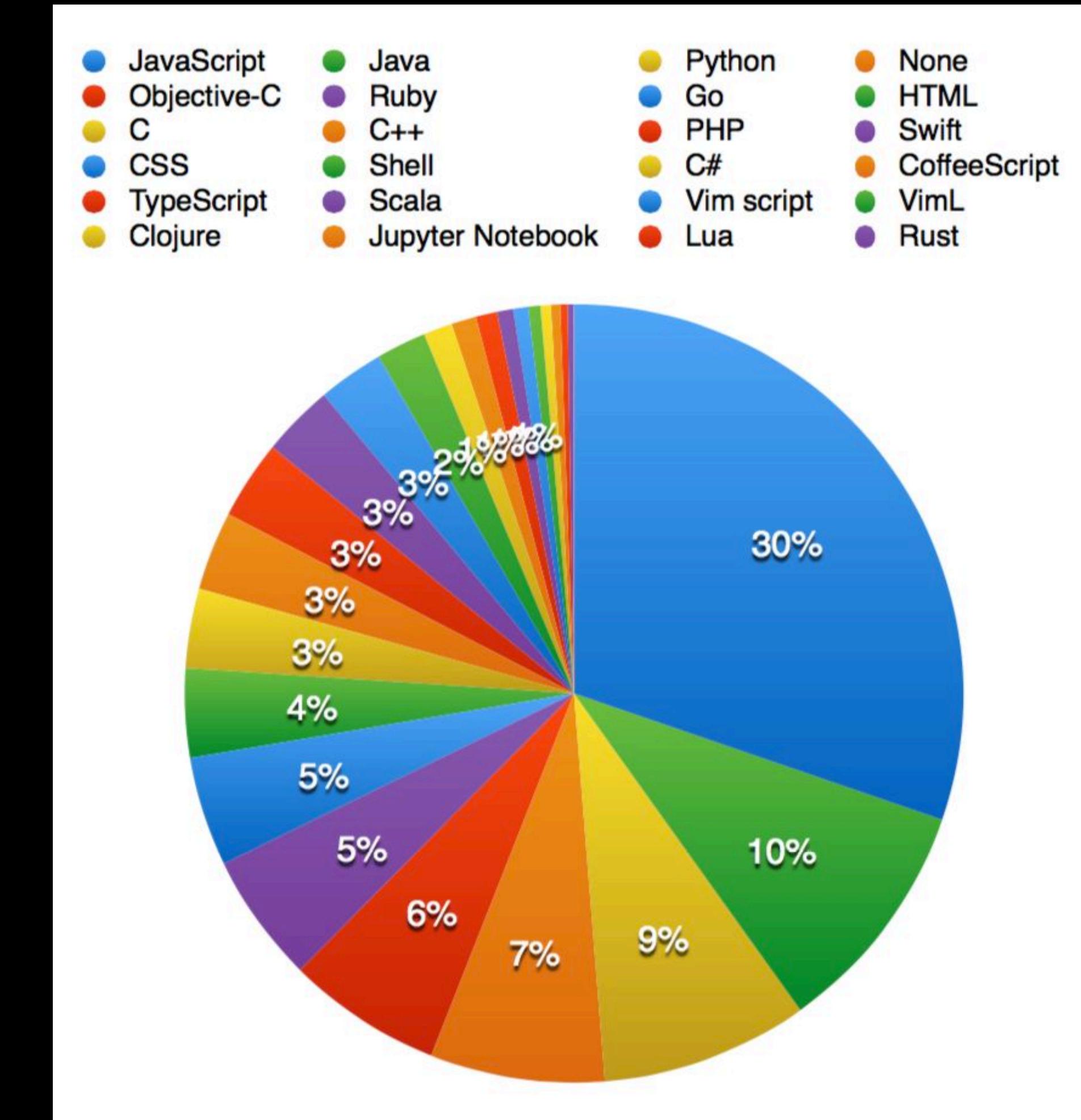


前端技术概览

开源软件的互联网+福利

Google
Facebook
Apache
Microsoft
Mozilla
Codrops
Twitter
Square
Google Samples
Netflix



从开源到包管理

npm ————— javascript
maven ————— java
nuget ————— .net
rubyGems ————— ruby
pypi ————— python
composer ————— PHP
bower ————— javascript、css

