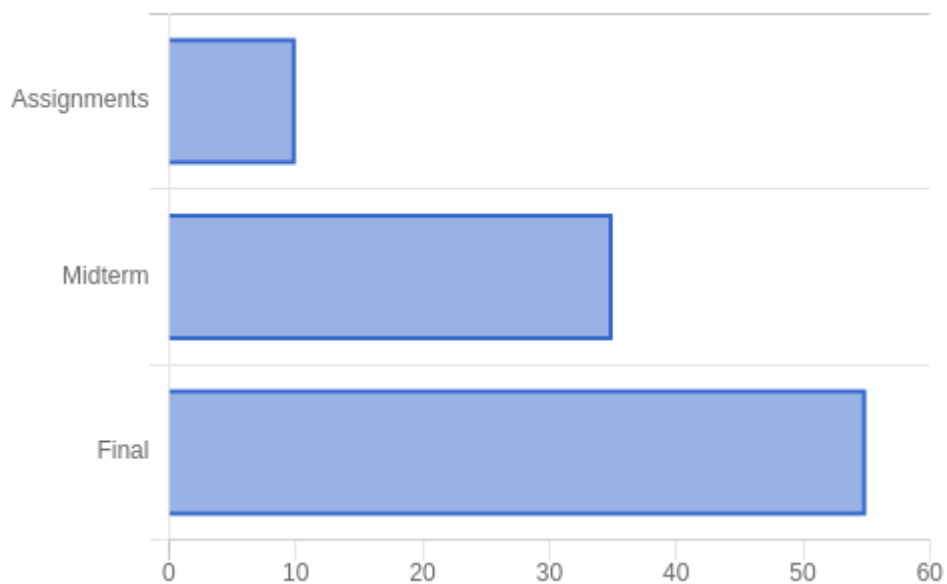


Charts

Bar chart

```
<bar-chart :data="[
  ['Assignments', 10],
  ['Midterm', 35],
  ['Final', 55]]"
/>
```

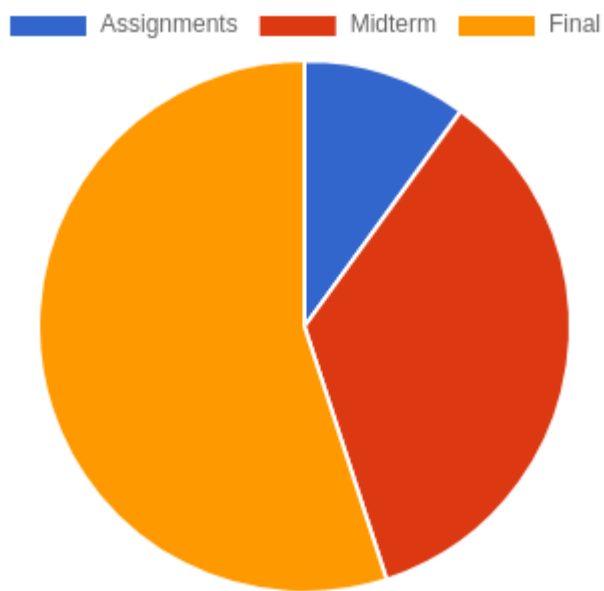
vue



Pie chart

```
<pie-chart :data="[
  ['Assignments', 10],
  ['Midterm', 35],
  ['Final', 55]]"
/>
```

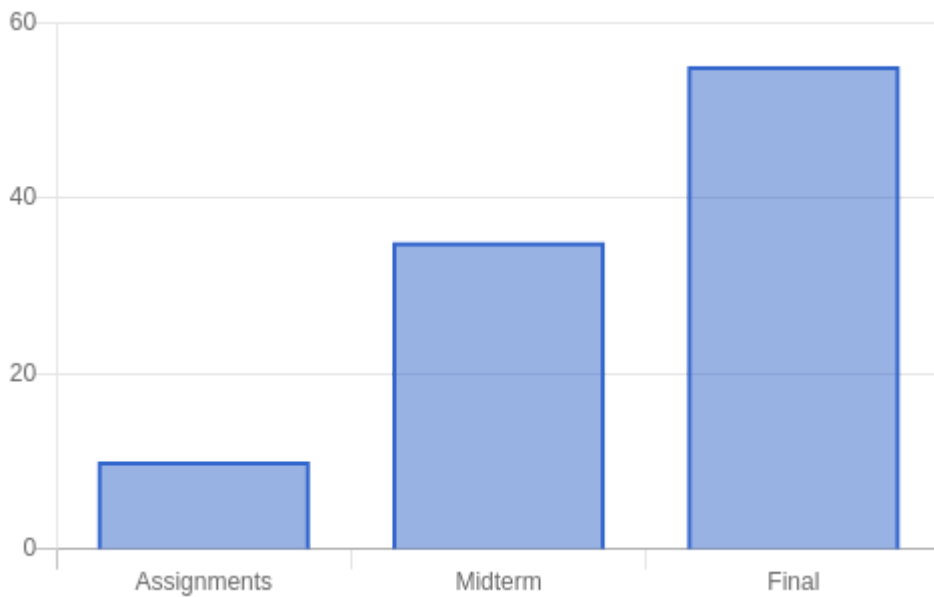
vue



Column chart

```
<column-chart :data="[
  ['Assignments', 10],
  ['Midterm', 35],
  ['Final', 55]]"
/>
```

