

手把手教你写VIRTUAL DOM

前端学堂FED123

小马老师

关于我

- 2012-2014 JAVA开发
- 2014-2018 前端开发



过眼云烟

- I. 研究生毕业于中国科技大学自动化专业，主修人工智能
- II. 曾就职于网易、阿里巴巴等知名互联网公司，参与核心项目架构建设、研发
- III. 撰写《我的JAVA不一般》电子书，翻译IONIC中文教程，创建达易科技工作室

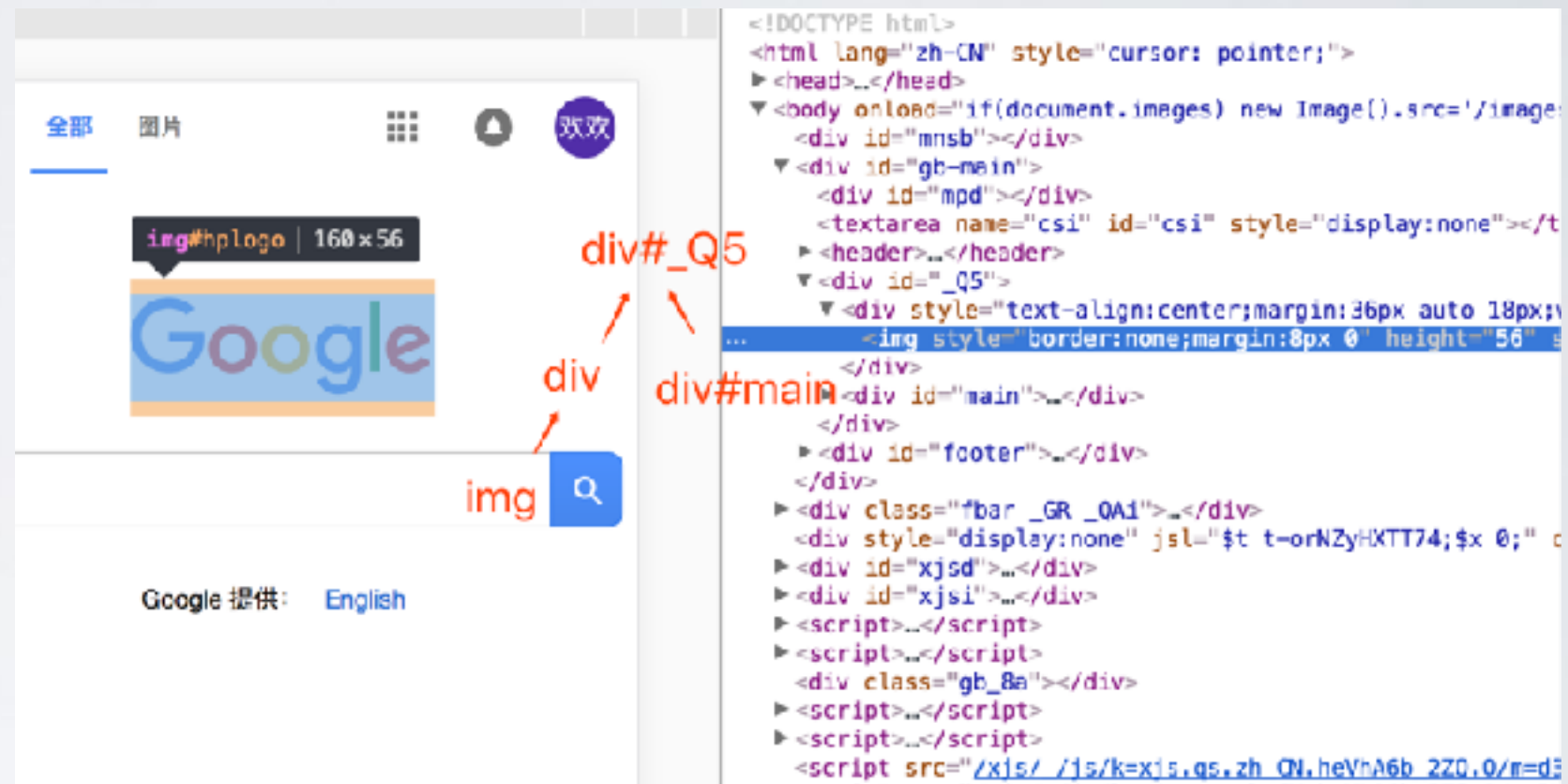
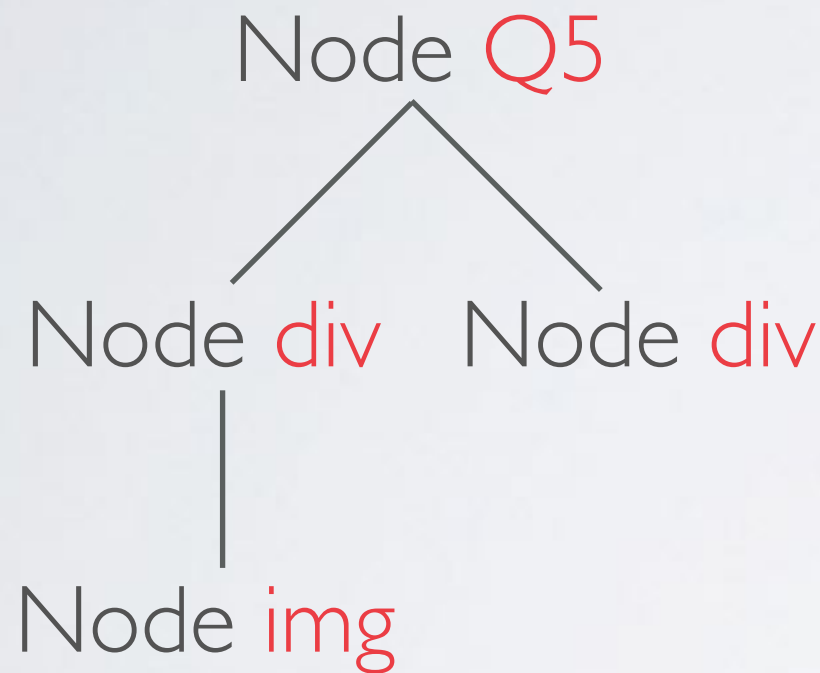
课程提纲

