

The fucking simple pigger document format

<https://github.com/ikey4u/pigger>

```
---
title: Pigger, Pigger, Pigger
date: 2018-12-14
author: The awesome author
---
```

pigger 首先会根据空行划分将文档分成为块,做一些特定处理后形成基本块.

```
# Head 1

This demo file outlines what a pigger doc should look like.

## Head 2

Bla bla bla ...
```

```
---
title: Pigger, Pigger, Pigger
date: 2018-12-14
author: The awesome author
---
```

```
# Head 1

This demo file outlines what a pigger doc should look like.

## Head 2

Bla bla bla ...
```

- code in list

```
 8  //:go
 4  func main() {
    key := brute(os.Args[1], os.Args[2], 5)
    fmt.Printf("Key is %s\n", key)
  }

Go language bla bla bla ...
bla bla bla ...
```

如果当前的 list 内嵌等级为 1,

那么该 list 下面 4 个空格缩进的为该 list 内容,

8 个空格缩进的为 list 内嵌代码块.

- Another list item

```
You can write several lines in adjacent way, they will be rendered into
one line!
bla bla bla ...

- A very very very very long long nested list title, you can break it in
multiple lines but rendered in one line...

- A nested list title
The nested list content

Even more lines ...

You can split thme in space line!
```

一个段落可以有多样行, 只要中间不空行, 可以写成多行, 在渲染时将会按照一行来渲染.

列表的小标题后面跟上一个空行, 那么渲染时将会另起一行.

列表的内容里面, 每一段文字可以用空行来间隔 渲染时每一段将会视作一个段落.

Pigger, Pigger, Pigger

by The awesome author

Publish → 2018-12-14 Update → 2019-01-12

1. Head 1

This demo file outlines what a pigger doc should look like.

1.1. Head 2

Bla bla bla ...

- code in list

```
func main() {
    key := brute(os.Args[1], os.Args[2], 5)
    fmt.Printf("Key is %s\n", key)
}
```

Go language bla bla bla ... bla bla bla ...

- Another list item

You can write several lines in adjacent way, they will be rendered into one line! bla bla bla ...

- A very very very very long long nested list title, you can break it in multiple lines but rendered in one line...

- A nested list title

The nested list content

Even more lines ...

You can split thme in space line!

A normal code `highlight`

A normal code `highlight`

行内高亮只需要用反引号括起来要高亮的部分即可.

```
#include <stdio.h>
int main() {
    printf("%s\n", "hello world");
}
```

单独的代码块需要缩进 4 个空格.

```
#include <stdio.h>
int main() {
    printf("%s\n", "hello world");
}
```