

# 前端小课

Smalle

温馨提示:好记性不如烂笔头



# 小课也有大纲

- 前端知识图谱
- JS相关技术演进
- Vue实战总结

# 知其 -> 知其然 -> 知其所以然



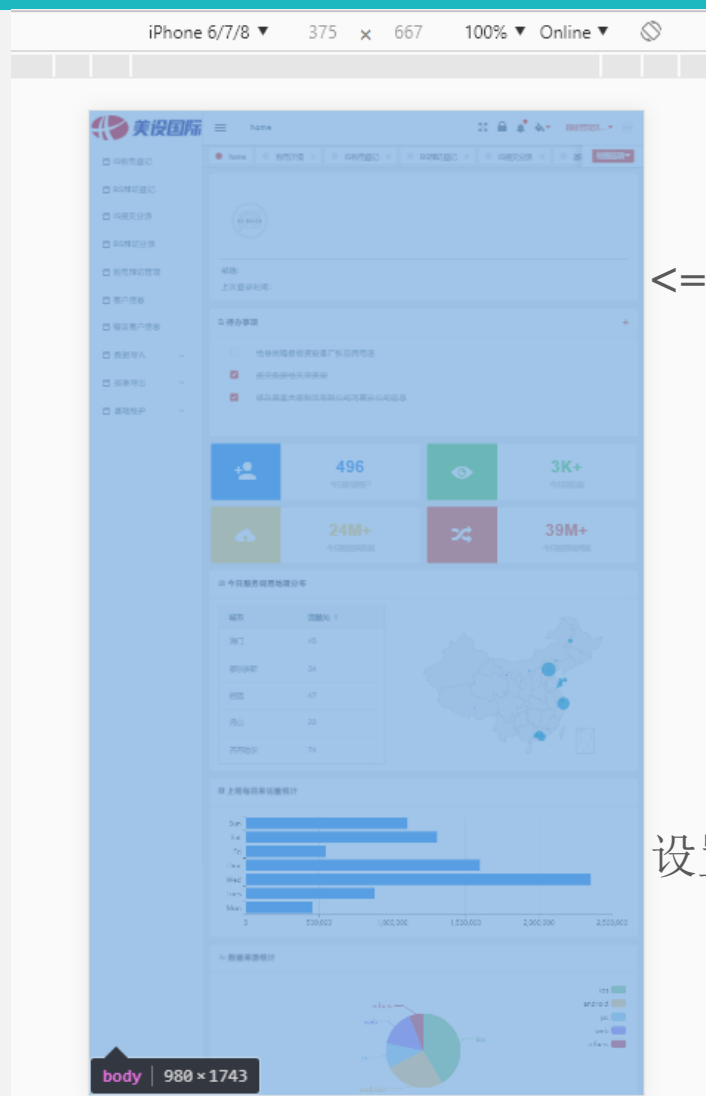
- [响应式布局](#)
- [模板引擎](#)
- [CSS设计模式](#)
- [WebPagetest](#)

# 响应式布局

- viewport属性配置
  - `<meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=0">`
- @media
  - ```
@media (min-width: 768px)
.ivu-col-span-sm-12 {
  display: block;
  width: 50%;
}
```

# 响应式布局：viewport

- css 1px != 屏幕 1px
- viewport
  - layout viewport
  - visual viewport
  - ideal viewport



