COMPANY NAME

//slogan //tagline //website

Awesome Mermaid Diagrams

Author: **Kuan Cheang**

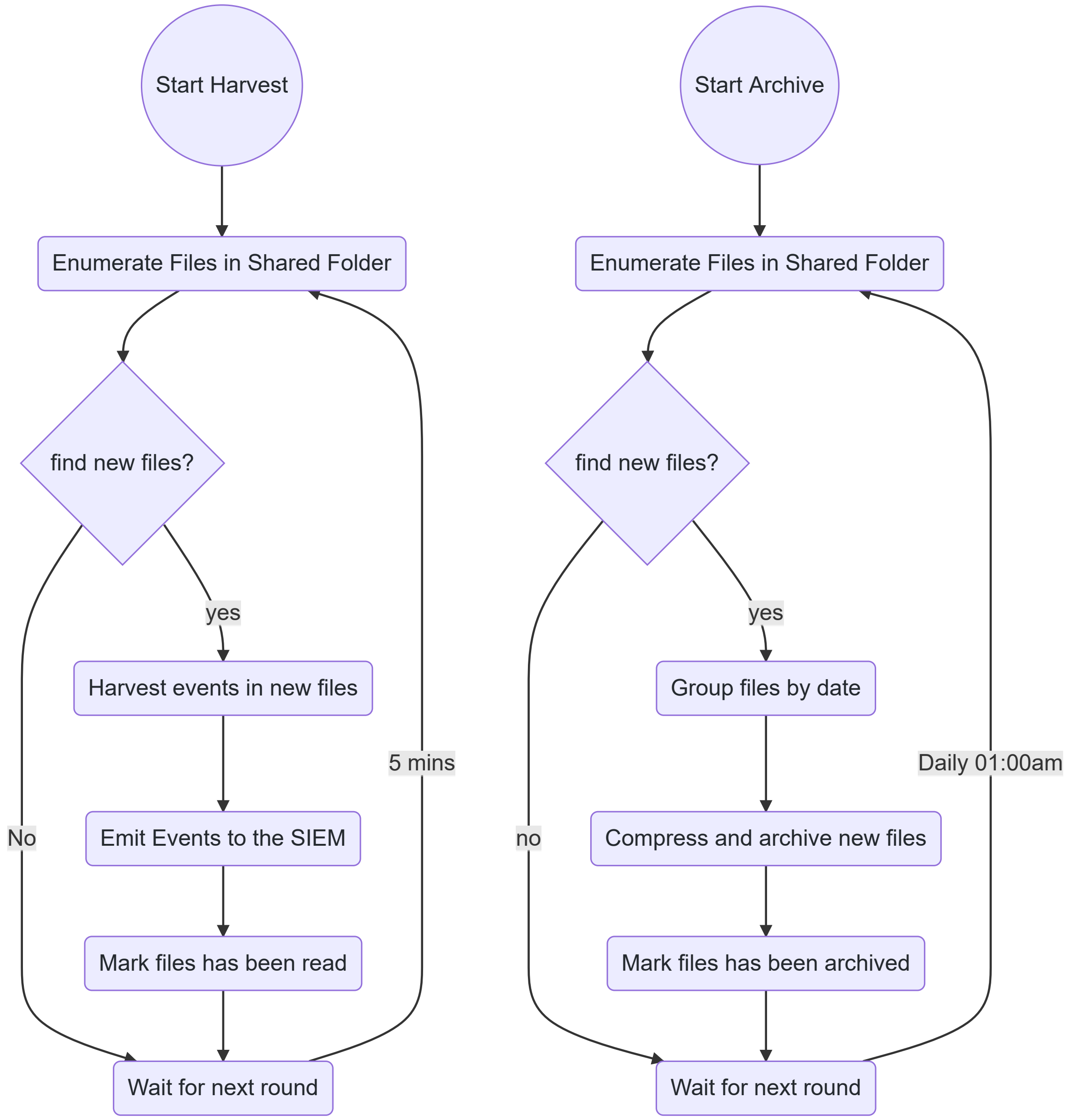
Version: **v1.0**

Date: **18/8/2021**

Generating Mermaid Diagrams

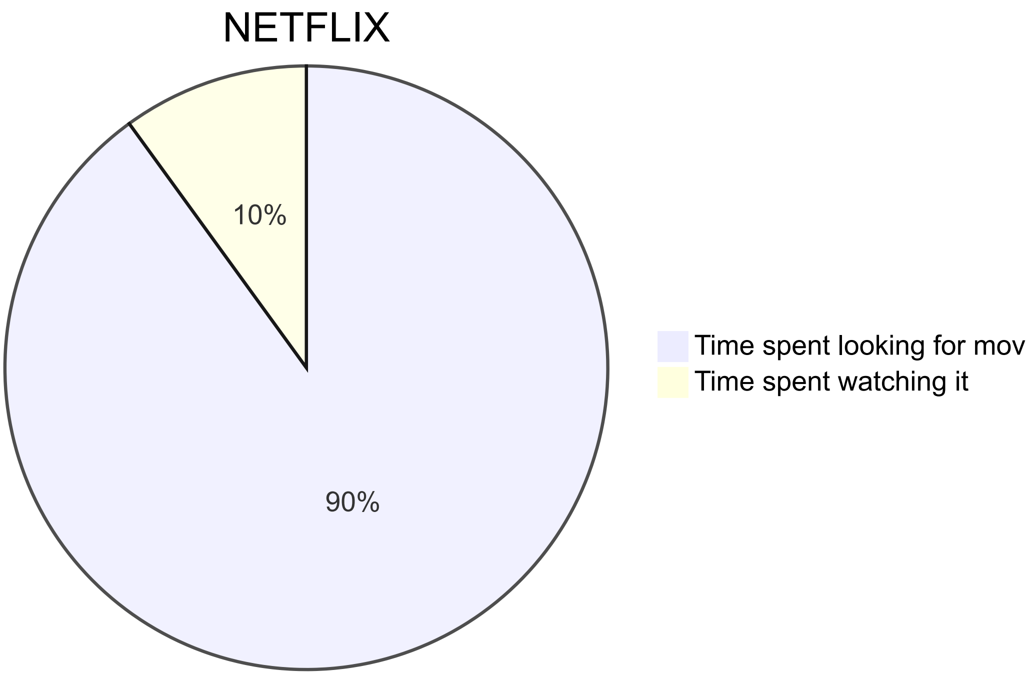