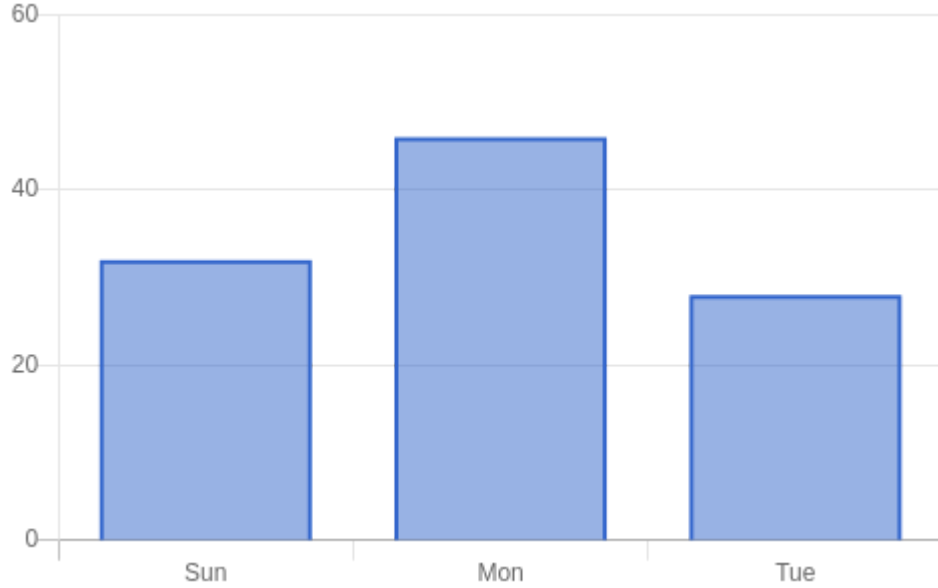
vue



another sample

```
<column-chart :data="[
  ['Sun', 32],
  ['Mon', 46],
  ['Tue', 28],
]" />
```

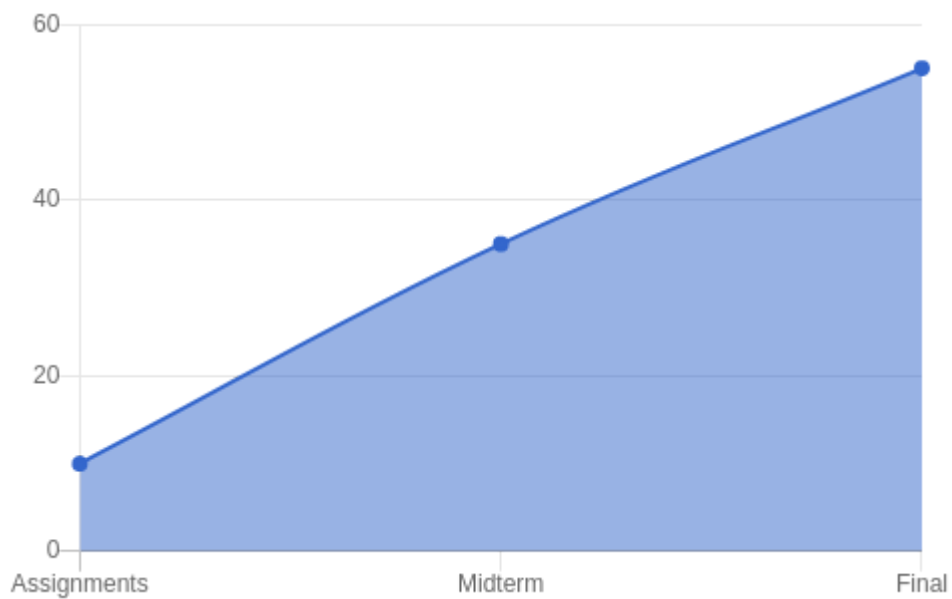
vue



Area chart

```
<area-chart :data="[
  ['Assignments', 10],
  ['Midterm', 35],
  ['Final', 55]
]" />
```