- DOM 与 Virtual DOM
- 构建 Virtual DOM
- DIFF Virtual DOM & Update
- Handle Props & Event
- 封装组件与应用



DOM与Virtual DOM

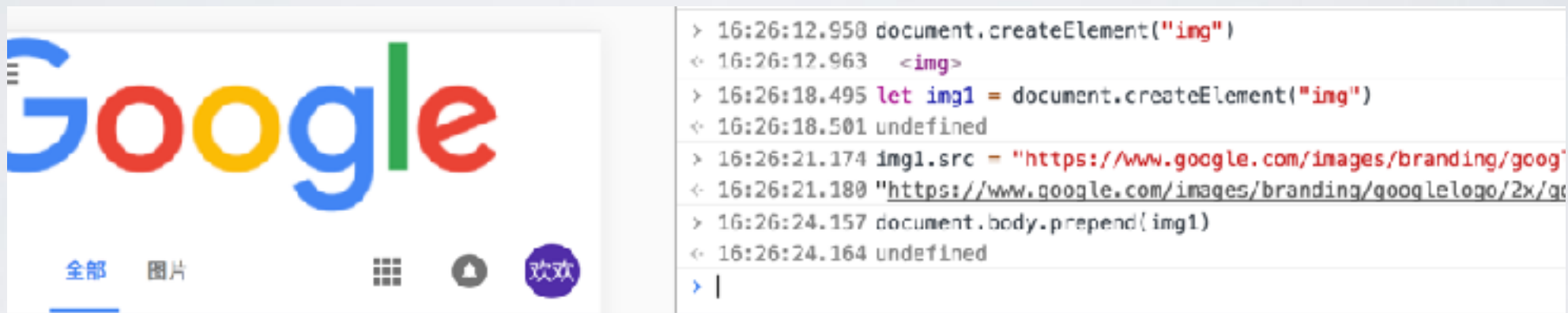
- DOM = Document Object Model



DOM = {node1, node2 ... noden}

DOM与Virtual DOM

- VDOM = Virtual Document Object Model
- VNODE = Virtual Node = Node in memory



Vnode **img1**

TIPS: 当Vnode**挂载**到页面上之后就不再是Vnode, 变成了node, 引用页面的真实node节点

DOM与Virtual DOM

- $VDOM = \{vnode1, vnode2 \dots vnode_n\}$



VNode img

```
> 16:35:46.055 let q5 = document.createElement("div")
< 16:35:46.074 undefined
> 16:35:59.427 q5.id = "_Q5"
< 16:35:59.431 "_Q5"
> 16:36:32.642 let div = document.createElement("div")
< 16:36:32.646 undefined
> 16:36:41.114 div.style = "text-align:center;margin:36px auto 18px;width:160px;line-height:0"
< 16:36:41.118 "text-align:center;margin:36px auto 18px;width:160px;line-height:0"
> 16:37:00.404 let img1 = document.createElement("img")
< 16:37:00.408 undefined
> 16:37:03.483 img1.src = "https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_160x56dp.png"
< 16:37:03.489 "https://www.google.com/images/branding/googlelogo/2x/googlelogo_color_160x56dp.png"
> 16:37:18.034 img1.style = "border:none;margin:8px 0"
< 16:37:18.039 "border:none;margin:8px 0"
> 16:37:27.610 img1.height = "56"
< 16:37:27.615 "56"
> 16:37:50.770 div.appendChild(img1)
< 16:37:50.774 
> 16:38:04.951 q5.appendChild(div)
< 16:38:04.957 <div style="text-align: center; margin: 36px auto 18px; width: 160px; line-height: 0;">
  
</div>
</div>
> 16:38:07.539 document
< 16:38:07.544 #document
> 16:38:14.925 q5
< 16:38:14.929 <div id="_Q5">
  <div style="text-align: center; margin: 36px auto 18px; width: 160px; line-height: 0;">
    
  </div>
</div>
> 16:38:39.911 document.body.prepend(q5)
< 16:38:39.916 undefined
> 16:39:11.543 q5.style = "border: 1px solid red;"
< 16:39:11.549 "border: 1px solid red;"
> 1
```

TIPS: 当VDOM挂载到页面上之后就不再是VDOM, 是渲染后真实DOM的引用

