section Go to work
      Make tea: 5: Me
      Go upstairs: 3: Me
      Do work: 1: Me, Cat
    section Go home
      Go downstairs: 5: Me
      Sit down: 5: Me
```



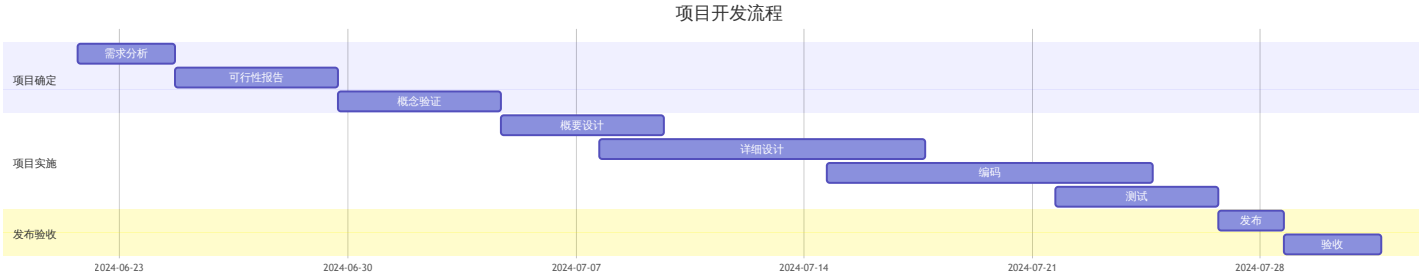
甘特图

甘特图是一种条形图，由 Karol Adamiecki 于 1896 年首次开发，Henry Gantt 于 20 世纪 10 年代独立开发，用于说明项目进度和完成任何一个项目所需的时间。甘特图显示了项目终端元素

和摘要元素的开始日期和结束日期之间的天数。

语法参考

```mermaid
gantt
    title 项目开发流程
    section 项目确定
        需求分析          :a1, 2024-06-22, 3d
        可行性报告        :after a1, 5d
        概念验证          : 5d
    section 项目实施
        概要设计         :2024-07-05  , 5d
        详细设计         :2024-07-08, 10d
        编码             :2024-07-15, 10d
        测试             :2024-07-22, 5d
    section 发布验收
        发布: 2d
        验收: 3d
```

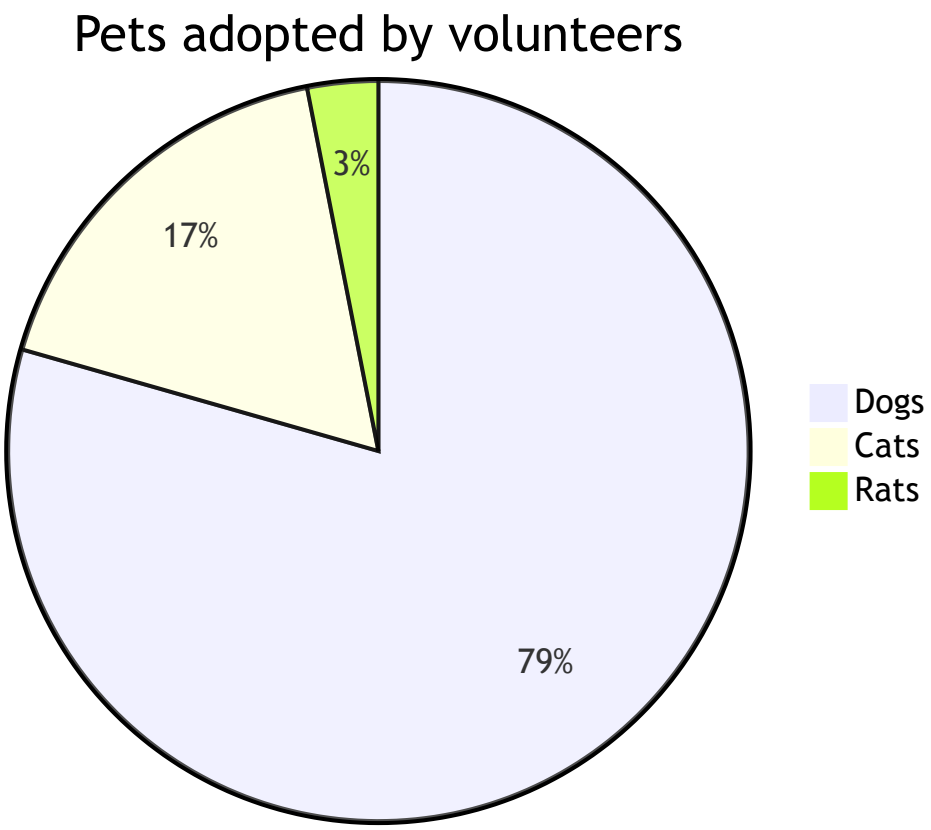


饼图

饼图（或圆形图）是一种圆形统计图形，被分成多个部分来表示数字比例。在饼图中，每个切片的弧长（以及其中心角和面积）与其所代表的数量成正比。虽然它因与切成薄片的饼相似而得名，但它的呈现方式却各不相同。最早的饼图通常归功于威廉·普莱费尔 1801 年的《统计简表》

语法参考