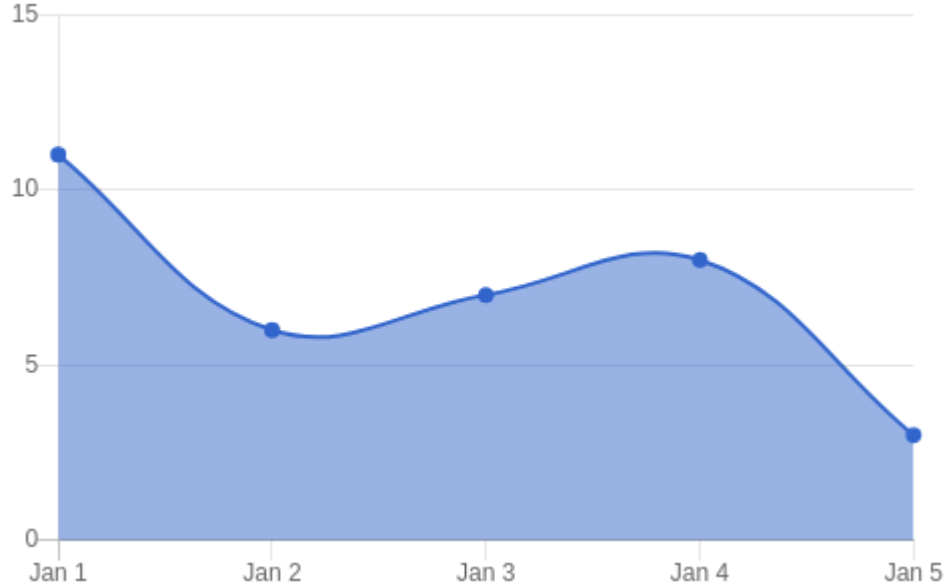
vue



another sample

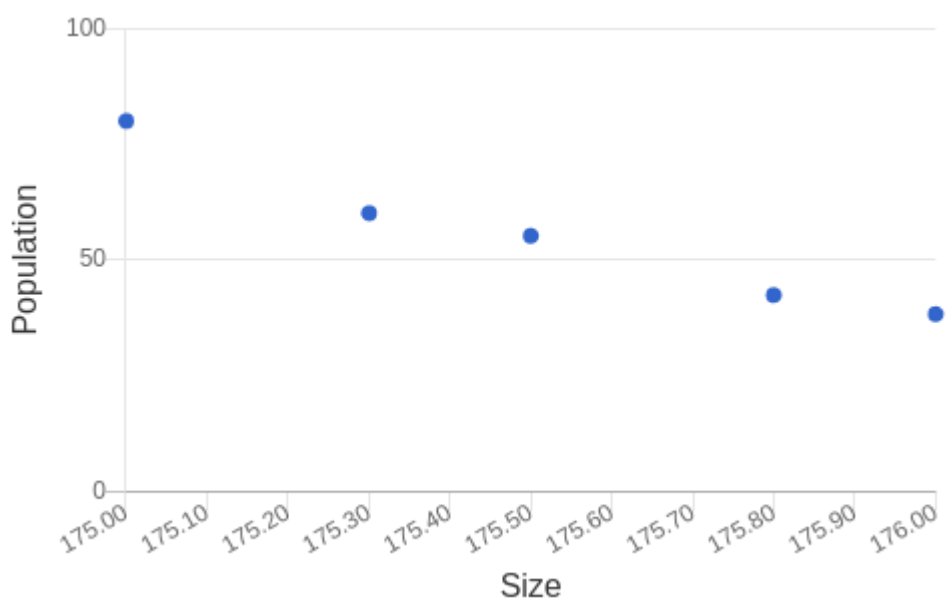
```
<area-chart :data="{
  '2017-01-01': 11,
  '2017-01-02': 6,
  '2017-01-03': 7,
  '2017-01-04': 8,
  '2017-01-05': 3,
}" />
```

vue



Scatter chart

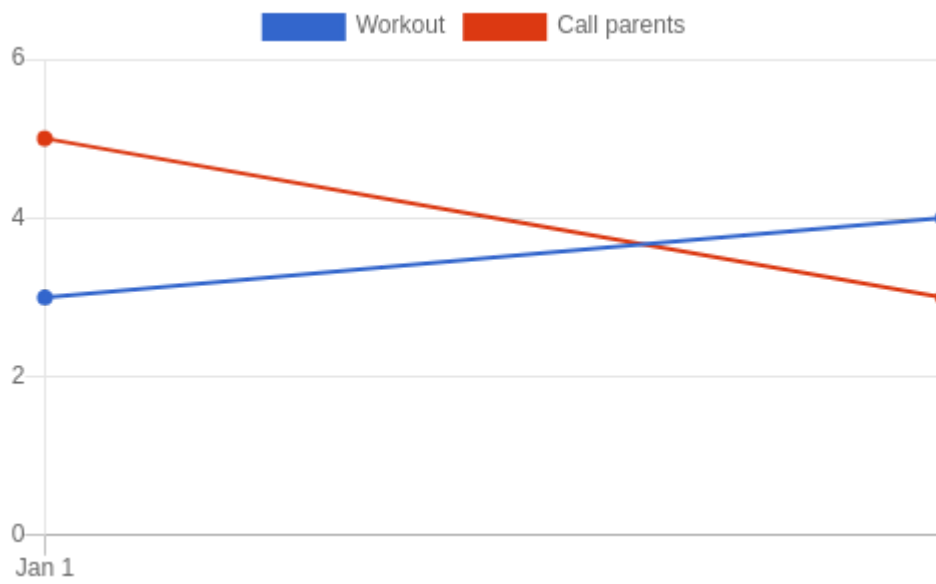
```
<scatter-chart
  xtitle="Size"
  ytitle="Population"
  :data="[
    [175.0, 80.0],
    [175.3, 60.1],
    [175.5, 55.2],
    [175.8, 42.4],
    [176.0, 38.3],
  ]"
/>
```



Line chart

```
<line-chart :data="[
  { name: 'Workout',
    data: { '2017-01-01': 3, '2017-01-02': 4 } },
```

```
    { name: 'Call parents', data: { '2017-01-01': 5, '2017-01-02': 3 } },  
  ]" />
```



Developer Guide

Quickstart new project

Start from documentation:

```
mkdir /tmp/1 && wget -qO- https://github.com/daggerok/vuepress-pdf/archive/master.zip
# wget -qO- https://github.com/daggerok/vuepress-pdf/archive/master.zip | tar xvf -
# tar -xvf <(curl -sL https://github.com/daggerok/vuepress-pdf/archive/master.zip)
# jar -xvf <(curl -sL https://github.com/daggerok/vuepress-pdf/archive/master.zip)
mv /tmp/1/vuepress-pdf /tmp/my-new-project
cd /tmp/my-new-project/
# find and replace: vuepress-pdf -> my-new-project
# such as: BASE='/my-new-project/'
vi package.json
vi .vuepress/config.js
```

Build and serve

```
npm i -E ; npm run build ; npx serve .vuepress/dist/
```

open <http://localhost:8080/>  url

Publish via CI

Checkout `.github/workflows/ci.yaml` file to see how easily you can deploy your VuePress docs on GitHub Pages:

```
name: 'GitHub actions'
on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]
jobs:
  github-pages:
    runs-on: ubuntu-latest
    steps:
```

```
- uses: actions/checkout@v2
- uses: actions/cache@v2
  id: cache-npm
  with:
    path: ~/.npm
    key: ${{ runner.os }}-${{ hashFiles('**/package.json') }}
- uses: actions/setup-node@v2
- run: npm i
- run: npm run gh-pages
- run: npm run pdf
- run: mv site.pdf src/.vuepress/dist/
- name: Deploy 🚀
  uses: JamesIves/github-pages-deploy-action@4.1.0
  with:
    branch: gh-pages # The branch the action should deploy to.
    folder: src/.vuepress/dist # The folder the action should deploy.
```