- 配置即引用
- 中央/代理/镜像库管理
- 依赖管理
- 版本管理
- 命令行工具
- IDE集成

NodeJS的前世今生

1996——Netscape设计javascript
1998——ECMA 1、2
2000——ECMA 3
2002——FireFox浏览器
2003——Safari浏览器
2008——Chrome浏览器
2009——Node.js
2010——ECMAScript 5 (ES5)
2015——ECMAScript 6 (ES6)

Javascript运行环境 (runtime)

基于Chrome V8引擎

在非浏览器环境下运行

速度非常快

性能非常好

模块化

事件驱动

非阻塞I/O

PHP=>250并发 NodeJS=>65535并发

用几行代码搭建web服务器

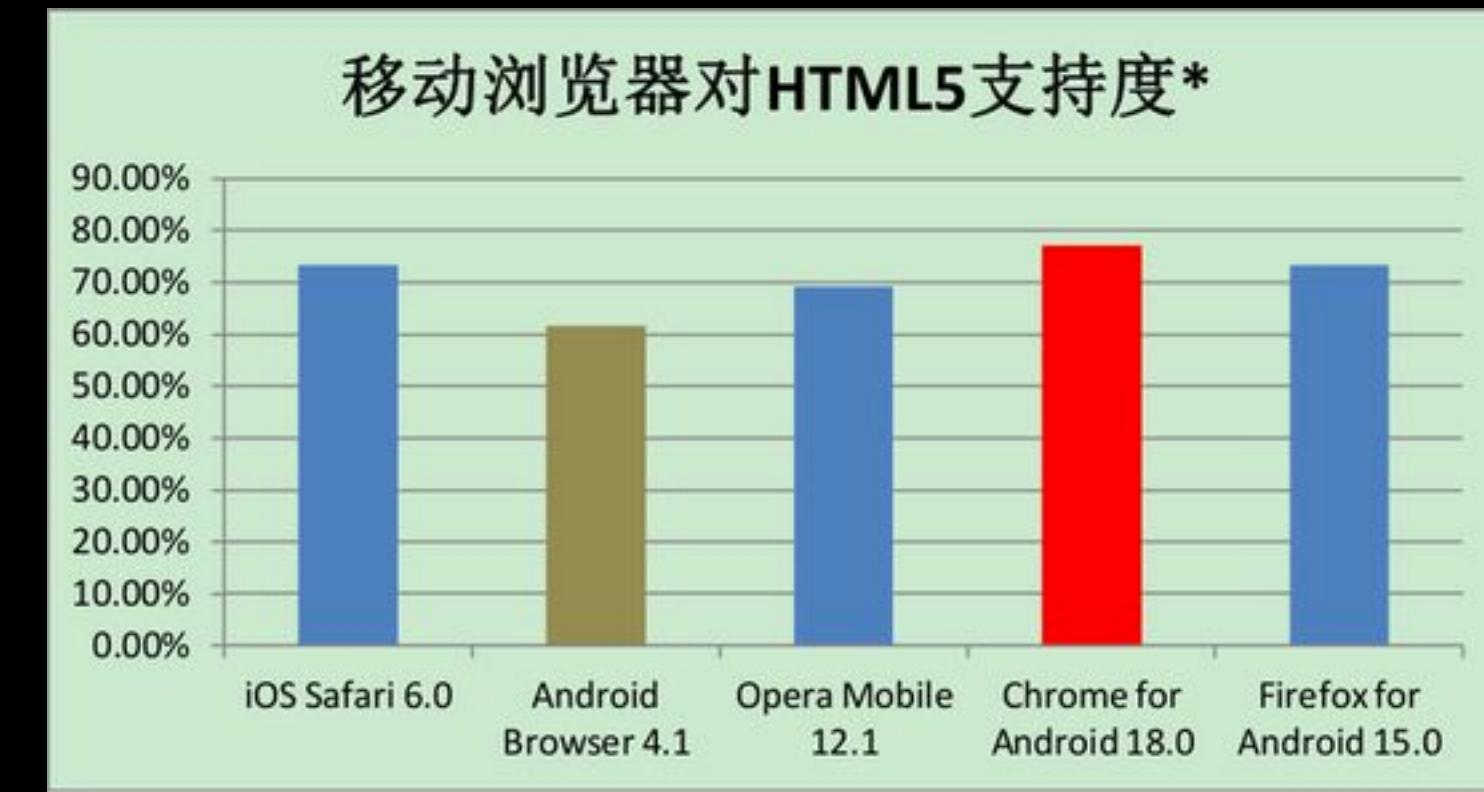
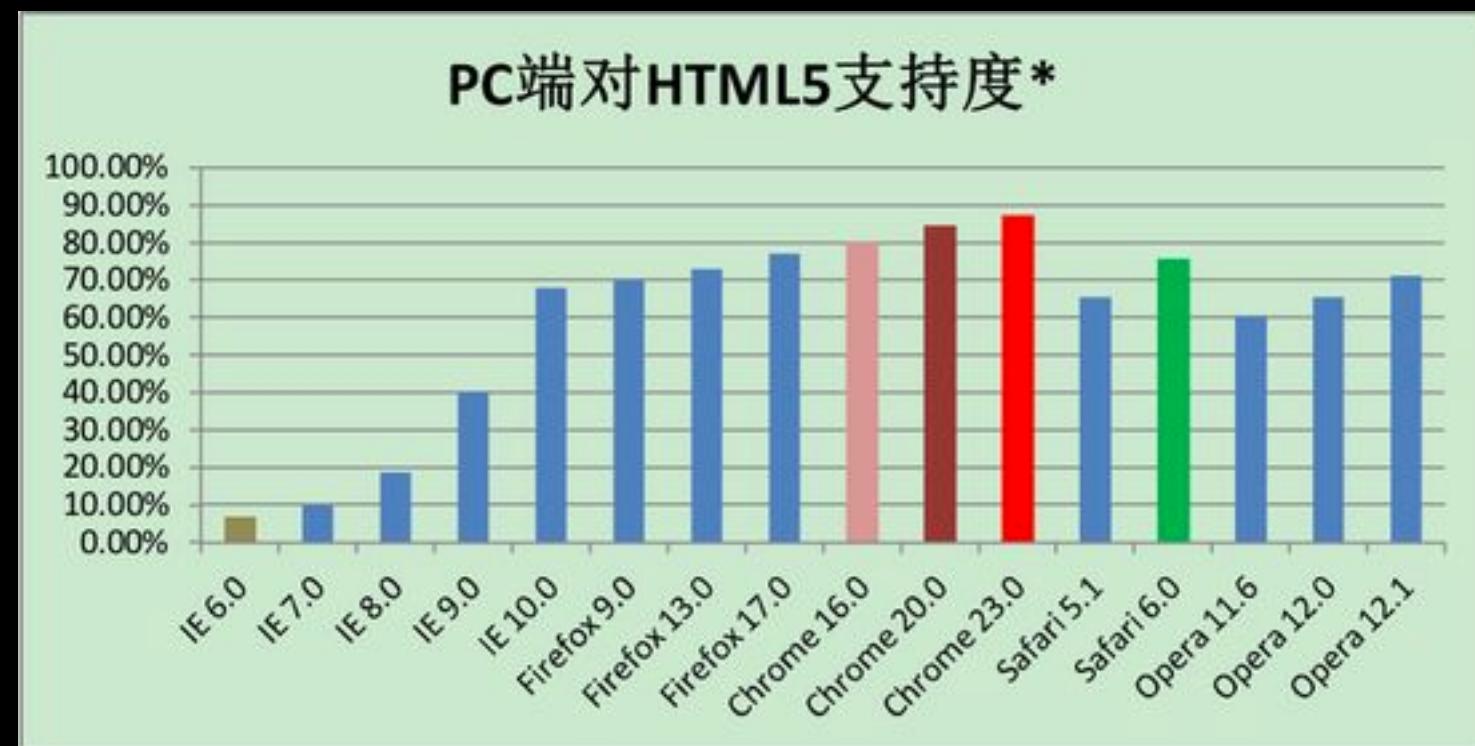
```
npm init  
npm install —save http
```

```
var http = require('http') ;  
var server = http.createServer(function(req,res){  
res.writeHead(200,{  
... 'Content-Type' : 'text/plain;charset=utf-8'  
});  
res.end("Hello,World! ") ;  
});  
server.listen(8888) ;  
console.log("http server running on port 8888 ...");
```

```
npm init  
npm install —save express
```

```
var express = require("express");  
var app = express();  
app.use(express.static(__dirname + '/public'));  
app.get('/',function(req,res){  
... res.send('Hello index!');  
});  
app.get('/hello',function(req,res){  
... res.send('Hello world!');  
});  
app.listen(3000);  
console.log("http server running on port 3000 ...")
```

移动互联网带来的浏览器变革



语义标签

多媒体

WebSocket

3D

WebGL

CSS3

本地存储

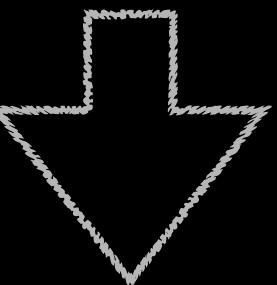
硬件支持 (定位, 麦克风, 摄像头)