```mermaid
pie title Pets adopted by volunteers
    "Dogs" : 386
    "Cats" : 85
    "Rats" : 15
```



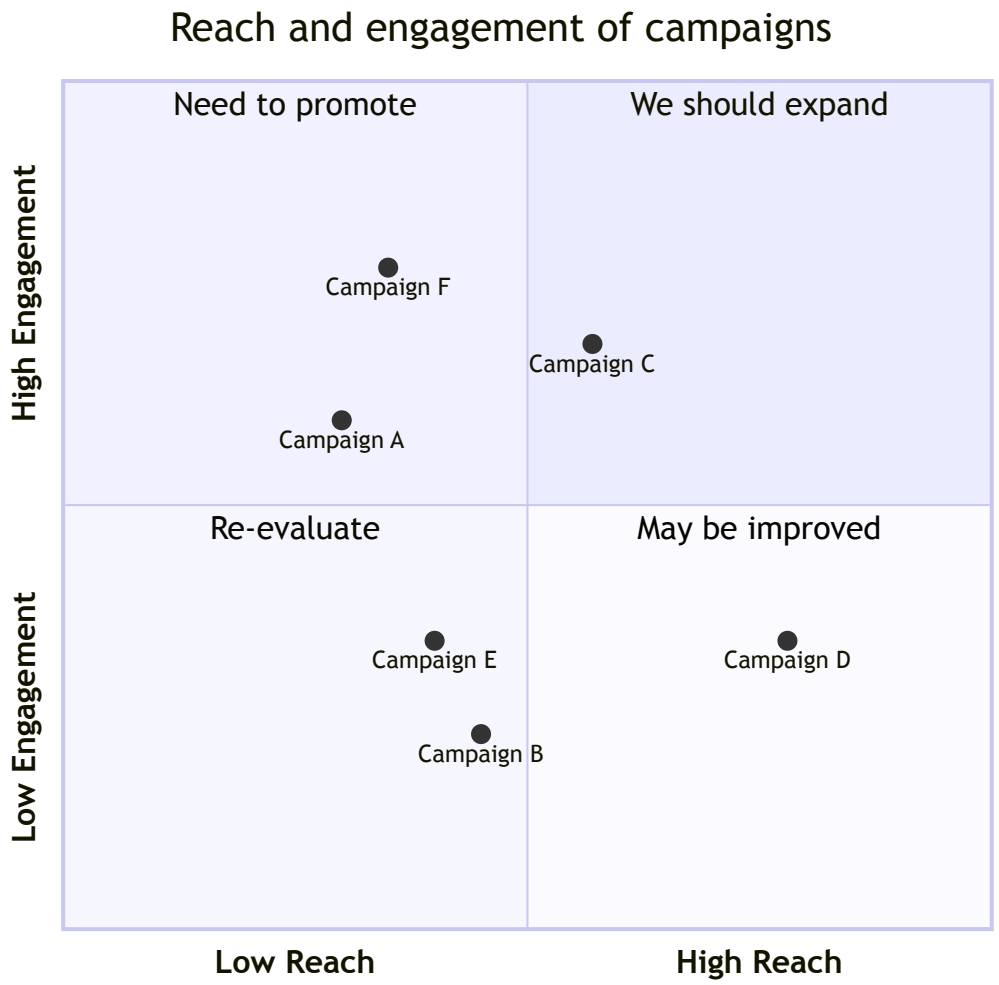
象限图

象限图是将数据划分为四个象限的直观表示。它用于在二维网格上绘制数据点，一个变量表示在 x 轴上，另一个变量表示在 y 轴上。根据一组特定于所分析数据的标准将图表划分为四个相等的部分来确定象限。象限图通常用于识别数据中的模式和趋势，并根据图表中数据点的位置确定操作的优先级。它们通常用于商业、营销和风险管理等领域。

语法参考