That README.md documentation file is compiled and deployed on [GitHub Pages](#)🔗

rtfm

- <https://friendlyuser.github.io/vuepress-theme-cool-starter>
- <https://github.com/mermaid-js/mermaid>
- <https://github.com/KaTeX/KaTeX>

VuePress

Diagrams quickstart

Get Started →

Mermaid

Mermaid diagrams, such as class, graph, sequence, gantt, state

UML

UML diagrams, like mind-maps, nodes, steps, flows, solidity diagrams

Charts and Math

Charts bar, pie, column, line, area, scatter are available, as well as math formulas

vuepress-pdf  GitHub actions passing 

Quick and fast rapid VuePress documentation diagrams development

Checkout [site.pdf](#) file

Made by Maksim Kostromin with ❤️ 🙌

Math

$\$x^2 + x_2 = x_2 + x^2\$$

$x_2+x_2=x_2+x_2$

yet another sample

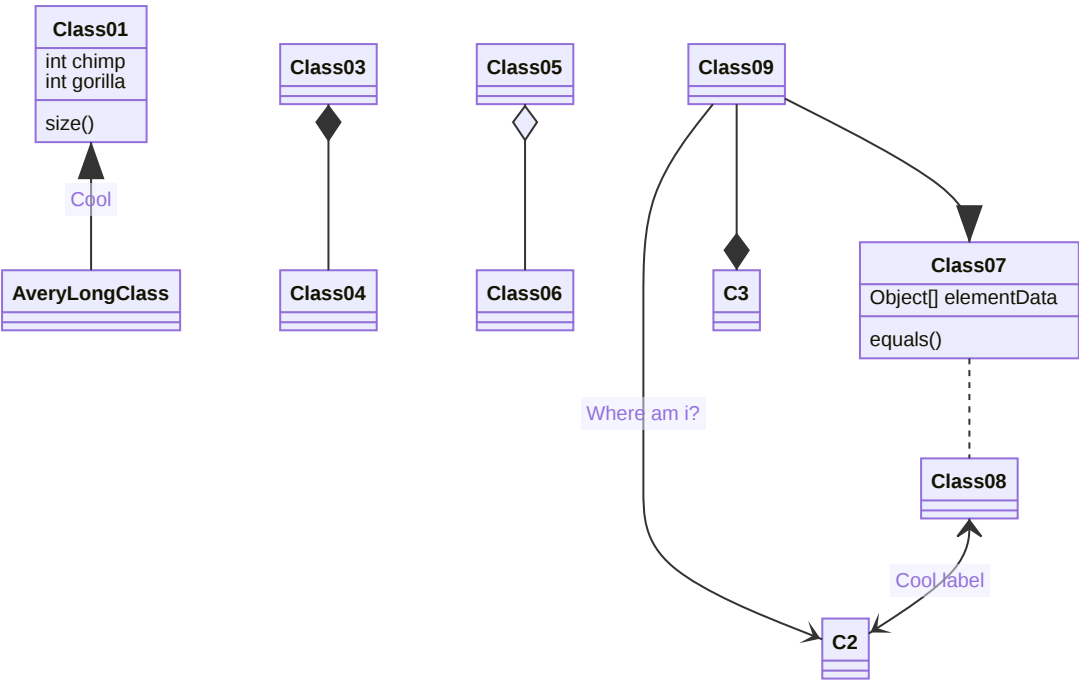
$\$\$x^2 + x = x + x^2\$\$$

$x_2+x=x+x_2$

Mermaid

Class diagram

```
<mermaid>
classDiagram
Class01 <|-- AveryLongClass : Cool
Class03 *-- Class04
Class05 o-- Class06
Class07 .. Class08
Class09 --> C2 : Where am i?
Class09 --* C3
Class09 --|> Class07
Class07 : equals()
Class07 : Object[] elementData
Class01 : size()
Class01 : int chimp
Class01 : int gorilla
Class08 <--> C2: Cool label
</mermaid>
```

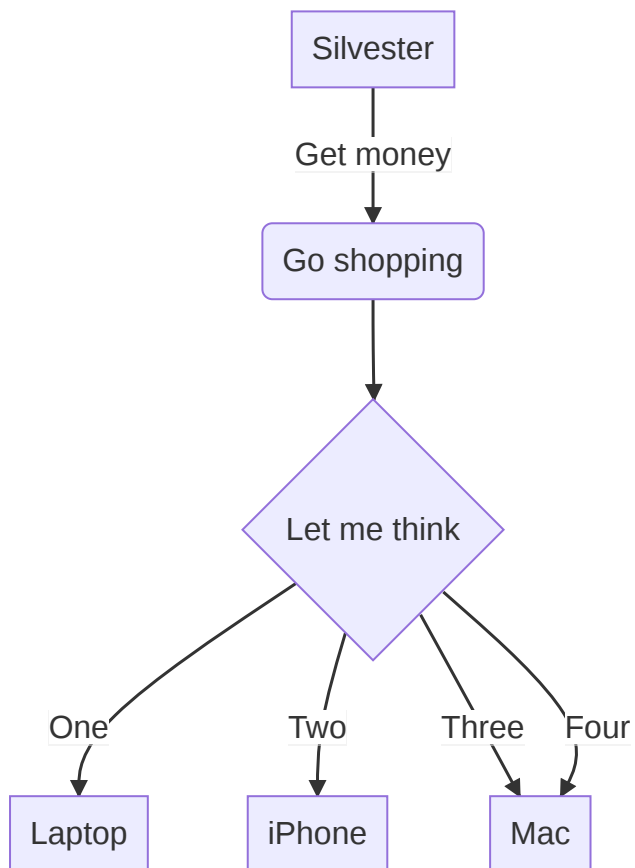


Graph

```

<mermaid>
graph TD
    A[Silvester] -->|Get money| B(Go shopping)
    B --> C{Let me think}
    C -->|One| D[Laptop]
    C -->|Two| E[iPhone]
    C -->|Three| F[Car]
    C -->|Four| F[Mac]
</mermaid>