SVG

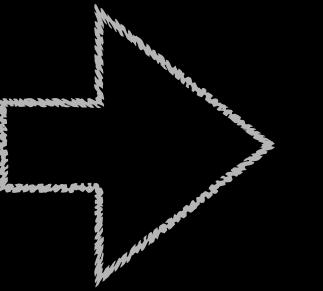
Canvas

前端代码打包与打包工具

合并、压缩、解决兼容问题

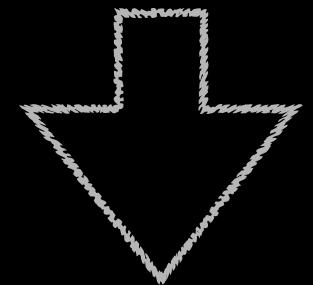


打包工具



gulp

以任务为核心



app.min.js、app.min.css

grunt

webpack

以成果物为核心

Web前端的蝴蝶效应



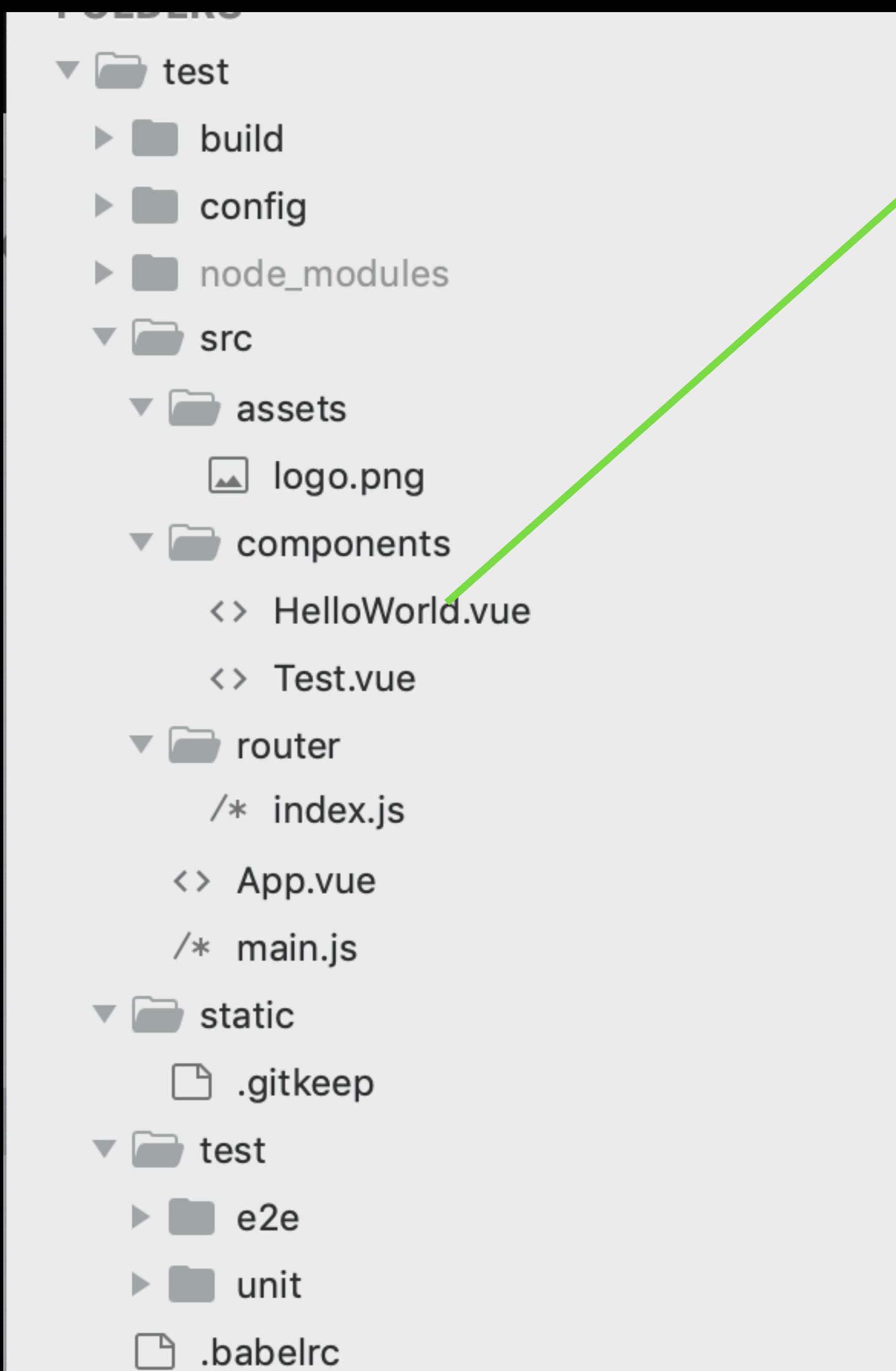
脚手架

```
gdw-mbp% vue-init webpack test

? Project name test
? Project description A Vue.js project
? Author gaodawei <gaodawei@linkme.org.cn>
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? Yes
? Pick an ESLint preset Standard
? Set up unit tests Yes
? Pick a test runner jest
? Setup e2e tests with Nightwatch? Yes
? Should we run `npm install` for you after the project has been created? (recommended) npm

vue-cli · Generated "test".
```