Converting code block which has a keyword `mermaid` to a diagram with `svg` format.

Examples



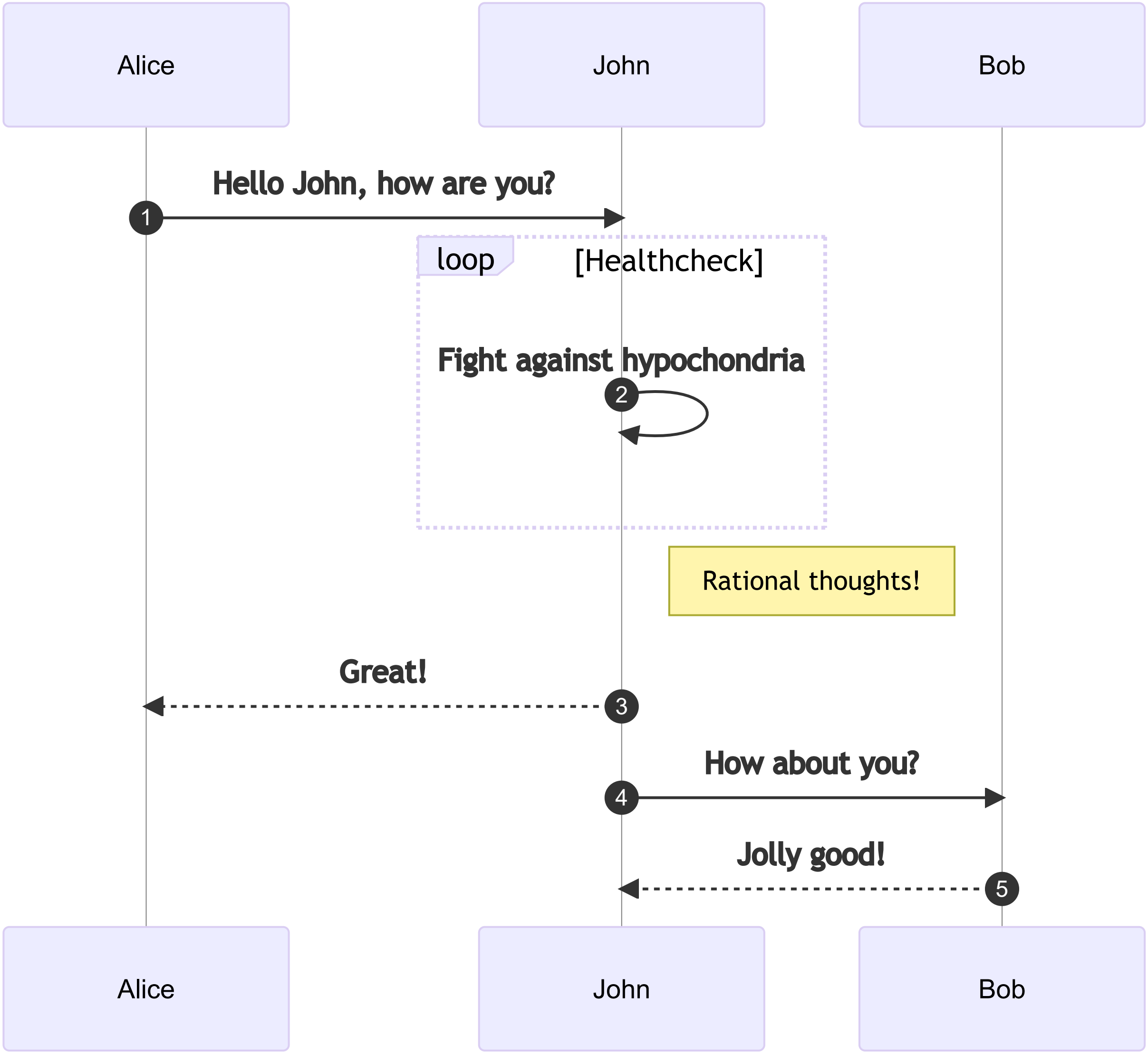
Basic Pie Chart

~~~
```mermaid
pie title NETFLIX
         "Time spent looking for movie" : 90
         "Time spent watching it" : 10
```
~~~



