

Mint UI - Unit01

1. 组件库基础

1.1 什么是组件?

组件(Component)是对于页面的结构(<template>)、表现(<style>)及行为(<script>)的封闭, 可以实现一次定义, 多次使用。

1.2 组件分类

根据组件针对的设备, 可以分为:

A. 移动端

- Mint UI (饿了么) -- <https://mint-ui.github.io/>
- Vant UI (有赞) -- <https://youzan.github.io/vant/#/zh-CN/>
- Cube UI (滴滴) -- <https://didi.github.io/cube-ui/#/zh-CN/>

B. 桌面端

- Element UI (饿了么) -- <https://element.eleme.cn/>
- AT-UI (凹凸实验室)
- View UI (视图更新)

1.3 Mint UI 安装

A. 在命令行中进入脚手架目录

本地磁盘 (D:) > webtn2008 > 01_MintUI > Day01 > demo > scaffolding				
名称	修改日期	类型	大小	
node_modules	2020/2/23 23:23	文件夹		
public	2020/2/23 23:15	文件夹		
src	2020/2/23 23:15	文件夹		
.gitignore	2020/2/23 23:15	文本文档	1 KB	
babel.config.js	2020/2/23 23:15	JavaScript 文件	1 KB	
package.json	2020/2/23 23:23	JSON 文件	1 KB	
package-lock.json	2020/2/23 23:23	JSON 文件	403 KB	
README.md	2020/2/23 23:16	Markdown File	1 KB	
vue.config.js	2020/2/23 23:25	JavaScript 文件	1 KB	

C:\Windows\System32\cmd.exe

Microsoft Windows [版本 10.0.18362.959]
(c) 2019 Microsoft Corporation。保留所有权利。
D:\webtn2008\01_MintUI\Day01\demo\scaffolding>

B.在命令行输入以下命令

```
npm install --save mint-ui
```

1.4 配置Mint UI

配置 Mint UI 需要在 main.js 中实现：

```
// 导入MintUI模块
import MintUI from 'mint-ui';

// 导入样式文件
import 'mint-ui/lib/style.min.css';

// 通过Vue.use()方法将MintUI注册为插件
Vue.use(MintUI);
```

2.Mint UI 组件库

• Header 组件

Header 组件为顶部导航组件，其语法结构是：

```
<mt-header title="标题信息" fixed>
</mt-header>
```

`fixed` 属性表示是否固定在页面顶部，布尔属性，示例代码如下：

```
<template>
<div>
  <mt-header title="学前端,到学问" fixed>
  </mt-header>
</div>
</template>
```

Header 组件的高度是 40px

在 Header 组件中可以嵌套子元素，此时必须为子元素添加 slot="left" 或 slot="right" 属性，示例代码如下：

```
<mt-header title="学前端,到学问" fixed>
  <div slot="left">左侧</div>
  <div slot="right">右侧</div>
</mt-header>
```

slot 称为插槽，是组件的设计者留给使用者实现自定义内容的区域，示例代码如下：

```
<div slot="header"></div>
```

• reset.css

每个浏览器对于标签都有默认样式，但是不同浏览器的默认样式是不同的。为了所有浏览器的默认样式是相同的，所以需要引入 reset.css 进行样式的重置。

在 public/index.html 中书写以下语句：

```
<link type="text/css" rel="stylesheet" href="/css/reset.css">
```

• Button 组件

Button 组件是按钮组件，其语法结构是：

```
<mt-button
  type="按钮类型"
  size="尺寸"
  icon="图标类型"
  disabled
  plain>
  ...
</mt-button>
```

`type` 属性值包括: `default` (默认)、`primary` (主要)、`danger` (危险)

`size` 属性值包括: `small` (小的)、`normal` (标准)、`large` (大的)

`icon` 属性值包括: `back` (返回)、`more` (更多)

`disabled` 属性用于控制按钮是否禁用, 布尔类型

`plain` 属性用于控制按钮是否为镂空, 布尔类型

• Field 组件

`Field` 组件用于实现表单编辑器, 其语法结构是:

```
<mt-field
  type="输入框的类型"
  label="标签文本"
  placeholder="占位符"
  :attr="原生属性"
  state="检测状态"
  disableClear
  readonly
  disabled>
</mt-field>
```

`type` 属性值包括: `text` (单行文本框)、`password` (密码框)、`textarea` (多行文本框)、`email` (电子邮箱)、`url` (URL 地址)等

`:attr` 属性用于控制输入框的原生属性, 对象类型, 示例代码如下:

```
<mt-field
  type="password"
  label="密码"
  :attr="{maxLength:'12',autocomplete:'off'}"
  placeholder="请输入密码">
</mt-field>
```

`state` 属性值包括: `success` (成功)、`warning` (警告)、`error` (错误)

`disableClear` 属性用于控制是否显示清理图标, 布尔属性



• Toast 组件

`toast` 组件用于显示短消息提示框, 其语法结构是:

```
//简捷方式
this.$toast("提示信息")

//标准方式
this.$toast({
  message: "提示信息",
  position: "位置(top|middle|bottom)",
  duration: 持续时长(单位:毫秒), Number类型
})
```

作业:

- A. 让用户名、密码及确认密码分别在失去焦点时完成校验
 - B. 在单击 **免费注册** 按钮时同样也要完成校验
 - C. 无论哪一种校验情况，都需要实时修改表单控件的检测状态(state)
-
- D. 参照用户注册页面来实现登录 (Login.vue) 页面的业务功能
 - E. 复习 Node.js 基础知识、MySQL 的基础知识