```

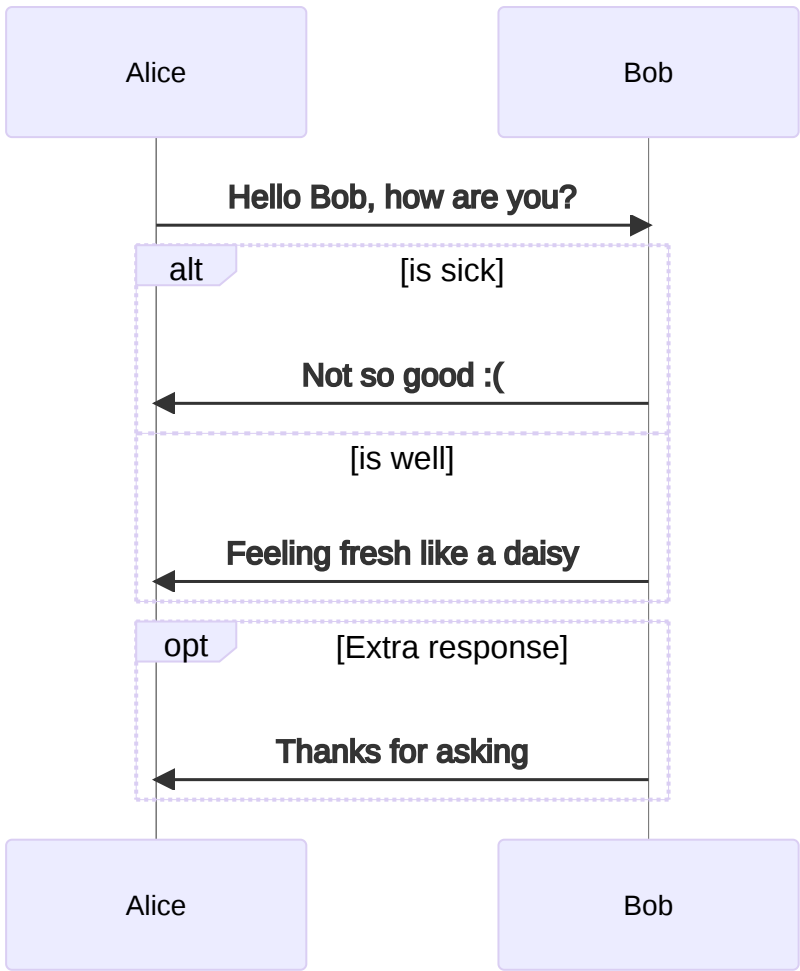


Sequence diagram

```

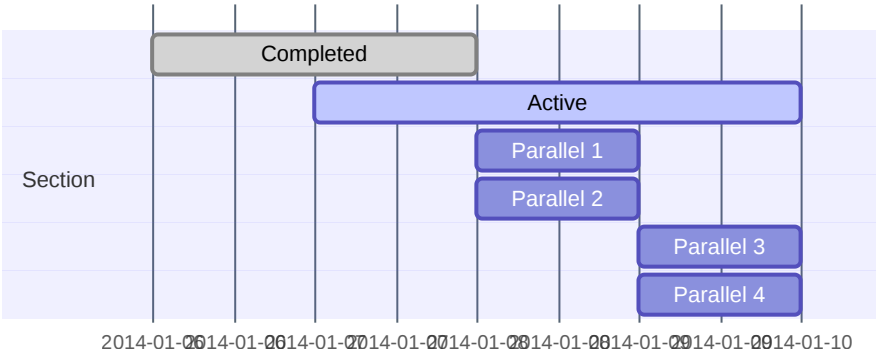
<mermaid>
sequenceDiagram
    Alice->>Bob: Hello Bob, how are you?
    alt is sick
        Bob->>Alice: Not so good :(
    else is well
        Bob->>Alice: Feeling fresh like a daisy
    end
    opt Extra response
        Bob->>Alice: Thanks for asking
    end
</mermaid>

```



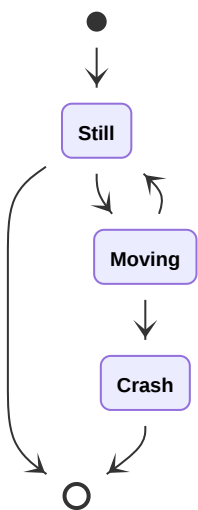
Gantt diagram

```
<mermaid>
gantt
    section Section
    Completed      :done,      des1, 2014-01-06, 2014-01-08
    Active          :active,    des2, 2014-01-07, 3d
    Parallel 1      :          des3, after des1, 1d
    Parallel 2      :          des4, after des1, 1d
    Parallel 3      :          des5, after des3, 1d
    Parallel 4      :          des6, after des4, 1d
</mermaid>
```



Mermaid 8.4.3+ => State diagram

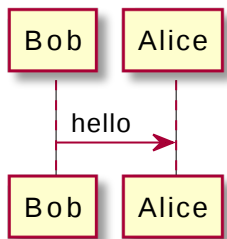
```
<mermaid>
stateDiagram
    [*] --> Still
    Still --> [*]
    Still --> Moving
    Moving --> Still
    Moving --> Crash
    Crash --> [*]
</mermaid>
```