代码结构

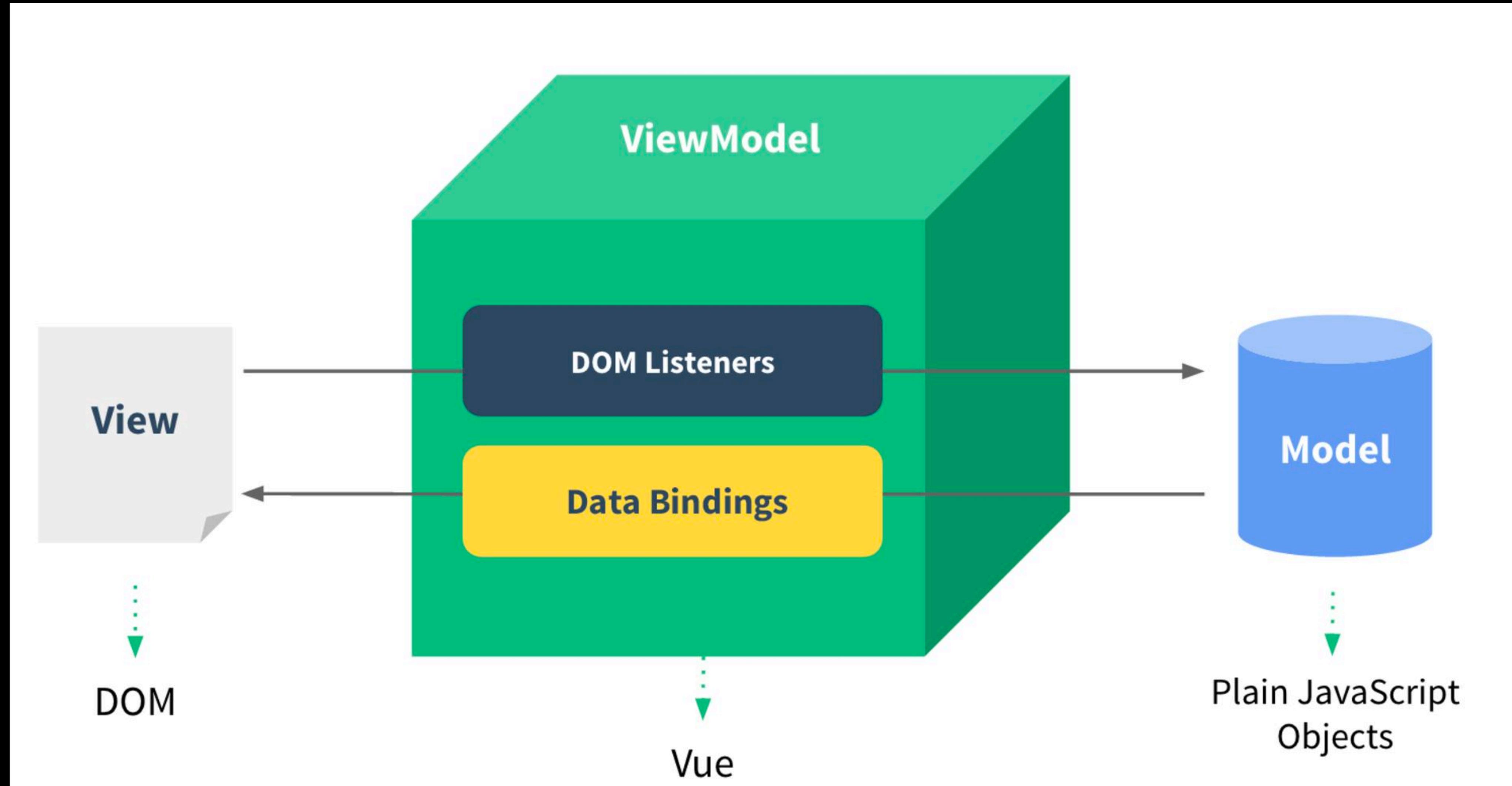


```
<template>
  <div class="hello">...
  </div>
</template>

<script>
export default {
}
</script>


<style scoped>
h1, h2 {
  font-weight: normal;
}
ul {
  list-style-type: none;
  padding: 0;
}
li {
  display: inline-block;
  margin: 0 10px;
}
a {
  color: #42b983;
}
</style>
```