课程提纲

- DOM 与 Virtual DOM
- 构建 Virtual DOM
- DIFF Virtual DOM & Update
- Handle Props & Event
- 封装组件与应用



构建Virtual DOM

构建Vnode

type: 元素类型

props: 元素属性

children: 子元素集合

IF:

SO:

```
q5.appendChild(div)
```

```
<div style="text-align: center; margin: 36px auto 1
  
</div>
```

```
{type: "div", props: {"style": "..."}, children: [
  {type: "img", props: {"src": "..."}}
]}
```

```
function vnode(type, props, ...children) {
  return { type, props, children }
}
```

```
vnode("div", {"style": "..."}, [
  vnode("img", {"src": "..."})
])
```


构建Virtual DOM

step I :

HTML → Vnodes 基于打包工具实现自动处理



GULP : 文件流管道操作



awesome

webpack

: 模块化松耦合处理



rollup.js

: ES6模块调用分析处理

构建Virtual DOM - Practice I

ES6

- babel-core
- babel-preset-env

JSX - transform html

- babel-plugin-transform-react-jsx

Rollup

- rollup
- Rollup-plugin-babel

```
npm install --save-dev rollup rollup-plugin-babel babel-core  
babel-plugin-transform-react-jsx babel-preset-env
```



构建Virtual DOM

step1: HTML → Vnodes

```
1  const vdom = [  
2    <div id="_Q5" style="border: 1px solid red;">  
3      <div style="text-align: center; margin: 36px 0px;">  
4        </img>  
5      </div>  
6    </div>]
```

```
1  /* fed123.com */  
2  'use strict';  
3  
4  var vdom = vnode(  
5    "div",  
6    { id: "_Q5", style: "border: 1px solid red;" },  
7    vnode(  
8      "div",  
9      { style: "text-align: center; margin: 36px 0px;" },  
10     vnode("img", { src: "https://www.google.com/images/logo.png", style: "border: none; margin: 8px 0px;" })  
11   )  
12 );  
13
```

step2: Vnodes → Vdom

```
8  function createElement(node) {  
9    if (typeof node === 'string') {  
10      return document.createTextNode(node);  
11    }  
12    const $el = document.createElement(node.type);  
13    node.children  
14      .map(createElement)  
15      .forEach($el.appendChild.bind($el));  
16    return $el;  
17  }
```

