

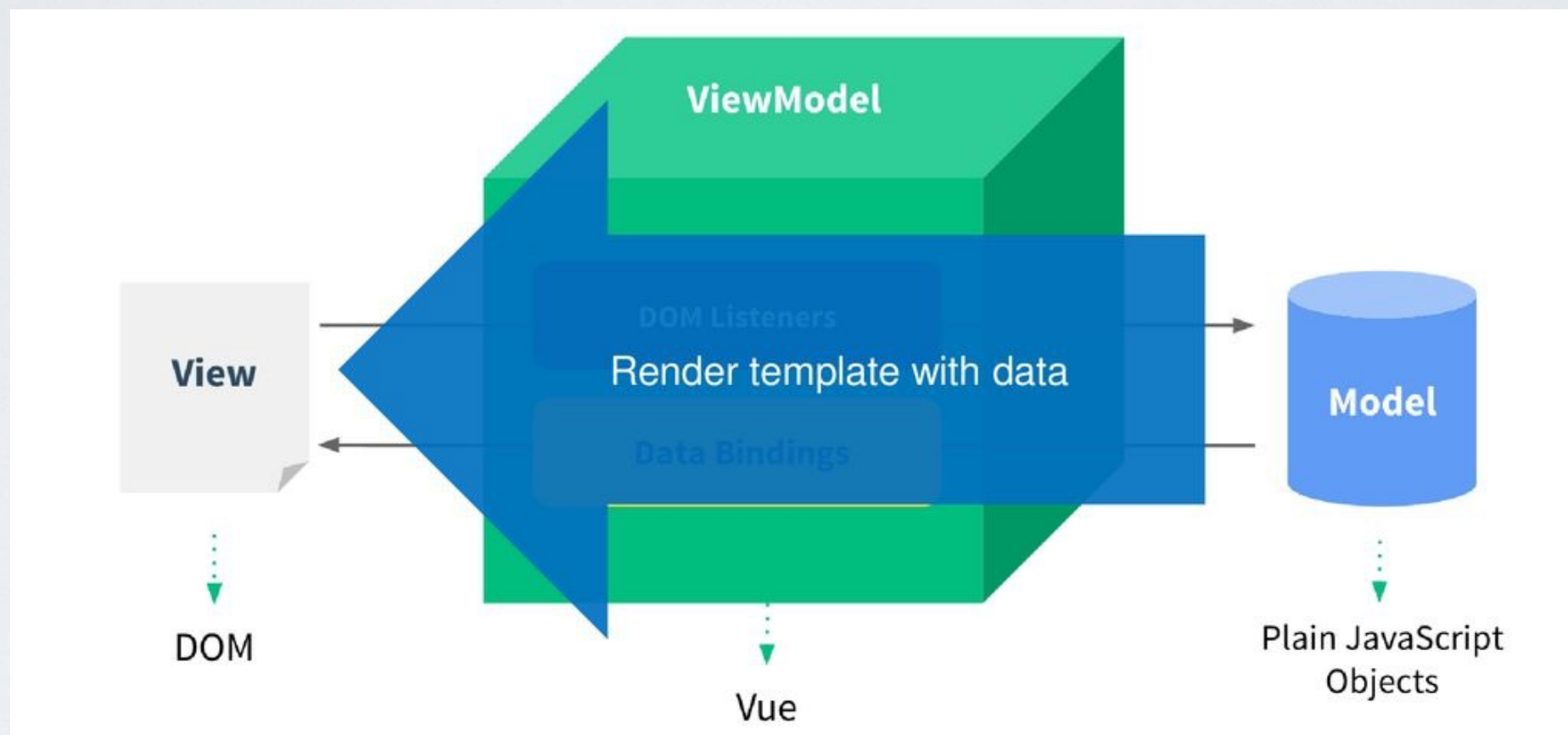
《Vue基础》第二单元

插值 / 数据绑定



Vue的核心

- 用简洁的 **模板语法** 声明式地将数据渲染进DOM



- 当数据发生变化时，Vue会自动更新DOM视图

哪些数据需要绑定？

```
..<div
  ..  title="属性"
  ..  class="wrapper"
  ..  style="text-align:center"
  ..>
  ..  <h1>标题文本</h1>
  ..  <p>
  ..    ..  我是
  ..    ..  <strong>前端</strong>
  ..    ..  工程师
  ..  </p>
..</div>
```

- 文本
- 属性
- 原始HTML

在HTML里写JS逻辑

- 条件
- 循环

```
if (condition1) {  
    ... // ...  
} else if (condition2) {  
    ... // ...  
} else {  
    ... // ...  
}
```

```
for (let i = 0; i < list.length; i++) {  
    ... const item = list[index];  
    ... // ...  
}
```

文本

- 文本插值：使用双大括号 **{{ }}**

```
<span>现在时间 · {{ time }}</span>
```

课堂案例：时钟

- 案例地址 demo.jxsjs.com/clock-1.html

Vue库 / 素材地址



<http://jxsjs.com/lib/vue.min.js>



<http://jxsjs.com/img/logo.png>



音乐时间 10:35:24

文本

- v-once 指令，一次性插值

```
· <span>现在时间 · {{ time }} </span>  
· <span v-once>最初的最初 · {{ time }} </span>
```