MVVM



VUE 数据绑定

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
    <h2>Essential Links</h2>
    <ul>
      <li>
        <a href="/test">
          Test
        </a>
      </li>
      <li>
```

```
<script>
export default {
  name: 'HelloWorld',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  }
}</script>
```

路由

```
1 import Vue from 'vue'  
2 import Router from 'vue-router'  
3 import HelloWorld from '@/components/HelloWorld'  
4 import Test from '@/components/Test'  
5  
6 Vue.use(Router)  
7  
8 export default new Router({  
9   mode: 'history',  
0   routes: [  
1     {  
2       path: '/',
3       name: 'HelloWorld',
4       component: HelloWorld
5     },
6     {
7       path: '/test',
8       name: 'Test',
9       component: Test
0     }
1   ]
2 })  
3
```

指令：循环、条件

```
<div v-for="(item,index) in list">
  <div v-if="index % 2 == 0" style="
    background-color:#EFEFEF">
    {{ item.name }}:{{ item.value }}
  </div>
  <div v-else style="background-color:#DDDDDD">
    {{ item.name }}:{{ item.value }}
  </div>
  <el-button v-on:click="ok(item.name)">test</
  el-button>
</div>
```

指令：事件绑定

```
<div v-for="(item, index) in list">
  <div v-if="index % 2 == 0" style="background-color:#EFEFEF">
    {{ item.name }}:{{ item.value }}
  </div>
  <div v-else style="background-color:#DDDDDD">
    {{ item.name }}:{{ item.value }}
  </div>
  <el-button v-on:click="ok(item.name)">test</el-button>
</div>
<\qT^>
  <el-button>
```

```
/* eslint-disable */
export default {
  name: 'Test',
  methods: {
    ok(v){
      alert(v)
    }
  },
}
```

指令：表单绑定

```
<template>
<div>
    <input v-model="user"></input>
    <el-button @click="getUser">get user</el-button>
    You input user is {{user}}
```

```
data () {
  return {
    user:"xxx",
    list: [
      {
        id: 1,
        name: "张三"
      }
    ]
}
```

外部组件

```
1 // The Vue build version to load with the `import`  
2 // (runtime-only or standalone) has been set in we  
3 import Vue from 'vue'  
4 import App from './App'  
5 import router from './router'  
6 import ElementUI from 'element-ui'  
7 import 'element-ui/lib/theme-chalk/index.css'  
8  
9 Vue.config.productionTip = false  
.0 Vue.use(ElementUI);  
.1 |  
.2 /* eslint-disable no-new */  
.3 new Vue({  
.4   el: '#app',  
.5   router,  
.6   components: { App },  
.7   template: '<App/>'  
.8 })  
.9
```

组件应用

```
<el-table
  :data="list"
  style="width: 100%">
  <el-table-column
    prop="name"
    label="名"
    width="180">
  </el-table-column>
  <el-table-column
    prop="value"
    label="值"
    width="180">
  </el-table-column>
  <el-table-column
    prop="value"
    label="操作"
    width="180">
    <template slot-scope="scope">
      <el-button @click="ok(scope.row.name)" type="text" size="small">操作</el-button>
    </template>
  </el-table-column>
</el-table>
...
```

自定义组件

```
1 import Vue from 'vue'  
2 export default Vue.component('button-counter', {  
3   data: function () {  
4     return {  
5       count: 0  
6     }  
7   },  
8   template: '<button v-on:click="count++">You clicked me {{ count }} times.  
9 })|
```

```
<template>  
  <div>  
    <input v-model="user"></input>  
    <button-counter />  
    <el-button @click="getUser">get user</el-button>  
    You input user is {{user}}  
    <el-table  
      :data="list">
```