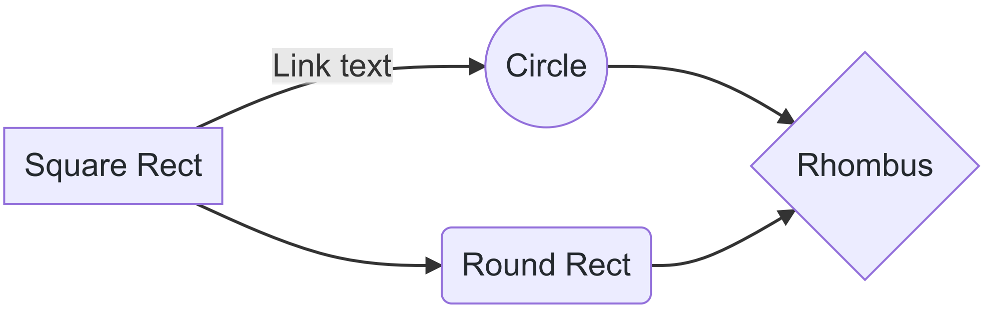
Basic sequence diagram

```
&#x60;&#x60;&#x60;mermaid
sequenceDiagram
    autonumber
    Alice->>John: Hello John, how are you?
    loop Healthcheck
        John->>John: Fight against hypochondria
    end
    Note right of John: Rational thoughts!
    John-->>Alice: Great!
    John->>Bob: How about you?
    Bob-->>John: Jolly good!
&#x60;&#x60;&#x60;
```



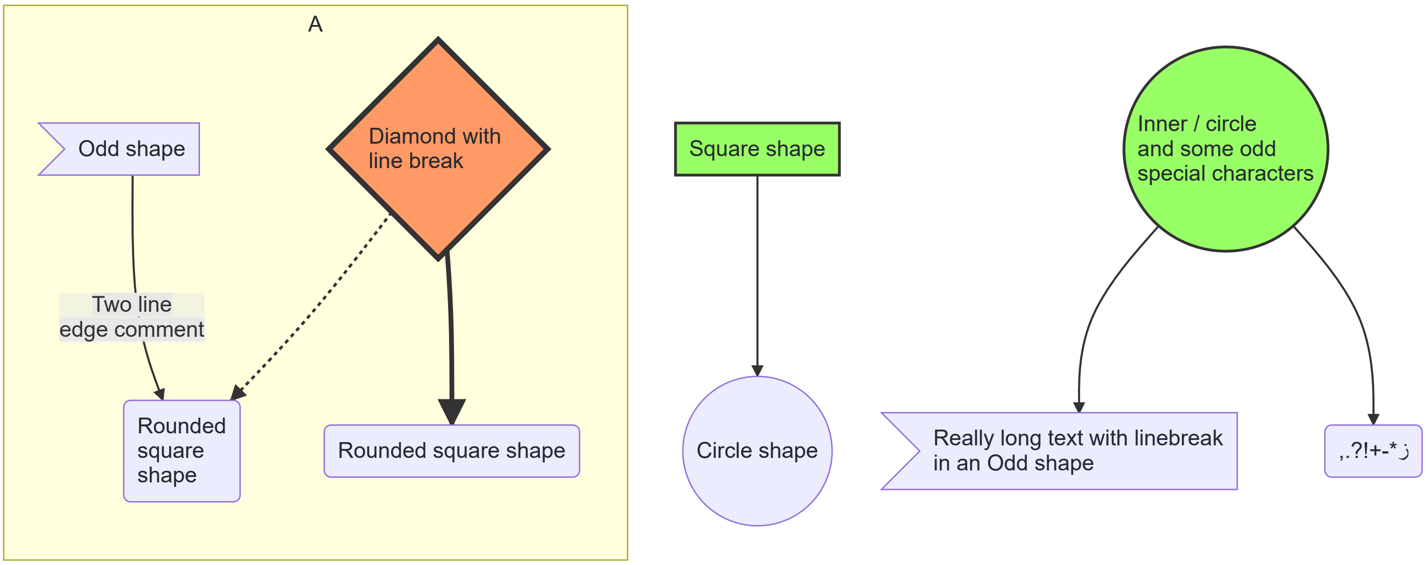
Basic flowchart

~~~
```mermaid
graph LR
    A[Square Rect] -- Link text --> B((Circle))
    A --> C(Round Rect)
    B --> D{Rhombus}
    C --> D
```
~~~