构建Virtual DOM - Practice2

step1: HTML → Vnodes

step2: Vnodes → Vdom



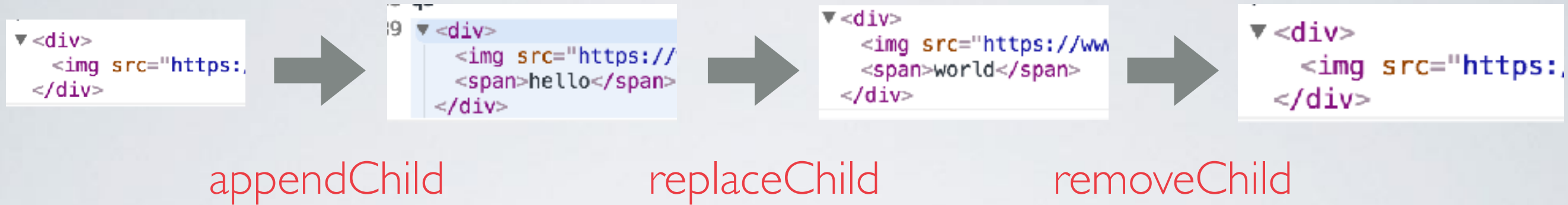
```
1  const vdom = (  
2    <div id="_Q5" style="border: 1px solid red;">  
3      <div style="text-align: center; margin: 36px au<  
4        </img>  
6          google  
7        </div>  
8      </div>  
9  
10   function vnode(type, props, ...children) {  
11     return { type, props, children };  
12   }  
13  
14   function createElement(node) {  
15     if (typeof node === 'string') {  
16       return document.createTextNode(node);  
17     }  
18     const $el = document.createElement(node.type);  
19     node.children  
20       .map(createElement)  
21       .forEach($el.appendChild.bind($el));  
22     return $el;  
23   }  
24   document.body.appendChild(createElement(vdom));
```


课程提纲

- DOM 与 Virtual DOM
- 构建 Virtual DOM
- DIFF Virtual DOM & Update
- Handle Props & Event
- 封装组件与应用



Diff Virtual DOM & Update



update VDOM = {update Vnode ... update Vnode}



Diff Virtual DOM & Update - Practice

diff children ➡ 增量更新

```
function updateElement($parent, newNode, oldNode, index = 0) {
  if (!oldNode) {
    $parent.appendChild(
      createElement(newNode)
    );
  } else if (!newNode) {
    $parent.removeChild(
      $parent.childNodes[index]
    );
  } else if (changed(newNode, oldNode)) {
    $parent.replaceChild(
      createElement(newNode),
      $parent.childNodes[index]
    );
  } else if (newNode.type) {
    const newLength = newNode.children.length;
    const oldLength = oldNode.children.length;
    for (let i = 0; i < newLength || i < oldLength; i++) {
      updateElement(
        $parent.childNodes[index],
        newNode.children[i],
        oldNode.children[i],
        i
      );
    }
  }
}
```



课程提纲

- DOM 与 Virtual DOM
- 构建 Virtual DOM
- DIFF Virtual DOM & Update
- Handle Props & Event
- 封装组件与应用



Handle Props & Event

```
{  
  type: "div",  
  props: { "style": "..."},  
  children: [  
    { type: "img", props: { "src": "..."}  
  ]  
}
```



```
//set props  
function setProp($target, name, value) {  
  $target.setAttribute(name, value);  
}  
function setProps($target, props) {  
  Object.keys(props).forEach(name => {  
    setProp($target, name, props[name]);  
  });  
}
```

```
const vdom = (  
  <div id="_Q5" style="border: 1px solid red;">  
    <div class="test" style="text-align: center; margin: 8px 0px;">  
        
      google  
      <input type="checkbox" checked="true" />  
    </div>  
  </div>  
)
```

```
var vdom = vnode(  
  "div",  
  { id: "_Q5", style: "border: 1px solid red;" },  
  vnode(  
    "div",  
    { class: "test", style: "text-align: center; margin: 8px 0px;" },  
    vnode("img", { src: "https://www.google.com/images/brand" },  
      "border: none; margin: 8px 0px;" ),  
    "google",  
    vnode("input", { type: "checkbox", checked: "true" })  
  )  
)
```