```mermaid
quadrantChart
    title Reach and engagement of campaigns
    x-axis Low Reach --> High Reach
    y-axis Low Engagement --> High Engagement
    quadrant-1 We should expand
    quadrant-2 Need to promote
    quadrant-3 Re-evaluate
    quadrant-4 May be improved
    Campaign A: [0.3, 0.6]
    Campaign B: [0.45, 0.23]
    Campaign C: [0.57, 0.69]
    Campaign D: [0.78, 0.34]
    Campaign E: [0.40, 0.34]
```

```
    Campaign F: [0.35, 0.78]
```



需求图

求图提供了需求及其相互之间以及其他已记录元素之间的联系的可视化。
语法参考

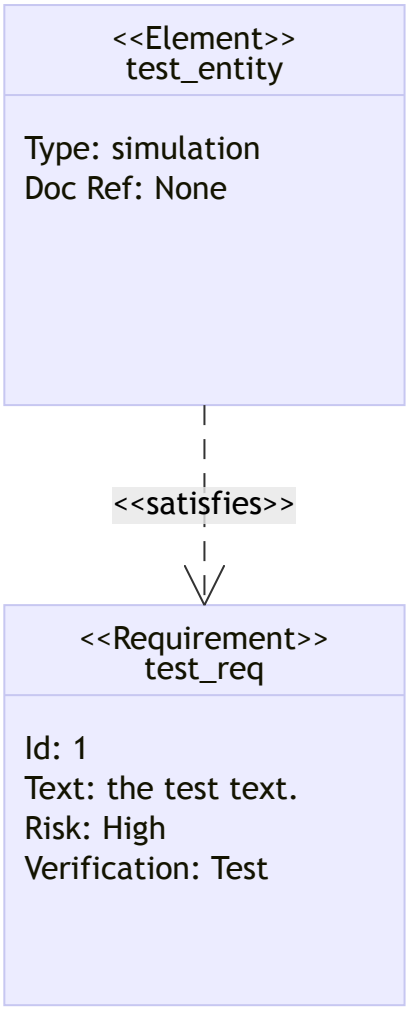
```mermaid
    requirementDiagram

    requirement test_req {
    id: 1
    text: the test text.
    risk: high
    verifymethod: test
    }

    element test_entity {
    type: simulation
```

```
    }

    test_entity - satisfies -> test_req
```



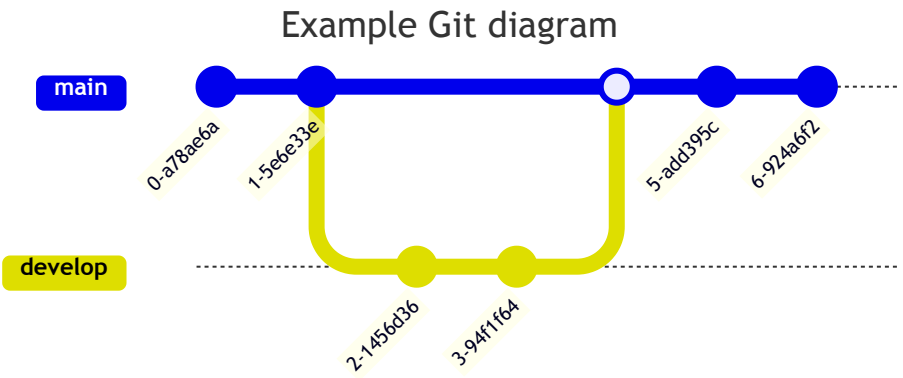
Git图

Git Graph 是各个分支上 git 提交和 git 操作（命令）的图形表示。这种图表对于开发人员和 DevOps 团队分享他们的 Git 分支策略特别有用。例如，它使可视化 git 流程的工作方式变得更加容易。

语法参考

```
```mermaid
---
title: Example Git diagram
---
gitGraph
   commit
   commit
```
```