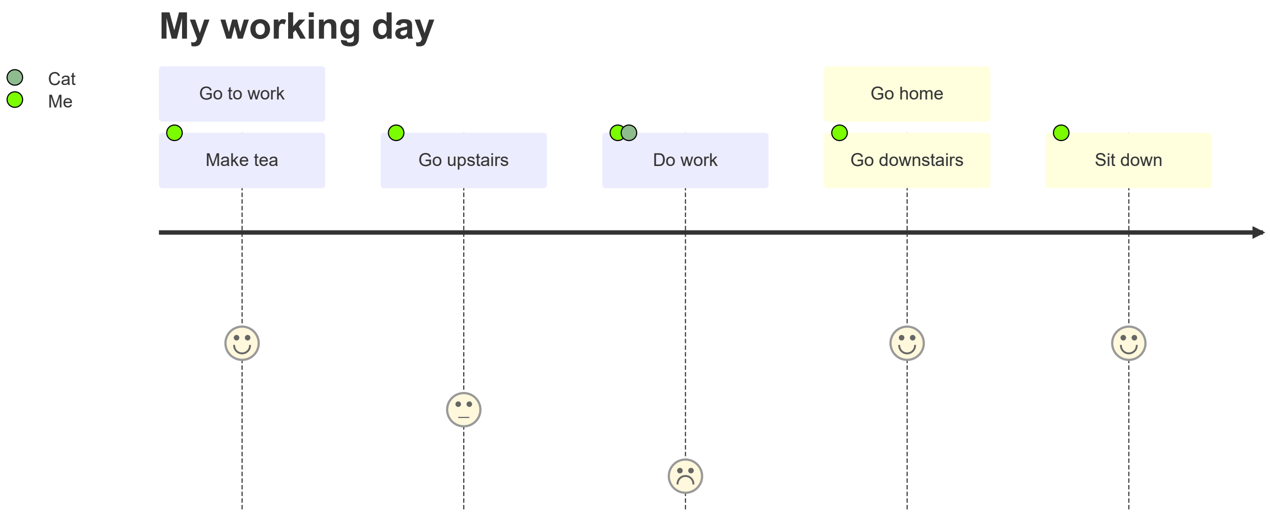
Larger flowchart with some styling

~~~
```mermaid
graph TB
    sq[Square shape] --> ci((Circle shape))

    subgraph A
        od>Odd shape]-- Two line
edge comment --> ro
        di{Diamond with
 line break} -.-> ro(Rounded
square
shape)
        di==>ro2(Rounded square shape)
    end

    %% Notice that no text in shape are added here instead that is appended further down
    e --> od3>Really long text with linebreak
in an Odd shape]

    %% Comments after double percent signs
    e((Inner / circle
and some odd
special characters)) --> f(,.?!+-*ز)

     classDef green fill:#9f6,stroke:#333,stroke-width:2px;
     classDef orange fill:#f96,stroke:#333,stroke-width:4px;
     class sq,e green
     class di orange
```
~~~