属性

- alt、 title
- id、 class、 style
- href、 src
- value、 checked、 disabled

属性

- 属性插值：使用 **v-bind** 指令

```
<div v-bind:id="divId"></div>  
<button v-bind:disabled="isDisabled">按钮</button>
```

```
data: {  
  divId: 'wrapper',  
  isDisabled: true  
}
```

原始HTML

innerHTML

- 输出HTML：使用 **v-html** 指令

```
<div>{{ rawHtml }}</div>
```



```
<div v-html="rawHtml"></div>
```



```
data: {  
  rawHtml: '我是一只<strong>小小鸟</strong>'  
}
```

表达式

- 所有插值处，均可使用单个JS表达式

```
<div>{{ number + 1 }}</div>  
<div>{{ ok ? 'YES' : 'NO' }}</div>  
<div>{{ message.split('').reverse().join('') }}</div>  
<div v-bind:id="'list-' + id"></div>
```

课堂案例：

一只鸭子一张嘴，两只眼睛两条腿

表达式

- 访问实例中data定义的变量
- 不可访问自定义的全局变量
- 只能访问白名单中的系统全局变量，
如 Math 和 Date 等

条件渲染

- **v-if、v-else-if、v-else** 指令
只有表达式为真时内容被渲染

```
<div v-if="name === 'zs'">我是张三</div>  
<div v-else-if="name === 'ls'">我是李四</div>  
<div v-else-if="name === 'ww'">我是王五</div>  
<div v-else>我是谁? </div>
```

条件渲染

- 用不可见的包裹元素 `<template>`
判断切换多个元素

```
<template v-if="ok">
  . . . . . <h1>标题</h1>
  . . . . . <p>段落 1</p>
  . . . . . <p>段落 2</p>
</template>
```



```
<div v-if="ok" class="wrapper">
  . . . . . <h1>标题</h1>
  . . . . . <p>段落 1</p>
  . . . . . <p>段落 2</p>
</div>
```



条件展示

- **v-show** 指令，显示 / 隐藏元素
- v-show 不支持 `<template>` 元素，也不支持 v-else

v-if 和 v-show 对比

v-if 条件渲染	v-show 条件展示
直到条件第一次变为真才渲染	不管条件是什么，元素总是被渲染，只是简单display切换
切换开销大 用于运行时条件很少改变	初始渲染开销大 用于频繁切换
切换过程中， 元素的事件监听和子组件会销毁和重建	-

案例

- {{ }}
- v-bind
- v-html
- 表达式
- v-if
- v-show

课堂案例：Tab 切换



- 案例地址 demo.jxsjs.com/tab.html

Vue库地址



<http://jxsjs.com/lib/vue.min.js>

我收藏的

最近听的

收藏内容

循环 / 列表渲染

- **v-for** 指令

基于数组渲染列表



```
[{
  "id": "0025Nh1N2yWrP4",
  "name": "周杰伦",
  "avatar": "https://y.gtimg.cn/mus
}, {
  "id": "002azErJ0UcDN6",
  "name": "张杰",
  "avatar": "https://y.gtimg.cn/mus
}, {
  "id": "0003ZpE43ypssl",
  "name": "张碧晨",
  "avatar": "https://y.gtimg.cn/mus
}, {
  "id": "004Be55m1SJalK",
  "name": "张学友",
  "avatar": "https://y.gtimg.cn/mus
}, {
  "id": "000aw4WC2EQYTv",
  "name": "张靓颖",
```

循环 / 列表渲染

- 使用 item **in** items 形式的语法
- v-for 可选第二个参数，即当前项的索引

```
<ul>  
  <li v-for="item in list">{{ item }}</li>  
</ul>
```

```
<li v-for="(item, index) in list">  
  {{ index }} - {{ item }}  
</li>
```


循环 / 列表渲染

- v-for 也可以遍历对象
- 第一参数为值，第二可选参数为键名
- v-for 也可以接受整数，表示重复次数

```
<span v-for="n in 10">{{ n }}</span>
```

循环 / 列表渲染

- 尽可能在使用 v-for 时提供 **key**

```
<div v-for="item in items" v-bind:key="item.id">
| ... <!-- 内容 -->
</div>

<div v-for="(item, index) in items" v-bind:key="index">
| ... <!-- 内容 -->
</div>
```

案例

课堂案例：歌手列表

- {{ }}
- v-bind
- v-for



- 案例地址 demo.jxsjs.com/singer-list-1.html

Vue库 / 数据地址



<http://jxsjs.com/lib/vue.min.js>

歌手
数据

<http://jxsjs.com/singer-list-1>



周杰伦



张杰



张碧晨



张学友

缩写

- v-bind:属性 → :属性

```
· · <!-- 完整语法 -->  
· · <a v-bind:href="url">...</a>  
· · <!-- 缩写 -->  
· · <a :href="url">...</a>
```

总结

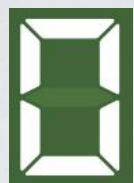
- {{ }}
- v-bind (缩写)
- v-html
- 表达式
- v-if / v-show
- v-for

课后作业 1

- 将时钟改进成液晶电子表
- 将时钟改进成带表盘的 (进阶)



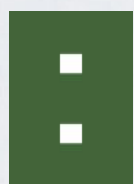
素材地址



<http://jxsjs.com/img/0.png>



1.png、 2.png 9.png



-.png



课后作业 2

- 带分组的歌手列表，循环嵌套

歌手数据

<http://jxsjs.com/singer-list-2>