```
   branch develop
   checkout develop
   commit
   commit
   checkout main
   merge develop
   commit
   commit
```



C4图

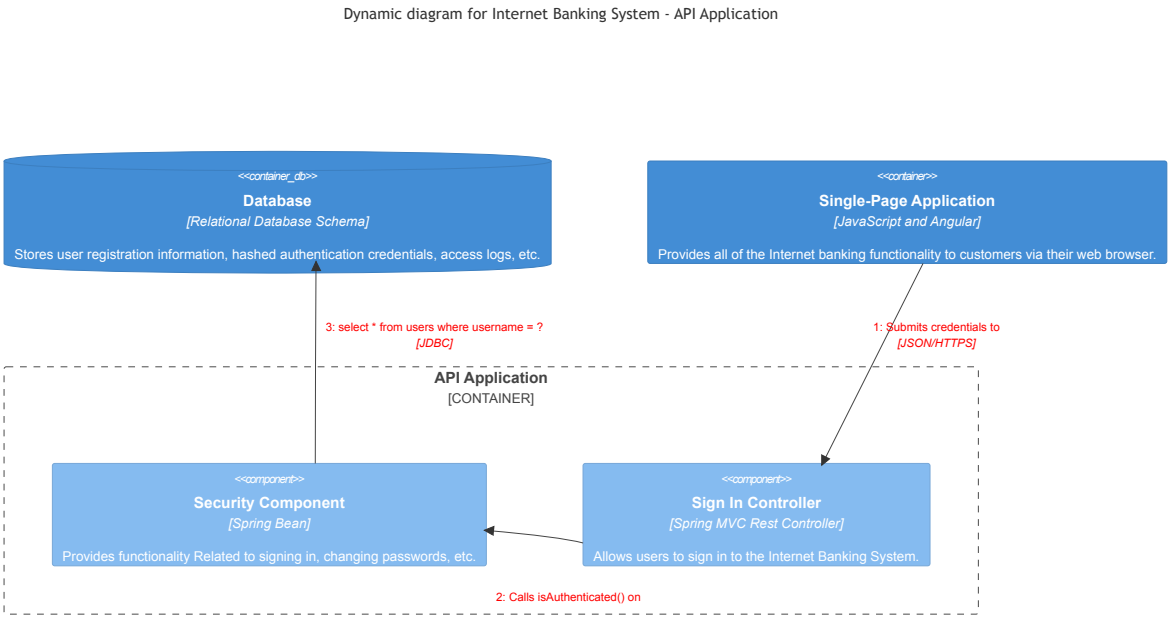
C4 图表：目前这是一个实验图表。语法和属性可能会在未来版本中发生变化。语法稳定后将提供适当的文档。
语法参考

```mermaid
    C4Dynamic
    title Dynamic diagram for Internet Banking System - API Application

    ContainerDb(c4, "Database", "Relational Database Schema", "Stores user registr
ation information, hashed authentication credentials, access logs, etc.")
    Container(c1, "Single-Page Application", "JavaScript and Angular", "Provides a
ll of the Internet banking functionality to customers via their web browser.")
    Container_Boundary(b, "API Application") {
      Component(c3, "Security Component", "Spring Bean", "Provides functionality R
elated to signing in, changing passwords, etc.")
      Component(c2, "Sign In Controller", "Spring MVC Rest Controller", "Allows us
ers to sign in to the Internet Banking System.")
    }
    Rel(c1, c2, "Submits credentials to", "JSON/HTTPS")
    Rel(c2, c3, "Calls isAuthenticated() on")
    Rel(c3, c4, "select * from users where username = ?", "JDBC")