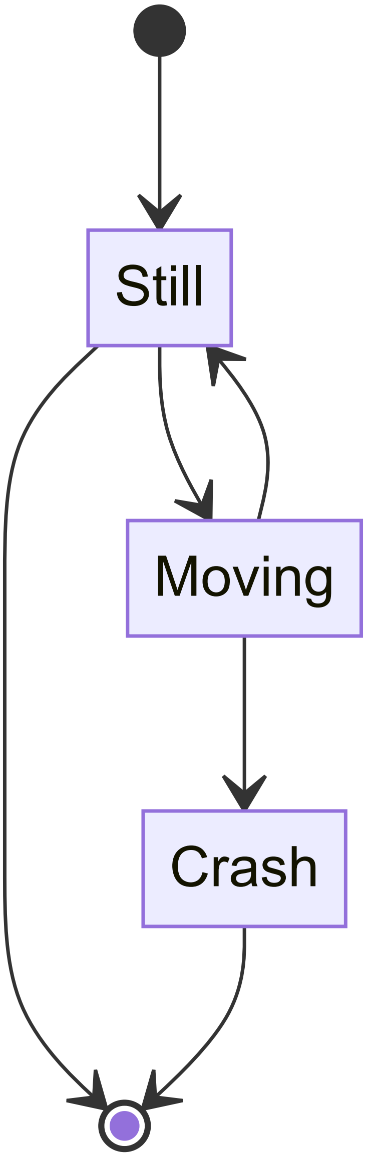
User Journey Diagram

~~~
```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
```
~~~



State diagrams

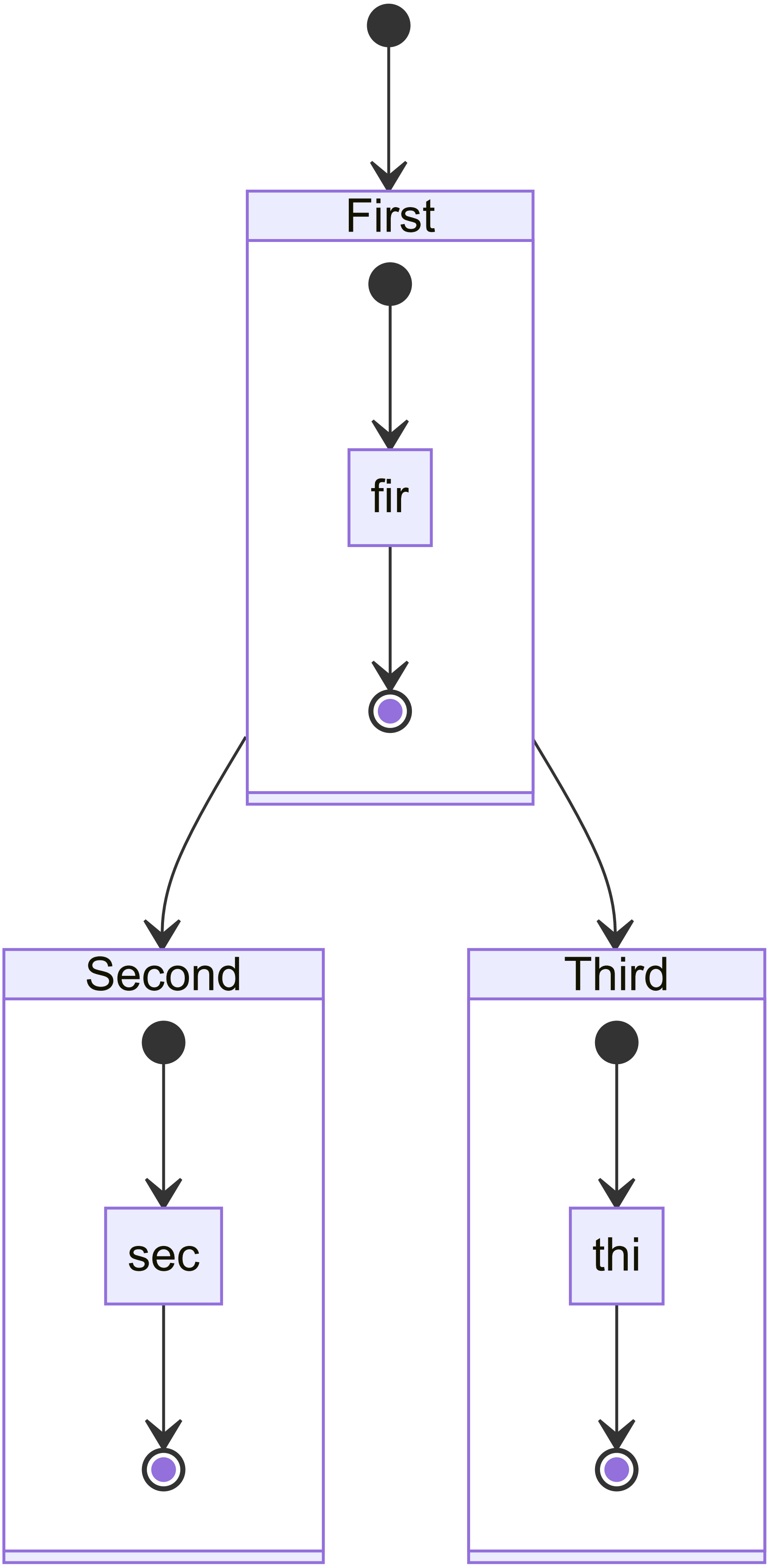
~~~
```mermaid
stateDiagram-v2
    [*] --> Still
    Still --> [*]