<== 未设置viewport



设置过viewport ==>



# 响应式布局: @media

The image displays a web browser window showing a responsive design interface. The browser's developer tools are open, highlighting the CSS styles for the page. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains a list of menu items, and the main content area displays a dashboard with various charts and data. The browser's address bar shows the URL "http://localhost:8080/". The developer tools show the following CSS styles:

```
element.style {
  margin-bottom: 10px;
  padding-left: 5px;
  padding-right: 5px;
}

@media (min-width: 768px) {
  .ivu-col-span-sm-12 {
    display: block;
    width: 50%;
  }
}

@media (min-width: 768px) {
  .ivu-col-span-sm-1, .ivu-col-span-sm-10, .ivu-col-span-sm-11, .ivu-col-span-sm-12, .ivu-col-span-sm-13, .ivu-col-span-sm-14, .ivu-col-span-sm-15, .ivu-col-span-sm-16, .ivu-col-span-sm-17, .ivu-col-span-sm-18, .ivu-col-span-sm-19, .ivu-col-span-sm-2, .ivu-col-span-sm-20, .ivu-col-span-sm-21, .ivu-col-span-sm-22, .ivu-col-span-sm-23, .ivu-col-span-sm-24 {
    display: block;
    width: 100%;
  }
}

.ivu-col {
  position: relative;
  display: block;
}

.ivu-col-span-xs-24 {
  display: block;
  width: 100%;
}
```

The browser's address bar shows the URL "http://localhost:8080/". The developer tools show the following CSS styles:

```
element.style {
  margin-bottom: 10px;
  padding-left: 5px;
  padding-right: 5px;
}

@media (min-width: 768px) {
  .ivu-col-span-sm-12 {
    display: block;
    width: 50%;
  }
}

@media (min-width: 768px) {
  .ivu-col-span-sm-1, .ivu-col-span-sm-10, .ivu-col-span-sm-11, .ivu-col-span-sm-12, .ivu-col-span-sm-13, .ivu-col-span-sm-14, .ivu-col-span-sm-15, .ivu-col-span-sm-16, .ivu-col-span-sm-17, .ivu-col-span-sm-18, .ivu-col-span-sm-19, .ivu-col-span-sm-2, .ivu-col-span-sm-20, .ivu-col-span-sm-21, .ivu-col-span-sm-22, .ivu-col-span-sm-23, .ivu-col-span-sm-24 {
    display: block;
    width: 100%;
  }
}

.ivu-col {
  position: relative;
  display: block;
}

.ivu-col-span-xs-24 {
  display: block;
  width: 100%;
}
```

# 响应式布局: @media (栅格系统)

```
<Col :xs="24" :sm="12" :md="6" :style="{marginBottom: '10px'}">
  <infor-card
    id-name="user_created_count"
    :end-val="count.createUser"
    iconType="android-person-add"
    color="#2d8cf0"
    intro-text="今日浏览量"
  ></infor-card>
</Col>
<Col :xs="24" :sm="12" :md="6" :style="{marginBottom: '10px'}">
  <infor-card
    id-name="visit_count"
    :end-val="count.visit"
    iconType="ios-eye"
    color="#64d572"
    :iconSize="50"
    intro-text="今日数据采集量"
  ></infor-card>
</Col>
```



# 模板引擎

```
!!! 5
html(lang="en")
  head
    title= pageTitle
    :javascript
      | if (foo) {
      |   bar()
      | }
  body
    h1 Jade - node template engine
    #container
      - if (youAreUsingJade)
        You are amazing
      - else
        Get on it!
        Get on it!
        Get on it!
        Get on it!
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      <![CDATA[
        if (foo) {
          bar()
        }
      </script>
    </head>
    <body>
      <h1>Jade - node template engine</h1>
      <div id="container">
        <p>You are amazing</p>
      </div>
    </body>
  </html>
```

Jade

layui

//第一步: 编写模版。你可以使用一个script标签存放模板, 如:

```
<script id="demo" type="text/html">
  <h3>{{ d.title }}</h3>
  <ul>
    {{# layui.each(d.list, function(index, item){ }}
      <li>
        <span>{{ item.modname }}</span>
        <span>{{ item.alias }}: </span>
        <span>{{ item.site || '' }}</span>
      </li>
    {{# }}; }}
    {{# if(d.list.length === 0){ }}
      无数据
    {{# }} }}
  </ul>
</script>
```

//第二步: 建立视图。用于呈现渲染结果。

```
<div id="view"></div>
```

//第三步: 渲染模版

```
var data = { //数据
  "title": "Layui常用模块"
  , "list": [{ "modname": "弹层", "alias": "layer", "site": "layer.layui.com" }, { "modname": "表单", "alias": "form" } ]
}
var getTpl = demo.innerHTML
, view = document.getElementById('view');
laytpl(getTpl).render(data, function(html){
  view.innerHTML = html;
});
```