    UpdateRelStyle(c1, c2, $textColor="red", $offsetY="-40")
```

```
    UpdateRelStyle(c2, c3, $textColor="red", $offsetX="-40", $offsetY="60")
        UpdateRelStyle(c3, c4, $textColor="red", $offsetY="-40", $offsetX="10")
```

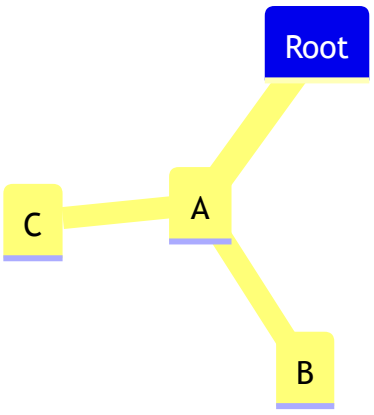


思维导图

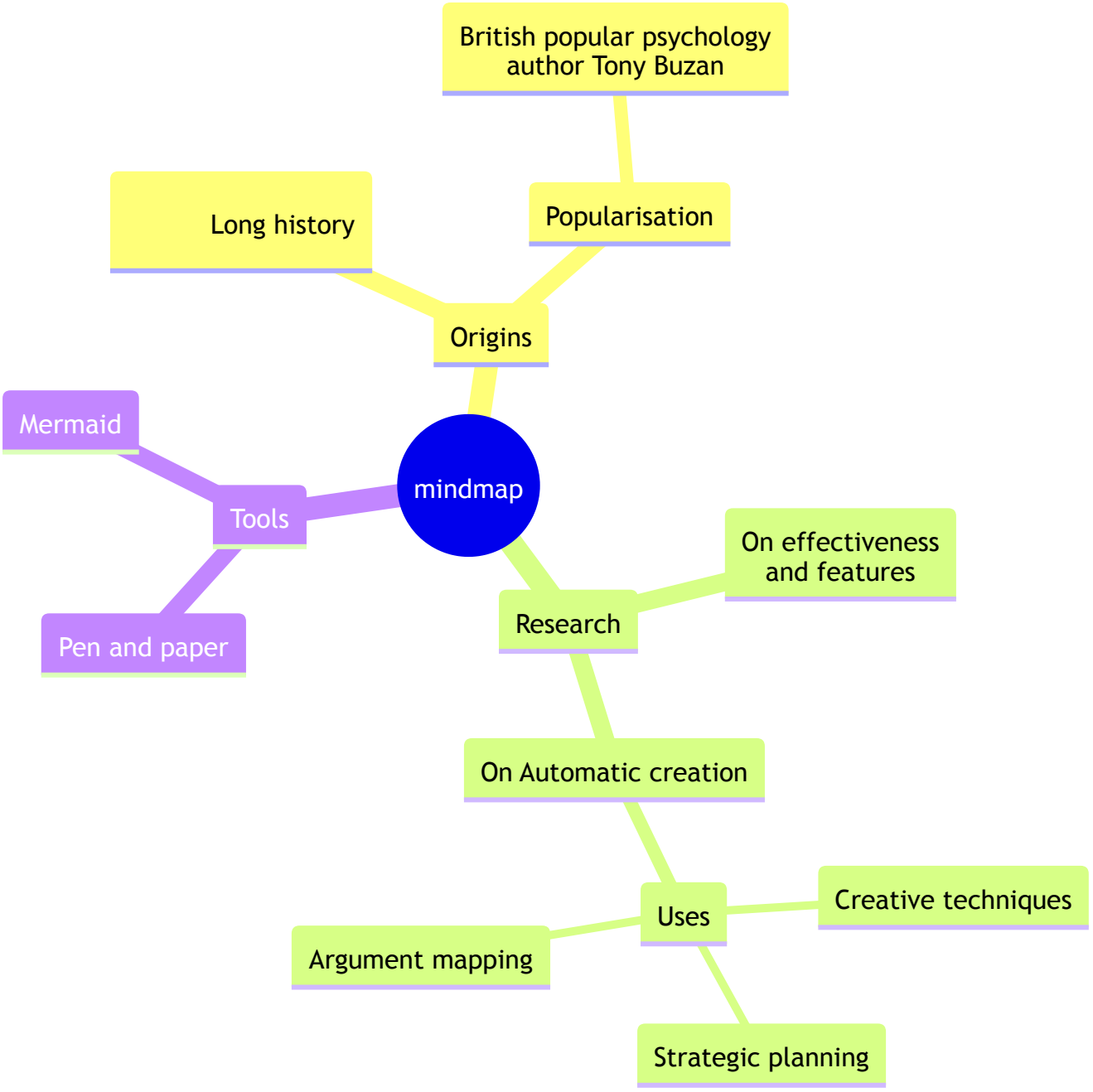
思维导图是一种将信息以层次结构直观地组织起来的图表，显示整体各部分之间的关系。它通常围绕一个概念创建，在空白页的中心绘制为图像，并在其中添加相关的想法表示，例如图像、单词和单词的部分。主要思想直接与中心概念相关，其他思想从这些主要思想中分支出来。

语法参考