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



Composite state diagram

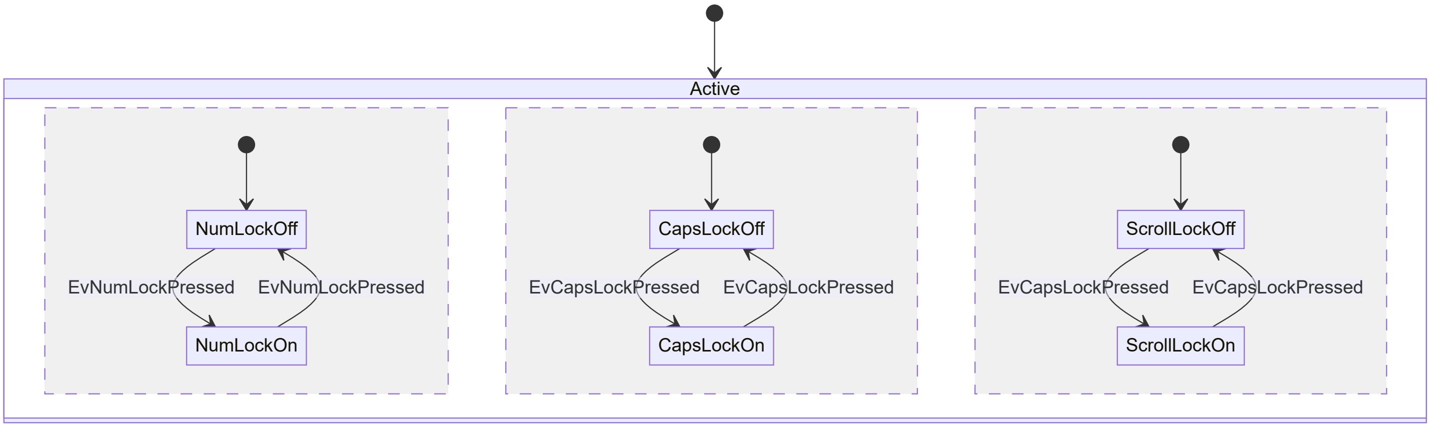
~~~
```mermaid
stateDiagram-v2
    [*] --> First
    First --> Second
    First --> Third

    state First {
        [*] --> fir
        fir --> [*]
    }
    state Second {
        [*] --> sec
        sec --> [*]
    }
    state Third {
~~~