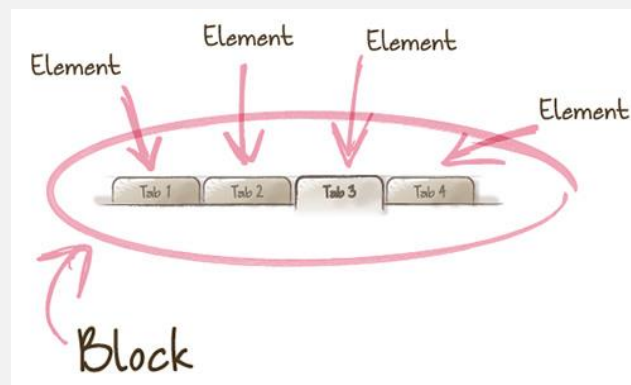
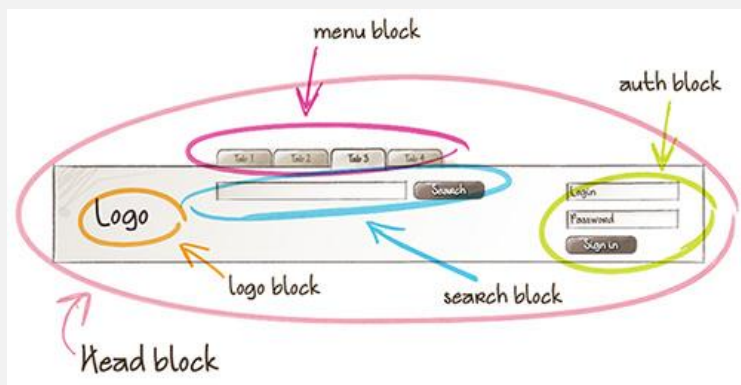




# CSS设计模式：OOCSS、SMACSS与BEM

- OOCSS：面向对象的CSS。如：bootstrap(btn-default)
- SMACSS (smacks css)
  - CSS分类规范：Base、Layout、Module、State、Theme
  - 命名规范：1-/layout-、模块本身、is-(.is-active、.is-hidden)、theme-
- BEM：Block、Element、Modifier

•



```
.sidebar {  
  .sidebar--left__section {  
    .sidebar--left__section--header {}  
    .sidebar--left__section--footer {}  
  }  
}
```

- Modifier是描述Block或Element的属性或状态
- Block作为起始开头; 不同 Block 和 Element 用 \_\_ 两个底线区隔开来; 不同的 Modifier 则用两个 - 区隔; 一个 - 则表示这个 class 不依赖任何 Block 或 Element



# JS相关技术演进



# JavaScript、jQuery

- JavaScript: 1995年, 由Netscape公司的Brendan Eich, 在网景浏览器上首次设计实现而成
- jQuery: 2006年8月John Resig等人发布了jQuery第一个稳定版本。相关框架
  - EasyUI
  - Bootstrap 响应式布局
  - Metronic
  - Layui (内置jQuery模块。异步模块加载, 不遵循AMD)

# 模块化: CommonJS

- NodeJS: 2009年, Ryan Dahl开发了node.js项目, 将javascript语言用于服务器端编程
- module、exports、require

```
// foo.js
module.exports = {
  multiply: function(n) {return n * 1000 }
};
```

```
// main.js
var foo = require("./foo");
foo.multiply(5);
```

```
var module = {
  exports: {}
};

(function(module, exports) {
  exports.multiply = function (n) { return n * 1000 };
  // .....
})(module, module.exports)
```

```
var f = module.exports.multiply;
f(5) // 5000
```