```
```mermaid
mindmap
Root
    A
      B
      C
```
```




```mermaid
mindmap
  root((mindmap))
    Origins
      Long history
      ::icon(fa fa-book)
      Popularisation
        British popular psychology author Tony Buzan
    Research
      On effectiveness<br/>and features
      On Automatic creation
        Uses
            Creative techniques
            Strategic planning
            Argument mapping
    Tools
      Pen and paper
      Mermaid
```



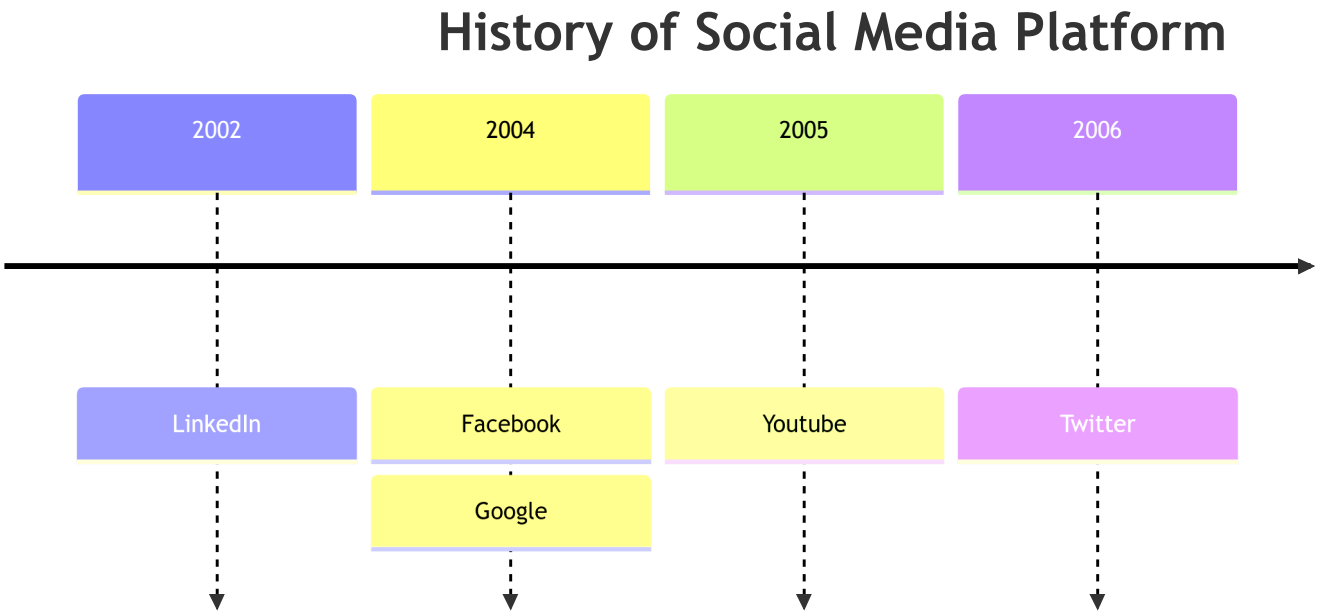
时间线图

时间线是一种图表，用于说明事件、日期或时间段的年表。它通常以图形方式呈现以指示时间的流逝，并且通常按时间顺序排列。基本时间线按时间顺序显示事件列表，通常使用日期作为标记。时间线还可用于显示事件之间的关系，例如一个人一生中事件之间的关系

语法参考

```
```mermaid
timeline
    title History of Social Media Platform
    2002 : LinkedIn
    2004 : Facebook
```
```

```
          : Google
    2005 : Youtube
    2006 : Twitter
```



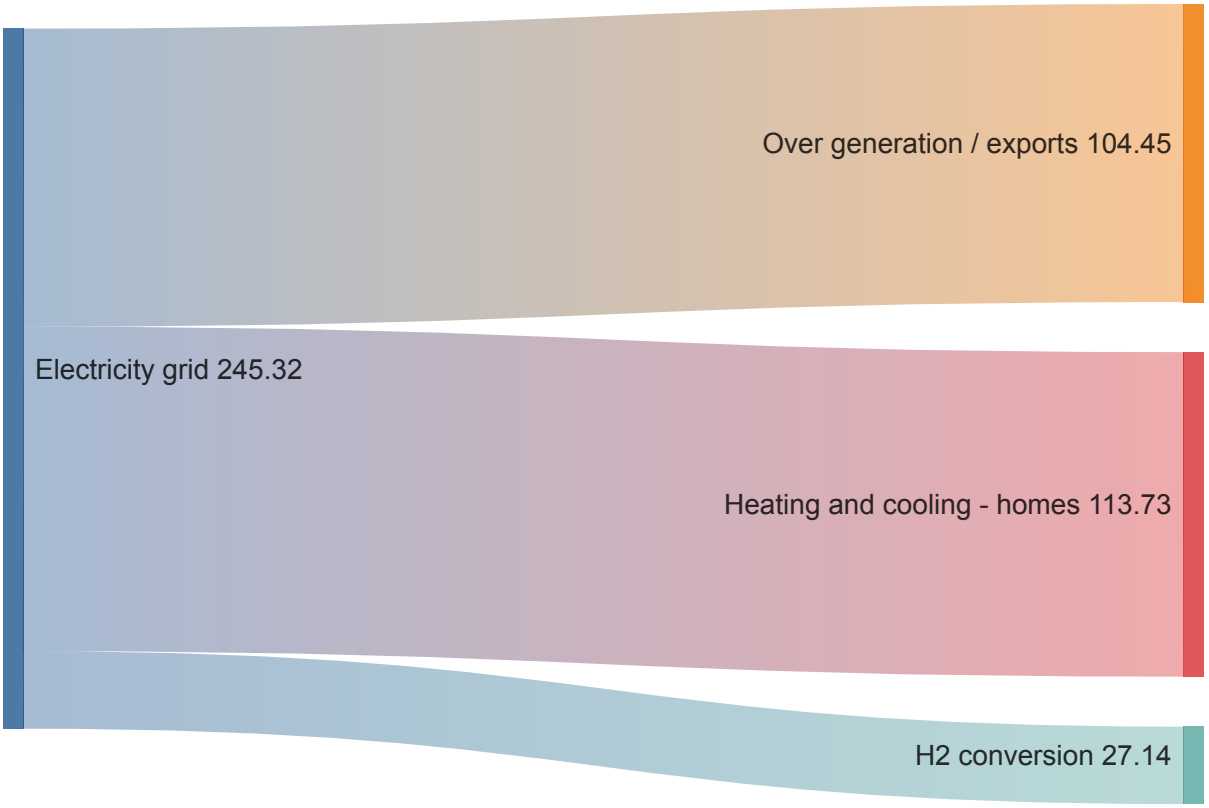
桑基图

桑基图是一种可视化方法，用于描绘从一组值到另一组值的流动。被连接的事物称为节点，连接称为链接。

语法参考