~~~
        [*] --> thi
        thi --> [*]
    }
```
~~~


First
fir
Second
sec
Third
thi

Concurrency

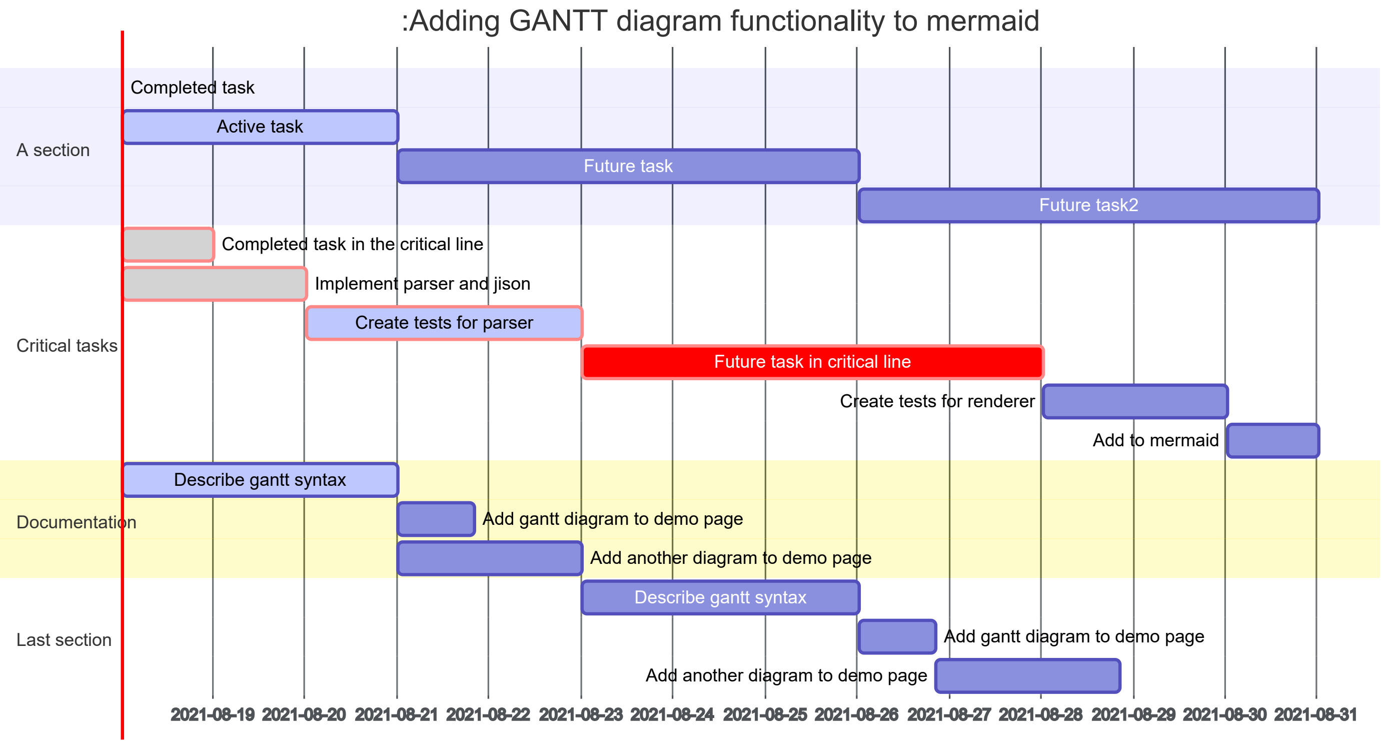
~~~
```mermaid
stateDiagram-v2
    [*] --> Active

    state Active {
        [*] --> NumLockOff
        NumLockOff --> NumLockOn : EvNumLockPressed
        NumLockOn --> NumLockOff : EvNumLockPressed
        --
        [*] --> CapsLockOff
        CapsLockOff --> CapsLockOn : EvCapsLockPressed
        CapsLockOn --> CapsLockOff : EvCapsLockPressed
        --
        [*] --> ScrollLockOff
        ScrollLockOff --> ScrollLockOn : EvCapsLockPressed
        ScrollLockOn --> ScrollLockOff : EvCapsLockPressed
    }
```
~~~



Gantt

~~~
```mermaid
gantt
    dateFormat                :YYYY-MM-DD
    title                     :Adding GANTT diagram functionality to mermaid
    excludes                  :excludes the named dates/days from being included i
    section A section
    Completed task            :done,    des1, 2014-01-06,2014-01-08
    Active task               :active,  des2, 2014-01-09, 3d
    Future task               :         des3, after des2, 5d
    Future task2              :         des4, after des3, 5d

    section Critical tasks
~~~

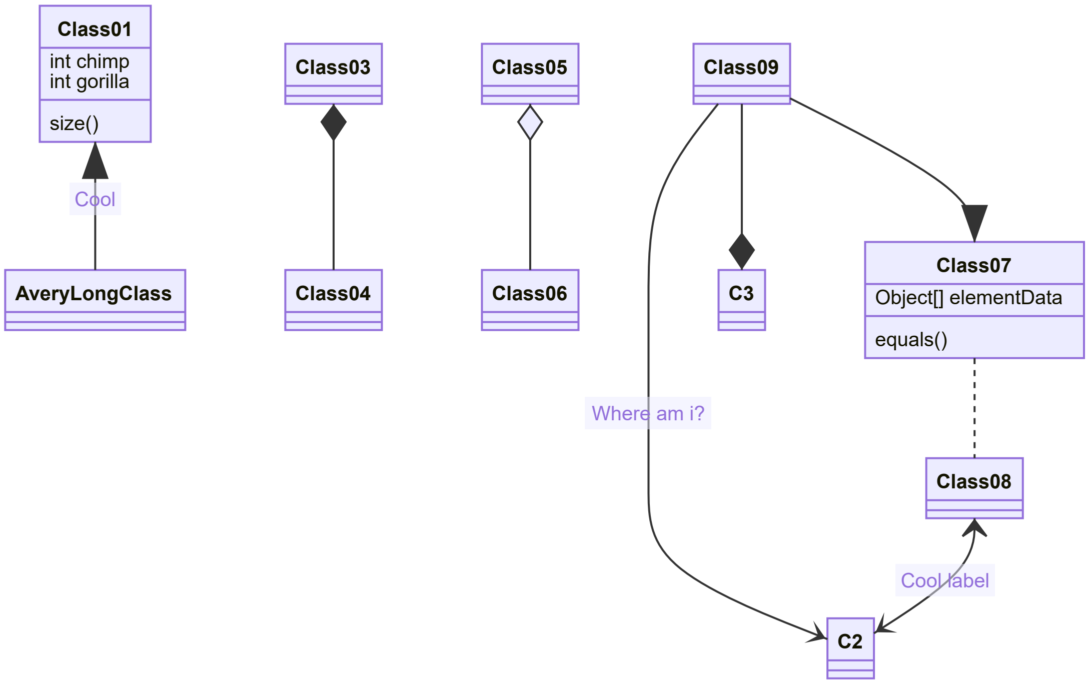
~~~
    Completed task in the critical line :crit, done, 2014-01-06,24h
    Implement parser and jison          :crit, done, after des1, 2d
    Create tests for parser             :crit, active, 3d
    Future task in critical line        :crit, 5d
    Create tests for renderer           :2d
    Add to mermaid                      :1d