UML

PlantUml

```
@startuml
Bob -> Alice : hello
@enduml
```



MindMap

```
@startuml
@startmindmap

title My super title

* <&flag>Debian
** <&globe>Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** <&graph>LMDE
** <&pulse>SolydXK
** <&people>SteamOS
** <&star>Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio

caption figure 1

@endmindmap
@enduml
```

My super title

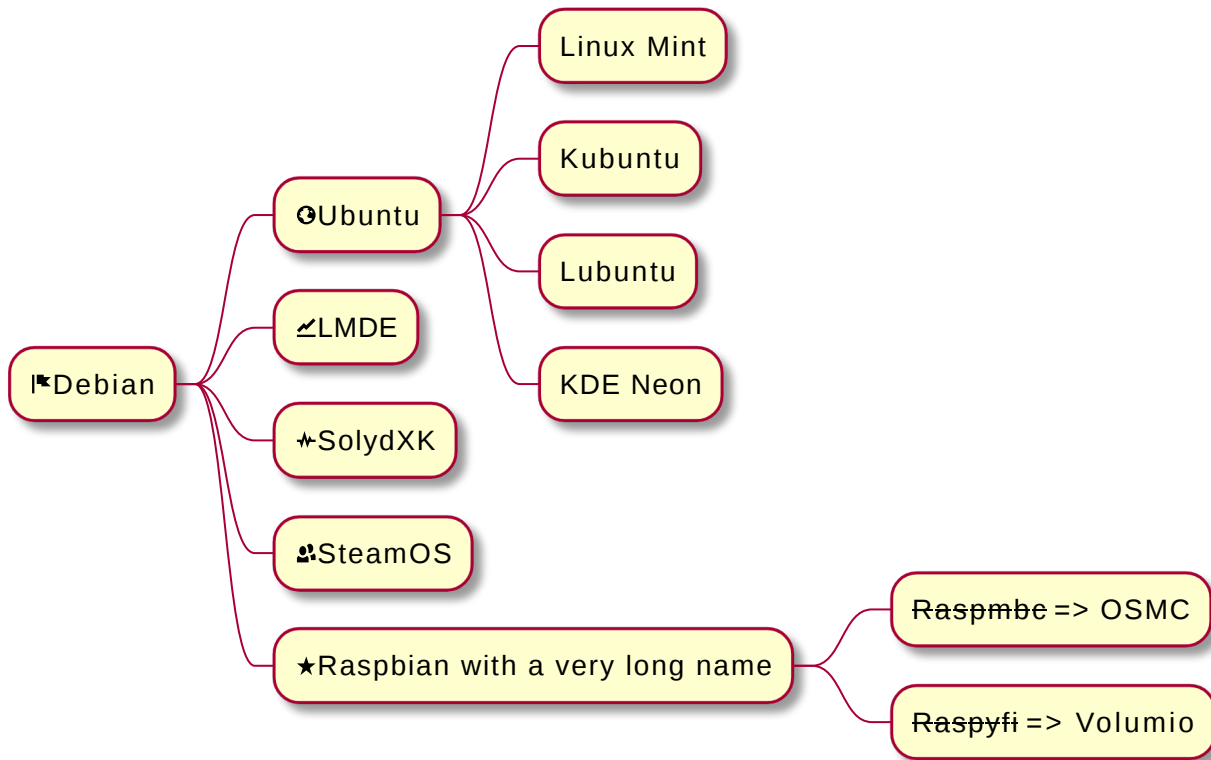
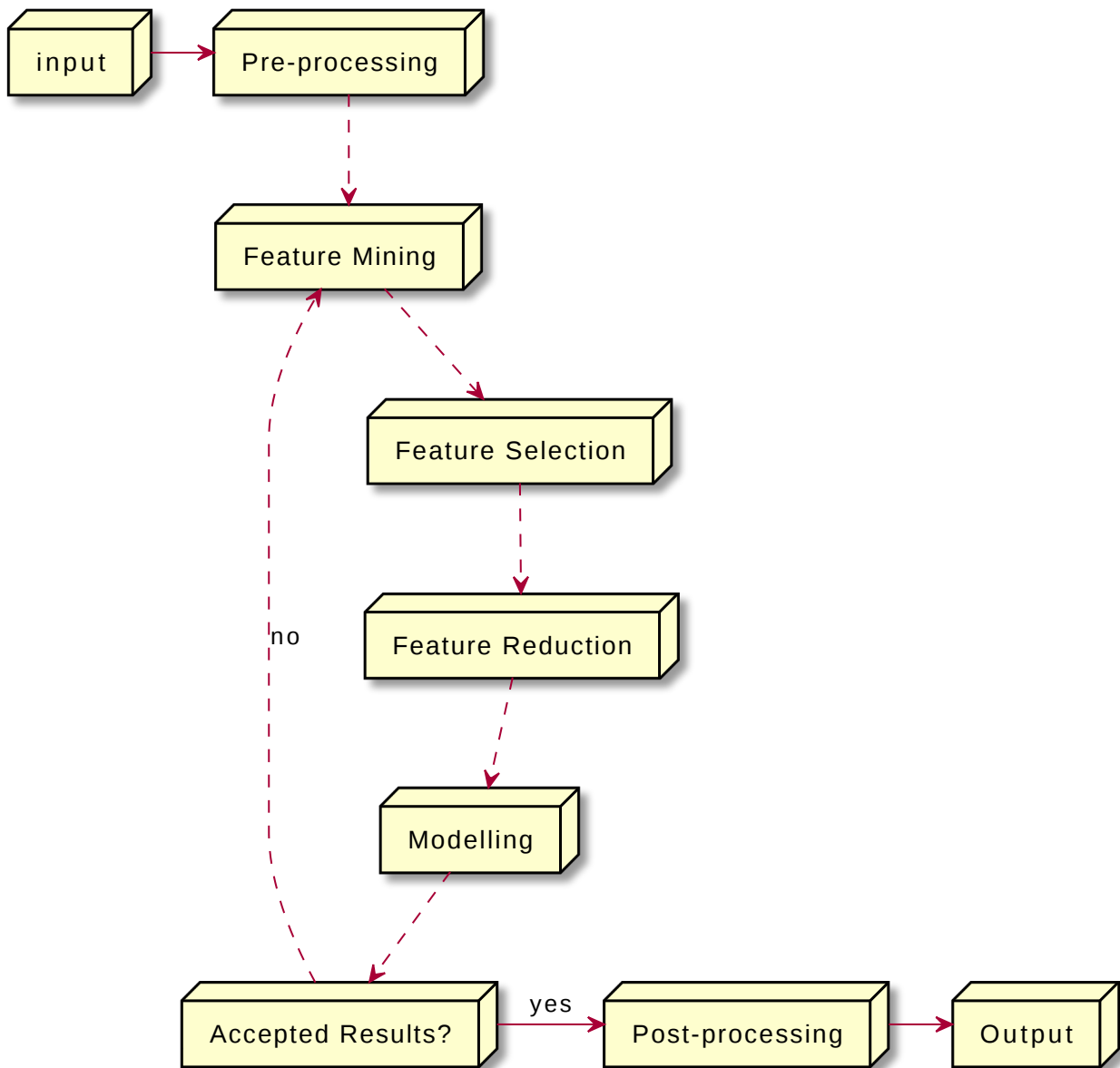