```mermaid
sankey-beta

Electricity grid,Over generation / exports,104.453
Electricity grid,Heating and cooling - homes,113.726
Electricity grid,H2 conversion,27.14
```

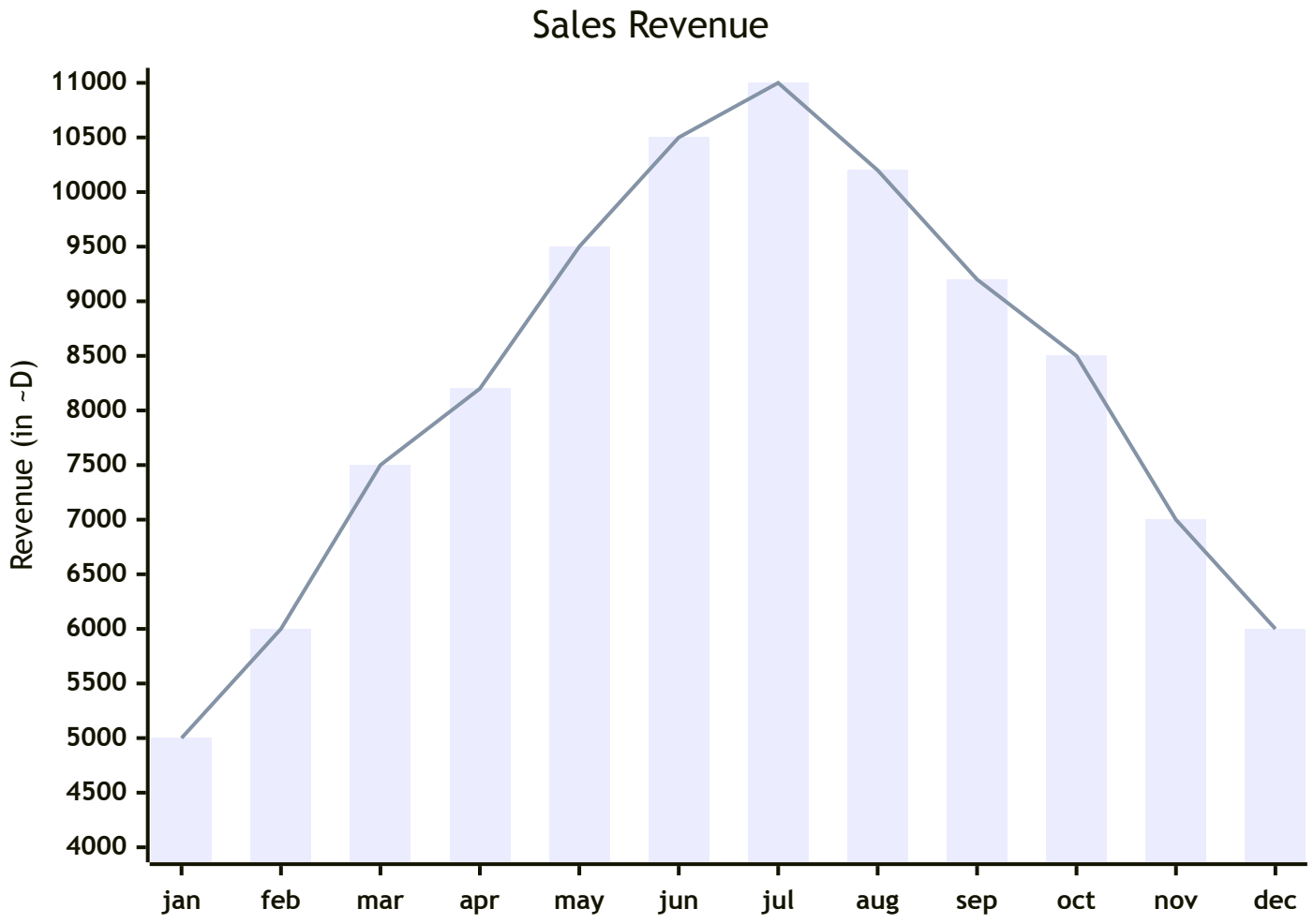


XY图表

XY 图表是一个综合图表模块，它包含各种类型的图表，这些图表使用 x 轴和 y 轴来表示数据。目前，它包括两种基本图表类型：条形图和折线图。这些图表旨在直观地显示和分析涉及两个数值变量的数据。

语法参考