    section Documentation
    Describe gantt syntax               :active, a1, after des1, 3d
    Add gantt diagram to demo page      :after a1  , 20h
    Add another diagram to demo page    :doc1, after a1  , 48h

    section Last section
    Describe gantt syntax               :after doc1, 3d
    Add gantt diagram to demo page      :20h
    Add another diagram to demo page    :48h
```
~~~



Class

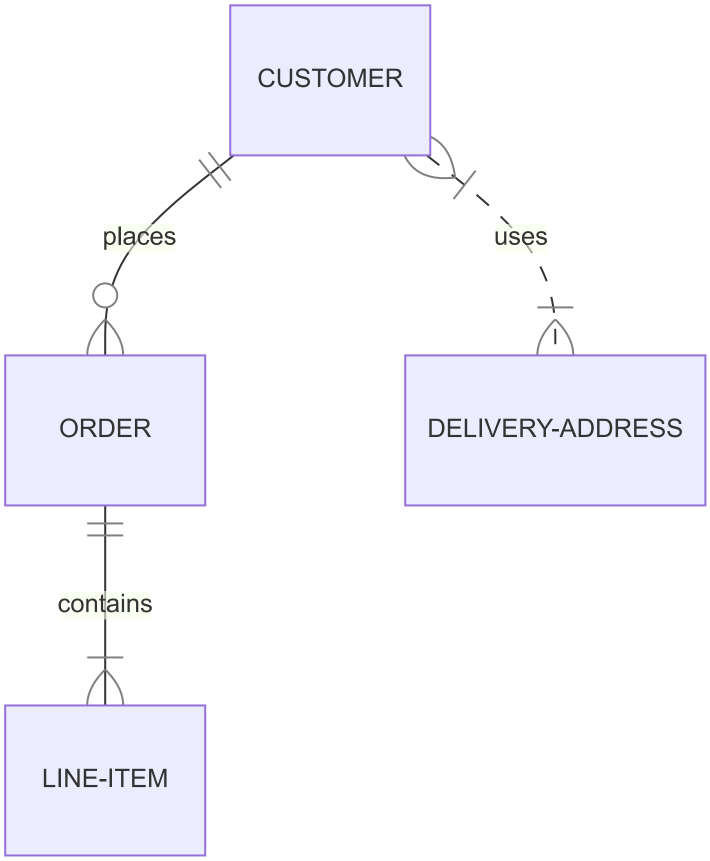
```
&#x60;&#x60;&#x60;mermaid
classDiagram
    Class01 &#x3C;|-- AveryLongClass : Cool
    Class03 *-- Class04
    Class05 o-- Class06
    Class07 .. Class08
    Class09 --&#x3E; C2 : Where am i?
```

```
    Class09 --* C3
    Class09 --|&#x3E; Class07
    Class07 : equals()
    Class07 : Object[] elementData
    Class01 : size()
    Class01 : int chimp
    Class01 : int gorilla
    Class08 &#x3C;--&#x3E; C2: Cool label
&#x60;&#x60;&#x60;
```



Git

~~~
```mermaid
gitGraph:
    options
    {
        "nodeSpacing": 150,
        "nodeRadius": 5
    }
    end
    commit
    branch newbranch
    checkout newbranch
    commit
    commit
    checkout master
    commit
    commit
~~~

```
    merge newbranch
```
```

ERDiagram

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



This document is auto generated, if you find any mistake on it. Please contact us, thank.
````