Handle Props & Event

`<input type="checkbox" checked="true" />`

`<input type="checkbox" />`

```
function setBooleanProp($target, name, value) {  
  if (value) {  
    $target.setAttribute(name, value);  
    $target[name] = true;  
  } else {  
    $target[name] = false;  
  }  
}
```

Diff Props - removeAttribute

```
<img style="border: none; margin: 8px 0" />
```



re-setAttribute

```
<img style="border: none;" />
```



removeAttribute

```
<img />
```



setAttribute

```
<img style="border: none;" />
```



Handle Props & Event - Practice

```
function removeBooleanProp($target, name) {
  $target.removeAttribute(name);
  $target[name] = false;
}
function removeProp($target, name, value) {
  if (isCustomProp(name)) {
    return;
  } else if (typeof value === "boolean") {
    removeBooleanProp($target, name);
  } else {
    $target.removeAttribute(name);
  }
}
function setProps($target, props) {
  Object.keys(props).forEach(name => {
    setProp($target, name, props[name]);
  });
}
function updateProp($target, name, newVal, oldVal) {
  if (!newVal) {
    removeProp($target, name, oldVal);
  } else if (!oldVal || newVal !== oldVal) {
    setProp($target, name, newVal);
  }
}
```

```
function updateProps($target, newProps, oldProps = {}) {
  const props = Object.assign({}, newProps, oldProps);
  Object.keys(props).forEach(name => {
    updateProp($target, name, newProps[name], oldProps[name]);
  });
}
```



Handle Props & Event

How about Events ?  Handle as props ?

```
<img src="" onClick={() => { alert("hello") }} />
```

TIPs: 约定基于on+原生事件名称(首字母大小) 绑定事件

```
//set event
function isEventProp(name) {
  return /^on/.test(name);
}
function extractEventName(name) {
  return name.slice(2).toLowerCase();
}
function addEventListeners($target, props) {
  Object.keys(props).forEach(name => {
    if (isEventProp(name)) {
      $target.addEventListener(
        extractEventName(name),
        props[name]
      );
    }
  });
}
```



Handle Props & Event - Practice

Diff Events ?



```
function test(){alert(1)}  
undefined  
test.toString()  
"function test(){alert(1)}"  
let test2 = test.bind({}, "")  
undefined  
test2.toString()  
"function () { [native code] }"
```

ForceUpdate when event changes

```
function changed(node1, node2) {  
  return typeof node1 !== typeof node2 ||  
    typeof node1 === 'string' && node1 !== node2 ||  
    node1.type !== node2.type ||  
    node.props.forceUpdate;  
}
```

