

## 简单说明

是一个可读性高，用来表达数据序列的格式。YAML 的意思其实是：仍是一种标记语言，但为了强调这种语言以数据做为中心，而不是以标记语言为重点

## 基本语法

- 缩进时不允许使用Tab键，只允许使用空格
- 缩进的空格数目不重要，只要相同层级的元素左侧对齐即可
- # 标识注释，从这个字符一直到行尾，都会被解释器忽略

## YAML 支持的数据结构

- 对象：键值对的集合，又称为映射（mapping） / 哈希（hashes） / 字典（dictionary）
- 数组：一组按次序排列的值，又称为序列（sequence） / 列表（list）
- 纯量（scalars）：单个的、不可再分的值

### 对象类型：对象的一组键值对，使用冒号结构表示

```
name: Steve
age: 18
```

Yaml 也允许另一种写法，将所有键值对写成一个行内对象

```
hash: { name: Steve, age: 18 }
```

### 数组类型：一组连词线开头的行，构成一个数组

```
animal
- Cat
- Dog
```

数组也可以采用行内表示法

```
animal: [Cat, Dog]
```

### 复合结构：对象和数组可以结合使用，形成复合结构

```
1 languages:
2 - Ruby
3 - Perl
4 - Python
5 websites:
6 YAML: yaml.org
7 Ruby: ruby-lang.org
8 Python: python.org
9 Perl: use.perl.org
```

## 纯量：纯量是最基本的、不可再分的值。以下数据类型都属于纯量

```
1 字符串 布尔值 整数 浮点数 Null
2 时间 日期
```

数值直接以字面量的形式表示

```
number: 12.30
```

布尔值用true和false表示

```
isSet: true
```

null用 ~ 表示

```
parent: ~
```

时间采用 ISO8601 格式

```
iso8601: 2001-12-14t21:59:43.10-05:00
```

日期采用复合 iso8601 格式的年、月、日表示

```
date: 1976-07-31
```

YAML 允许使用两个感叹号，强制转换数据类型

```
e: !!str 123
```

```
f: !!str true
```

## 字符串

### 字符串默认不使用引号表示

```
str: 这是一行字符串
```

### 如果字符串之中包含空格或特殊字符，需要放在引号之中

```
str: '内容： 字符串'
```

### 单引号和双引号都可以使用，双引号不会对特殊字符转义

```
s1: '内容\n字符串'  
s2: "内容\n字符串"
```

单引号之中如果还有单引号，必须连续使用两个单引号转义

```
str: 'labor''s day'
```

字符串可以写成多行，从第二行开始，必须有一个单空格缩进。换行符会被转为 空格

```
str: 这是一段  
    多行  
    字符串
```

多行字符串可以使用|保留换行符，也可以使用>折叠换行

```
this: |  
Foo  
Bar  
that: >  
Foo  
Bar
```

+ 表示保留文字块末尾的换行，- 表示删除字符串末尾的换行

```
s1: |  
    Foo  
  
s2: |+  
    Foo  
  
s3: |-  
    Foo
```