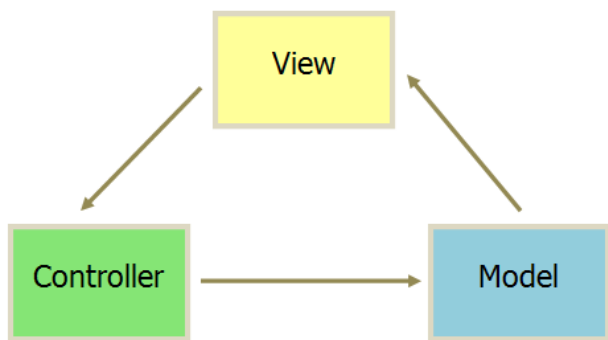
# 模块化: AMD/CMD

- AMD: Asynchronous Module Definition, 异步模块定义。require([module], callback);
- 实现: require.js

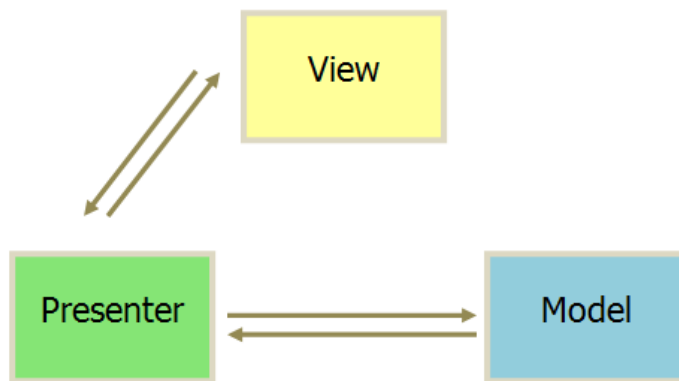
```
// main.js
require(['jquery', 'moduleA', 'moduleB'], function ($, moduleA, moduleB) {
  // some code here
});
```

- CMD: Common Module Definition, 通用模块定义。CMD规范是国内发展出来的
- CMD实现如: SeaJS
- 打包工具: Browserify、Webpack

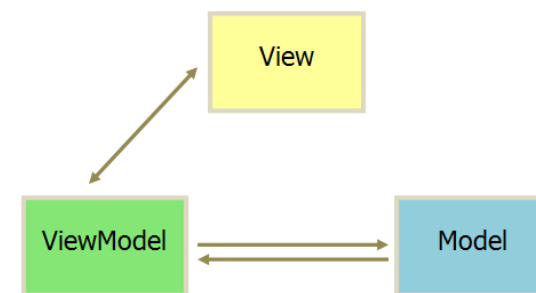
# 数据绑定



MVC



MVP



MVVM



# ES6

- CoffeeScript
- Typescript
  - 示例: class、 interface
- ECMAScript 6 (简称ES6)是于2015年6月正式发布的JavaScript语言的标准，正式名为ECMAScript 2015 (ES2015)
- ES6新特性: let、箭头操作符(=>)、扩展运算符(...)、 promises 、 class、 module等

# 移动互联网的那点事

- H5
  - [Amaze UI](#)、[MUI](#)
- App: 原生、H5、混合
  - dcloud、apicloud、cordova
  - React Native、Angular2、Weex
- 微信小程序
  - 官方规范、mpvue、wepy

# Vue实战总结



# Vue今世情缘

- AngularJS: 诞生于2009年, 由Misko Hevery 等人创建, 后为Google所收购 (基于JS)
  - Angular2/ Angular4 (基于TS)
- React: 起源于 Facebook 的内部项目, 用来架设 Instagram 的网站, 并于 2013 年 5 月开源
  - React Native
- Vue: 2016年, 国产开源框架
  - Weex
  - UI框架: [Element](#)、[iView](#)、Vuetify.....

# Vue技术栈

- 开发环境: nodejs
- 包管理工具: npm/yarn
- 打包工具: webpack
- 路由: vue router
- 状态管理器: vuex
- UI框架: element、iview
- Ajax: axios
- 测试: mocha
- Api管理: [yapi](#)
- 代码风格校验: eslint
- 文档整理: markdown
- 代码管理: git
- .....

# 知其？





# 谢谢！

敬请期待《后端小课》