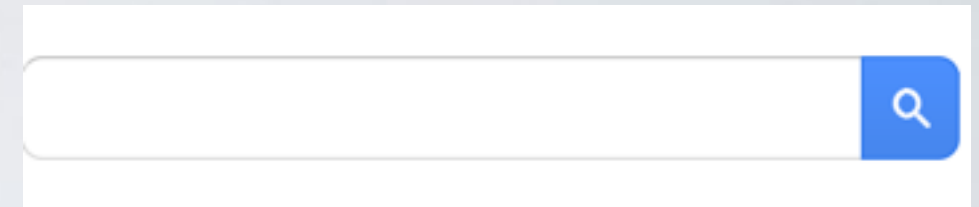
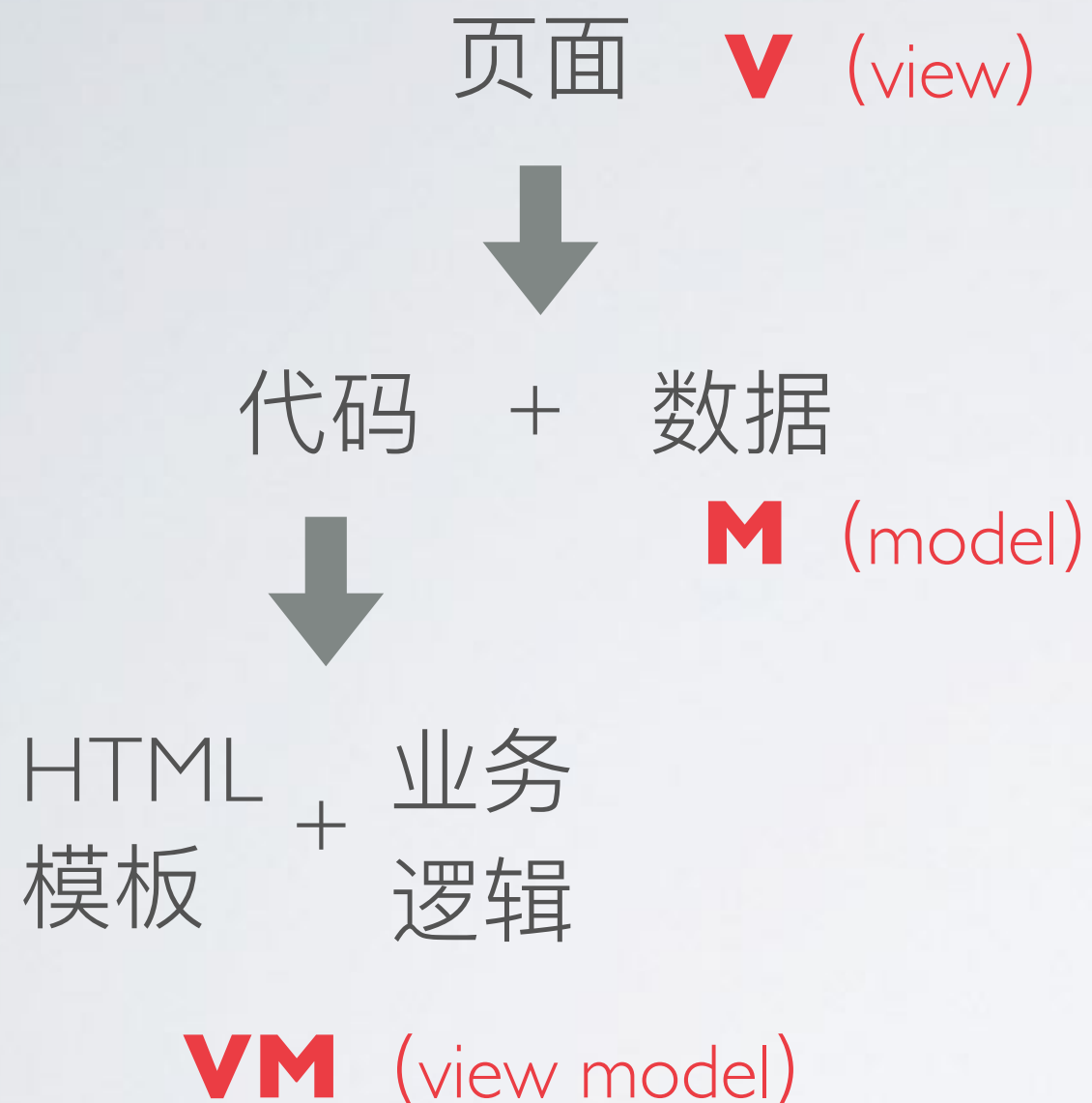


课程提纲

- DOM 与 Virtual DOM
- 构建 Virtual DOM
- DIFF Virtual DOM & Update
- Handle Props & Event
- 封装组件与应用



封装组件与应用



MV-VM组件



MV-VM页面
= {组件1...组件n}

封装组件与应用

如何构建MV-VM开发模式？

- 模块化 - 松耦合

ES6 Module

- 渲染模板 - 业务处理

Virtual DOM

- 打包处理 - 代码检查、压缩等

WebPack,

Rollup,

Gulp

VM (view model)



ES6
Module + Virtual
DOM

+ rollup打包工具



封装组件与应用

函数式编程？

- 函数

```
export const isNumber = (val) => {  
  return typeof val == "number"  
}
```

- 函数 - 通用化处理

```
export const isNumber = (val, type) => {  
  return typeof val == type  
}
```

- 函数 - 柯里化curry 传入相对确定性参数，返回不确定性参数构成的函数

```
const isType = (type) => (val) => {  
  return typeof val == type  
}  
export const isNumber = isType("number")  
export const isString = isType("string")
```

TIPS: 欢迎大家关注前端学堂FED123后续《手把手教你学函数式编程》教程

手把手教你写VIRTUAL DOM

官方博客：fed123.com



公众号



小程序