figure 1

Nodes

```
@startuml
node in as "input"
node p as "Pre-processing"
node fm as "Feature Mining"
node fs as "Feature Selection"
node fr as "Feature Reduction"
node m as "Modelling"
node a as "Accepted Results?"
node pp as "Post-processing"
node o as "Output"
```

```
in -> p
p ..> fm
fm ..> fs
fs ..> fr
fr ..> m
m ..> a
a ..> fm : no
a -> pp : yes
pp -> o
@enduml
```

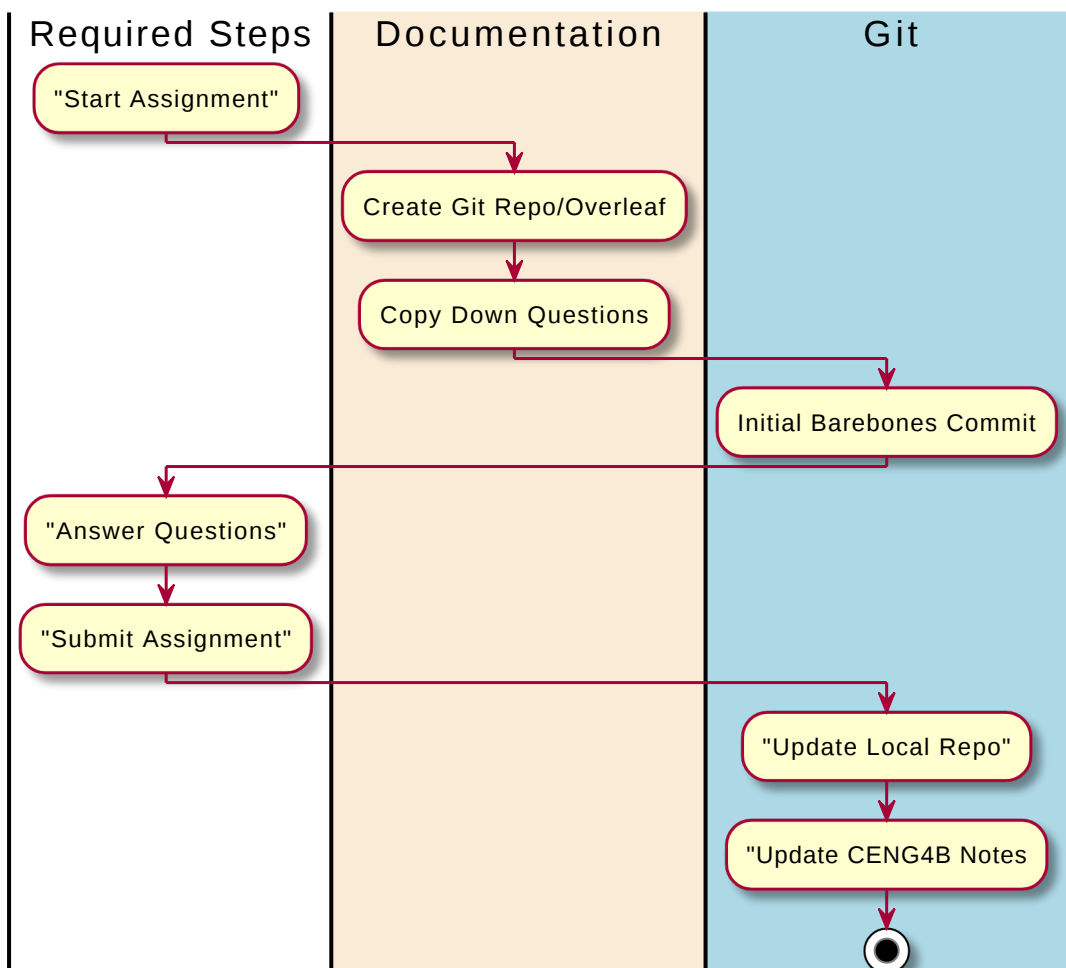


Steps flow

```

@startuml
  |Required Steps|
  :Start Assignment;
  |#AntiqueWhite|Documentation|
  :Create Git Repo/Overleaf;
  :Copy Down Questions;
  |#LightBlue|Git|
  :Initial Barebones Commit;
  |Required Steps|
  :Answer Questions;
  :Submit Assignment;
  |Git|
  :Update Local Repo;
  :Update CENG4B Notes;
  stop
@enduml

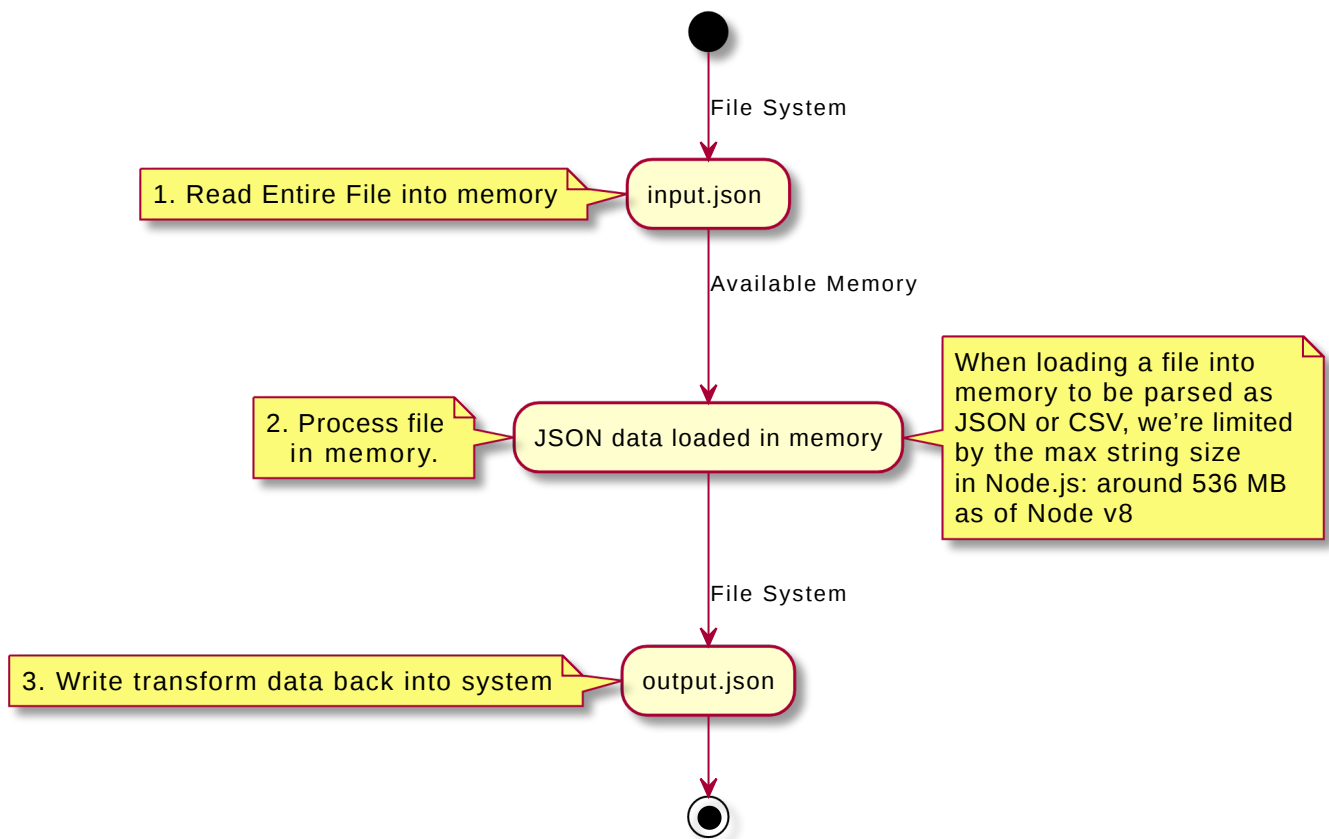
```

Notes flow

```
@startuml
(*) --> [File System] "input.json "
note left
1. Read Entire File into memory
end note
--> [Available Memory] "JSON data loaded in memory"