```
```mermaid
xychart-beta
    title "Sales Revenue"
    x-axis [jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec]
    y-axis "Revenue (in $)" 4000 --> 11000
    bar [5000, 6000, 7500, 8200, 9500, 10500, 11000, 10200, 9200, 8500, 7000, 600
0]
    line [5000, 6000, 7500, 8200, 9500, 10500, 11000, 10200, 9200, 8500, 7000, 600
0]
```
```



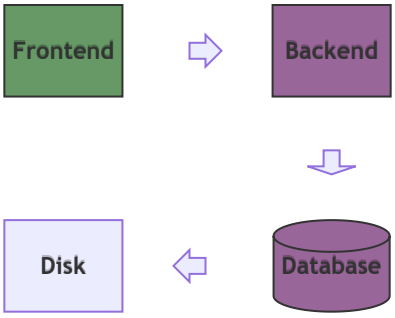
框图

框图是一种直观且有效的方法，可以直观地表示复杂的系统、流程或架构。它们由块和连接器组成，其中块表示基本组件或功能，连接器显示这些组件之间的关系或流程。这种绘图方法在工程、软件开发和流程管理等各个领域都是必不可少的。

语法参考

```mermaid
block-beta
  columns 3
  Frontend blockArrowId6<[" "]>(right) Backend
  space:2 down<[" "]>(down)
  Disk left<[" "]>(left) Database[("Database")]

  classDef front fill:#696,stroke:#333;
  classDef back fill:#969,stroke:#333;
  class Frontend front
  class Backend,Database back
```

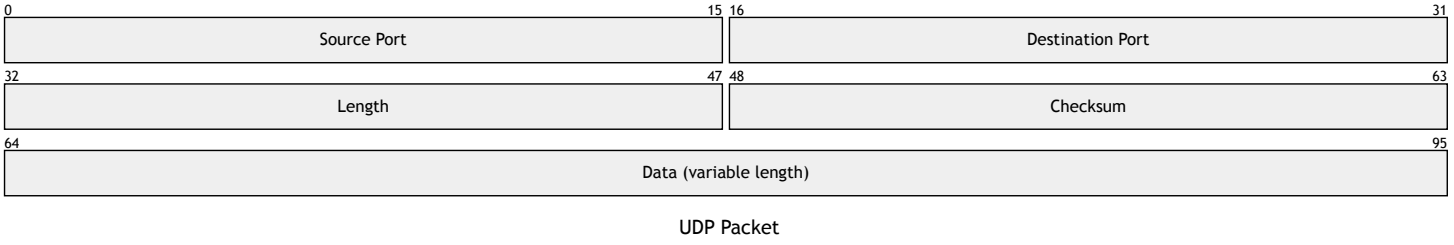


包裹图

数据包图是一种直观的表示方法，用于说明网络数据包的结构和内容。网络数据包是通过网络传输数据的基本单位。

语法参考

```
```mermaid
packet-beta
title UDP Packet
0-15: "Source Port"
16-31: "Destination Port"
32-47: "Length"
48-63: "Checksum"
64-95: "Data (variable length)"
```
```



UDP Packet

架构图

构图用于显示云或 CI/CD 部署中常见的服务和资源之间的关系。在架构图中，服务（节点）通过边连接。相关服务可以放在组中，以更好地说明它们的组织方式。

语法参考

```mermaid
architecture-beta
    group api(cloud)[API]

    service db(database)[Database] in api
    service disk1(disk)[Storage] in api
    service disk2(disk)[Storage] in api
    service server(server)[Server] in api

    db:L -- R:server
    disk1:T -- B:server
    disk2:T -- B:db
```