note left
2. Process file
in memory.
end note
note right
When loading a file into
memory to be parsed as
JSON or CSV, we're limited
by the max string size
in Node.js: around 536 MB
as of Node v8
end note
--> [File System] "output.json"
note left
3. Write transform data back into system
end note
```

--> (*)
@endum1

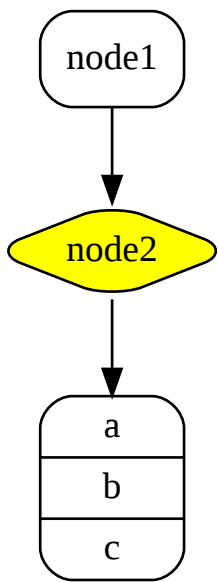


Solidity Diagram

sample 1

```
@startuml
digraph foo {
    node [style=rounded]
    node1 [shape=box]
    node2 [fillcolor=yellow, style="rounded,filled", shape=diamond]
    node3 [shape=record, label="{ a | b | c }"]

    node1 -> node2 -> node3
}
@enduml
```



sample 2

```
@startuml
strict digraph cool {
  exists [color=blue]
  authenticate [color=blue]
  require
  create
  UserCreated
  destroy
  UserDestroyed
  get [color=blue]
  authenticate -> require
  create -> UserCreated
  destroy -> require
  destroy -> UserDestroyed
  get -> require
}
@enduml
```

